

数字图像处理第一次作业

学生姓名：雷润泽

班级：自动化少 61

学号：2140506016

提交日期：2019 年 3 月 5 日

摘 要：在第一次作业中，对 BMP 文件格式进行了分析，并且对图像完成了灰度级递减、计算图像均值和方差、分别使用最近邻插值、双线性插值和双三次插值实现图像的尺度变换以及对图像进行仿射变换等处理。

1. BMP 图像格式简介

BMP 是一种常见的位图文件格式。

1.1 BMP 文件格式的基本结构

以位图文件“7.bmp”为例，其结构和内容如表 1 所示

结构体名称	是否可以省略	大小	作用
位图文件头	否	14 字节	存储位图文件通用信息
DIB 头	否	与文件版本有关，取固定的几个值，这里为 40	存储位图详细信息及像素格式
附加位掩码	是，该文件省略	12 或 16Byte	定义像素格式
调色板	色彩位深不超过 8bit 时不能省略	可变	定义像素数组所用颜色
填充区 A	是，该文件省略	可变	结构体对齐
像素数组	否	可变	定义实际的像素数值
填充区 B	是，该文件省略	可变	结构体对齐
ICC 色彩特性数据	是，该文件省略	可变	定义色彩特性

表 1 BMP 文件的整体结构

1.2 BMP 文件各部分内容说明

以位图文件“7.bmp”为例，其各结构体内容如下：

1.2.1 位图文件头

偏移	大小 (Byte)	值	含义
000H	2	424DH	字符“BM”，这一部分用于识别 BMP 文件类型，以“BM”为开头的文件为 Windows 操作系统下的 BMP 文件
002H	4	0000046EH	BMP 文件的大小，此处为 1134Byte
006H	2	0000H	保留
008H	2	0000H	保留
00AH	4	00000436H	像素数组的偏移地址，此处为 436H

表 2 BMP 文件的位图文件头的内容

1.2.2 DIB 头

偏移	大小 (Byte)	值	含义
00EH	4	00000028H	DIB 头的大小，这里为 40Byte
012H	4	00000007H	位图文件的像素宽度，这里为 7
016H	4	00000007H	位图文件的像素高度，这里为 7
01AH	2	0001H	色彩平面数，这个值必须为 1
01CH	2	0008H	每个像素所用的位数，这里为 8

01EH	4	00000000H	使用的压缩方法，这里为 0，即无压缩
022H	4	00000038H	原始位图数据的大小，这里为 56Byte
026H	4	00000000H	图像的水平分辨率（像素/米），这里为 0
02AH	4	00000000H	图像的垂直分辨率（像素/米），这里为 0
02EH	4	00000000H	调色板中的颜色数量，这里为 0，表示默认值
032H	4	00000000H	重要颜色的数量，这里为 0，表示所有颜色都是重要的

表 3 BMP 文件的 DIB 头内容

1.2.3 调色板

偏移为 036H，大小可变。由于“7.bmp”为 8 位位深的灰度图像，故其调色板大小为 1024Byte，包括了 256 个数据，每个数据大小为 4Byte，地址从小到大每个字节的数据分别为 GBRA 四个通道的值。

1.2.4 像素数组

偏移为 436H（由位图文件头可以得到），大小为 56Byte（由 DIB 头可以得到）。这一部分每个数据的大小为 1Byte（由 DIB 头可以得到），表示一个像素点上的颜色，其顺序为按行存储，同一行内地址小的数据在图像左侧，不同行之间地址小的行在图像下侧。每行的大小为 8Byte，与位深和像素宽度的乘积不等，这是因为 BMP 文件每行的大小必须为 4Byte 的整数倍，若位深和像素宽度的乘积不是 4Byte 的整数倍，则在每行中地址较大的部分进行填充对齐，地址较小的部分为有效数据。数据共计 7 列，与像素高度相等。

2. 把 lena 512*512 图像灰度级逐级递减 8-1 显示

思路分析：“lena.bmp”为 512*512, 8 位位深的灰度图像。在 matlab 中，使用 imread 函数可以得到该图像的像素矩阵，每个像素的灰度值为 0 到 255 之间的无符号 8 位整数，因此，只需对像素矩阵的每一个元素右移 n 位，即可得到灰度级减少 n 的灰度图像。

程序代码：使用 matlab 编写，详见附件“intensity.txt”。

处理结果：如图 1 所示。



图1 对“lena.bmp”进行灰度逐级递减的结果
具体的每一张图片详见附件“intensity_1.bmp” -
“intensity_8.bmp”。

观察结果发现，图像的灰度级越多，其能显示的灰度值就越多，当灰度级为8时，共可以显示 $2^8 = 256$ 种不同的灰度值，而当灰度级为1时，则只能显示 $2^1 = 2$ 种不同的灰度值。

3. 计算 lena 图像的均值方差

思路分析：对“lena.bmp”的像素矩阵的每个元素求和并除以总像素数量即为图像的均值。其计算公式为：

$$\bar{f} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N f(i,j)$$

其中， \bar{f} 为图像均值， M 为图像的像素高度， N 为图像的像素宽度， $f(i,j)$ 为图像第*i*行第*j*列的灰度值。

图像方差则可以由以下公式计算得出：

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

其中 $Var(f)$ 为图像方差。

程序代码：使用 matlab 编写，详见附件“mean_var.txt”。

处理结果：图像的均值为 99.051216，方差为 2796.031839。

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

思路分析：

最近邻插值：对于变换后的图像的每一个像素点，计算其尺度变换前的对应点，将据该点最近的原图像的像素的值作为变换后的的像素点的值。

双线性插值：对于变换后图像的每一个像素点，计算其尺度变换前的对应点，并取其 4 个近邻的像素点的值按照下式进行插值：

$$v(x, y) = \sum_{i=0}^1 \sum_{j=0}^1 a_{ij} x^i y^j$$

通过四个近邻点的坐标及像素点的值建立线性方程组求解出 a_{ij} 的值，并将对应点的坐标代入得到尺度变换后的像素点的值。

双三次插值：对于变换后图像的每一个像素点，计算其尺度变换前的对应点，并取其 16 个近邻的像素点的值按照下式进行插值：

$$v(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j$$

通过四个近邻点的坐标及像素点的值建立线性方程组求解出 a_{ij} 的值，并将对应点的坐标代入得到尺度变换后的像素点的值。

程序代码：使用 matlab 编写，参见附件 “zoom.txt”。

处理结果：



图 2 最近邻插值的处理结果（附件 “nearest.bmp”）



图 3 双线性插值的处理结果（附件“bilinear.bmp”）



图 4 双三次插值的处理结果（附件“bicubic.bmp”）

观察处理结果可知，从总体上来说，双三次插值的处理效果好于双线性插值，而双线性插值又好于最近邻插值。但是三种插值方式的计算速度与其处理效果恰好相反，最近邻插值最快，双线性插值其次，双三次插值最慢。此外，可以发现双三次插值的结果的图像右下角有较多噪点，这是因为双三次插值的插值公式为六次插值公式，次数较高，容易出现龙格现象，使插值点的值与临近点的值相差较大，并且，右下角的像素点横纵坐标均过大也导致计算过程中的精度下降，进一步加剧了该区域的噪声。

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

思路分析：设原始图像的某个像素点的坐标为 (v, w) ，对其进行仿射变换，则其变换后对应点的坐标 (x, y) 由下式确定：

$$\begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} u & v & 1 \end{bmatrix} T$$

其中， T 为一线性变换矩阵。

对于水平偏移变换，其仿射矩阵为

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

其中 s_h 为水平偏移参数。

对于旋转变换，其仿射矩阵为

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

其中 θ 为逆时针旋转的弧度。

对于尺度变换，其仿射矩阵为

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

其中 c_x 为垂直方向尺度变换系数， c_y 为水平方向尺度变换系数。

当多种变换依次进行时，只需对 $[u \ v \ 1]$ 依次右乘变换的仿射矩阵即可。

在程序实现时，采用反向映射的方式进行处理，即先给定变换后的像素点的坐标 (x, y) ，再由逆变换

$$[x \ y \ 1]T^{-1} = [u \ v \ 1]$$

计算对应的原始图像像素点。

程序代码：使用 matlab 编写，详见附件“affine.txt”。

处理结果：分别对“lena.bmp”和“elain1.bmp”做参数为 1.5 的水平偏移，再顺时针旋转 30 度，最后通过尺度变换将图像变换至 2048×2048 。

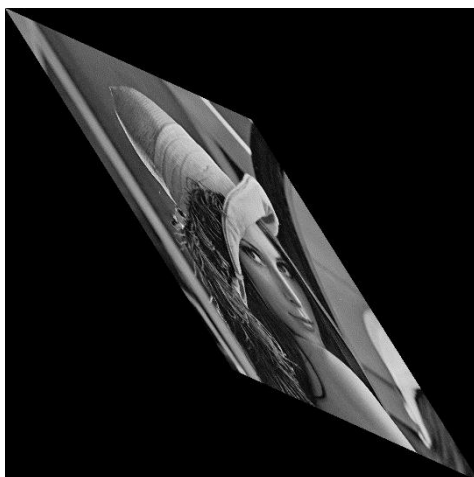


图 5 “lena.bmp” 经过变换和最近邻插值的处理结果
(附件 “lena_affine_nearest.bmp”)

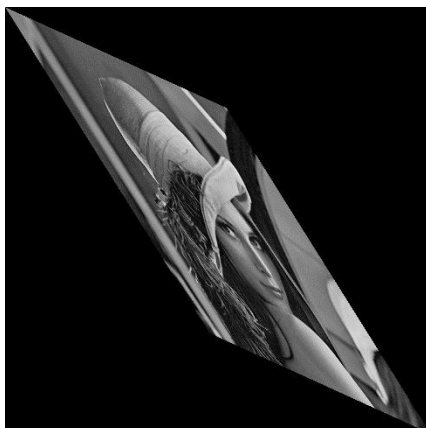


图 6 “lena.bmp” 经过变换和双线性插值的处理结果
(附件 “lena_affine_bilinear.bmp”)

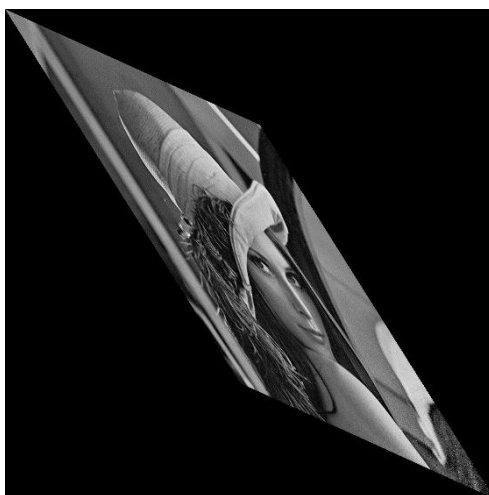


图 7 “lena.bmp” 经过变换和双三次插值的处理结果
(附件 “lena_affine_bicubic.bmp”)

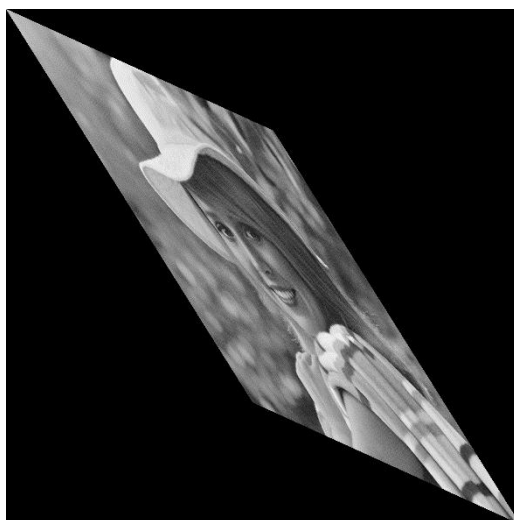


图 8 “elain1.bmp” 经过变换和最近邻插值的处理结果
(附件 “elain_affine_nearest.bmp”)

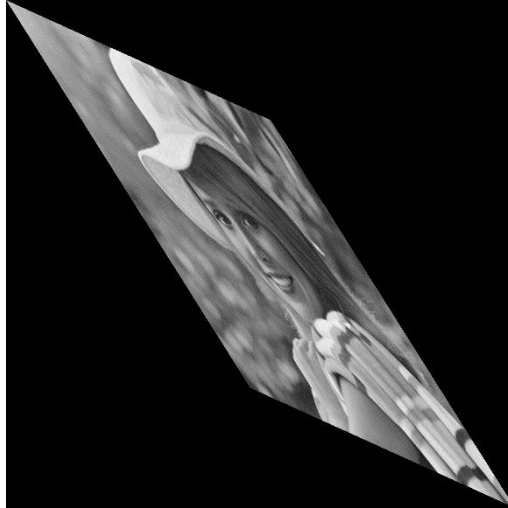


图 9 “elain1.bmp” 经过变换和双线性插值的处理结果（附件
“elain_affine_bilinear.bmp”）

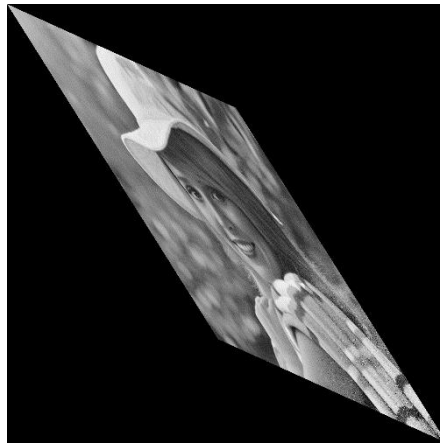


图 10 “elain1.bmp” 经过变换和双三次插值的处理结果（附件
“elain_affine_bicubic.bmp”）