数字图像及视频处理第一次作业

自动化64张晟

2160504107

2019年3月4日

摘要

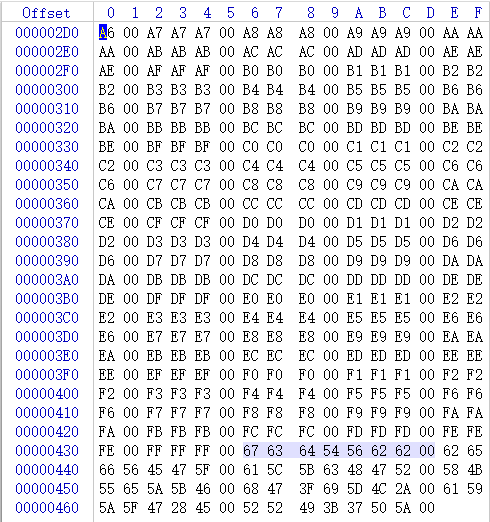
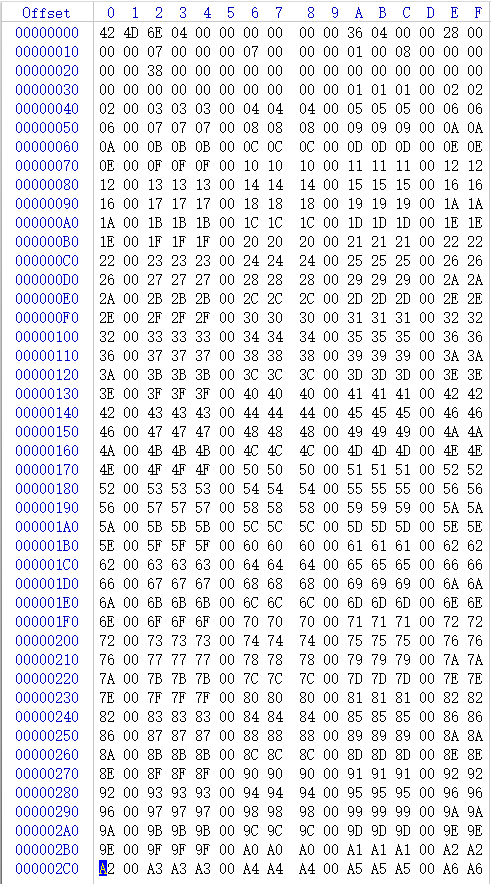
本报告完成了数字图像与视频处理课程的第一次作业任务，主要包括bmp图像格式的分析，图像的灰度级变换，计算均值方差，图像的大小变化，图像的矩阵变换操作等等。报告叙述了解决问题的思路，题目答案以及解题的一些领悟。

第一题：Bmp图像格式简介,以7.bmp为例说明；

解题思路：从网络上查找bmp文件格式的相关内容，使用具有16进制文本编辑功能的文本编辑器打开7.bmp并对其进行分析。

结果：BMP文件格式，又称为Bitmap（位图）或是DIB(Device-Independent Device，设备无关位图)，是Windows系统中广泛使用的图像文件格式。由于它可以不作任何变换地保存图像像素域的数据，因此成为我们取得RAW数据的重要来源。Windows的图形用户界面（graphical user interfaces）也在它的内建图像子系统GDI中对BMP格式提供了支持。

首先用winhex打开7.bmp，如下所示：



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：

**bmp文件头(bmp file header)：**提供文件的格式、大小等信息

**位图信息头(bitmap information)：**提供图像数据的尺寸、位平面数、压缩方式、颜色索引等信息

**调色板(color palette)：**可选，如使用索引来表示图像，调色板就是索引与其对应的颜色的映射表

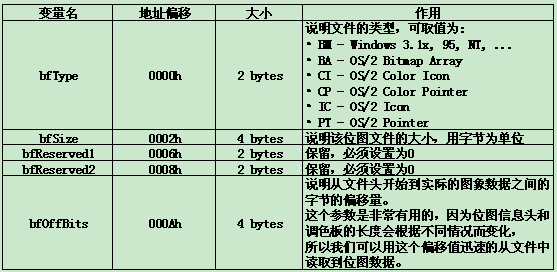
**位图数据(bitmap data)：**图像数据

在分析bmp图像时，有几个原则：

1、在BMP文件中，如果一个数据需要用几个字节来表示的话，那么该数据的存放字节顺序为“低地址存放低位数据，高地址存放高位数据”。

2、所有分析均以字节为序号单位进行。

Bmp文件头：



1-2  ：424dh = 'BM'（ASCII码）,表示这是Windows支持的位图格式。

3-5  ：0000046Eh = 1134 B = 1.134 kB，通过查询文件属性发现一致。

6-9  ：这是两个保留段，为0。

A-D：00000436h = 1078。即从文件头到位图数据需偏移1078字节。稍后将验证这个数据。

共有14个字节。

位图信息头：



0E-11：00000028h = 40,这就是说我这个位图信息头的大小为40个字节。

12-15：00000007h = 7，图像宽为7像素，与文件属性一致。

16-19：00000007h = 7，图像高为7像素，与文件属性一致。这是一个正数，说明图像数据是从图像左下角到右上角排列的。

1A-1B：0001h, 该值总为1。

1C-1D：0008h = 8, 表示每个像素占8个比特，即该图像共有256种颜色。

1E-21：00000000h，BI\_RGB， 说明本图像不压缩。

22-25：00000000h，图像的大小，因为使用BI\_RGB，所以设置为0。

26-29：00000000h，水平分辨率，缺省。

2A-2D：00000000h，垂直分辨率，缺省。

2E-31：00000000h = 0,说明本位图实际使用的颜色索引数为0，与1C-ID得到的结论一致。（8位图像共计256种颜色，索引数为零表示使用所有颜色）

32-35：00000000h = 0,说明本位图重要的颜色索引数为0，与前面得到的结论一致。

调色板：

调色板其实是一张映射表，标识颜色索引号与其代表的颜色的对应关系。它在文件中的布局就像一个二维数组palette[N][4],其中N表示总的颜色索引数，每行的四个元素分别表示该索引对应的B、G、R和Alpha的值，每个分量占一个字节。如不设透明通道时，Alpha为0。因为前面知道，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。这里截取一些数据来说明：

观察从35到435的数据不难发现以连续四个字节为一组，其值呈现1，1，1，0；2，2，2，0；3，3，3，0的递增趋势，直至FF，FF，FF，0，共计256种颜色，由于这些颜色的RGB值都相等，所以颜色为灰色，最深的灰色即黑色，最浅的灰色即白色。

图像数据：

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。注意：由于位图信息头中的图像高度是正数，所以位图数据在文件中的排列顺序是从左下角到右上角，以行为主序排列的。图中436-46D为图像数据，每个字节表示一个点的颜色。

结果讨论：

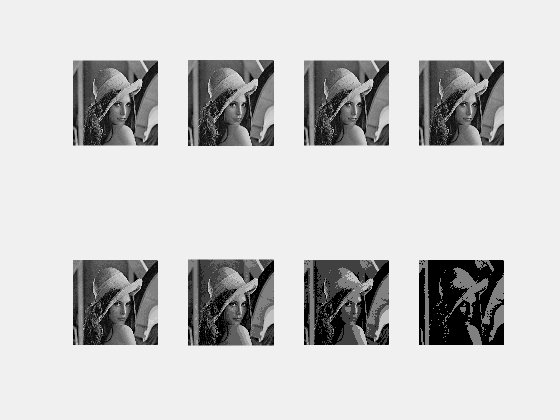
在分析数据时，结果与网上资料相符较好，但是位图数据部分理论上是7\*7=49字节，实际上是56字节。经过分析，我认为是系统读写时以8个字节（64位）或4个字节（32位）为一组。可以观察到位图数据部分每8个字节就有一个是00，用于补齐，不带有实际意义，仅仅是方便读写。

第二题：把lena 512\*512图像灰度级逐级递减8-1显示；

解题思路：

在matlab中读入灰度图片，结果为uint8类型的矩阵，需要转换成数值类型。所谓灰度级的概念，根据冈萨雷斯的数字图像处理一书中相关介绍，可做如下分析。首先将原图像的数值矩阵除以255得到归一化的图像，即取值范围在[0,1]的图像，接着将该矩阵乘以所需灰度级的颜色数，再取整，即可得到指定灰度级数的图片。由于常用的灰度级为2^k，因此考虑使用将原图像除以2^(8-k)并取整，此时灰度级数正确但是由于显示是在[0,255]区间，因此需要乘上2^(8-k)得到正确答案。

结果：



从左上到右下依次为k=8，7，6……3，2，1.

结果讨论：

512\*512的图像已经可以体现较多的细节，当灰度级变化时，k=8，7，6，5时与原图像差别不大。随着k的继续减小，图像产生了明显的变化，在灰度缓变区，出现一些极细微的山脊状结构，同时伴随细节丢失，部分边界变得不平滑等等，图像质量不断变差。从对比中不难得出结论，灰度级数越大，图像色彩越丰富，表达效果越好。

第三题：计算lena图像的均值方差；

解题思路：在matlab中读入灰度图片，结果为uint8类型的矩阵，需要转换成数值类型。此时问题转化为对矩阵求均值和方差，查询资料后发现，matlab提供mean2和std2来计算方差和标准差。将标准差平方即可得到方差。

结果：

均值为99.05方差为2796.04

结果讨论：

均值影响人对图像的整体感受，均值越大，则图像整体感受越亮，均值越小则图像整体感受越暗。方差影响图像对比度，方差越大，图像对比越强烈，方差越小，则图像对比越弱。

第四题：把lena图像用近邻、双线性和双三次插值法zoom到2048\*2048；

解题思路：在matlab中有imresize函数，该函数可以根据所选用的参数设置新图像的尺寸和插值方法。分别使用三种插值方法进行插值即可。

结果：

最近邻：



双线性插值：



双三次插值：



结果讨论：

最近邻插值，双线性插值，双三次插值计算方法不相同，按从前向后的顺序，这三种计算方法使用的点数依次增多，相应的，所需计算时间也越长。最近邻插值在灰度变化剧烈的区域边界极易产生锯齿，双线性的表现要好一些，双三次插值的效果最好，图像的边缘较为平滑。

第五题：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；

解题思路：使用matlab中的仿射变换方法进行水平shear，构造相应的矩阵进行变换，矩阵为[1 1.5 0;0 1 0;0 0 1].使用imrotate可以旋转图像。Zoom图像的方法和上一题相同。

结果：

水平shear——lena

最近邻



双线性插值

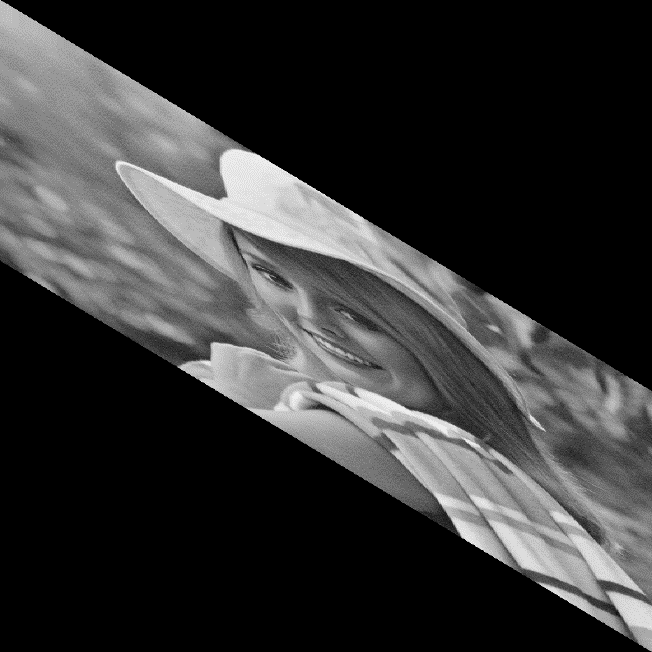


双三次插值

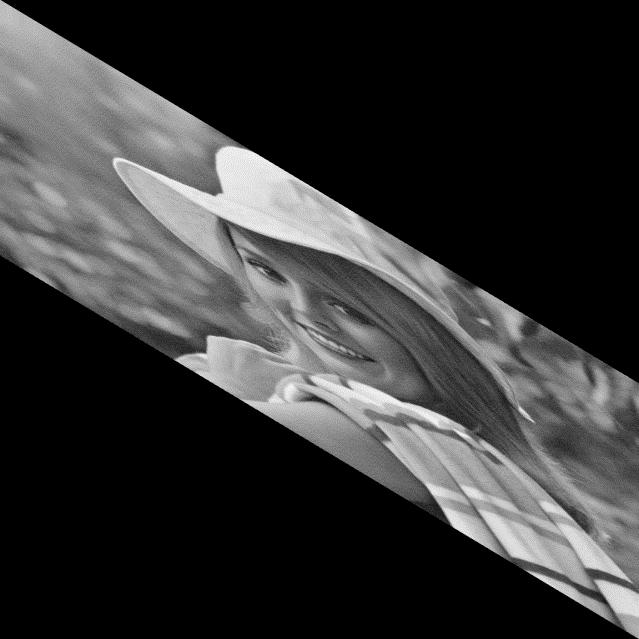


水平shear——elain

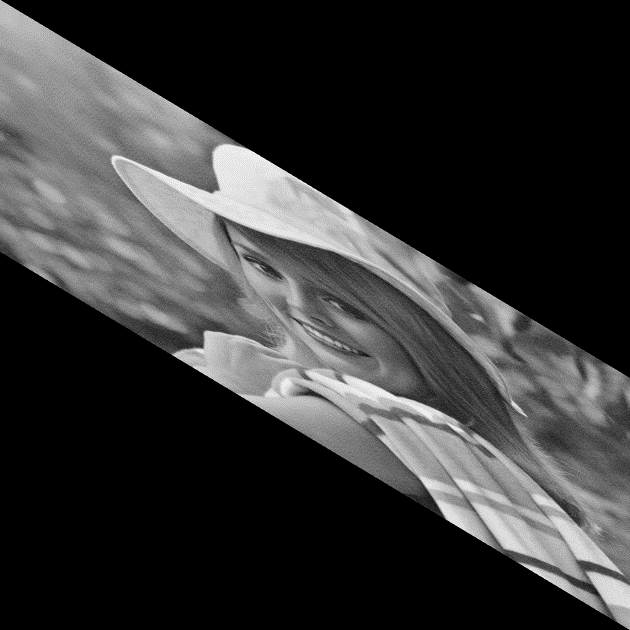
最近邻插值



双线性插值



双三次插值



旋转——lena

最近邻



双线性插值



双三次插值



旋转——elain

最近邻



双线性插值



双三次插值



结果讨论：

本题目主要考察图像的仿射变换，常见的方法有shear和rotate。与上一题相同。最近邻方法得到的图像细节最差，有锯齿边缘，双三次插值效果最好，但用时最长。

附录

Gray\_Level\_Transformation.m

img=[];

img=imread('lena.bmp');

imgt=[];

t=double(img);

for i=1:-1:0

for j=4:-1:1

t=double(img);

k=i\*4+j;

subplot(2,4,9-k);

title(['k=' num2str(k)]);

t=2^(8-k)\*fix(t/(2^(8-k)));

imgt=uint8(t);

imshow(imgt);

imwrite(imgt,['k=' num2str(k) '.bmp']);

end

end

p=getframe(gcf);

imwrite(p.cdata,['total.bmp']);

MeanValueAndVarianceCalculation.m

img=imread('lena.bmp');

t=double(img);

mean\_value=mean2(t)

variance=std2(t)^2

t4\_zoom.m

img=imread('lena.bmp');

img1=imresize(img,[2048 2048],'nearest');

imwrite(img1,'t4\_nearest.bmp');

img2=imresize(img,[2048 2048],'bilinear');

imwrite(img2,'t4\_bilinear.bmp');

img3=imresize(img,[2048 2048],'bicubic');

imwrite(img3,'t4\_bicubic.bmp');

subplot(1,3,1)

imshow(img1);

subplot(1,3,2)

imshow(img2);

subplot(1,3,3)

imshow(img3);

shear.m

img1=imread('lena.bmp');

img2=imread('elain1.bmp');

h=[1,1.5,0;0,1,0;0,0,1];

t=maketform('affine',h);

img10=imtransform(img1,t);

img11=imresize(img10,[2048 2048],'nearest');

imwrite(img11,'lean\_shear\_nearest.bmp');

img12=imresize(img10,[2048 2048],'bilinear');

imwrite(img12,'lean\_shear\_bilinear.bmp');

img13=imresize(img10,[2048 2048],'bicubic');

imwrite(img13,'lean\_shear\_bicubic.bmp');

img20=imtransform(img2,t);

img21=imresize(img20,[2048 2048],'nearest');

imwrite(img21,'elain\_shear\_nearest.bmp');

img22=imresize(img20,[2048 2048],'bilinear');

imwrite(img22,'elain\_shear\_bilinear.bmp');

img23=imresize(img20,[2048 2048],'bicubic');

imwrite(img23,'elain\_shear\_bicubic.bmp');

rotate.m

img1=imread('lena.bmp');

img2=imread('elain1.bmp');

img110=imrotate(img1,30,'nearest');

img111=imresize(img110,[2048 2048],'nearest');

imwrite(img111,'lena\_rotate\_nearest.bmp');

img120=imrotate(img1,30,'bilinear');

img121=imresize(img120,[2048 2048],'bilinear');

imwrite(img121,'lena\_rotate\_bilinear.bmp');

img130=imrotate(img1,30,'bicubic');

img131=imresize(img130,[2048 2048],'bicubic');

imwrite(img131,'lena\_rotate\_bicubic.bmp');

img210=imrotate(img2,30,'nearest');

img211=imresize(img210,[2048 2048],'nearest');

imwrite(img211,'elain\_rotate\_nearest.bmp');

img220=imrotate(img2,30,'bilinear');

img221=imresize(img220,[2048 2048],'bilinear');

imwrite(img221,'elain\_rotate\_bilinear.bmp');

img230=imrotate(img2,30,'bicubic');

img231=imresize(img230,[2048 2048],'bicubic');

imwrite(img231,'elain\_rotate\_bicubic.bmp');