

数字图像与视频处理

第一次作业

姓 名：孔维乐

班 级：自动化 65

学 号：2160504122

提交日期：2019 年 2 月

摘要：数字图像与视频处理技术在近年来受到了越来越多的关注，各种图像与视频的处理方法不断地出现与完善。本报告主要对图片的 **BMP** 格式作简要介绍，然后基于 **MATLAB** 中的图像处理方法对灰度图像进行降低灰度、缩放、旋转、水平错切等操作。然后对通过最近邻插值、双线性插值、双三次插值得到的图像进行对比分析。

关键词：降低灰度 最近邻插值 双线性插值 双三次插值

1. BMP 图像格式

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。

BMP 图像格式由四部分组成：位图头文件、位图信息、调色板、位图数据
7.bmp 的图像信息如下：

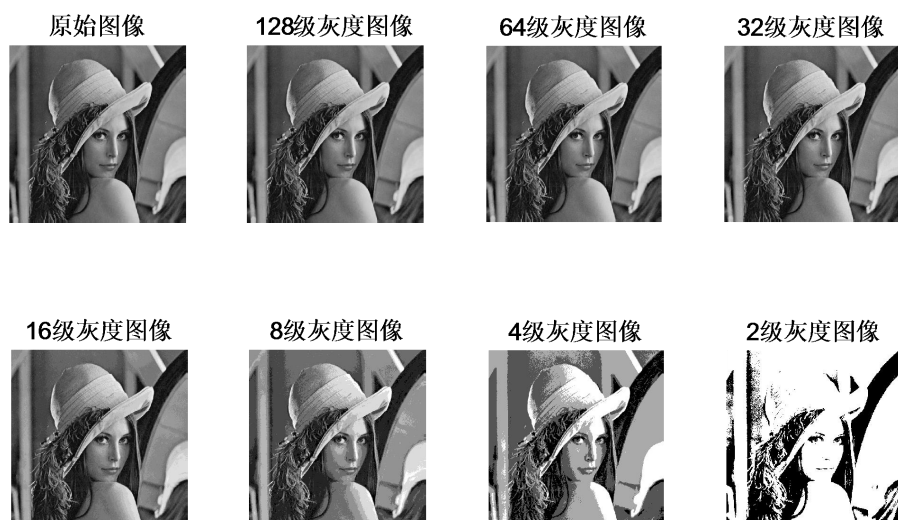
```
Filename: 'E:\数字图像处理\workspace1\7.bmp'
FileModDate: '03-Jun-2011 11:17:32'
FileSize: 1134
Format: 'bmp'
FormatVersion: 'Version 3 (Microsoft Windows 3.x)'
Width: 7
Height: 7
BitDepth: 8
ColorType: 'indexed'
FormatSignature: 'BM'
NumColormapEntries: 256
Colormap: [256×3 double]
RedMask: []
GreenMask: []
BlueMask: []
ImageDataOffset: 1078
BitmapHeaderSize: 40
NumPlanes: 1
CompressionType: 'none'
BitmapSize: 56
HorzResolution: 0
VertResolution: 0
NumColorsUsed: 0
NumImportantColors: 0
```

2. 降低图像灰度

2.1 问题分析

把 lena 512*512 图像灰度级逐级递减 8-1 显示，首先应该保持样本数恒定，而将灰度级数以 2 的整数次幂从 256 减少到 2。

2.2 实验结果



2.3 结果分析

由以上降低灰度所得到的 8 幅图可以看出，灰度级为 256、128、64、32、16 的几幅图基本上没有什么区别，而从 8 级开始图像开始出现较为明显的伪轮廓，这是由于数字图像在平滑区域灰度级数不足导致的。最后一张 2 级图像中仅有黑白两色，即灰度值 0 和 1。

3. 计算图像均值方差

3.1 问题分析

计算灰度图像“lena.bmp”的均值与方差。
均值即对图像中的每一点的灰度值求和后平均，其计算公式如下：

$$\text{aver} = \frac{1}{n * m} \sum_{x=1}^n \sum_{y=1}^m f(x, y)$$

方差即每一点灰度值与均值差的平方和后，求其平均，公式如下：

$$\text{var} = \frac{1}{n * m} \sum_{x=1}^n \sum_{y=1}^m (f(x, y) - \text{aver})^2$$

其中，n 和 m 是 bmp 图像的尺寸。

MATLAB 中提供图像均值函数 mean2 () 和方差函数 var ()，从而可以简单快捷的求出 bmp 图像的均值与方差。

3.2 实验结果

均值：

```
aver =
```

```
99.0512
```

方差：

```
sqr =
```

```
2.7960e+03
```

即：灰度图像“lena.bmp”的均值为 99.0512 方差为 2796。

4. 图像缩放

4.1 图像内插

内插是在图像的放大、收缩、旋转和几何校正等任务中广泛应用的基本工具。本质上，它是用已知数据来估计未知位置的数值的处理。下面简要介绍一下本实验所要用到的三种图像内插方法。

4.1.1 最近邻内插

最近邻内插法就是令变换后像素的灰度值等于距它最近的输入像素的灰度值。假设图像的缩放比为 r ，则

$$f_{new}(x, y) = f(\text{ceil}(\frac{x_0}{r}), \text{ceil}(\frac{y_0}{r}))$$

其中， $f_{new}(x, y)$ 是新坐标点 (x, y) 的灰度值， $f(x_0, y_0)$ 为原坐标点 (x_0, y_0) 灰度值。

4.1.2 双线性内插

双线性内插法并不是一种线性内插方法，它利用 4 个最近邻去估计给定位置的灰度。其公式如下：

$$f(x, y) = ax + by + cxy + d$$

其中，4 个系数可由点 (x, y) 的 4 个最近邻写出的未知方程确定。

4.1.3 双三次内插

双三次内插法是一种复杂度较高的方法，它利用 16 个最近邻去估计给定位置的灰度。其公式如下：

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

其中，16 个系数可由点 (x, y) 的 16 个最近邻写出的未知方程确定。

4.2 问题分析

把 lena 图像用近邻、双线性和双三次插值法 zoom 到 2048*2048。根据以上介绍的三种内插方法即可完成图像的放大。

MATLAB 中提供了图像缩放函数 `imresize(image, [m, n], 'method')` 可以方便快捷的利用三种内插方法实现 bmp 图像从 512*512 到 2048*2048 的放大。

其中 `image` 为输入的图像，`[m, n]` 为图像的尺寸，`'method'` 为所需的内插方法。

4.3 实验结果



对嘴唇部分局部放大后得到下图：



4.4 结果分析

对比以上三幅完整图很难发现什么差异，因此也无法判别三种方法的优劣。于是分别对三幅图进行局部放大，对比三幅图可发现由最近邻内插得到的图具有边缘锯齿和马赛克效应，在明暗交界处失真严重。双线性内插和双三次内插法得到的图很大程度上消除了边缘锯齿和马赛克效应，故其效果均优于最近邻内插法。

5. 图像坐标变换

5.1 仿射变换

坐标变换公式为：

$$(x, y) = T(v, w)$$

其中 (v, w) 为原图像中像素的坐标， (x, y) 为变换后图像中像素的坐标。

旋转变换矩阵如下：

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

水平偏移变换矩阵如下：

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

5.2 问题分析

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；

利用上面讲到的仿射变换可知，将 s_h 设置成 1.5 即可实现图像的水平偏移，

将 θ 设置为 30 即可实现图像旋转 30 度操作。然后进一步实现放大功能。

本试验中为了操作简单旋转操作直接在 MATLAB 中调用 `imrotate()` 函数，而水平偏移利用 `imtransform()` 函数。

5.3 实验结果

对两幅图分别水平偏移和旋转 30 度后的结果如下：

lena的错切



elain1的错切



lena旋转30度

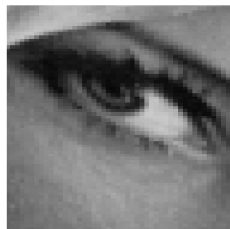


elain1旋转30度



对 lena 进行偏移旋转后放大得到的图像如下：

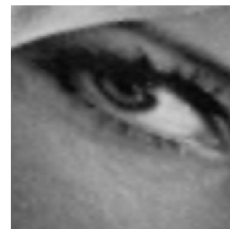
shear nearest



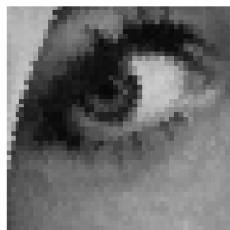
shear bilinear



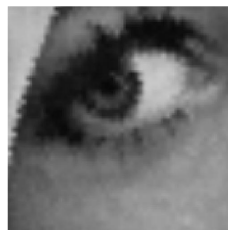
shear bicubic



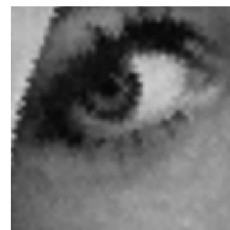
rotate nearest



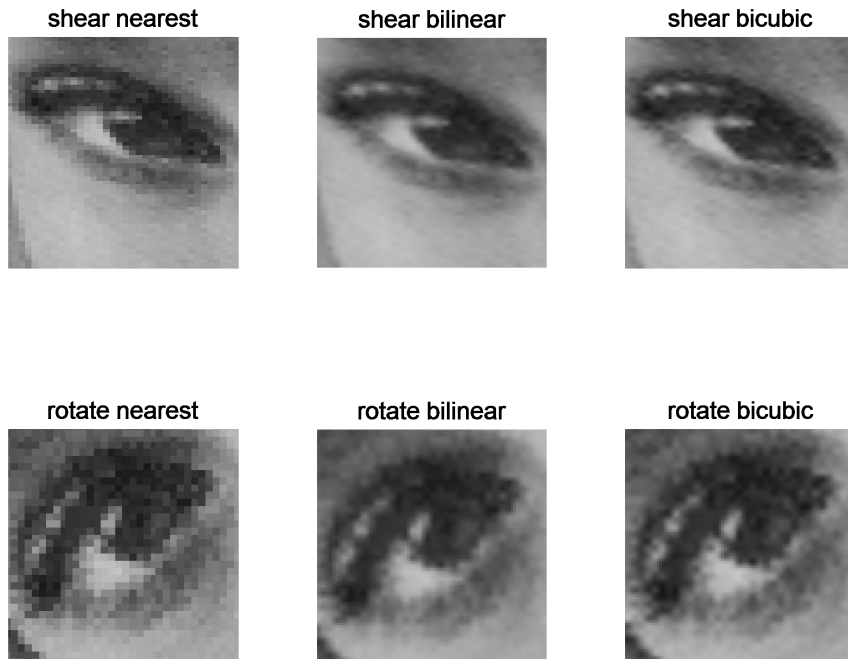
rotate bilinear



rotate bicubic



对 elain1 进行偏移旋转后放大得到的图像如下：



5.4 结果分析

根据实验得到的图像可以看出其结论与 4.4 结论相同，即最近邻内插得到的图具有边缘锯齿和马赛克效应，在明暗交界处失真严重。双线性内插和双三次内插法得到的图很大程度上消除了边缘锯齿和马赛克效应，故其效果均优于最近邻内插法。

附录

参考文献

- [1] Rafael C. Gonzalez, Richard E. Woods. 数字图像处理 阮秋琦, 阮宇智等译, 北京: 电子工业出版社, 2016.
- [2] 赵小川 MATLAB 图像处理—程序实现与模块化仿真, 北京: 北京航空航天大学出版社, 2014.
- [3] Kenneth R. Castleman. 数字图像处理 朱志刚, 林学间, 石定机, 等译, 北京: 电子工业出版社, 2001.

源代码

主程序

```
clc
I=imread('7.bmp');
Im = imread('lena.bmp');
Im2 = imread('elain1.bmp');

%读取 bmp 图像信息
imfinfo('7.bmp')

%逐级递减的灰度图像
figure(1);
subplot(2,4,1);
imshow(Im);
title('原始图像');

for i = 1 : 1 : 7
    subplot(2,4,i+1);
    imshow(reduce_gray(Im,i),[0,power(2,8-i)-1]);
    title(sprintf('%d 级灰度图像',power(2,8-i)));
end

%快速计算图像均值与方差的方法
Im1 = double(Im);
aver = mean2(Im1)
sqr = var(Im1(:))
```



```

% aver = average (Im); %计算图像的均值

% sqr = variance (Im); %计算图像的方差

%对 lena 图像放大

figure(2);
subplot(1,3,1);
lena_near=imresize(Im,[2048,2048],'nearest');
imshow(lena_near);
title('nearest');

subplot(1,3,2);
lena_linear=imresize(Im,[2048,2048],'bilinear');
imshow(lena_linear);
title('bilinear');

subplot(1,3,3);
lena_cubic=imresize(Im,[2048,2048],'bicubic');
imshow(lena_cubic);
title('bicubic');

%分别对 lena 和 elain1 进行水平错切和旋转 30 度

tform=maketform('affine',[1,0.5,0;0,1,0;0,0,1]);
Lena1=imtransform(Im,tform);
Elain11=imtransform(Im2,tform);

Lena2=imrotate(Im,30);
Elain12=imrotate(Im2,30);

figure(3);
subplot(2,2,1);
imshow(Lena1);

title('lena 的错切');

subplot(2,2,2);
imshow(Elain11);

title('elain1 的错切');

subplot(2,2,3);

```

```
imshow(Lena2);

title('lena 旋转 30 度');


subplot(2,2,4);
imshow(Elain12);

title('elain1 旋转 30 度');


figure(4);
subplot(2,3,1);
L11=imresize(Lena1,[2048,2048],'nearest');
imshow(L11);
title('shear nearest');


subplot(2,3,2);
L12=imresize(Lena1,[2048,2048],'bilinear');
imshow(L12);
title('shear bilinear');


subplot(2,3,3);
L13=imresize(Lena1,[2048,2048],'bicubic');
imshow(L13);
title('shear bicubic');


subplot(2,3,4);
L21=imresize(Lena2,[2048,2048],'nearest');
imshow(L21);
title('rotate nearest');


subplot(2,3,5);
L22=imresize(Lena2,[2048,2048],'bilinear');
imshow(L22);
title('rotate bilinear');


subplot(2,3,6);
L23=imresize(Lena2,[2048,2048],'bicubic');
imshow(L23);
title('rotate bicubic');


figure(5);
subplot(2,3,1);
E11=imresize(Elain11,[2048,2048],'nearest');
imshow(E11);
```

```

title('shear nearest');

subplot(2,3,2);
E12=imresize(E1ain11,[2048,2048],'bilinear');
imshow(E12);
title('shear bilinear');

subplot(2,3,3);
E13=imresize(E1ain11,[2048,2048],'bicubic');
imshow(E13);
title('shear bicubic');

subplot(2,3,4);
E21=imresize(E1ain12,[2048,2048],'nearest');
imshow(E21);
title('rotate nearest');

subplot(2,3,5);
E22=imresize(E1ain12,[2048,2048],'bilinear');
imshow(E22);
title('rotate bilinear');

subplot(2,3,6);
E23=imresize(E1ain12,[2048,2048],'bicubic');
imshow(E23);
title('rotate bicubic');

```

灰度降低函数

```

function new_image = reduce_gray(image , level)

[m,n]=size(image);
new_image=zeros(m,n);
de_level = uint8(power(2 , level));
for x=1:m
    for y=1:n
        new_image(x,y)=floor(image(x,y)/de_level);
    end
end
return

```