

数字图像处理

第一次作业

李明哲

自动化 64

2160504096

摘要

本次实验有以下几个内容：首先通过查看一张 7*7 的 bmp 图像的二进制数据信息了解了 bmp 图像格式，了解了四个部分数据的含义。其次，把给定图片灰度级按 8-1 显示，对不同图像灰度级进行了比对，得出结论：灰度级越高，图片越逼真。同时计算了图像方差 (std2 函数) 和均值 (mean 函数)。并运用 imresize 函数进行近邻、双线性插值和双三次插值，得到了 3 个放大后的图片。最后利用 imtransform 函数把图像进行了仿射变换，例如旋转变换和水平 shear 变换，并对图像进行了插值。本次基础实验，为以后数字图像处理的学习打下了良好的基础。

一. Bmp 图像格式简介：

1. 简介

Bmp(Bitmap-File) 图形文件是 Windows 采用的图形文件格式，在 Windows 环境下运行的所有图像处理软件都支持 bmp 图象文件格式。Windows 系统内部各图像绘制操作都是以 bmp 为基础的。Windows 3.0 以前的 bmp 图文件格式与显示设备有关，因此把这种 bmp 图象文件格式称为设备相关位图 DDB(device-dependent bitmap) 文件格式。Windows 3.0 以后的 bmp 图象文件与显示设备无关，因此把这种 bmp 图象文件格式称为设备无关位图 DIB(device-independent bitmap) 格式（注：Windows 3.0 以后，在系统中仍然存在 DDB 位图，象 BitBlt() 这种函数就是基于 DDB 位图的，只不过如果你想将图像以 BMP 格式保存到磁盘文件中时，微软极力推荐你以 DIB 格式保存），目的是为了让 Windows 能够在任何类型的显示设备上显示所存储的图象。BMP 位图文件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)。

2. 文件结构

位图文件可看成由 4 个部分组成：位图文件头、位图信息头、彩色表和定义位图的字节阵列。

3. 格式解读

用 Ultra Edit 打开 bmp 二进制文件后，其中表示文件大小、宽高等数值的字节是从低字节到高字节排序，而字节本身又是高位到低位的两位 16 进制数。

a. 位图文件头(14 字节)

BMP 文件头数据结构含有 BMP 文件的类型、文件大小和位图起始位置等信息。其结构定义如下：

```
typedef struct tagBITMAPFILEHEADER
{
    WORD bfType; // 位图文件的类型，必须为 BM(0-1 字节)
    DWORD bfSize; // 位图文件的大小，以字节为单位(2-5 字节)
    WORD bfReserved1; // 位图文件保留字，必须为 0(6-7 字节)
    WORD bfReserved2; // 位图文件保留字，必须为 0(8-9 字节)
    DWORD bfOffBits; /* 位图数据的起始位置，以相对于位图(10-13 字节)
                      */ 文件头的偏移量表示，以字节为单位
} BITMAPFILEHEADER;
```

- 1) 0-1: 图像文件头。424Dh='BM', 表示是 Windows 支持的 BMP 格式。
- 2) 2-5: 整个文件大小。6E04 0000, 为 0000046Eh=1134。
- 3) 6-9: 位图文件保留字, 必须设置为 0。
- 4) 10-13: 从文件开始到位图数据之间的偏移量。3604 0000, 为 00000436h=1078, 上面的文件头就是 539 字=1078 字节。

b. 位图信息头

000Eh 是位图信息头的长度, 用来描述位图的颜色、压缩方法等。图中为 28h - Windows 3.1x, 95, NT, ...

【注】在 Windows95、98、2000 等操作系统中, 位图信息头的长度并不一定是 28h, 因为微软已经制定出了新的 BMP 文件格式, 其中的信息头结构变化比较大, 长度加长。所以最好不要直接使用常数 28h, 而是应该从具体的文件中读取这个值。这样才能确保程序的兼容性。

0012h : 位图的宽度。0700 0000 为 0000 0007h=7, 以像素为单位。

0016h : 位图的高度。0700 0000 为 0000 0007h=7, 以像素为单位

【注】这个值除了用于描述图像的高度之外, 它还有另一个用处, 就是指明该图像是倒向的位图, 还是正向的位图。如果该值是一个正数, 说明图像是倒向的, 如果该值是一个负数, 则说明图像是正向的。大多数的 BMP 文件都是倒向的位图, 也就是时, 高度值是一个正数。(注: 当高度值是一个负数时 (正向图像), 图像将不能被压缩 (也就是说 biCompression 成员将不能是 BI_RLE8 或 BI_RLE4)。

001Ah : 位图的位面数。0100 为 0001h=1 (注: 该值将总是 1)

001Ch : 每个像素的位数。0800 为 0008h=8。表示该图为 256 色位图。

001Eh : 压缩说明。00h=0, 表示不压缩 (使用 BI_RGB 表示)

0022h : 用字节数表示的位图数据的大小。该数必须是 4 的倍数。3800 0000 为 0000 0038h=56。

0026h : 用像素/米表示的水平分辨率。0000 0000h=0。

002Ah : 用像素/米表示的垂直分辨率。0000 0000h=0。

002Eh : 位图使用的颜色数。0000 0000h=0

0032h : 指定重要的颜色数。0000 0000h=0, 表示所有颜色都一样重要。

c. 彩色表

颜色表用于说明位图中的颜色, 它有若干个表项, 每一个表项是一个 RGBQUAD 类型的结构, 定义一种颜色。颜色表中 RGBQUAD 结构数据的个数由 biBitCount 来确定:

当 biBitCount=1, 4, 8 时, 分别有 2, 16, 256 个表项;

当 biBitCount=24 时, 没有颜色表项。图中 biBitCount=8, 说明有 256 个表项。

RGBQUAD 结构的定义如下:

```
typedef struct tagRGBQUAD
{
    BYTE rgbBlue; // 蓝色的亮度(值范围为 0-255)
    BYTE rgbGreen; // 绿色的亮度(值范围为 0-255)
    BYTE rgbRed; // 红色的亮度(值范围为 0-255)
    BYTE rgbReserved; // 保留, 必须为 0
} RGBQUAD;
```

调色板数据根据 BMP 版本的不同而不同。根据 Palette $N * 4$ byte 调色板规范，对于调色板中的每个表项，这 4 个字节用下述方法来描述 RGB 的值：1 字节用于蓝色分量，1 字节用于绿色分量，1 字节用于红色分量，1 字节用于填充符（设置为 0）。

对于 24-位真彩色图像就不使用彩色表，因为位图中的 RGB 值就代表了个像素的颜色。但是 16 位 r5g6b5 位域彩色图像需要彩色表，将掩码跟像素值进行“与”运算再进行移位操作就可以得到各色分量值。看看掩码，就可以明白事实上在每个像素值的两个字节 16 位中，按从高到低取 5、6、5 位分别就是 r、g、b 分量值。取出分量值后把 r、g、b 值分别乘以 8、4、8 就可以补齐第个分量为一个字节，再把这三个字节按 rgb 组合，放入存储器（同样要反序），就可以转换为 24 位标准 BMP 格式了。

d. 定义位图的字节阵列

位图数据根据 BMP 版本及调色板尺寸的不同而不同。该域的大小取决于压缩方法及图像的尺寸和图像的位深度，它包含所有的位图数据字节，这些数据可能是彩色调色板的索引号，也可能是实际的 RGB 值，这将根据图像信息头中的位深度值来决定。

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；

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，图中，位图数据位置范围为：0000 0436h-0000 046Dh。

二进制信息截图如下：

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
00000000h:	42	4D	6E	04	00	00	00	00	00	36	04	00	00	28	00	;	B Mn.....6...(. 00000010h:	00	00	07	00	00	00	07	00	00	00	01	00	08	00	00	00	; 00000020h:	00	00	38	00	00	00	00	00	00	00	00	00	00	00	00	00	;	..8..... 00000030h:	00	00	00	00	00	00	00	00	00	01	01	01	00	02	02	; 00000040h:	02	00	03	03	03	00	04	04	04	00	05	05	05	00	06	06	; 00000050h:	06	00	07	07	07	00	08	08	08	00	09	09	09	00	0A	0A	; 00000060h:	0A	00	0B	0B	0B	00	0C	0C	0C	00	0D	0D	0D	00	0E	0E	; 00000070h:	0E	00	0F	0F	0F	00	10	10	10	00	11	11	11	00	12	12	; 00000080h:	12	00	13	13	13	00	14	14	14	00	15	15	15	00	16	16	; 00000090h:	16	00	17	17	17	00	18	18	18	00	19	19	19	00	1A	1A	; 000000a0h:	1A	00	1B	1B	1B	00	1C	1C	1C	00	1D	1D	1D	00	1E	1E	; 000000b0h:	1E	00	1F	1F	1F	00	20	20	20	00	21	21	21	00	22	22	;!!!."" 000000c0h:	22	00	23	23	23	00	24	24	24	00	25	25	25	00	26	26	;	".###.\$\$.%%.&& 000000d0h:	26	00	27	27	27	00	28	28	28	00	29	29	29	00	2A	2A	;	&.''.(((.)).** 000000e0h:	2A	00	2B	2B	2B	00	2C	2C	2C	00	2D	2D	2D	00	2E	2E	;	*.+++.,.,---... 000000f0h:	2E	00	2F	2F	2F	00	30	30	30	00	31	31	31	00	32	32	;	..///.000.111.22 00000100h:	32	00	33	33	33	00	34	34	34	00	35	35	35	00	36	36	;	2.333.444.555.66 00000110h:	36	00	37	37	37	00	38	38	38	00	39	39	39	00	3A	3A	;	6.777.888.999.: 00000120h:	3A	00	3B	3B	3B	00	3C	3C	3C	00	3D	3D	3D	00	3E	3E	;	.;.;.;.<<<===.>> 00000130h:	3E	00	3F	3F	3F	00	40	40	40	00	41	41	41	00	42	42	;	>.???.@@@.AAA.BB 00000140h:	42	00	43	43	43	00	44	44	44	00	45	45	45	00	46	46	;	B.CCC.DDD.EEE.FF 00000150h:	46	00	47	47	47	00	48	48	48	00	49	49	49	00	4A	4A	;	F.GGG.HHH.III.JJ 00000160h:	4A	00	4B	4B	4B	00	4C	4C	4C	00	4D	4D	4D	00	4E	4E	;	J.KKK.LLL.MMM.NN 00000170h:	4E	00	4F	4F	4F	00	50	50	50	00	51	51	51	00	52	52	;	N.OOO.PPP.QQQ.RR 00000180h:	52	00	53	53	53	00	54	54	54	00	55	55	55	00	56	56	;	R.SSS.TTT.UUU.VV 00000190h:	56	00	57	57	57	00	58	58	58	00	59	59	59	00	5A	5A	;	V.WWW.XXX.YYY.ZZ 000001a0h:	5A	00	5B	5B	5B	00	5C	5C	5C	00	5D	5D	5D	00	5E	5E	;	Z.[[.\\.\.]].^^ 000001b0h:	5E	00	5F	5F	5F	00	60	60	60	00	61	61	61	00	62	62	;	^._.````.aaa.bb 000001c0h:	62	00	63	63	63	00	64	64	64	00	65	65	65	00	66	66	;	b.ccc.ddd.eee.ff 000001d0h:	66	00	67	67	67	00	68	68	68	00	69	69	69	00	6A	6A	;	f.ggg.hhh.iii.jj 000001e0h:	6A	00	6B	6B	6B	00	6C	6C	6C	00	6D	6D	6D	00	6E	6E	;	j.kkk.lll.mmm.nn 000001f0h:	6E	00	6F	6F	6F	00	70	70	70	00	71	71	71	00	72	72	;	n.ooo.ppp.qqq.rr 00000200h:	72	00	73	73	73	00	74	74	74	00	75	75	75	00	76	76	;	r.sss.ttt.uuu.vv 00000210h:	76	00	77	77	77	00	78	78	78	00	79	79	79	00	7A	7A	;	v.www.xxx.yyy.zz 00000220h:	7A	00	7B	7B	7B	00	7C	7C	7C	00	7D	7D	7D	00	7E	7E	;	z.{{{. .}}}.~~ 00000230h:	7E	00	7F	7F	7F	00	80	80	80	00	81	81	81	00	82	82	;	~.!!!.€€€.三?脩 00000240h:	82	00	83	83	83	00	84	84	84	00	85	85	85	00	86	86	;	?傻?剥?厖?俺 00000250h:	86	00	87	87	87	00	88	88	88	00	89	89	89	00	8A	8A	;	?噠?盜?塿?婀 00000260h:	8A	00	8B	8B	8B	00	8C	8C	8C	00	8D	8D	8D	00	8E	8E	;	?媼?瘡?峒?帛

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
00000260h:	8A	00	8B	8B	8B	00	8C	8C	8C	00	8D	8D	8D	00	8E	8E	;	?媼?瘡?峒?帛 00000270h:	8E	00	8F	8F	8F	00	90	90	90	00	91	91	91	00	92	92	;	?寿?悖?憫?挾 00000280h:	92	00	93	93	93	00	94	94	94	00	95	95	95	00	96	96	;	?搗?敏?噉?枷 00000290h:	96	00	97	97	97	00	98	98	98	00	99	99	99	00	9A	9A	;	?棗?鳩?檀?旻 000002a0h:	9A	00	9B	9B	9B	00	9C	9C	9C	00	9D	9D	9D	00	9E	9E	;	?涑?潜?溟?瀋 000002b0h:	9E	00	9F	9F	9F	00	A0	A0	A0	00	A1	A1	A1	00	A2	A2	;	?煙?賤? ?ii 000002c0h:	A2	00	A3	A3	A3	00	A4	A4	A4	00	A5	A5	A5	00	A6	A6	;	?#?い?ッ?Z 000002d0h:	A6	00	A7	A7	A7	00	A8	A8	A8	00	A9	A9	A9	00	AA	AA	;	?Ė?à?---? 000002e0h:	AA	00	AB	AB	AB	00	AC	AC	AC	00	AD	AD	AD	00	AE	AE	;	?驱?郢?綜?簪 000002f0h:	AE	00	AF	AF	AF	00	B0	B0	B0	00	B1	B1	B1	00	B2	B2	;	?绝?鞍?北?膊 00000300h:	B2	00	B3	B3	B3	00	B4	B4	B4	00	B5	B5	B5	00	B6	B6	;	?吵?创?档?抖 00000310h:	B6	00	B7	B7	B7	00	B8	B8	B8	00	B9	B9	B9	00	BA	BA	;	?販?父?枸?汉 00000320h:	BA	00	BB	BB	BB	00	BC	BC	BC	00	BD	BD	BD	00	BE	BE	;	?換?技?澆?揪 00000330h:	BE	00	BF	BF	BF	00	C0	C0	C0	00	C1	C1	C1	00	C2	C2	;	?靠?览?亮?侶 00000340h:	C2	00	C3	C3	C3	00	C4	C4	C4	00	C5	C5	C5	00	C6	C6	;	?妹?哪?排?破 00000350h:	C6	00	C7	C7	C7	00	C8	C8	C8	00	C9	C9	C9	00	CA	CA	;	?乔?热?縊?适 00000360h:	CA	00	CB	CB	CB	00	CC	CC	CC	00	CD	CD	CD	00	CE	CE	;	?怱?燙?屯?挝 00000370h:	CE	00	CF	CF	CF	00	D0	D0	D0	00	D1	D1	D1	00	D2	D2	;	?舷?行?艇?乙 00000380h:	D2	00	D3	D3	D3	00	D4	D4	D4	00	D5	D5	D5	00	D6	D6	;	?佑?栽?照?种 00000390h:	D6	00	D7	D7	D7	00	D8	D8	D8	00	D9	D9	D9	00	DA	DA	;	?鬟?刎?儻?譚 000003a0h:	DA	00	DB	DB	DB	00	DC	DC	DC	00	DD	DD	DD	00	DE	DE	;	?圻?苘?葦?椰 000003b0h:	DE	00	DF	DF	DF	00	E0	E0	E0	00	E1	E1	E1	00	E2	E2	;	?哌?噲?後?忖 000003c0h:	E2	00	E3	E3	E3	00	E4	E4	E4	00	E5	E5	E5	00	E6	E6	;	?沔?滢?邈?駘 000003d0h:	E6	00	E7	E7	E7	00	E8	E8	E8	00	E9	E9	E9	00	EA	EA	;	?玦?梃?殫?覲 000003e0h:	EA	00	EB	EB	EB	00	EC	EC	EC	00	ED	ED	ED	00	EE	EE	;	?脛?袄?肭?铤 000003f0h:	EE	00	EF	EF	EF	00	F0	F0	F0	00	F1	F1	F1	00	F2	F2	;	?鐐?痧?縶?蝟 00000400h:	F2	00	F3	F3	F3	00	F4	F4	F4	00	F5	F5	F5	00	F6	F6	;	?僕?趙?紹?蝟 00000410h:	F6	00	F7	F7	F7	00	F8	F8	F8	00	F9	F9	F9	00	FA	FA	;	?駟? ? ? 00000420h:	FA	00	FB	FB	FB	00	FC	FC	FC	00	FD	FD	FD	00	FE	FE	;	? ? ? ? 00000430h:	FE	00	FF	FF	FF	00	67	63	64	54	56	62	62	00	62	65	;	? .gcdTVbb.be 00000440h:	66	56	45	47	5F	00	61	5C	5B	63	48	47	52	00	58	4B	;	fVEG_.a\[cHGR.XK 00000450h:	55	65	5A	5B	46	00	68	47	3F	69	5D	4C	2A	00	61	59	;	UeZ[F.hG?i]L*.aY 00000460h:	5A	5F	47	28	45	00	52	52	49	3B	37	50	5A	00			;	Z_G(E.RRI;7PZ.

二. lena 512*512 图像灰度级逐级递减 8-1 显示:

源代码: Gray_level.m

```
clear all;
clc;
I_8 = imread('lena.bmp');
for i = 1:512
    for j = 1:512
        I_7(i,j) = uint8(I_8(i,j)/2);
        I_6(i,j) = uint8(I_8(i,j)/4);
        I_5(i,j) = uint8(I_8(i,j)/8);
        I_4(i,j) = uint8(I_8(i,j)/16);
        I_3(i,j) = uint8(I_8(i,j)/32);
        I_2(i,j) = uint8(I_8(i,j)/64);
        I_1(i,j) = uint8(I_8(i,j)/128);
    end
end

subplot(2,4,1);imshow(I_8,[0,255]);title('8 灰度');

subplot(2,4,2);imshow(I_7,[0,127]);title('7 灰度');

subplot(2,4,3);imshow(I_6,[0,63]);title('6 灰度');

subplot(2,4,4);imshow(I_5,[0,31]);title('5 灰度');

subplot(2,4,5);imshow(I_4,[0,15]);title('4 灰度');

subplot(2,4,6);imshow(I_3,[0,7]);title('3 灰度');

subplot(2,4,7);imshow(I_2,[0,3]);title('2 灰度');

subplot(2,4,8);imshow(I_1,[0,1]);title('1 灰度');
```

结果截图:



结论：灰度级越高，图像色彩越逼真。

三. 计算 lena 图像的均值方差：

源代码：Mean_var.m

```
clear all;
clc;
I_origin = imread('lena.bmp');
Mean_image = mean2(I_origin);
Var_image = std2(I_origin)^2;
fprintf('The mean is %f.\nThe var
is %f.',Mean_image,Var_image)
```

运行结果：

The mean is 99.051216.

The var is 2796.042505.

四. 把 lena 图像用近邻、双线性性和双三次插值法 zoom 到 2048*2048:

源代码：Inter_image.m

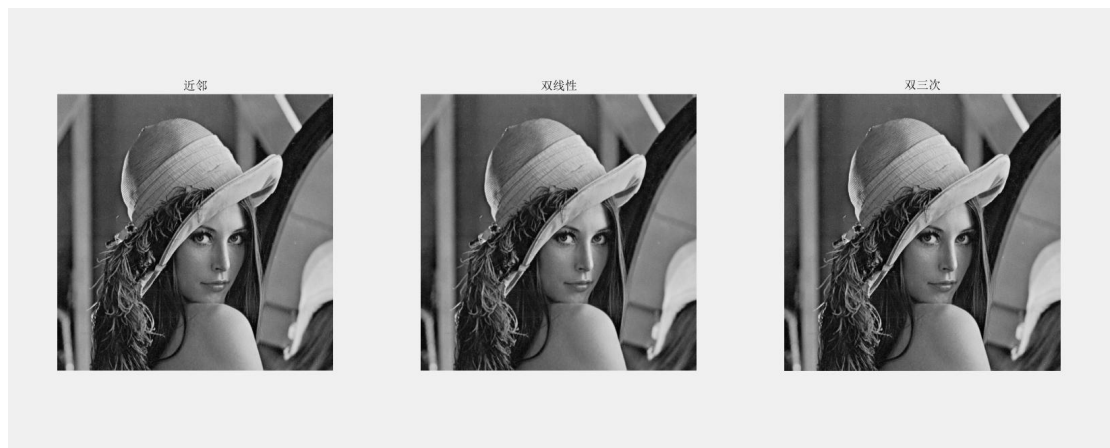
```
clear all;
clc;
I_origin = imread('lena.bmp');
I_near = imresize(I_origin,4);
I_bili = imresize(I_origin,4,'bilinear');
I_bicu = imresize(I_origin,4,'bicubic');

subplot(1,3,1);imshow(I_near);title('近邻');

subplot(1,3,2);imshow(I_bili);title('双线性');

subplot(1,3,3);imshow(I_bicu);title('双三次');
```

结果截图：



结论：从结果看，三种插值方法得出的结果差别不大，后两种插值方法结果稍好。

五. 把 lena 和 elain 图像分别进行水平 shear 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:

源代码: Shear_image.m

```
clear all;
clc;
I_ori_lena = imread('lena.bmp');
I_ori_elain = imread('elain1.bmp');
type = 'affine';
form = [1,1.5,0;0,1,0;0,0,1];
Transform = maketform(type,form);
I_shear_lena = imtransform(I_ori_lena,Transform);
I_shear_elain = imtransform(I_ori_elain,Transform);
figure(1);
subplot(1,2,1);imshow(I_shear_lena);title('shear lena');
subplot(1,2,2);imshow(I_shear_elain);title('shear elain');
I_near_lena = imresize(I_shear_lena,[2048,2048]);
I_bili_lena =
imresize(I_shear_lena,[2048,2048],'bilinear');
I_bicu_lena =
imresize(I_shear_lena,[2048,2048],'bicubic');
figure(2);

subplot(1,3,1);imshow(I_near_lena);title('近邻 lena');

subplot(1,3,2);imshow(I_bili_lena);title('双线性 lena');

subplot(1,3,3);imshow(I_bicu_lena);title('双三次 lena');

I_near_elain = imresize(I_shear_elain,[2048,2048]);
I_bili_elain =
imresize(I_shear_elain,[2048,2048],'bilinear');
```



```

I_bicu_elain =
imresize(I_shear_elain,[2048,2048],'bicubic');
figure(3);

subplot(1,3,1);imshow(I_near_elain);title('近邻 elain');

subplot(1,3,2);imshow(I_bili_elain);title('双线性 elain');

subplot(1,3,3);imshow(I_bicu_elain);title('双三次 elain');

Rotate_image.m
clear all;
clc;
I_ori_lena = imread('lena.bmp');
I_ori_elain = imread('elain1.bmp');
I_rotate_lena = imrotate(I_ori_lena,30);
I_rotate_elain = imrotate(I_ori_elain,30);
figure(1);
subplot(1,2,1);imshow(I_rotate_lena);title('rotate lena');
subplot(1,2,2);imshow(I_rotate_elain);title('rotate
elain');
I_near_lena = imresize(I_rotate_lena,[2048,2048]);
I_bili_lena =
imresize(I_rotate_lena,[2048,2048],'bilinear');
I_bicu_lena =
imresize(I_rotate_lena,[2048,2048],'bicubic');
figure(2);

subplot(1,3,1);imshow(I_near_lena);title('近邻 lena');

subplot(1,3,2);imshow(I_bili_lena);title('双线性 lena');

subplot(1,3,3);imshow(I_bicu_lena);title('双三次 lena');

I_near_elain = imresize(I_rotate_elain,[2048,2048]);
I_bili_elain =
imresize(I_rotate_elain,[2048,2048],'bilinear');
I_bicu_elain =
imresize(I_rotate_elain,[2048,2048],'bicubic');
figure(3);

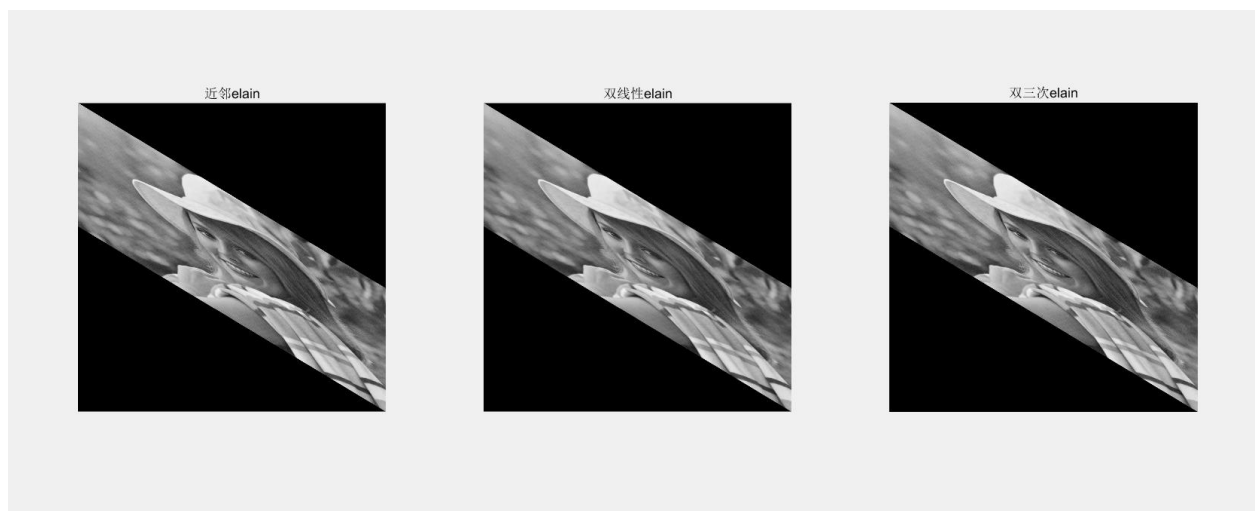
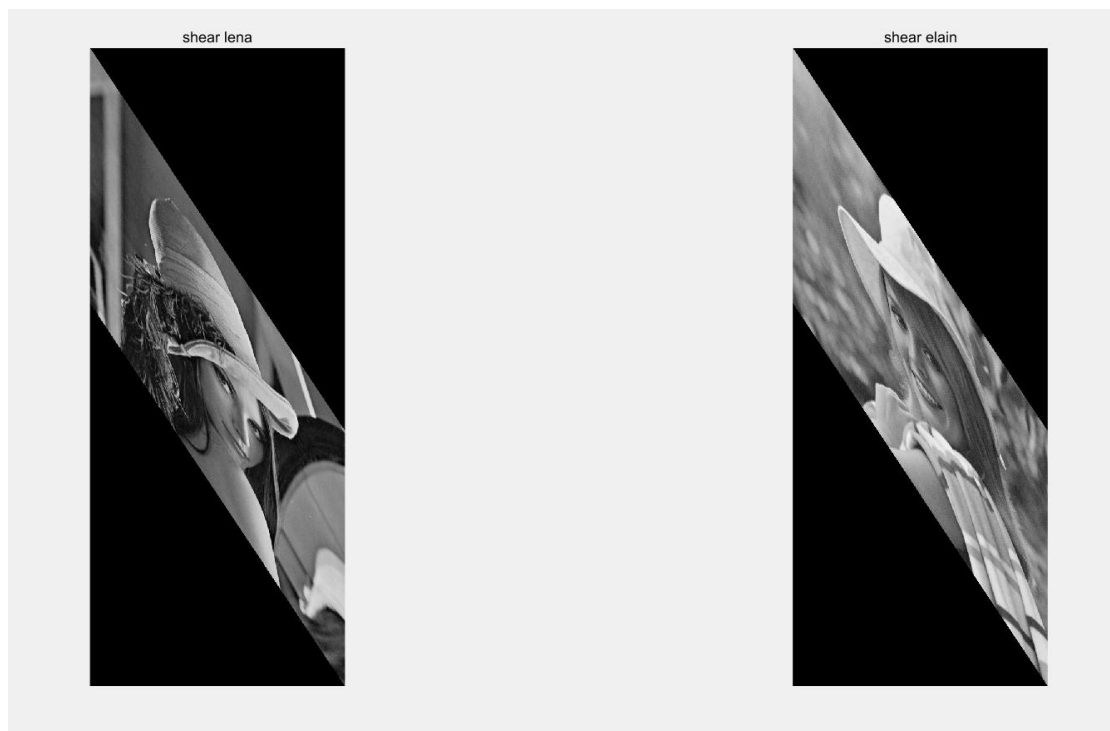
subplot(1,3,1);imshow(I_near_elain);title('近邻 elain');

subplot(1,3,2);imshow(I_bili_elain);title('双线性 elain');

subplot(1,3,3);imshow(I_bicu_elain);title('双三次 elain');

```

结果截图：





结论：对两个图像成功进行了 Shear（参数 1.5）变换和旋转变换。三种差值的结果并不大。

六. 心得体会：

本次数字图像基础实验让我对课上学习的知识进行了一次实践。了解了 Matlab 中有关数字图像处理的相关基础函数,包括 imread、imshow、imtransform 等函数。在使用这些函数时,一定要仔细阅读它的使用文档,确保每一个参数都是符合标准的,比如在进行 affine 仿射变换时,对变换矩阵的形式要求很严格,最后一列必须为 0、0、1。

同时，我也对函数内部的算法进行了一定的了解，了解了函数封装内部的算法，比如插值方法。通过这次作业，我对数字图像处理有了更好地认识，也掌握了一些程序调试的方法。