



南京邮电大学  
Nanjing University of Posts and Telecommunications

通信与信息工程学院

2015/2016 学年第一学期

# 实 验 报 告

实验课程名称 信息技术综合实验-图像实验

专 业 电子信息工程

学 生 学 号 Q12010221

学 生 姓 名 张 磊

指 导 教 师 庄文芹

指 导 单 位 图像与广播电视系

日 期: 2015 年 09 月 22 日

# 实验名称：数字图像处理实验

## 一、实验目的：

- 1、认识数字图像处理系统构成，了解系统主要功能以及基本性能
- 2、了解数字图像信号的采集方法和存储格式，了解图像文件格式和数据量之间的关系。
- 3、了解图像处理软件的额相关知识，掌握图像处理的初步操作方法。

## 二、实验内容：

- 1、使用数字化设备采集数字图像信息。
- 2、常用图像压缩格式之间转换。
- 3、数字图像的读取、显示和基本处理操作。

## 三、实验方法及编程：

读取

```
X1=imread('lena','BMP')
subplot(2,2,1)
imshow(X1,256)
title('数字图像1')
colorbar
```

相加

```
X1=imread('lena','BMP')
subplot(2,2,1)
imshow(X1,256)
title('数字图像1')
colorbar
X2=imread('Couple','BMP')
subplot(2,2,2)
imshow(X2,256)
title('数字图像2')
colorbar
Z=imadd(X1,X2);
title('1、2图像相加')
subplot(2,2,3)
imshow(Z,256)
```

旋转

```
X=imread('lena','BMP')
subplot(2,2,1)
imshow(X,256)
```

```
title('数字图像1')
colorbar
J=imrotate(X,180)
subplot(2,2,2)
imshow(J,256)
title('旋转之后的图像')
colorbar
```

#### 数字图像截取

```
X=imread('lena','BMP')
subplot(2,2,1)
imshow(X,256)
title('数字图像1')
colorbar
J=imcrop
subplot(2,2,2)
imshow(J,256)
title('截取之后图像')
colorbar
```

## 四、实验结果及分析：

### 1、实验结果：

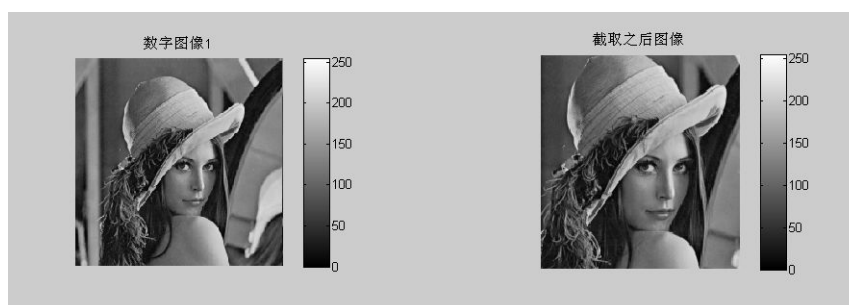
#### 数字图像的读取



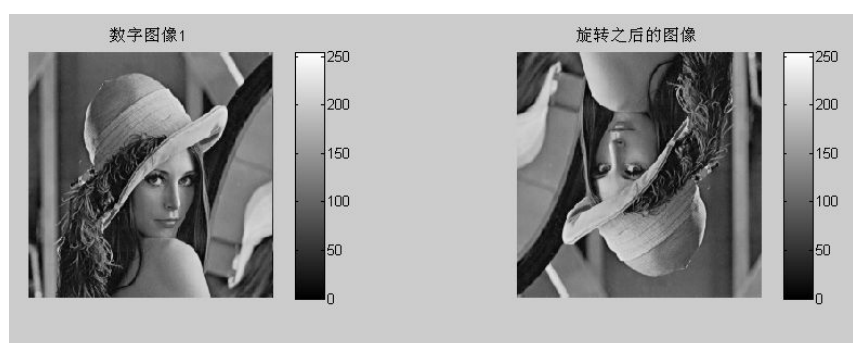
#### 数字图像的相加



数字图像的截取



数字图像的旋转



## 2、图像实验课程小结和思考

实验一，相对而言还是比较简单的，书中给出了我们在做实验过程中所需要的一些程序，我们只需要敲进电脑，然后对图像进行分析就行了。在之前我们上图像这些课程的时候，只是很系统的，很模糊，笼统的讲解了一下抽象的知识，这是我们第一次真正意义上的进行图像的实验，在我们的手中，图像进行了相加，截取或者其他的一些改变，让我们对图像的处理的概念更加深入的了解，对图像这门课程也有了更进一步的深层次的认识。

# 实验名称：数字图像增强实验

## 一、实验目的：

1. 建立数字图像处理的概念，通过练习，加深对图像处理基础知识的认识和理解
2. 了解和熟悉常见的几种图像增强技术如图像灰度变换、图像平滑、图像边缘提取等基本原理和基本方法，并观察上述方法的图像处理效果。

## 二、实验内容：

1. 图像灰度变换
2. 图像直方图均衡
3. 图像的平滑滤波
4. 图像的锐化处理

## 三、实验方法及编程：

灰度变化

```
X=imread('VAX256','BMP')
subplot(2,2,1)
imshow(X,256)
title('数字图像1')
colorbar
J=imadjust(X,[0 0.4],[0 0.8])
subplot(2,2,2)
imshow(J,256)
title('灰度变换后的图像')
colorbar
%low=stretchlim(J)
K=imadjust(J,[0 1],[0 0.5])
subplot(2,2,3)
imshow(K,256)
title('反变换后的图像')
colorbar
```

直方图和均衡化后的直方图

```
X=imread('VAX256','BMP')
subplot(2,2,1)
imshow(X,256)
title('数字图像1')
colorbar
subplot(2,2,2)
imhist(X)
```

```
title('直方图')
K=histeq(X,128)
subplot(2,2,3)
imshow(K,256)
title('均衡化后图像')
colorbar
subplot(2,2,4)
imhist(K)
title('均衡化后的直方图')
```

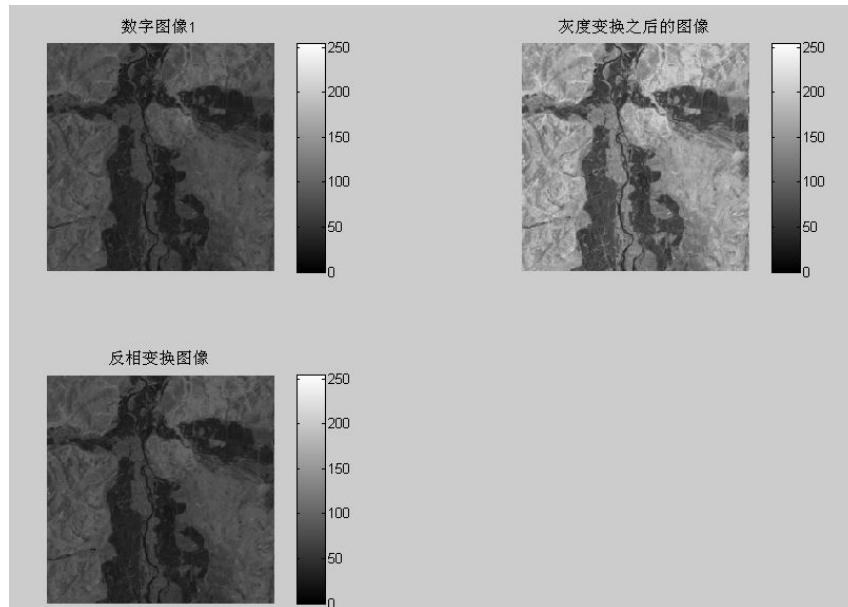
### 滤波

```
X=imread('lena','BMP')
J=imnoise(X,'salt & pepper',0.05)
subplot(3,3,1)
imshow(J,256)
title('椒盐噪声之后图像')
h1=ones(3,3)/(3*3)
h2=ones(5,5)/(5*5)
h3=ones(7,7)/(7*7)
K1=imfilter(J,h1)
subplot(3,3,4)
imshow(K1,256)
title('均值滤波邻域尺度3')
K2=imfilter(J,h2)
subplot(3,3,5)
imshow(K2,256)
title('均值滤波邻域尺度5')
K3=imfilter(J,h3)
subplot(3,3,6)
imshow(K3,256)
title('均值滤波邻域尺度7')
K4=medfilt2(J,[3 3])
subplot(3,3,7)
imshow(K4,256)
title('中值滤波邻域尺度3')
K5=medfilt2(J,[5 5]);
subplot(3,3,8)
imshow(K5,256)
title('中值滤波邻域尺度5')
K6=medfilt2(J,[7 7]);
subplot(3,3,9)
imshow(K6,256)
title('中值滤波邻域尺度7')
```

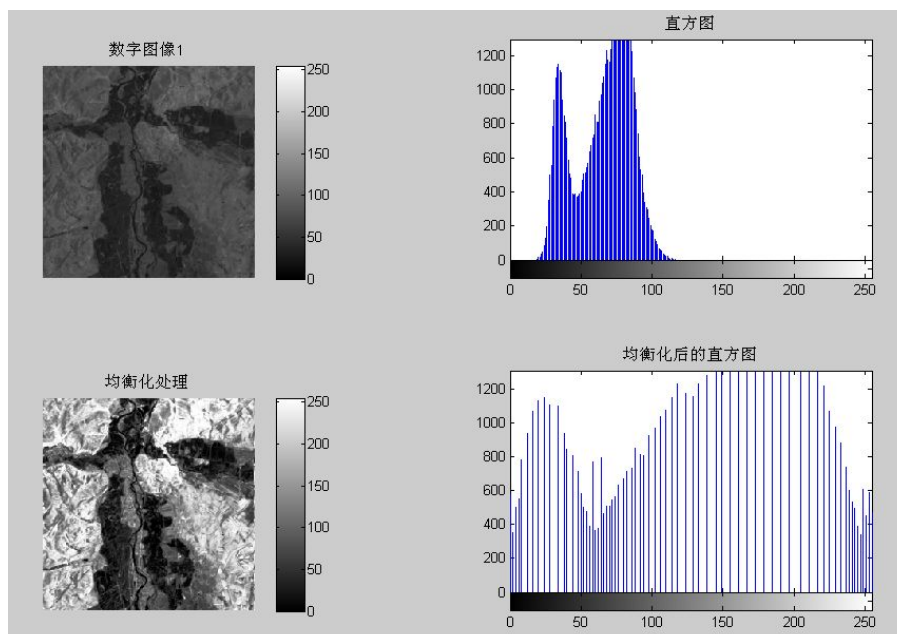
## 四、实验结果及分析：

### 1、实验结果：

#### 灰度变换以及反变换



#### 图像的直方图均衡



#### 图像平滑滤波



图像锐化处理



### 3、图像实验课程小结和思考

1. 各种增强算法的图像处理结果如上述图所示。
2. 图像的直方图是指将图像中的高频和低频部分映射到频率域，通过直方图的形式表现出来。均衡后的直方图表面上看是变得亮了。实际上，这是通过程序把原本集中在低频段的图像，通过直方图均衡的方式，在各个频段分布相对均匀，从而使低频段减少，高频段的数量相应的增加，在图像上的直观表现就是图像变得更加亮。频率分布不是差别不是太大的图像可以使用直方图均衡，但是对一些特别亮或者特别暗的图像这不适合进行直方图均衡。
3. 中值滤波属于非线性滤波器，比较适合用来处理椒盐噪声这一类的噪声。
4. 平滑滤波是对邻域的像素点进行求均值或者求中值等操作，如果邻域选的比较大，这处理的结果比较准确，相反的，如果邻域比较小，则可能产生比较大的误差，处理的效果很不好。
5. 锐化与平滑滤波相反，是对图像的边缘进行的一个增强，锐化就可以增强图像边缘，让模糊图像变得清晰



## 实验名称：运动模糊图像的复原处理实验

### 一、实验目的：

1. 通过实验了解运动模糊图像（图像降质现象）的物理本质
2. 掌握降质图像的维纳滤波复原方法

### 二、实验内容：

1. 编程实现无噪声的运动模糊图像复原
2. 编程实现有噪声的运动模糊图像复原

### 三、实验方法及编程：

```
运动模糊加噪 维纳滤波
X=imread('lena','BMP')
subplot(2,2,1)
imshow(X,256)
title('数字图像lena')
PSF=fspecial('motion',10,78)
G=imfilter(X,PSF,20,30)
subplot(2,2,2)
imshow(G,256)
title('模糊图像')
J=imnoise(G,'salt & pepper',0.05)
subplot(2,2,3)
imshow(J,256)
title('模糊且加噪图像')
K=deconvwnr(J,PSF,1)
subplot(2,2,4)
imshow(K,256)
title('维纳滤波图像')
```

### 四、实验结果及分析：

- 1、实验结果：

运动模糊图像以及维纳滤波



## 2、图像实验课程小结和思考

1. 在运动过程中，观察者与景物之间存在足够大的相对运动，就会造成照片的模糊，称之为运动模糊。
2. 再通过改变 psf 的情况下，观察有噪声和无噪声图像复原的差别，我们可以发现，无噪声的图像复原相对而言更好。

# 实验名称：数字图像编码实验

## 一、实验目的：

数字图像正交变换的方法，观察并比较图像频谱、了解 FFT、DCT 的幅度分布情况。

## 二、实验内容：

对图像进行 FFT、DCT 评语变换和反变换，并显示图像频谱。

## 三、实验方法及编程：

傅里叶等变换

```
X=imread('lena','BMP')
subplot(2,3,1)
imshow(X,256)
title('数字图像lena')
Y1=fft2(X)
Y11=fftshift(Y1)
```

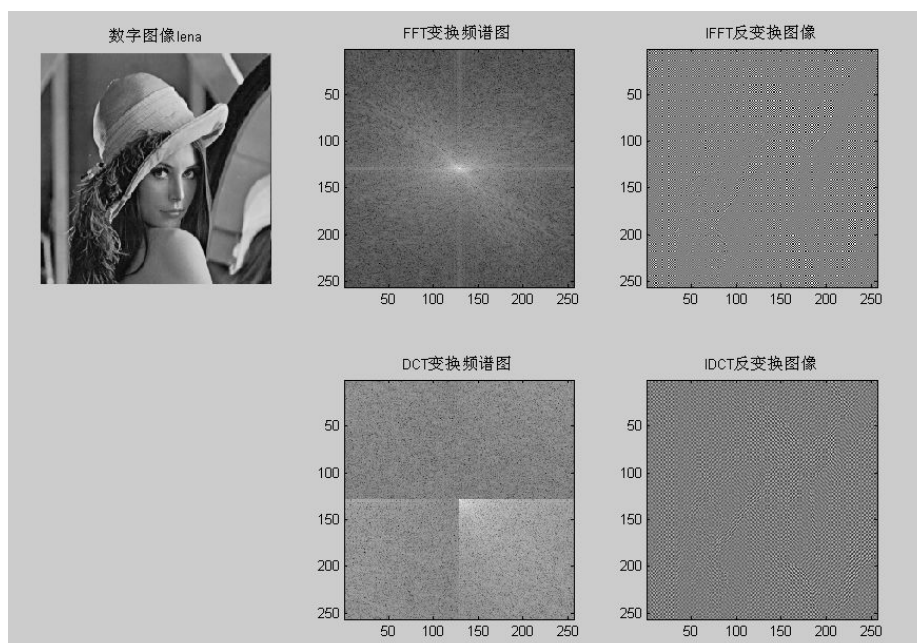
```

y1=log(abs(Y11))
subplot(2,3,2)
imagesc(y1)
title('FFT变换频谱图')
Y2=ifft2(Y11)
subplot(2,3,3)
imagesc(Y2)
title('IFFT反变换图像')
D1=dct2(X)
subplot(2,3,5)
D11=fftshift(D1)
d1=log(abs(D11))
imagesc(d1)
title('DCT变换频谱图')
D2=idct2(D11)
subplot(2,3,6)
imagesc(D2)
title('IDCT反变换图像')

```

## 四、实验结果及分析：

### 1、实验结果：



### 2、图像实验课程小结和思考

1. FFT 频谱系数能量的分布呈现的状态时中间大，四周小。可以看出该图具有较大的高频分量，而低频分量相对而言占有较少的比例。DCT 频谱系数分布，低频系数的值较大，高频系数的值较小，即能量集中在低频。
2. 正交变换可以使得图像能量主要集中分布在低频率成分上，边缘和线信息反映在高频率成分上。正交变换广泛应用在图像增强、图像恢复、特征提取、图像编码压缩和形状分析等方面。

## 实验名称：JPEG 压缩编码实验

### 一、实验目的：

熟悉和了解 JPEG 基本系统的编解码基本原理和方法

### 二、实验内容：

1. 编程实现  $n \times n$  子块 DCT 变换的图像频谱显示
2. 编程实现近似 JPEG 的图像压缩编码

### 三、实验方法及编程：

```
function JPEG_main()
% ***** JPEG 压缩编码算法函数
% *****
% 函数名称：
%   JPEG_main()
% 说明：
%   编程实现 子块 DCT 变换的图像频谱显示； $8 \times 8$  子块 DCT 变换系数按“Z”（Zig-Zag）
扫描的图像重建；
% JPEG 压缩编码（进行  $8 \times 8$  子块的 DCT 图像变换，JPEG 量化矩阵的量化与反量化， $8 \times 8$ 
子块 DCT 的图像重建）；
% 计算图像的均方根误差、显示误差图像和误差直方图。
%
% 调用函数：
%   Block=DctBlock(oldbuf,Block)      DCT  $n \times n$  块频谱函数
%   DCTch=DctCode(oldbuf,DCTch)      DCT  $8 \times 8$  块系数“Z”字扫描图像压缩函数
%   Scale=JpegCode(oldbuf,Scale)      JPEG 近似基准编码函数
%   eimag(oldbuf,newbuf)              计算均方根误差、显示误差图像和误差直方图
函数
%
% *****
% *****
clear;close;clc;
```

```

Block=3;
DCTch=1;
Scale=1;
set(gcf,'name','JPEG 压缩编码实验','numbertitle','off',...
    'unit','normalized','position',[0.05,0.1,0.9,0.75],...
    'menubar','none');
K=0;
while (K<6) % 菜单选择
K=menu('JPEG 菜单','读入图像','n*n DCT 频谱','8*8 DCT 压缩','JPEG 压缩编
码+', 'CLOSE');
switch K
    case 1
        oldbuf=ReadImage; %调用读图像函数
    case 2
        %Block=DctBlock(oldbuf,Block); %调用 DCT n*n 块频谱函数(Block
取值范围: 3~8)
        DctBlock(oldbuf,Block);
    case 3
        %DCTch=DctCode(oldbuf,DCTch); %调用 DCT 8*8 块系数“Z”字扫描
图像压缩函数(DCTch 取值范围: 1~60)
        DctCode(oldbuf,DCTch);
    case 4
        Scale=JpegCode(oldbuf,Scale); %调用 JPEG 近似基准编码函数
    case 5
        close;clear;
end
end;

%-----
-----
function oldbuf=ReadImage;
%
%*****
%*****
% 函数名称:
% ReadImage()
% 说明:
% 读入原图像。
%
%*****
%*****

oldbuf=imread('lena.bmp');
subplot(2,2,1);
imshow(oldbuf);

```

```

    title('原图像');
%-----
%-----

function Block=DctBlock(oldbuf,Block);
%
%*****
%*****
% 函数名称:
%   DctBlock()      DCT n*n 块频谱函数
%
% 参数:
%   oldbuf          原图像数组
%   Block           DCT n*n 当前块选择值
%   newbuf          存放处理后的图像二维数组
%
% 说明:
%   根据 Block 块的当前选择值, 计算原图像的 n*n 块 DCT 变换, 并转换为可视频谱图,
%   有利于频谱的观察。
%
%*****
%*****

if Block>=9 Block=3; end
n=pow2(Block);
x=double(oldbuf);
t=dctmtx(n);
dctfre=blkproc(x,[n n],'P1 * x * P2',t,t);
newbuf=log(abs(dctfre)*5+1);
subplot(2,2,2),imshow(newbuf,[]);
ch = num2str(n);
strch = strcat(ch, '*',ch, '频谱显示');
title(strch);
subplot(2,2,3),imshow([]);
subplot(2,2,4),imshow([]);
%Block=Block+1;
%if Block=9 Block=3; end
%-----
%-----

function DCTch=DctCode(oldbuf,DCTch);
%
%*****
%*****
% 函数名称:
%   DctCode()      DCT 8*8 块系数“Z”字扫描图像压缩函数
%

```

```

% 参数:
%   oldbuf           原图像数组
%   DCTch            DCT 8*8 块“Z”扫描当前系数选择值
%   newbuf           存放处理后的图像二维数组
%
% 说明:
%   计算图像的 8×8 子块 DCT 变换, 按“Z”字扫描顺序, 根据 DCTch 参数, 只保留 64
个
% DCT 系数中的前 DCTch 个系数, 对修改后的 DCT 系数用逆 DCT 变换重建图像, 得到 DCT
变
% 换的压缩图像。计算重建图像的均方根误差 RMSE ; 显示误差图像和误差直方图。
%
*****
*****

zigzag=[1 2 6 7 15 16 28 29           %设置 z 扫描顺序
        3 5 8 14 17 27 30 43
        4 9 13 18 26 31 42 44
        10 12 19 25 32 41 45 54
        11 20 24 33 40 46 53 55
        21 23 34 39 47 52 56 61
        22 35 38 48 51 57 60 62
        36 37 49 50 58 59 63 64];

tbuf=ones(8);
maskbuf=tbuf .* zigzag<=DCTch;

x=double(oldbuf);
t=dctmtx(8);
dctfre=blkproc(x,[8 8], 'P1 * x * P2',t,t');
y=blkproc(dctfre,[8 8], ' x.* P1',maskbuf);
newbuf=blkproc(y,[8 8], 'P1 * x * P2',t',t);

subplot(2,2,2),imshow(uint8(newbuf));
ch= num2str(DCTch);
strch = strcat('取',ch,'个 DCT 系数时的压缩图像');
title(strch);
eimag(oldbuf,newbuf);
%DCTch=DCTch+1;
%if DCTch==11 DCTch=1; end

%-----
-----
function Scale=JpegCode(oldbuf,Scale);

```

```

%
*****

% 函数名称:
%   JpegCode()   JPEG 近似基准编码函数
%
% 参数:
%   oldbuf       原图像数组
%   newbuf       存放处理后的图像二维数组
%   Scale        乘积因子, 通过对标准量化表 AC 系数的 Q 倍乘, 对图像作不同的量化,
%               得到不同的压缩比。
%
% 说明:
%   实现 JPEG 压缩编码。进行  $8 \times 8$  子块 DCT 变换, JPEG 量化矩阵的量化与反量化,  $8 \times 8$  子块 DCT 的
%   图像重建。计算重建图像的均方根误差 RMSE ; 显示误差图像和误差直方图。该函数没有
%   进行熵编码,
%   只是一个近似 JPEG 的编解码处理。
%
*****

Z = [16 11 10 16 24 40 51 61.
      12 12 14 19 26 58 60 55
      14 13 16 24 40 57 69 56
      14 17 22 29 51 87 80 62
      18 22 37 56 68 109 103 77
      24 35 55 64 81 104 113 92
      49 64 78 87 103 121 120 101
      72 92 95 98 112 100 103 99] * Scale ;

Z(1)=16;
x=double(oldbuf);
t=dctmtx(8);
dctfre=blkproc(x,[8 8], 'P1 * x * P2',t,t');
quacoe=blkproc(dctfre,[8 8], 'round(x ./ P1)',Z);
iquacoe=blkproc(quacoe,[8 8], 'x .* P1',Z);
newbuf=blkproc(iquacoe,[8 8], 'P1 * x * P2',t',t);
subplot(2,2,2),imshow(uint8(newbuf));
ch= num2str(Scale);
strch = strcat('Scale 为',ch,'时的 JPEG 压缩图像');
title(strch);
eimag(oldbuf,newbuf);
Scale=Scale+1;

```



```

if Scale==6  Scale=1; end

%-----
-----

function eimag(oldbuf,newbuf)
%
%*****
%*****
% 函数名称:
%    eimag()
% 说明:
%    计算重建图像的均方根误差 RMSE ; 显示误差图像和误差直方图。
%
%*****
%*****

e = double(oldbuf)-newbuf;
[m,n]=size(e);RMSE=sqrt(sum(e(:).^2) / (m*n));
if RMSE
    emax = max(abs(e(:)));
    [h,x] = hist(e(:),emax);
    if length(h) >= 1
        s=max(h(:));
        subplot(2,2,3),bar(x,h/s,'k');
        RMSE = num2str(RMSE);
        strRMSE = strcat('图像差值直方图 均方根误差 RMSE= ',RMSE);
        title(strRMSE);
        e = mat2gray(e, [-emax,emax]);
        subplot(2,2,4),imshow(e);
        title('原图像与压缩图像的差值图像');
    end
end

end

%-----
-----

```

## 四、实验结果及分析:

### 1、实验结果:

