

# 实验二

## 实验基本要求：

### 1. 图像几何运算

- (1) 对图像进行缩放、旋转等几何操作
- (2) 对比采用不同的插值法的效果
- (3) 对图像进行水平、垂直镜像操作
- (4) 对图像进行反相操作

### 2. 图像代数运算

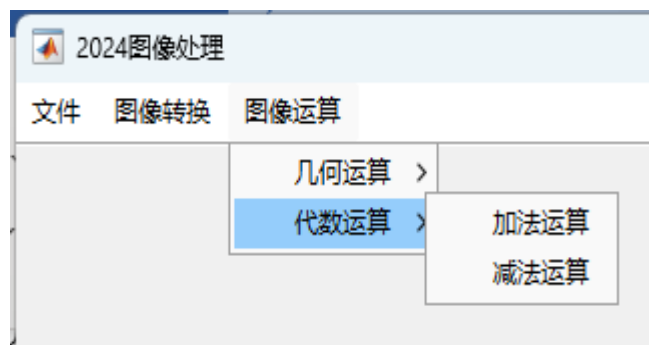
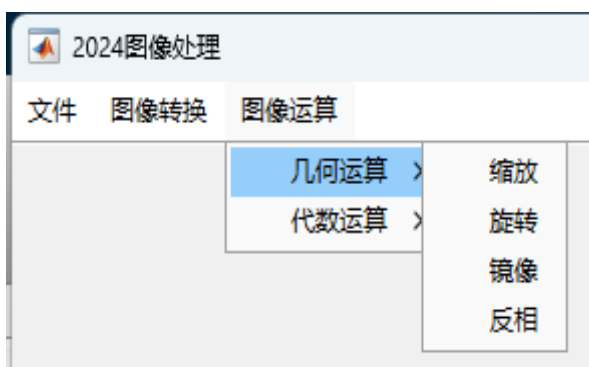
#### (1) 图像加法运算

图像相加一般用于对同一场景的多幅图像求平均效果（平均法降噪）。一个静止场景的  $M$  幅图像，假定每幅噪声图像来自于同一个互不相关的，噪声均值等于 0 的随机噪声图像的样本集，通过将  $M$  幅图像相加，然后求平均值，即对  $M$  幅图像进行平均，使图像中每一点的平方信噪比提高了  $M$  倍。

- (a) 原图（没有噪声）。
  - (b) 加入“椒盐(salt&pepper)”噪声的图像。
  - (c)  $M$  幅噪声图像均值降噪得到的图像。
- 对比不同的  $M$  值的降噪效果。

#### (2) 图像减法运算

### 参考界面：



## 实验提示：

### Matlab 图像处理工具箱的使用

#### 1. `imresize()` 调整图像的大小

**B=imresize(A,m,method)**

用 **method** 指定的插补方法返回大小等于 A 的 m 倍的图像 B。

参数 **method** 用于指定插值的方法，可选的值为 'nearest'(最近邻法)、'bilinear'(双线性插值)、'bicubic'(双三次插值)，默认值为 'nearest'。

例：使用不同插值方法放大图像：

```
I=imread('rice.tif');  
imshow(I);  
I1=imresize(I,1.5,'nearest'); %最近邻插值  
figure;imshow(I1)  
I2=imresize(I,1.5,'bilinear');  
figure;imshow(I2);
```

#### 2. 图像的旋转 `imrotate()`

**B=imrotate(A,angle,method)**

A 是图像的数据矩阵，**angle** 是图像的旋转角度。

参数 **method** 用于指定插值的方法，可选的值为 'nearest'(最近邻法)、'bilinear'(双线性插值)、'bicubic'(双三次插值)，默认值为 'nearest'。

例：使用最近邻插值方法旋转图像：

```
I=imread('lenna.bmp');  
I1=imrotate(I,30,'nearest');  
figure,imshow(I1);
```

#### 3. 图像的镜像

**flip(X,dim)**函数    matlab 中针对矩阵翻转变换的函数，其中 X 表示待操作矩阵，**dim** 指定翻转方式，**dim** 为 1 时矩阵上下翻转，**dim** 为 2 时矩阵左右翻转。

#### 4. 平均法降噪

##### a. `imnoise` 函数

Matlab 图像处理工具箱提供了模拟噪声生成的函数 **imnoise**，它可以对图像添加一些典型的噪声。

**imnoise** 的语法格式为：

**J=imnoise(I, type)**

**J=imnoise(I, type, parameters)**

对原图像 **I** 添加典型噪声得到含噪声图像 **J**，

参数 **type** 和 **parameters** 用于确定噪声的类型和相应的参数。

参数 **Type** 对应的噪声类型如下：

'gaussian'	高斯噪声
'localvar'	0 均值高斯白噪声
'poisson'	泊松噪声
'salt & pepper'	盐椒噪声
'speckle'	乘性噪声

**I=imread('cat.bmp');**

**J1=imnoise(I, 'salt & pepper', 0.02); %对图像 I 添加椒盐噪声**

## b. imadd 函数

**Z=imadd(X, Y)**

其中，**X** 和 **Y** 表示需要相加的两幅图像，返回值 **Z** 表示得到的加法操作结果。

若 **Y** 为一常量，则表示对图像 **X** 亮度的整体调整；图像的加法运算主要用来得到图像的叠加效果和消除叠加性噪声。

## 5. 图像减法

图像减法也称为差分方法，是一种常用于检测图像变化及运动物体的图像处理方法,图像减法可以作为许多图像处理工作的准备步骤。例如，可以使用图像减法来检测一系列相同场景图像的差异。

### imsubtract() 函数

**Z=imsubtract(x,y) %Z 为 x-y 操作的结果**

## 6. 图像数据类型：

**uint8    uint16    double**

MATLAB 使用三种存储格式来存储图像：**uint8**(8 位无符号整数)、**uint16**(16 位无符号整数)和双精度 **double**。大多数图像文件格式采用 8 位数据存储像素值，将这些文件读入内存后，MATLAB 都将其存储为 **uint8** 类型。

matlab 中矩阵在运算时要求所有的变量为 **double** 型

```
x1=imread(...);  
x2=double(x1);  
....  
x2=uint8(x2);  
imshow(x2);
```

**%x1 为图像矩阵**  
**%将 X1 转换为 double 类型**  
**%各种运算**  
**%将 X2 转换为 uint8 类型**