江西理工大學 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

Digital Image Processing

数字图像处理

Lecture 011:

Review some concept based on matlab code

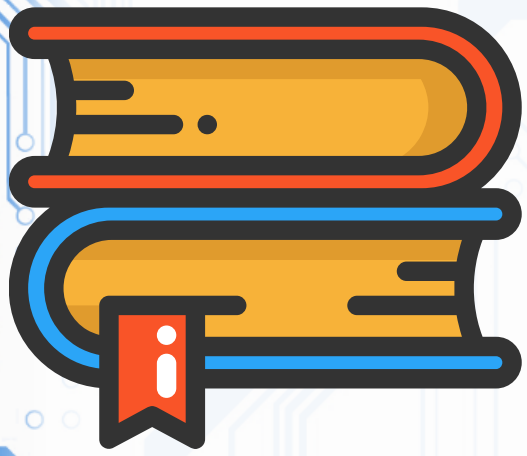
Dr Ata Jahangir Moshayedi

Prof Associate ,
School of information engineering Jiangxi university of science and technology, China

EMAIL: ajm@jxust.edu.cn

Autumn _2021

江西理工大學信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



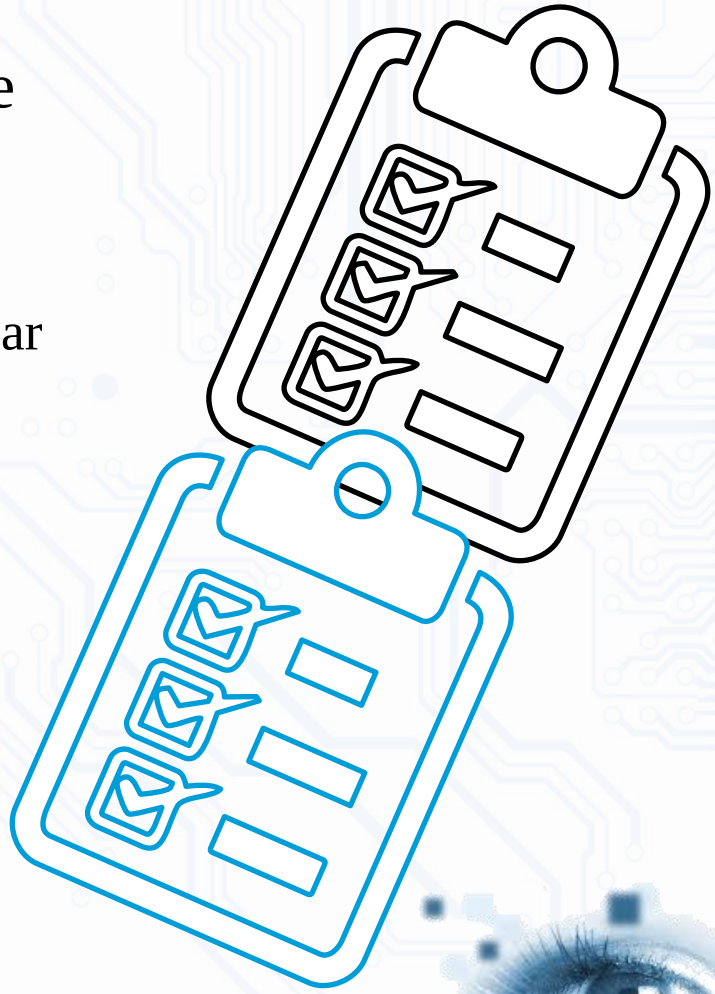
Digital Image Processing

LECTURE 15: **Using MATLAB Image-processing tool-box**

Review on pervious concept in matlab code _filtering

Agenda 议程

- **Noise:** Imnoise: Add noise to image
  噪声:Imnoise:给图像添加噪点
- Pixel neighbourhoods 像素邻域
  - Filter kernels and the mechanics of linear filtering 滤波器内核和线性滤波机制
  - Mean filtering 均值滤波
  - Median filtering 中值滤波
  - Rank filtering 级滤波
  - Gaussian filtering 高斯滤波



Imnoise: Add noise to image

Imnoise：向图像添加噪点

- J = imnoise(I,'gaussian') adds zero-mean, Gaussian white noise with variance of 0.01 to grayscale image I. You optionally can add noise using a GPU (requires Parallel Computing Toolbox™).
- J = imnoise(I,'gaussian',m) adds Gaussian white noise with mean m and variance of 0.01.
- J = imnoise(I,'gaussian',m,var_gauss) adds Gaussian white noise with mean m and variance var_gauss.
- J = imnoise(I,'localvar',var_local) adds zero-mean, Gaussian white noise of local variance var_local.
- J =imnoise(I,'localvar',intensity_map,var_local) adds zero-mean, Gaussian white noise. The local variance of the noise, var_local, is a function of the image intensity values in I. The mapping of image intensity value to noise variance is specified by the vector intensity_map.
- J = imnoise(I,'poisson') generates Poisson noise from the data instead of adding artificial noise to the data. See Algorithms for more information.
- J = imnoise(I,'salt & pepper') adds salt and pepper noise, with default noise density 0.05. This affects approximately 5% of pixels.
- J = imnoise(I,'salt & pepper',d) adds salt and pepper noise, where d is the noise density. This affects approximately d*numel(I) pixels.
- J = imnoise(I,'speckle') adds multiplicative noise using the equation J = I+n*I, where n is uniformly distributed random noise with mean 0 and variance 0.05.
- J = imnoise(I,'speckle',var_speckle) adds multiplicative noise with variance var_speckle.

Review on all 回顾所有

- J = imnoise(A,’noise’,m)

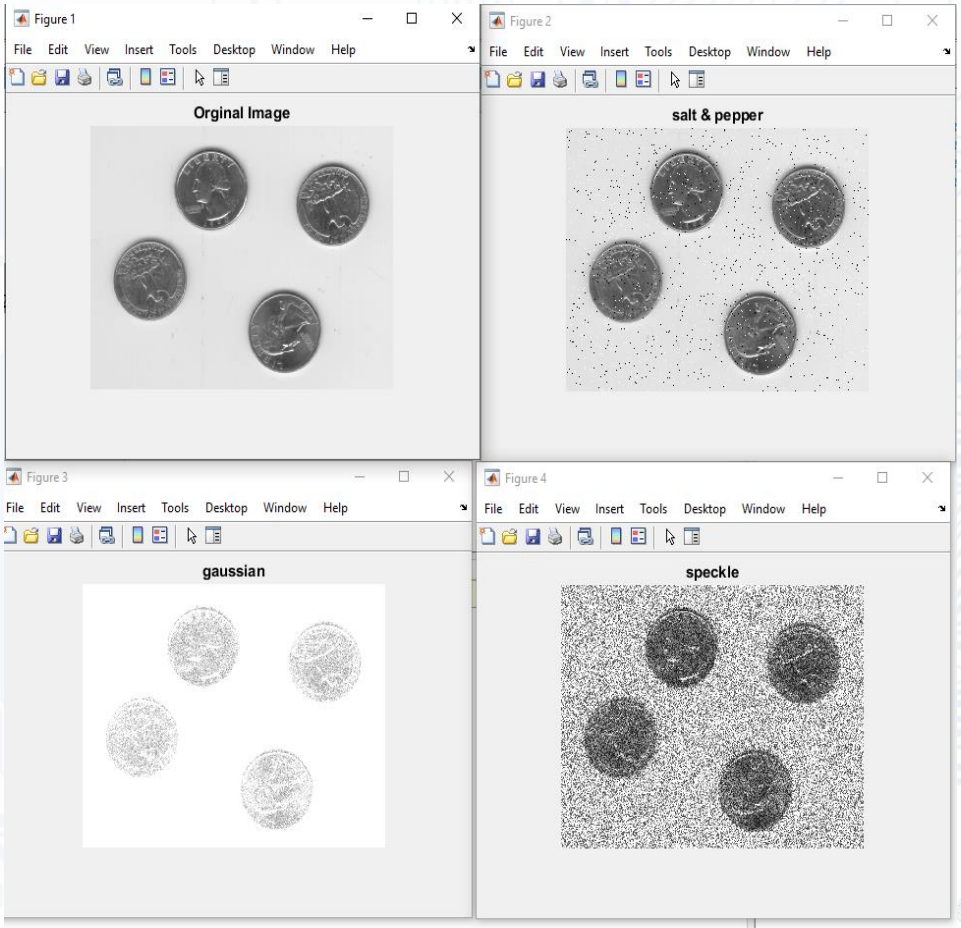
- Noise: noise name
  - Gaussian: white noise with the average contrast of image
  - salt & pepper:noise on pixels
  - Speckle:adds multiplicative noise using the equation J = I+n*I, where n is uniformly distributed random noise with mean 0 and variance 0.05.
- A: image
- M:mean m and variance

  平均m和方差

高斯:白噪点与图像的平均对比度
盐和胡椒:像素上的噪点
散斑:使用方程J = I+n*I添加乘性噪点，其中n是均匀分布的随机噪点，均值0，方差0.05。

Noise on image 图像上的噪点

```
clc
clear all
I = imread('eight.tif');
figure(1);imshow(I)
title('Orginal Image')
 %Add salt and pepper noise, with a noise
density of 0.02, to the image. Display the result.
%添加盐和胡椒的噪声，噪声密度为0. 02。
显示结果。
J2 = imnoise(I,'salt & pepper',0.02);
figure(2);imshow(J2)
title('salt & pepper')
J3 = imnoise(I,'gaussian',0.5);
figure(3);imshow(J3)
title('gaussian')
J4 = imnoise(I,'speckle',0.2);
figure(4);imshow(J4)
title('speckle')
```



Pixel neighbourhoods

像素邻域xiàng sù lín yù

- An important measure in images is the concept of connectivity.
- Many operations in image processing use the concept of a local image neighbourhood to define a local area of influence, relevance or interest.
- Central to this theme of defining the local neighbourhood is the notion of pixel connectivity, i.e. deciding which pixels are connected to each other.
- When we speak of 4-connectivity, only pixels which are N, W, E, S of the given pixel are connected.
- However, if, in addition, the pixels on the diagonals must also be considered, then we have 8-connectivity (i.e. N, NW, W, NE, SE, E, SW, S are all connected see Figure).

像素邻域

- 图像中的一个重要度量是连通性的概念。
- 图像处理中的许多操作使用局部图像邻域的概念来定义影响、相关或感兴趣的局部区域。
- 这个定义局部邻域的主题核心是像素连接的概念，即决定哪些像素相互连接。
- 当我们谈到4连通时，只有给定像素的 N、W、E、S是连通的。
- 但是，如果另外还必须考虑对角线上的像素，那么我们就有了 8 个连通性（即 N、NW、W、NE、SE、E、SW、S 都是连通的，见图）。

Pixel neighbourhoods

像素邻域

```
clc
clear all
close all
A=imread('cameraman.tif'); % Read in image
subplot(1,2,1), imshow(A); % Display image
func = @(x) max(x(:)); % set filter to apply
B = nlfilter(A,[3 3],func); % apply over 3 x 3 neighbourhood
subplot(1,2,2), imshow(B); % Display result image B
```

Comments

Here we specify func() as the max() filter function to apply over each and every 3 *3 neighbourhood of the image.

This replaces every input pixel in the output image with the maximum pixel value of the input pixel neighbourhood.

You may wish to experiment with the effects of varying the neighbourhood dimensions and investigating the Matlab min() and mean() (for the latter, a type conversion will be required to display the double output type of the Matlab mean() function as an 8 bit image specify the filter function as uint8(mean())).

注释
这里我们将 func() 指定为 max() 过滤器函数，以应用于图像的每个 3 * 3 邻域。
这将用输入像素邻域的最大像素值替换输出图像中的每个输入像素。
您可能希望试验改变邻域维度的影响并研究 Matlab min() 和 mean()（对于后者，需要进行类型转换以将 Matlab mean() 函数的双输出类型显示为 8 位图像指定过滤函数为 uint8(mean())）。

Filter kernels and the mechanics of linear filtering

滤波器内核和线性滤波机制

- In linear spatial filters the new or filtered value of the target pixel is determined as some linear combination of the pixel values in its neighbourhood.

在线性空间滤波器中，目标像素的新值或过滤值被确定为其邻域中像素值的某种线性组合。

- Any other type of filter is, by definition, a nonlinear filter.

根据定义，任何其他类型的滤波器都是非线性滤波器。

- The specific linear combination of the neighbouring pixels that is taken is determined by the filter kernel (often called a mask).

所采用的相邻像素的特定线性组合由滤波器内核（通常称为掩码）决定。

Filter kernels and the mechanics of linear filtering

滤波器内核和线性滤波机制

```
clc
clear all
close all
A=imread('peppers.png'); % Read in image
subplot(1,2,1), imshow(A); % Display image
k = fspecial('motion', 50, 54); % create a 5x5 convolution kernel
B = imfilter(A, k, 'symmetric'); % apply using symmetric mirroring at edges
subplot(1,2,2), imshow(B); % Display result image B
```

Comments

Here we specify the fspecial() function to construct a kernel that will mimic the effect of motion blur (of specified length and angle) onto the image. Option 3) from our earlier discussion is used to deal with image edges during filtering.

You may wish to investigate the use of other kernel filters that can be generated with the fspecial() function and the edge region filtering options available with the imfilter() function.

Type doc imfilteratthe Matlab prompt for details). How do they effect the image filtering result?

注释

这里我们指定 fspecial() 函数来构建一个内核，该内核将模拟运动模糊（指定长度和角度）对图像的影响。 我们之前讨论的选项 3) 用于在过滤过程中处理图像边缘。

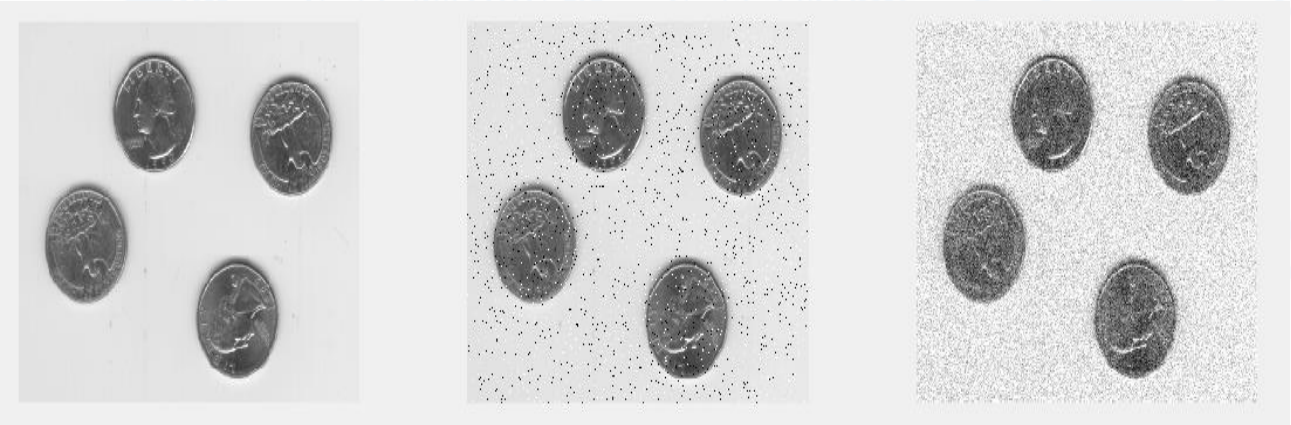
您可能希望研究可以使用 fspecial() 函数生成的其他内核过滤器的使用以及 imfilter() 函数提供的边缘区域过滤选项。

输入 doc imfilteratthe Matlab 提示以获得详细信息）。 它们如何影响图像过滤结果？

Filtering for noise removal

滤波去噪lǜ bō qù zào

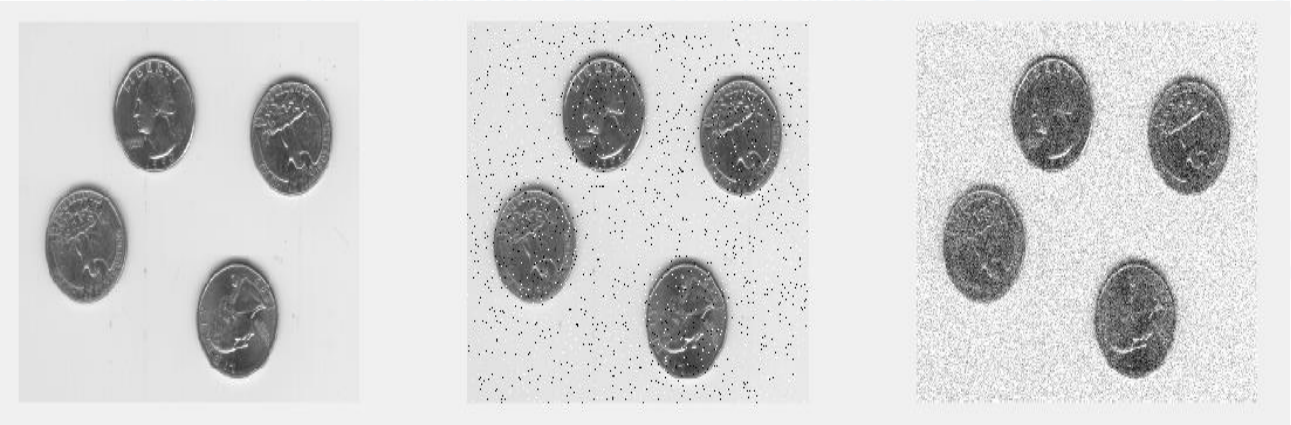
- One of the primary uses of both linear and nonlinear filtering in image enhancement is for noise removal. We will now investigate the application of a number of different filters for removing typical noise, such as additive ‘salt and pepper’ and Gaussian noise.
- The results of the noise addition from Example are shown in Figure. These images will form the basis for our comparison of noise removal filters in the following sections of this chapter.



(a) Original image with (b) ‘salt and pepper’ noise and (c) Gaussian noise added

滤波去噪

- 线性滤波和非线性滤波在图像增强中的主要用途之一是去除噪声。 现在，我们将研究一些不同的过滤器的应用，以去除典型的噪声，如添加剂“盐和胡椒”和高斯噪声。
- 例中噪声添加的结果如图所示。 这些图像将构成我们在本章接下来的章节中比较噪声去除滤波器的基础。



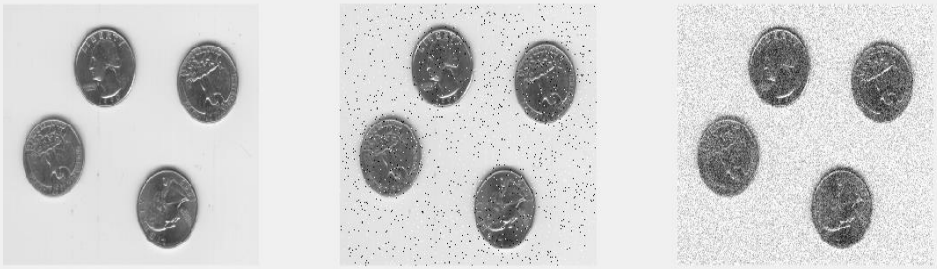
(a) Original image with (b) ‘salt and pepper’ noise and (c) Gaussian noise added

Filtering for noise removal

滤波去噪

```
clc
clear all
close all
I=imread('eight.tif'); % Read in image
subplot(1,3,1), imshow(I); % Display image
Isp = imnoise(I,'salt & pepper',0.03); % add 3% (0.03) salt and pepper noise
subplot(1,3,2), imshow(Isp); % Display result image Isp
Ig = imnoise(I,'gaussian',0.02); % add Gaussian noise (with 0.02 variance)
subplot(1,3,3), imshow(Ig); % Display result image Ig
```

Here we use the imnoise() function to add both 'salt and pepper' noise and Gaussian noise to the input image.
The strength is specified using the percentage density and variance (with zeromean)respectively.Thereadermaywishtoexperimentwiththeeffectsofchangingthese noise parameters and also explore the other noise effects available using this function (type doc imnoise at the Matlab prompt for details)



这里我们使用imnoise()函数为输入图像添加“盐和胡椒”噪声和高斯噪声。
强度分别使用百分比密度和方差(具有零均值)指定。 读者可以尝试改变这些噪声参数的效果，也可以使用这个函数探索其他可用的噪声效果(在Matlab提示符中输入doc imnoise获取详细信息)

Mean filtering

均值滤波jūn zhí lǜ bō

- The mean filter is perhaps the simplest linear filter and operates by giving equal weight wK to all pixels in the neighbourhood.
- A weight of WK=1/(NM) is used for an N M neighbourhood and has the effect of smoothing the image, replacing every pixel in the output image with the mean value from its N * M neighbourhood.
- This weighting scheme guarantees that the weights in the kernel sum to one over any given neighbourhood size.
- Mean filters can be used as a method to suppress noise in an image (although the median filter which we will discuss shortly usually does a better job).
- Another common use is as a preliminary processing step to smooth the image in order that some subsequent processing operation will be more effective.

均值滤波

- 均值滤波器可能是最简单的线性滤波器，其操作方法是给邻近的所有像素赋予相等的权重wK。
- NM邻域的权值为WK=1/(NM)，具有平滑图像的效果，将输出图像中的每个像素替换为其N * M邻域的均值。
- 这种加权方案保证了核和中的权值为1 /任何给定的邻域大小。
- 均值滤波器可以作为一种方法来抑制图像中的噪声(尽管我们稍后将讨论的中值滤波器通常做得更好)。
- 另一个常用的用途是作为平滑图像的初步处理步骤，以使后续处理操作更加有效。

Median filtering

中值滤波zhōng zhí lǜ bō

- The **median filter** is a non-linear digital filtering technique, often used to remove noise from an image or signal.

中值滤波是一种非线性数字滤波技术，常用于去除图像或信号中的噪声。

- Such noise reduction is a typical pre-processing step to improve the results of later processing (for example, edge detection on an image).

这样的降噪是一个典型的预处理步骤，以改善后期处理的结果(例如，图像的边缘检测)。

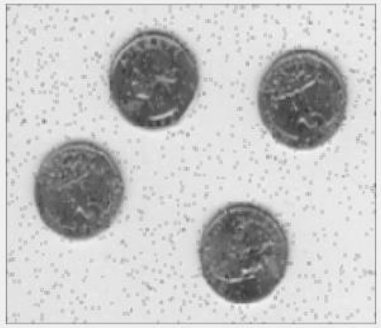
- Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise (but see the discussion below), also having applications in signal processing.

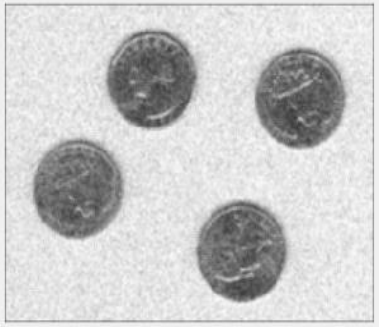
中值滤波在数字图像处理中应用非常广泛，因为在某些条件下，它可以在去除噪声的同时保留边缘(但见下面的讨论)，在信号处理中也有应用。

Mean filtering

均值滤波

```
clc
clear all
close all
I=imread('eight.tif'); % Read in image
Isp = imnoise(I,'salt & pepper',0.03); % add 3% (0.03) salt and pepper noise
Ig = imnoise(I,'gaussian',0.02); % add Gaussian noise (with 0.02 variance)
figure();subplot(1,3,1), imshow(I); % Display image
subplot(1,3,2), imshow(Isp); % Display result image Isp
subplot(1,3,3), imshow(Ig); % Display result image Ig
k = ones(3,3) / 9;  % define mean filter
I_m = imfilter(I,k); % apply to original image
Isp_m = imfilter(Isp,k); % apply to salt and pepper image
Ig_m = imfilter(Ig,k); % apply tp gaussian image
figure();
subplot(1,3,1), imshow(I_m); % Display result image
subplot(1,3,2), imshow(Isp_m); % Display result image
subplot(1,3,3), imshow(Ig_m); % Display result image
```

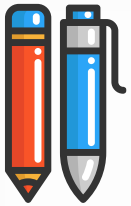




Filtering

Rank filtering 级滤波 jí lǜ bō

- Rank filters are non-linear filters using the local gray-level ordering to compute the filtered value.
- This ensemble of filters share a common base: the local gray-level histogram is computed on the neighborhood of a pixel (defined by a 2-D structuring element). If the filtered value is taken as the middle value of the histogram, we get the classical median filter.
- Rank filters can be used for several purposes such as:
  - ***image quality enhancement e.g. image smoothing, sharpening***
  - ***image pre-processing e.g. noise reduction, contrast enhancement***
  - ***feature extraction e.g. border detection, isolated point detection***
  - ***post-processing e.g. small object removal, object grouping, contour smoothing***



级滤波

- 等级滤波器是利用局部灰度排序来计算过滤值的非线性滤波器。
- 这个滤波器集合有一个共同的基础:局部灰度直方图是在一个像素的邻域上计算的(由一个二维结构元素定义)。 将滤波后的值作为直方图的中值，得到经典中值滤波。
- 等级过滤器可以用于以下几个目的:
  - 图像质量增强，如图像平滑、锐化
  - 图像预处理，如降噪，对比度增强
  - 特征提取，如边界检测、孤立点检测
  - 后期处理，如小物体移除，物体分组，轮廓平滑

Rank filtering 级滤波

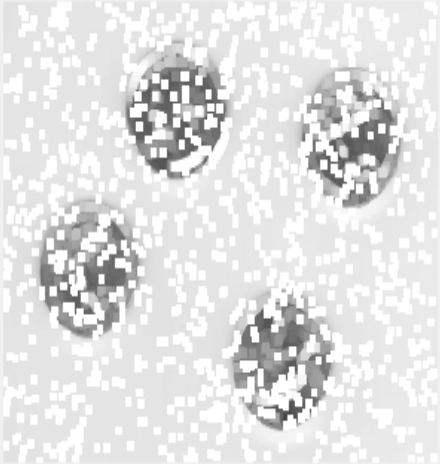
The median filter is really just a special case of a generalized order (or rank) filter. The general order filter is a nonlinear filter comprising the following common steps:

(1) Define the neighbourhood of the target pixel (N * N).

(2) Rank them in ascending order (first is lowest value, $(N*N)_{th}$ is highest value).

(3) Choose the order of the filter (from 1 to N).

(4) Set the filtered value to be equal to the value of the chosen rank pixel.



级滤波

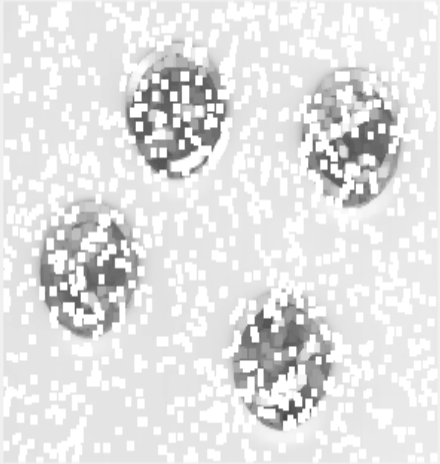
秩过滤器实际上只是广义顺序(或秩)过滤器的一种特殊情况。 一般阶滤波器是一种非线性滤波器，其一般步骤如下:

(1)定义目标像素的邻域(N * N)。

(2)将它们按升序排列(第1个值最低，第N*N个值最高)。

(3)选择滤波器的顺序(从1到N)。

(4)将过滤后的值设置为与所选rank像素的值相等。



Rank filtering 级滤波

```
clc
clear all
close all
I=imread('eight.tif'); % Read in image
Isp = imnoise(I,'salt & pepper',0.03); % add 3% (0.03) salt and pepper noise
Ig = imnoise(I,'gaussian',0.02); % add Gaussian noise (with 0.02 variance)
figure();subplot(1,3,1), imshow(I); % Display image
subplot(1,3,2), imshow(Isp); % Display result image Isp
subplot(1,3,3), imshow(Ig); % Display result image Ig
k = ones(3,3) / 9;  % define mean filter
I_m = imfilter(I,k); % apply to original image
Isp_m = imfilter(Isp,k); % apply to salt and pepper image
Ig_m = imfilter(Ig,k); % apply tp gaussian image
figure();
subplot(1,3,1), imshow(I_m); % Display result image
subplot(1,3,2), imshow(Isp_m); % Display result image
subplot(1,3,3), imshow(Ig_m); % Display result image

I_m = medfilt2(I,[3 3]); % apply to original image
Isp_m = medfilt2(Isp,[3 3]); % apply to salt and pepper image
Ig_m =medfilt2(Ig,[3 3]); % apply tp gaussian image
figure();
subplot(1,3,1), imshow(I_m); % Display result image
subplot(1,3,2), imshow(Isp_m); % Display result image
subplot(1,3,3), imshow(Ig_m); % Display result image
I_m = ordfilt2(I,25,ones(5,5)); % apply to original image
Isp_m = ordfilt2(Isp,25,ones(5,5)); % apply to salt and pepper image
Ig_m =ordfilt2(Ig,25,ones(5,5)); % apply tp gaussian image
figure();
subplot(1,3,1), imshow(I_m); % Display result image
subplot(1,3,2), imshow(Isp_m); % Display result image
subplot(1,3,3), imshow(Ig_m); % Display result image
```

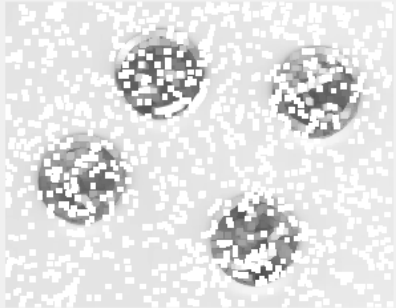
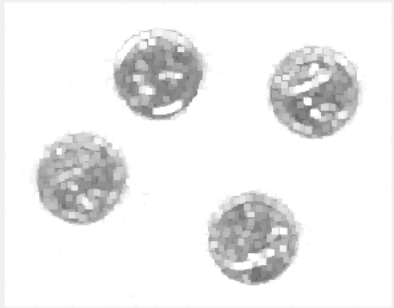
Rank filtering 级滤波

- Here we define a 5 * 5 max filter and apply it to the three images generated in Example and shown in Figure.

这里我们定义了一个5 * 5的最大过滤器，并将其应用到Example中生成的3张图片中，如图所示。

- Experiment with using larger neighbourhood sizes for this filter, varying the rank of the filter (second parameter of ordfilt2() function) and the effect it has on the resulting image.

实验使用更大的邻域尺寸，改变过滤器的等级(ordfilt2()函数的第二个参数)以及它对结果图像的影响。

Median filtering 中值滤波

- The **median filter** is a non-linear digital filtering technique, often used to remove noise from an image or signal. Median filtering is a nonlinear process useful in reducing impulsive, or salt-and-pepper noise.Such noise reduction is a typical pre-processing step to improve the results of later processing (for example, edge detection on an image).

- *Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise (but see the discussion below), also having applications in signal processing.*

中值滤波

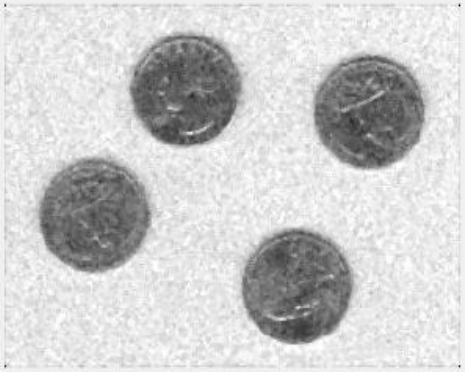
- 中值滤波是一种非线性数字滤波技术，常用于去除图像或信号中的噪声。中值滤波是一种非线性过程，在减少脉冲或椒盐噪声方面很有用。这样的降噪是一个典型的预处理步骤，以改善后期处理的结果(例如，图像的边缘检测)。

- 中值滤波在数字图像处理中应用非常广泛，因为在某些条件下，它可以在去除噪声的同时保留边缘(但见下面的讨论)，在信号处理中也有应用。

Median filtering 中值滤波

- Another commonly used filter is the median filter. Median filtering overcomes the main limitations of the mean filter, albeit at the expense of greater computational cost.
- As each pixel is addressed, it is replaced by the statistical median of its N * M neighbourhood rather than the mean.
- The median filter is superior to the mean filter in that it is better at preserving sharp high-frequency detail (i.e. edges) whilst also eliminating noise, especially isolated noise spikes (such as ‘salt and pepper’ noise).

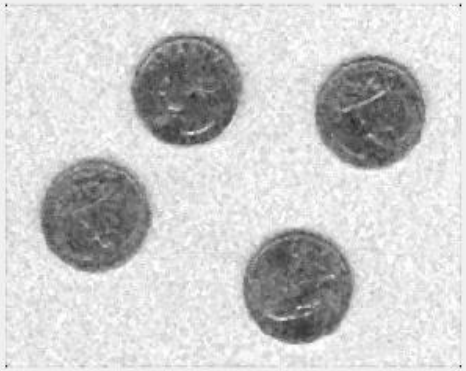




中值滤波

- 另一个常用的过滤器是中值过滤器。 中值滤波克服了均值滤波的主要局限性，尽管代价是更大的计算成本。
- 当每个像素被寻址时，它将被其N * M邻域的统计中值取代，而不是平均值。
- 中值滤波器优于均值滤波器，因为它能更好地保留尖锐的高频细节(如边缘)，同时还能消除噪声，特别是孤立的噪声峰值(如“盐和胡椒”噪声)。





Median filtering

中值滤波

```
clc
clear all
close all
I=imread('eight.tif'); % Read in image
Isp = imnoise(I,'salt & pepper',0.03); % add 3% (0.03) salt and pepper noise
Ig = imnoise(I,'gaussian',0.02); % add Gaussian noise (with 0.02 variance)
figure();subplot(1,3,1), imshow(I); % Display image
subplot(1,3,2), imshow(Isp); % Display result image Isp
subplot(1,3,3), imshow(Ig); % Display result image Ig
k = ones(3,3) / 9;  % define mean filter
I_m = imfilter(I,k); % apply to original image
Isp_m = imfilter(Isp,k); % apply to salt and pepper image
Ig_m = imfilter(Ig,k); % apply tp gaussian image
figure();
subplot(1,3,1), imshow(I_m); % Display result image
subplot(1,3,2), imshow(Isp_m); % Display result image
subplot(1,3,3), imshow(Ig_m); % Display result image

I_m = medfilt2(I,[3 3]); % apply to original image
Isp_m = medfilt2(Isp,[3 3]); % apply to salt and pepper image
Ig_m =medfilt2(Ig,[3 3]); % apply tp gaussian image
figure();
subplot(1,3,1), imshow(I_m); % Display result image
subplot(1,3,2), imshow(Isp_m); % Display result image
subplot(1,3,3), imshow(Ig_m); % Display result image
```

Removing noise in RGB image

去除 RGB 图像中的噪点

- The filter we used to remove the "salt & pepper" type noise was **medfilt2()**.
- However, as the "2" in the name indicates it's for 2-D array, it won't work for RGB image unless we decomposed each RGB channel and concatenate after the filtering each channel.

J = medfilt2(I)
J = medfilt2(I,[m n])
J = medfilt2(___,padopt)

- J = medfilt2(I)

performs median filtering of the image I in two dimensions. Each output pixel contains the median value in a 3-by-3 neighborhood around the corresponding pixel in the input image. You optionally can compute the normalized cross-correlation using a GPU (requires Parallel Computing Toolbox™).

- J = medfilt2(I,[m n])

performs median filtering, where each output pixel contains the median value in the m-by-n neighborhood around the corresponding pixel in the input image.

- J = medfilt2(___,padopt)

controls how medfilt2 pads the image boundaries.This syntax is not supported on a GPU.

去除 RGB 图像中的噪点

- 我们用来去除“盐和胡椒”类型噪点的过滤器是 medfilt2()。
- 但是，由于名称中的“2”表示它用于二维数组，因此除非我们分解每个 RGB 通道并在过滤每个通道后进行连接，否则它不适用于 RGB 图像。

```
J = medfilt2(I)
J = medfilt2(I,[m n])
J = medfilt2(___,padopt)
```

- J = medfilt2(I)

  对图像I进行二维中值滤波。 每个输出像素包含输入图像中对应像素的3 × 3邻域的中值。 你可以选择使用GPU计算标准化的互相关(需要并行计算工具箱™)。
- J = medfilt2(I,[m n])

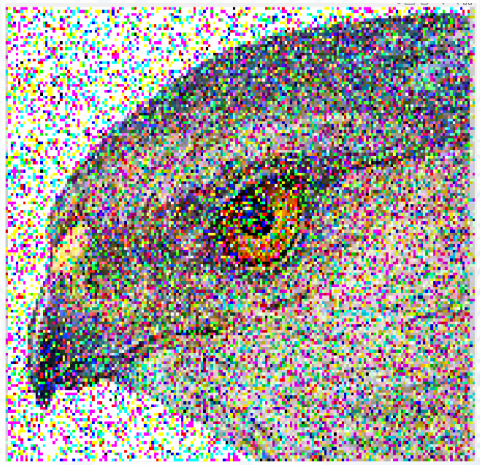
  执行中值滤波，其中每个输出像素包含输入图像中对应像素的m × n邻域的中值。
- J = medfilt2(___,padopt)

  控制medfilt2如何填充图像边界。 GPU不支持此语法。

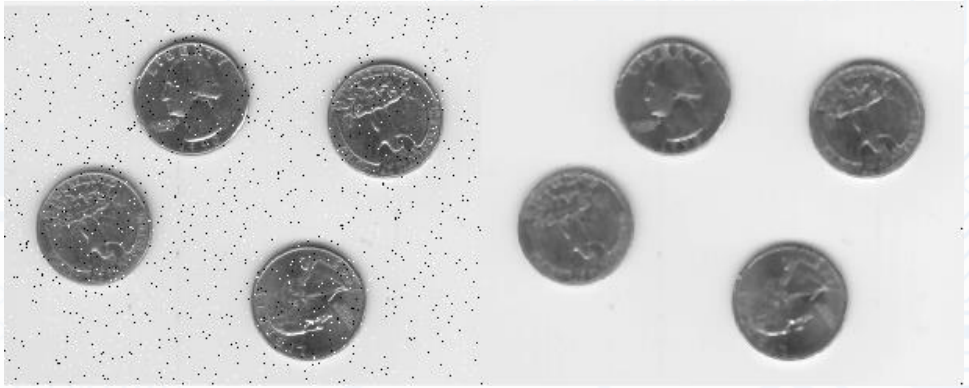
medfilt2:
2-D median filtering

medfilt2：二维中值滤波

```
clc
clear all
close all
 I = imread('C:\Users\AJM\Desktop\pic sourse\hawk.png');
J = imnoise(I,'salt & pepper',0.2);

% filter each channel separately
%分别过滤每个通道
r = medfilt2(J(:, :, 1), [3 3]);
g = medfilt2(J(:, :, 2), [3 3]);
b = medfilt2(J(:, :, 3), [3 3]);

% reconstruct the image from r,g,b channels
%从r,g,b通道重建图像
K = cat(3, r, g, b);
 figure
subplot(121);imshow(J);
subplot(122);imshow(K);
```

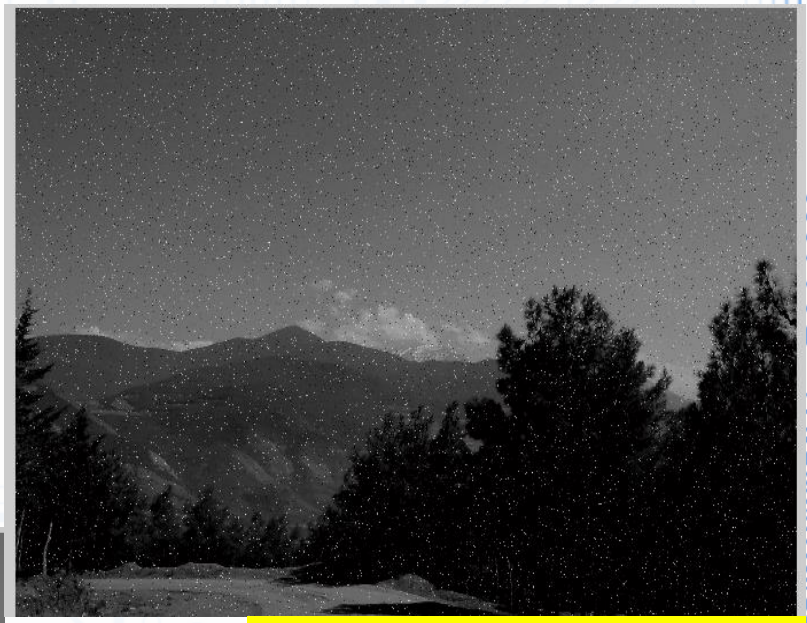


```
clc
clear all
close all

 I = imread('eight.tif');
figure, imshow(I)
%Add salt and pepper noise.

J = imnoise(I,'salt & pepper',0.02);
%Use a median filter to filter out the noise.

K = medfilt2(J); %Display results, side-by-side.
 imshowpair(J,K,'montage')
```



N=medfilt2(….

```
clc
clear all
close all

I = imread('C:\Users\AJM\Desktop\35.bmp');
N=rgb2gray(I)
N=medfilt2(N)
subplot(1,2,1),imshow(I)
 subplot(1,2,2),imshow(N)
```



Original Image with noise

After Filtering

wiener2

2-D adaptive noise-removal filtering

wiener2 2-D 自适应降噪滤波

```
J = wiener2(I,[m n],noise)
[J,noise_out] = wiener2(I,[m n])
```

J = wiener2(I,[m n],noise)

- filters the grayscale image I using a pixel-wise adaptive low-pass Wiener filter.
- [m n] specifies the size (m-by-n) of the neighborhood used to estimate the local image mean and standard deviation.
- The additive noise (Gaussian white noise) power is assumed to be noise.
- The input image has been degraded by constant power additive noise.
- wiener2 uses a pixelwise adaptive Wiener method based on statistics estimated from a local neighborhood of each pixel.

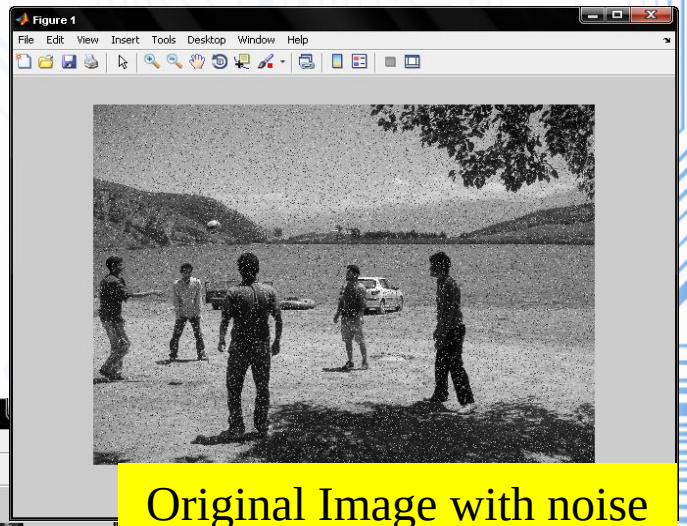
filters 灰度图像 I 使用像素级自适应低通维纳滤波器。
[m n] 指定用于估计局部图像均值和标准差的邻域大小（m×n）。
加性噪声（高斯白噪声）功率被假定为噪声。
输入图像已被恒功率附加噪声降级。
wiener2 使用基于从每个像素的局部邻域估计的统计数据的逐像素自适应 Wiener 方法。

wiener2

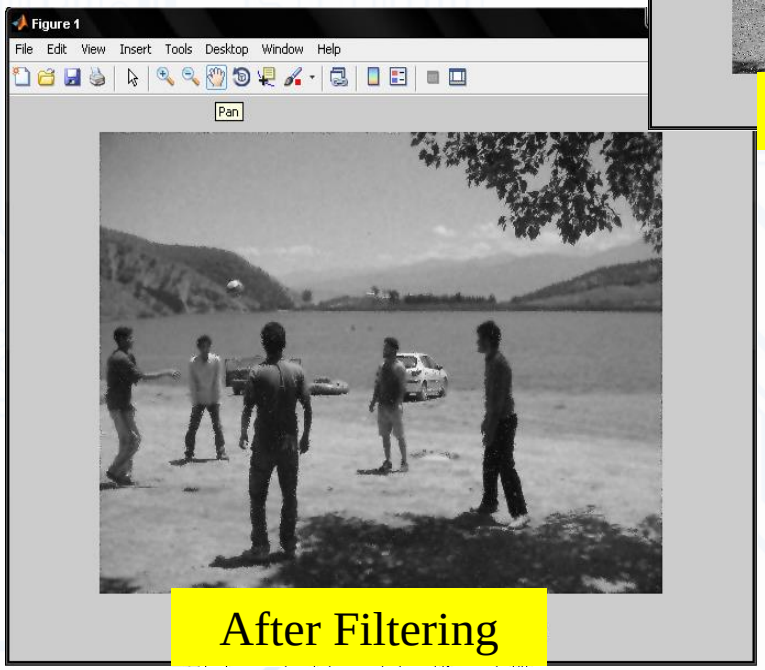
2-D adaptive noise-removal filtering

wiener2 2-D 自适应降噪滤波

```
clc
clear all
close all
 I = imread('C:\Users\AJM\Desktop\38.bmp');
N=rgb2gray(I)
N=wiener2(N,[15,15]);
imshow(I),figure,imshow(N)
```



Original Image with noise



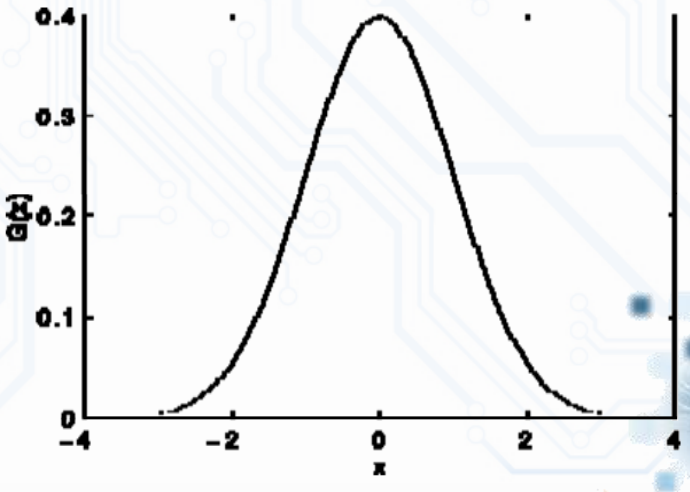
After Filtering

Gaussian filtering 高斯滤波

- Gaussian filtering is used to blur images and remove noise and detail. In one dimension, the Gaussian function is:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

- Where σ is the standard deviation of the distribution The distribution is assumed to have a mean of 0.
- Shown graphically, we see the familiar bell shaped Gaussian distribution.



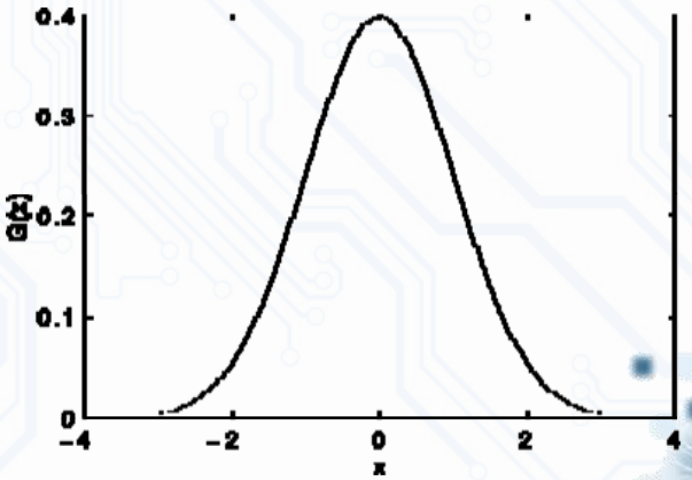
Gaussian distribution with mean 0 and σ = 1

高斯滤波

- 高斯滤波用于模糊图像，去除噪声和细节。 在一维情况下，高斯函数为:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

- 其中，σ为分布的标准差。假设分布的均值为0。
- 如图所示，我们看到了熟悉的钟形高斯分布。



高斯分布，均值为0，σ = 1

Gaussian filtering

高斯滤波

- The Gaussian filter is a very important one both for theoretical and practical reasons.

  高斯滤波器是一个非常重要的理论和实践原因。

- Here, we filter the image using a discrete kernel derived from a radially symmetric form of the continuous 2-D Gaussian function defined as follows:

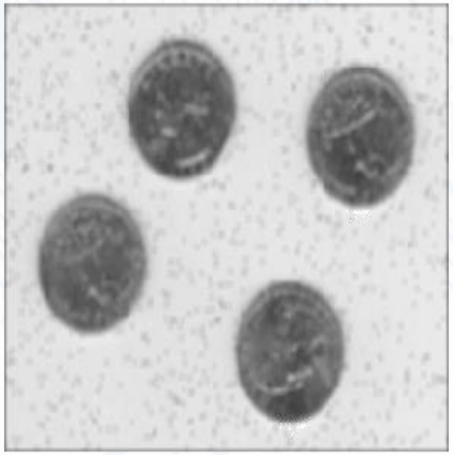
  在这里，我们使用由径向对称形式的连续二维高斯函数导出的离散核来过滤图像，定义如下：

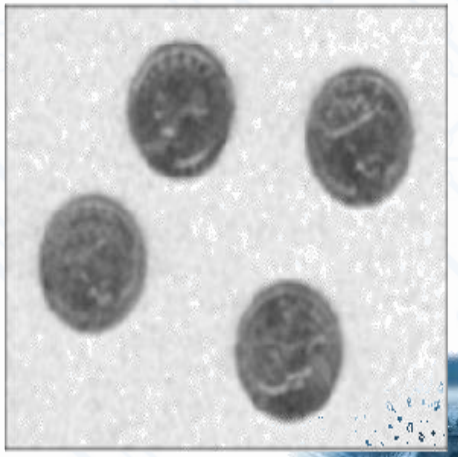
$$f(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

Gaussian filtering

高斯滤波

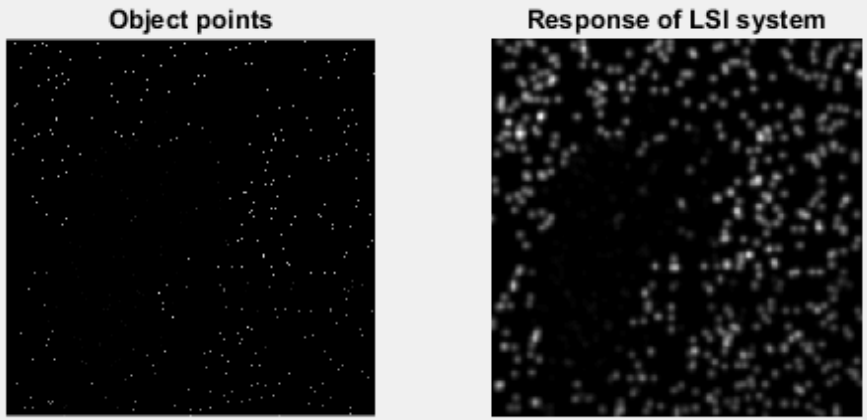
```
clc
clear all
close all
I=imread('eight.tif'); % Read in image
Isp = imnoise(I,'salt & pepper',0.03); % add 3% (0.03) salt and pepper noise
Ig = imnoise(I,'gaussian',0.02); % add Gaussian noise (with 0.02 variance)
 k = fspecial('gaussian', [5 5], 2); % define Gaussian filter
I_g = imfilter(I,k); % apply to original image
Isp_g = imfilter(Isp,k); % apply to salt and pepper image
Ig_g = imfilter(Ig,k); % apply tp gaussian image
figure();
subplot(1,3,1), imshow(I_g); % Display result image
subplot(1,3,2), imshow(Isp_g); % Display result image
subplot(1,3,3), imshow(Ig_g); % Display result image
```





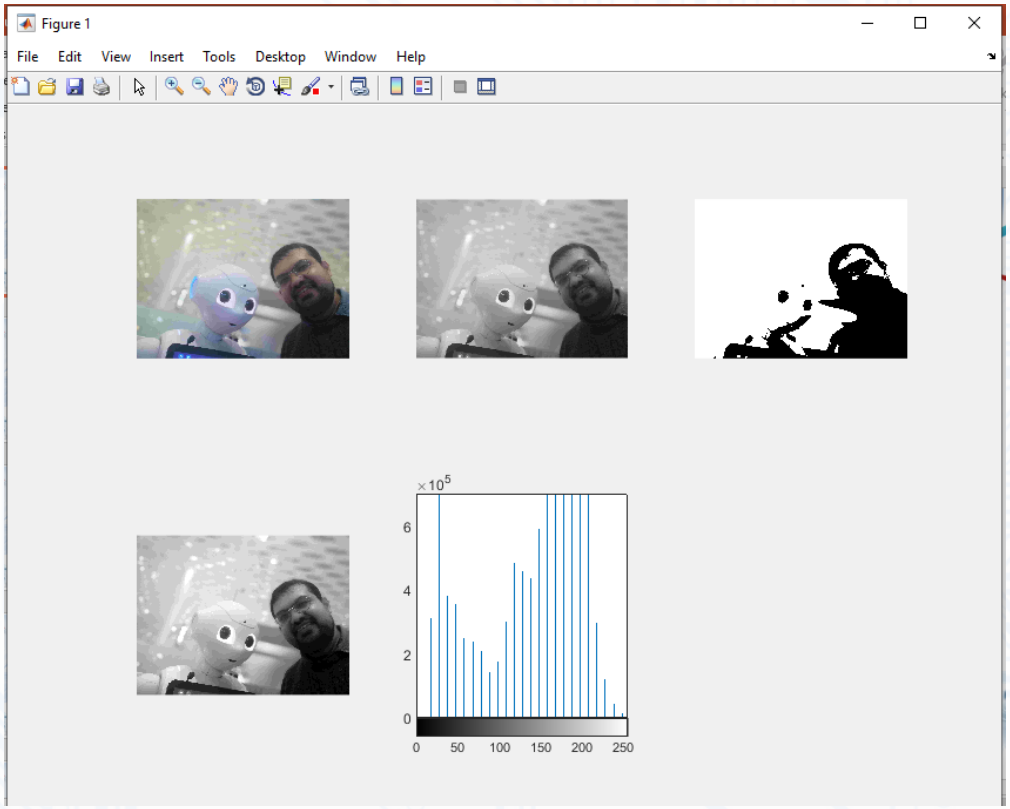
Gaussian filtering 高斯滤波

```
A=imread('cameraman.tif'); % read in an image
[rows dims]=size(A);        % get image dimensions
Abuild=zeros(size(A));     % construct zero image of equal size
                           %Randomly sample 1% of points only and convolve with gaussian PSF
sub=rand(rows.*dims,1)<0.01;
Abuild(sub)=A(sub);
h=fspecial('gaussian',[10 10],2);
B10=filter2(h,Abuild);
subplot(1,2,1), imagesc(Abuild); axis image; axis off;colormap(gray); title('Object points')
subplot(1,2,2), imagesc(B10); axis image; axis off;colormap(gray); title('Response of LSI system')
```



One review example

```
a = imread('F:\ATA.jpg');
subplot(2,3,1);imshow(a);
 b = rgb2gray(a);
subplot(2,3,2);imshow(b);
 c = im2bw(a);
subplot(2,3,3);imshow(c);
 d = imadjust(b);
subplot(2,3,4);imshow(d);
 e = a;
e=rgb2gray(e);
subplot(2,3,5);imhist(e);
 imfinfo('ATA.jpg')
[height, width, colour_planes] = size(a)
 %colormap('spring')
```



fspecial
Create predefined 2-D filter

fspecial 创建预定义的二维过滤器

h = fspecial(type)

h = fspecial('average',hsize)

h = fspecial('disk',radius)

h = fspecial('gaussian',hsize,sigma)

h = fspecial(type)

creates a two-dimensional filter h of the specified type. Some of the filter types have optional additional parameters, shown in the following syntaxes. fspecial returns h as a correlation kernel, which is the appropriate form to use with imfilter.

h = fspecial('average',hsize)

returns an averaging filter h of size hsize.

h = fspecial('disk',radius)

returns a circular averaging filter (pillbox) within the square matrix of size 2*radius+1.

h = fspecial('gaussian',hsize,sigma)

returns a rotationally symmetric Gaussian lowpass filter of size hsize with standard deviation sigma.

Not recommended.

Use imgaussfilt or imgaussfilt3 instead.

创建预定义的二维过滤器

h = fspecial(type)

h = fspecial('average',hsize)

h = fspecial('disk',radius)

h = fspecial('gaussian',hsize,sigma)

h = fspecial(type)

创建指定类型的二维筛选器h。 一些过滤器类型有可选的附加参数，如下面的语法所示。 Fspecial返回h作为相关核，这是imfilter使用的合适形式。

h = fspecial('average',hsize)

返回大小为hsize的平均滤波器h。

h = fspecial('disk',radius)

返回大小为2*半径+1的方阵内的圆形平均滤波器(药盒)。

h = fspecial('gaussian',hsize,sigma)

返回一个大小为hsize、标准差为sigma的旋转对称高斯低通滤波器。
不推荐。
使用imgaussfilt或imgaussfilt3代替 .

h = fspecial(……

h = fspecial('laplacian',alpha)

h = fspecial('log',hsize,sigma)

h = fspecial('motion',len,theta)

h = fspecial('laplacian',alpha)

returns a 3-by-3 filter approximating the shape of the two-dimensional Laplacian operator, alpha controls the shape of the Laplacian.

返回一个近似于二维拉普拉斯算子形状的3 × 3滤波器，alpha控制拉普拉斯算子的形状。

h = fspecial('log',hsize,sigma)

returns a rotationally symmetric Laplacian of Gaussian filter of size hsize with standard deviation sigma.

返回一个大小为hsize的带有标准差sigma的高斯滤波器的旋转对称拉普拉斯算子。

h = fspecial('motion',len,theta)

returns a filter to approximate, once convolved with an image, the linear motion of a camera. len specifies the length of the motion and theta specifies the angle of motion in degrees in a counter-clockwise direction. The filter becomes a vector for horizontal and vertical motions. The default len is 9 and the default theta is 0, which corresponds to a horizontal motion of nine pixels.

返回一个近似的过滤器，一旦与图像卷积，相机的线性运动。 Len表示运动的长度，theta表示逆时针方向的角度。 过滤器成为水平和垂直运动的矢量。 默认的len是9，默认的theta是0，这对应于9个像素的水平运动

h = fspecial(……

h = fspecial('prewitt’)

h = fspecial('sobel')

h = fspecial('prewitt’)

returns a 3-by-3 filter that emphasizes horizontal edges by approximating a vertical gradient. To emphasize vertical edges, transpose the filter h’.

返回一个3乘3的过滤器，它通过近似垂直梯度来强调水平边缘。 为了强调垂直边缘，调换滤波器h’。

[1 1 1
0 0 0
-1 -1 -1]

h = fspecial('sobel’)

returns a 3-by-3 filter that emphasizes horizontal edges using the smoothing effect by approximating a vertical gradient.

To emphasize vertical edges, transpose the filter h'.

[1 2 1
0 0 0
-1 -2 -1]

返回一个3乘3的过滤器，它通过近似垂直梯度的平滑效果来强调水平边缘。为了强调垂直边缘，调换滤波器h’。

h = fspecial(……

h = fspecial(type) creates a two-dimensional filter **h** of the specified type.

- It returns **h** as a correlation kernel, which is the appropriate form to use with **imfilter()**.
- The **type** is a string having one of these values:

average'	**Averaging filter**
'disk'	**Circular averaging filter (pillbox)**
'gaussian'	**Gaussian lowpass filter. Not recommended. Use imgaussfilt or imgaussfilt3 instead.**
'laplacian'	**Approximates the two-dimensional Laplacian operator**
'log'	**Laplacian of Gaussian filter**
'motion'	**Approximates the linear motion of a camera**
'prewitt'	**Prewitt horizontal edge-emphasizing filter**
'sobel'	**Sobel horizontal edge-emphasizing filter**

originalRGB

average filter

h = fspecial(……

originalRGB

h = fspecial(type) 创建指定类型的二维过滤器 h。

它返回 h 作为相关内核，这是与 imfilter() 一起使用的适当形式。

类型是具有以下值之一的字符串：

average'	**Averaging filter**
'disk'	**Circular averaging filter (pillbox)**
'gaussian'	**Gaussian lowpass filter. Not recommended. Use imgaussfilt or imgaussfilt3 instead.**
'laplacian'	**Approximates the two-dimensional Laplacian operator**
'log'	**Laplacian of Gaussian filter**
'motion'	**Approximates the linear motion of a camera**
'prewitt'	**Prewitt horizontal edge-emphasizing filter**
'sobel'	**Sobel horizontal edge-emphasizing filter**

average filter

imfilter

N-D filtering of multidimensional images

多维图像的 N-D 滤波

B = imfilter(A,h)

B = imfilter(A,h)

filters the multidimensional array A with the multidimensional filter h and returns the result in B.

You optionally can filter a multidimensional array with a 2-D filter using a GPU (requires Parallel Computing Toolbox™).

使用多维过滤器h过滤多维数组A，并返回结果在B中。

您可以选择使用GPU用2d过滤器过滤多维数组(需要并行计算工具箱™)。

B = imfilter(A,h,options,...)

B = imfilter(A,h,options,...)

performs multidimensional filtering according to one or more specified options.

根据一个或多个指定选项执行多维筛选。

h = fspecial('average', …);

```
clc
clear all
close all
originalRGB = imread('peppers.png');
imshow(originalRGB)
h = fspecial('average', 25);
X=imfilter(originalRGB,h)
subplot(1,2,1),imshow(originalRGB);title('originalRGB')
subplot(1,2,2),imshow(X);title('average filter')
```

Averaging filter

平均滤波器

h = fspecial('disk', …);

```
clc
clear all
close all
originalRGB = imread('peppers.png');
imshow(originalRGB)
h = fspecial('disk', 25);
X=imfilter(originalRGB,h)
subplot(1,2,1),imshow(originalRGB);title('originalRGB')
subplot(1,2,2),imshow(X);title('disk filter')
```

Circular averaging filter (pillbox)

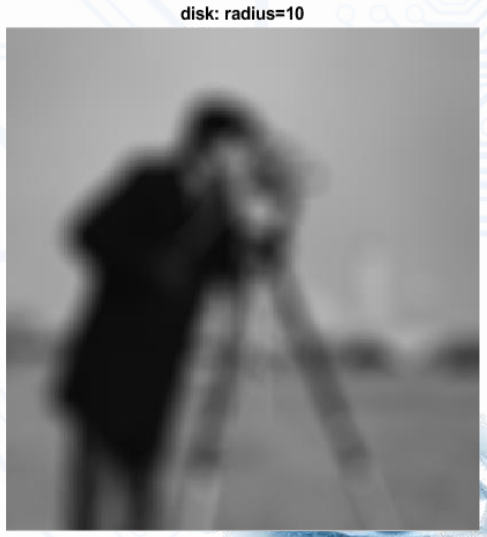
循环平均滤波器

h = fspecial('disk', …);

```
clc
clear all
close all
 I = imread('cameraman.tif');
radius = 1;
J1 = fspecial('disk', radius);
K1 = imfilter(I,J1,'replicate');
radius = 10;
J10 = fspecial('disk', radius);
K10 = imfilter(I,J10,'replicate');
subplot(131);imshow(I);title('original');
subplot(132);imshow(K1);title('disk: radius=1');
subplot(133);imshow(K10);title('disk: radius=10');
```

Different value for Disk

磁盘的不同值



h = fspecial('gaussian', …,…);

```
clc
clear all
close all
originalRGB = imread('peppers.png');
imshow(originalRGB)
h = fspecial('gaussian', 15,10);
X=imfilter(originalRGB,h)
subplot(1,2,1),imshow(originalRGB);title('originalRGB')
subplot(1,2,2),imshow(X);title('gaussian')
```

originalRGB

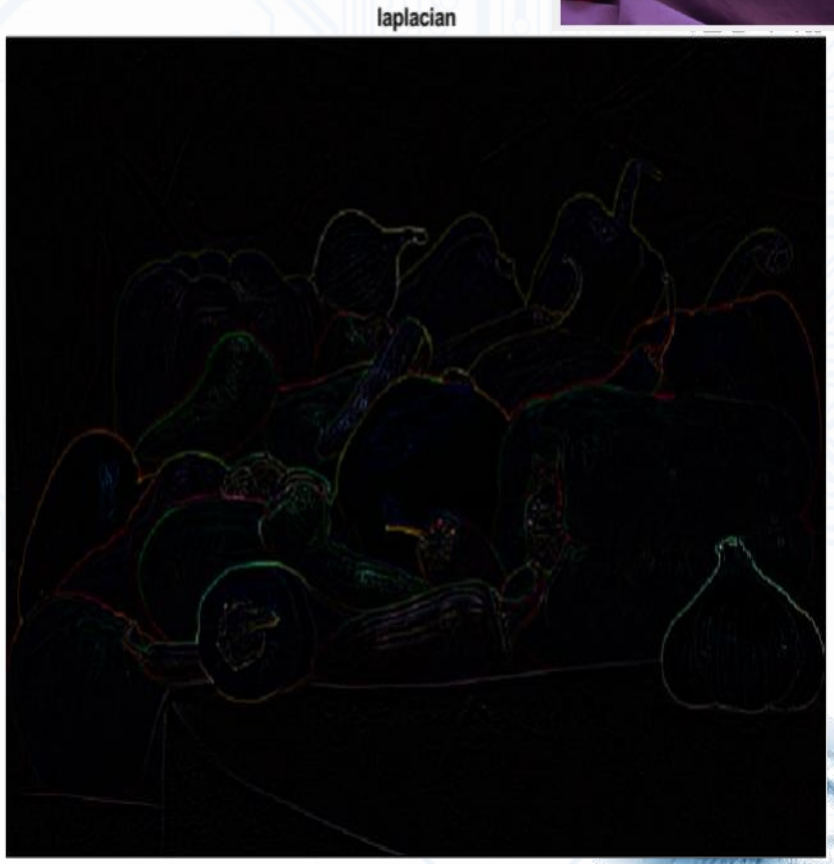
gaussian

Lowpass Gaussian filter

高斯低通滤波器

h = fspecial('laplacian', …);

```
clc
clear all
close all
originalRGB = imread('H:\chess.jpg');
imshow(originalRGB)
h = fspecial('laplacian', 0.5);
X=imfilter(originalRGB,h)
subplot(1,2,1),imshow(originalRGB);title('originalRGB')
subplot(1,2,2),imshow(X);title('laplacian')
```

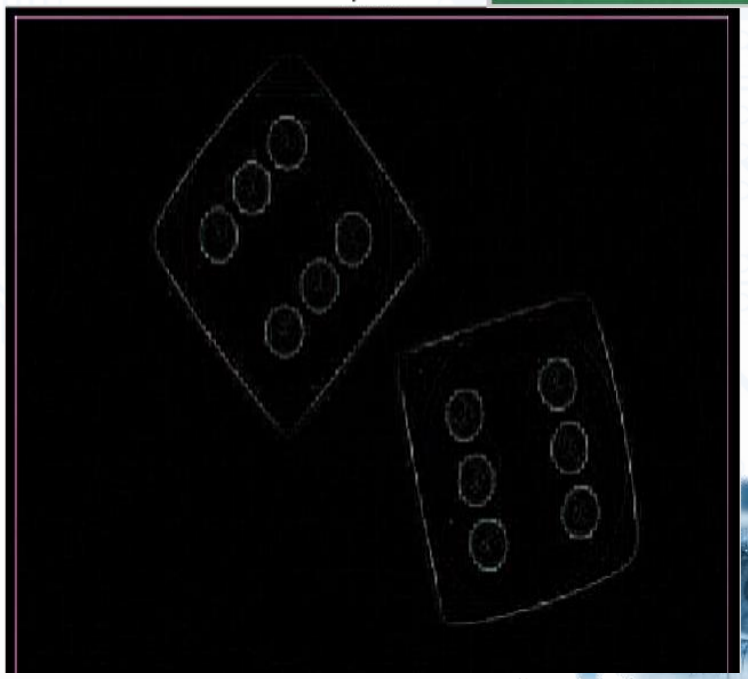


Laplacian filter

拉普拉斯算子的过滤器

h = fspecial('laplacian', …);

```
clc
clear all
close all
originalRGB = imread('H:\chess.png');
imshow(originalRGB)
h = fspecial('laplacian', 0.5);
X=imfilter(originalRGB,h)
subplot(1,2,1),imshow(originalRGB);title('originalRGB')
subplot(1,2,2),imshow(X);title('laplacian')
```

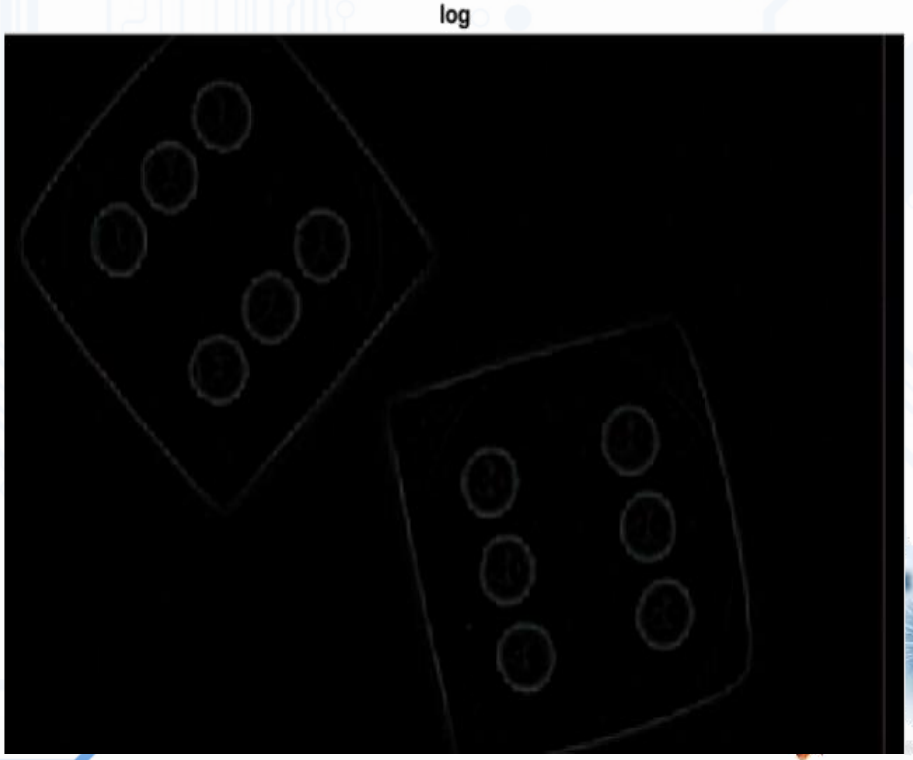


Laplacian filter

拉普拉斯算子的过滤器

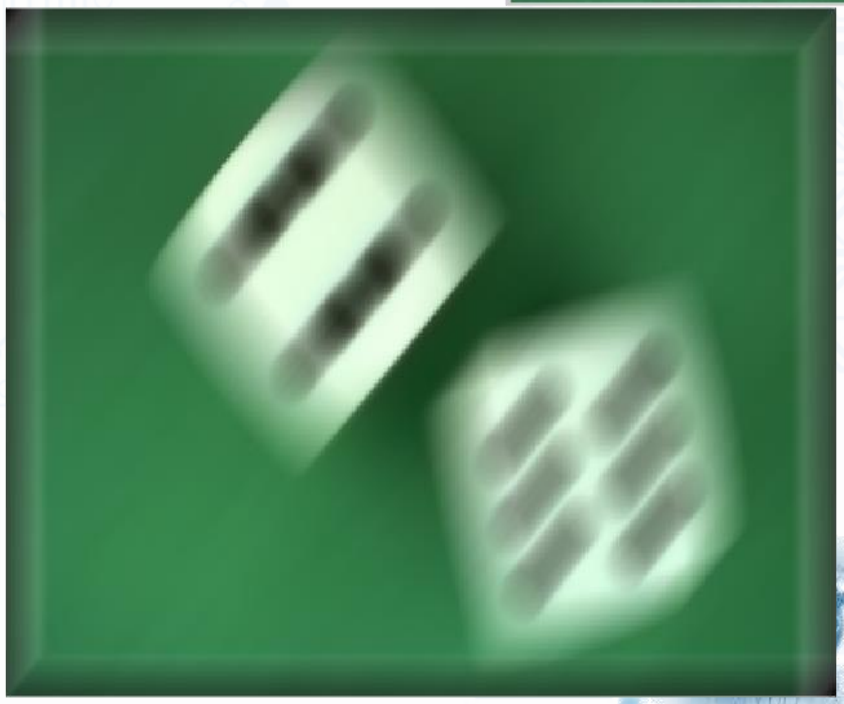
h = fspecial('log', .,,..);

```
clc
clear all
close all
originalRGB = imread('H:\chess.png');
imshow(originalRGB)
h = fspecial('log', 70,1);
X=imfilter(originalRGB,h)
subplot(1,2,1),imshow(originalRGB);title('originalRGB')
subplot(1,2,2),imshow(X);title('log')
```



h = fspecial('motion', ..,…);

```
clc
clear all
close all
originalRGB = imread('H:\chess.png');
imshow(originalRGB)
h = fspecial('motion', 50,45);
X=imfilter(originalRGB,h)
subplot(1,2,1),imshow(originalRGB);title('originalRGB')
subplot(1,2,2),imshow(X);title('log')
```

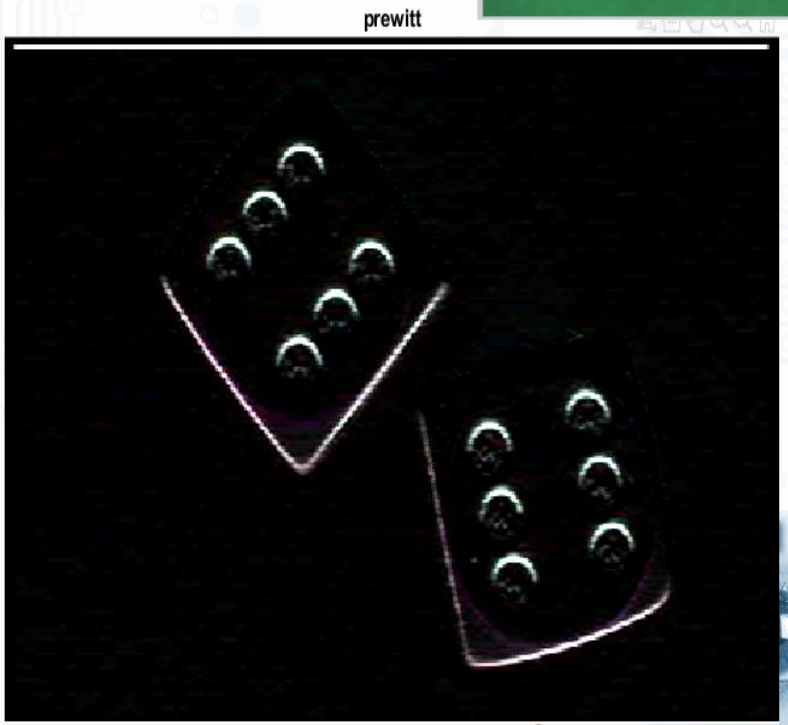


Movement filter

运动滤波器

h = fspecial('prewitt');

```
clc
clear all
close all
originalRGB = imread('H:\chess.png');
imshow(originalRGB)
h = fspecial('prewitt');
X=imfilter(originalRGB,h)
subplot(1,2,1),imshow(originalRGB);title('originalRGB')
subplot(1,2,2),imshow(X);title('prewitt')
```

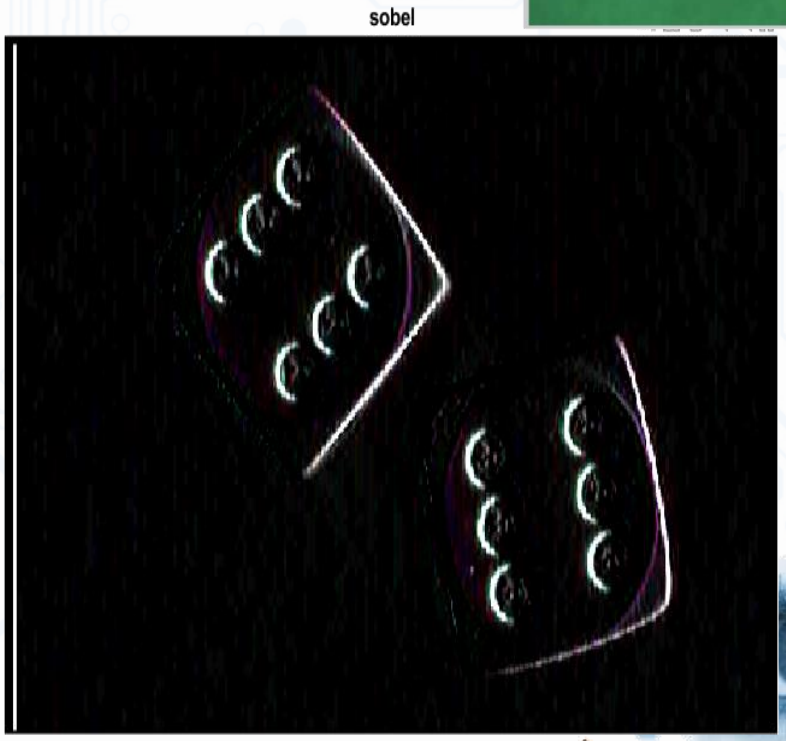


Edge improvement filter

边改进滤波器

h = fspecial('sobel');

```
clc
clear all
close all
originalRGB = imread('H:\chess.png');
imshow(originalRGB)
h = fspecial('sobel');
X=imfilter(originalRGB,rot90(h))
subplot(1,2,1),imshow(originalRGB);title('originalRGB')
subplot(1,2,2),imshow(X);title('sobel')
```

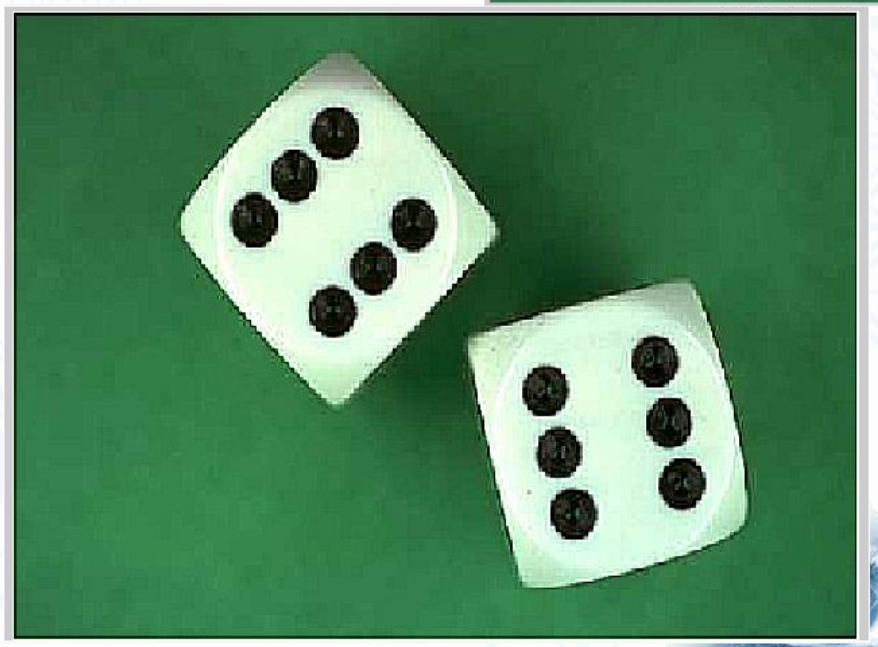


Vertical and horizontal edge

垂直和水平边

h = fspecial('unsharp',….);

```
clc
clear all
close all
originalRGB = imread('H:\chess.png');
imshow(originalRGB)
h = fspecial('unsharp',0.9);
X=imfilter(originalRGB,h)
subplot(1,2,1),imshow(originalRGB);title('originalRGB')
subplot(1,2,2),imshow(X);title('unsharp')
```



Increment of edge and light intensity

边缘和光强度的增加

Reference 参考

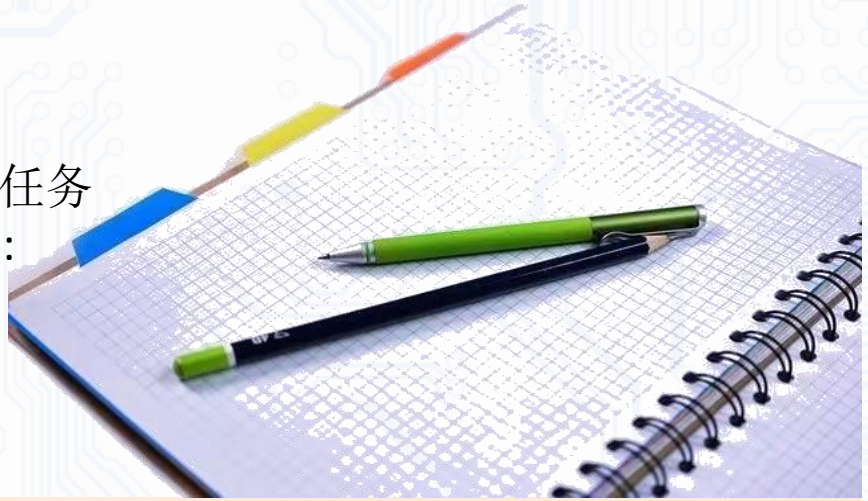
1. www.mathworks.com/products/**image**/ www.mathtools.net/**MATLAB**/**Imaqe Processing**/
2. www.amath.colorado.edu/courses/4720/2000Spr/Labs/Worksheets/**Matlab** tutorial/**matlab**impr.html www.**imaqeprocessinq**place.com/DIPUM/dipum book description/book description.htm

Student Task_7: DIP

- Repeat all the example in this PPT with your personal photo用你的个人照片重复PPT中的所有例子

Send for Next lecture

请听下一节课



您有时间在北京时间上午9点之前发送任务
将文件以PPT(PPT格式)发送至以下邮箱:
drajm@ yahoo.com
您的文件应该具有这种格式的名称
<任务号><学生名><学生ID>.ppt

- **You have time to send your task before 9 am (bejing time) of lecture**
- **Send the file in PPT(power point format) to this email :**

drajm@ yahoo.com

- **Your file should have this format of name**

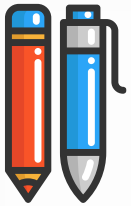
<Task number><student name><Student ID>.ppt

THE DISTANCE
BETWEEN YOUR
DREAMS AND REALITY
IS DISCIPLINE.

江西理工大学
Jiangxi University of Science and Technology
信息工程学院
School of information engineering

Digital Image Processing

THANK YOU



"BE HUMBLE. BE HUNGRY. AND ALWAYS BE THE HARDEST WORKER IN THE ROOM."