Digital Image Processing

Image Enhancement in the Spatial Domain

Low and High Pass Filtering

Topics

- Low Pass Filtering
  - Averaging
  - Median Filter
- High Pass Filtering
  - Edge Detection
- Line Detection

Low Pass Filtering

- Low pass filters block high frequency content of the image
- High frequency content correspond to boundaries of the objects

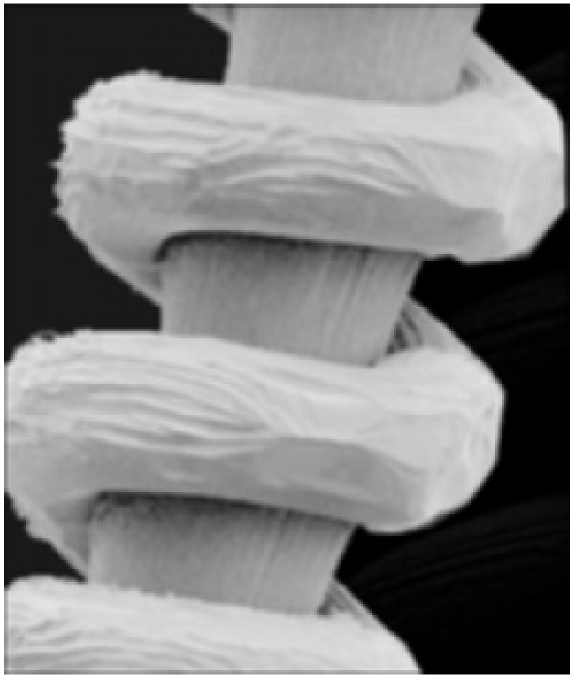


Image Averaging

- Image averaging is obtained by finding the average of K images. It is applied in de-noising the images.

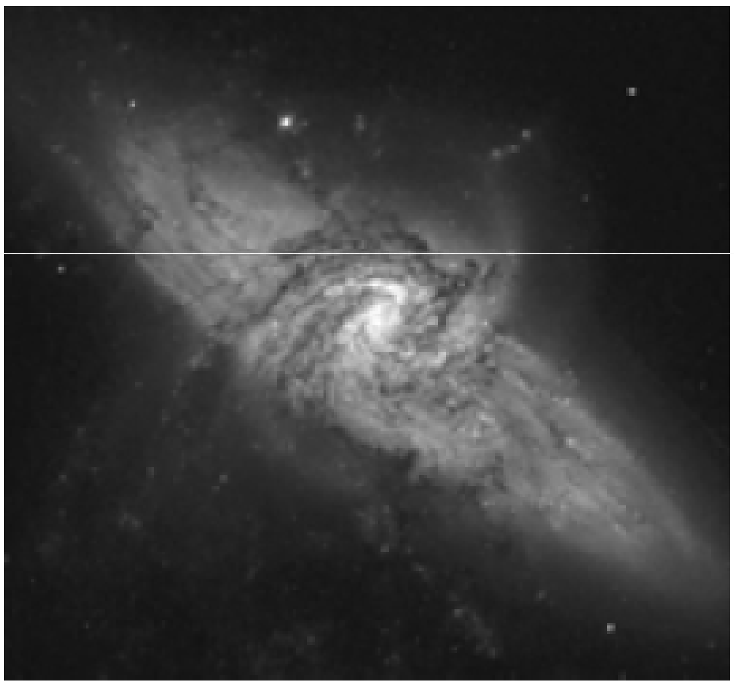
$$\bar{g}(x, y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x, y)$$

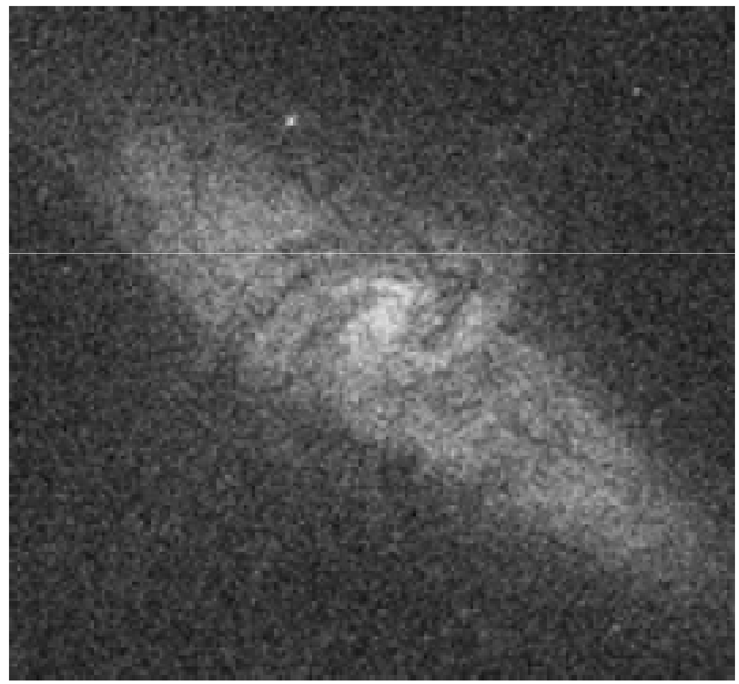
- A noisy image is defined by:

$$g(x, y) = f(x, y) + \eta(x, y)$$

Assuming that the noise is uncorrelated with zero mean

Image Averaging





Average Images (8, 16, 64, and 128)

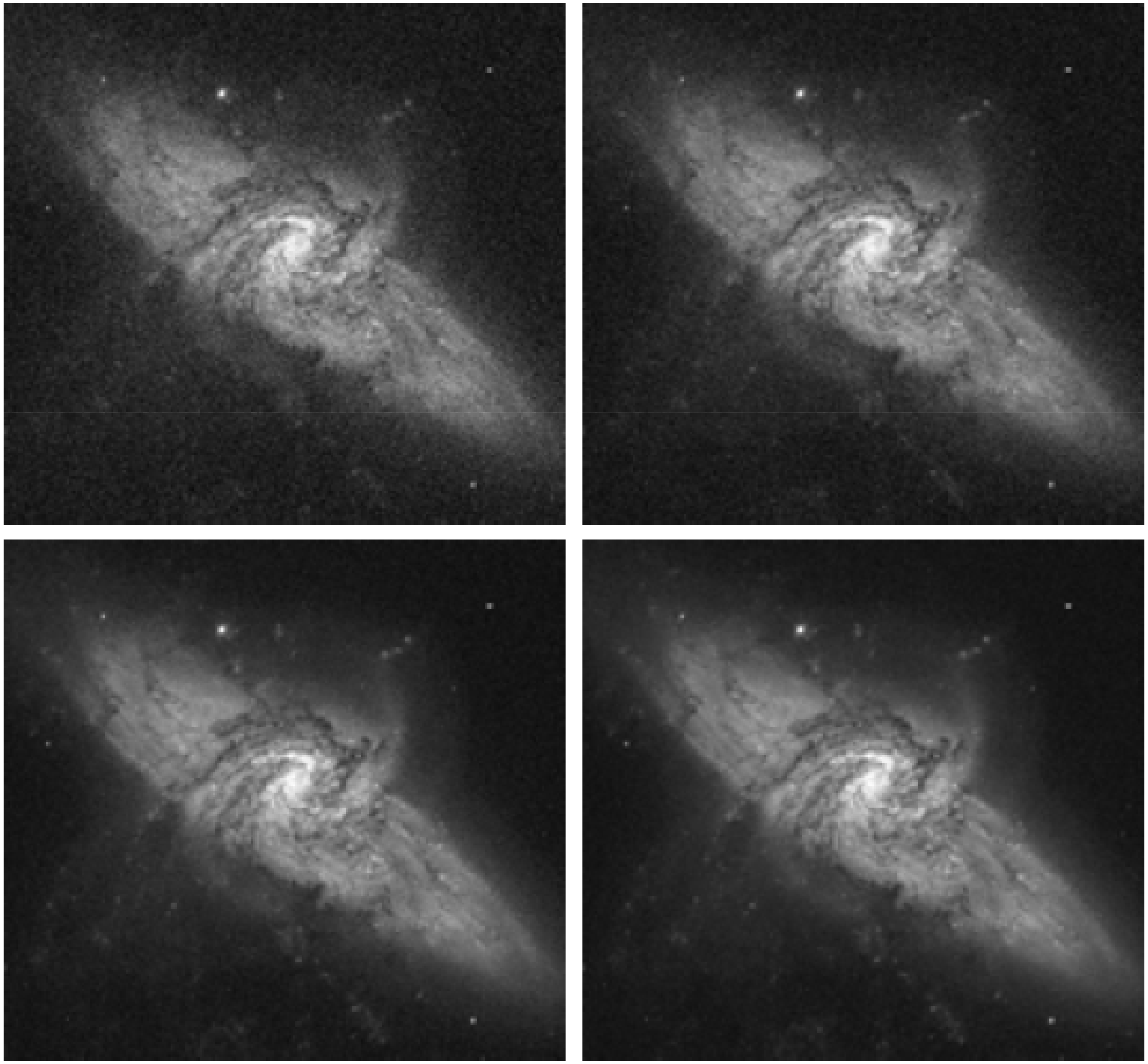
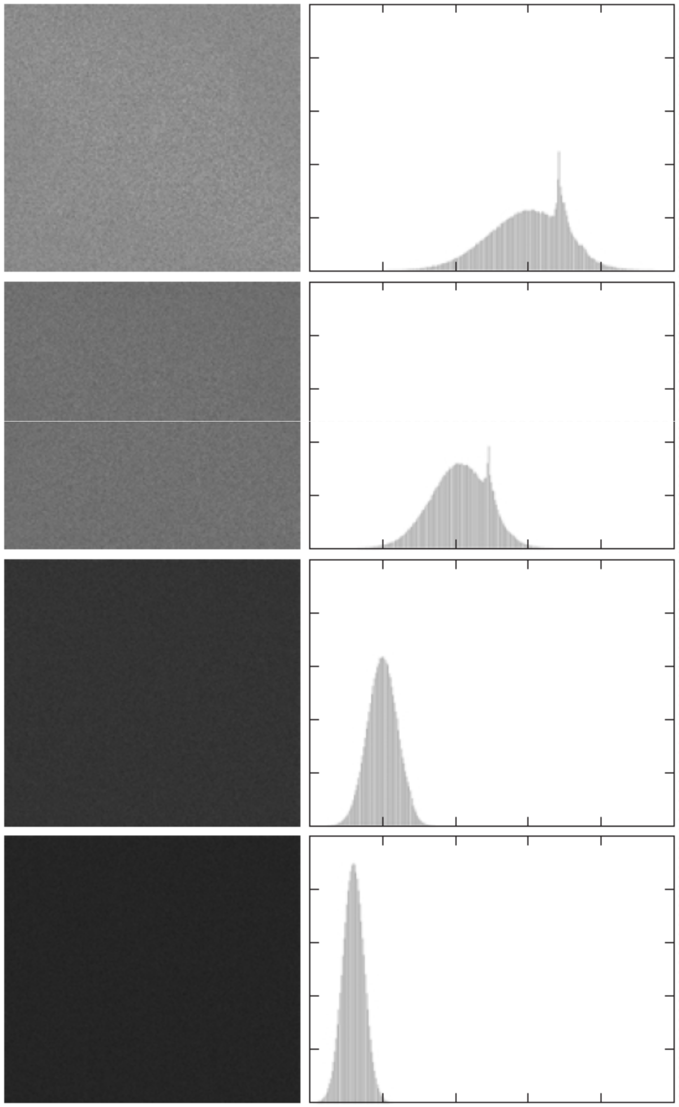


Image averaging in MATLAB

- Open the first image file   im1 = imread('test1.bmp');
- Open the second image file   im2 = imread('test2.bmp');
- Open the third image file   im3 = imread('test3.bmp');
- Find the average imAvg = (im1 + im2 + im3)/3;
- Display the image imshow(imAvg)

De-noising Effect on Histogram

Difference of Original Image and Average Images

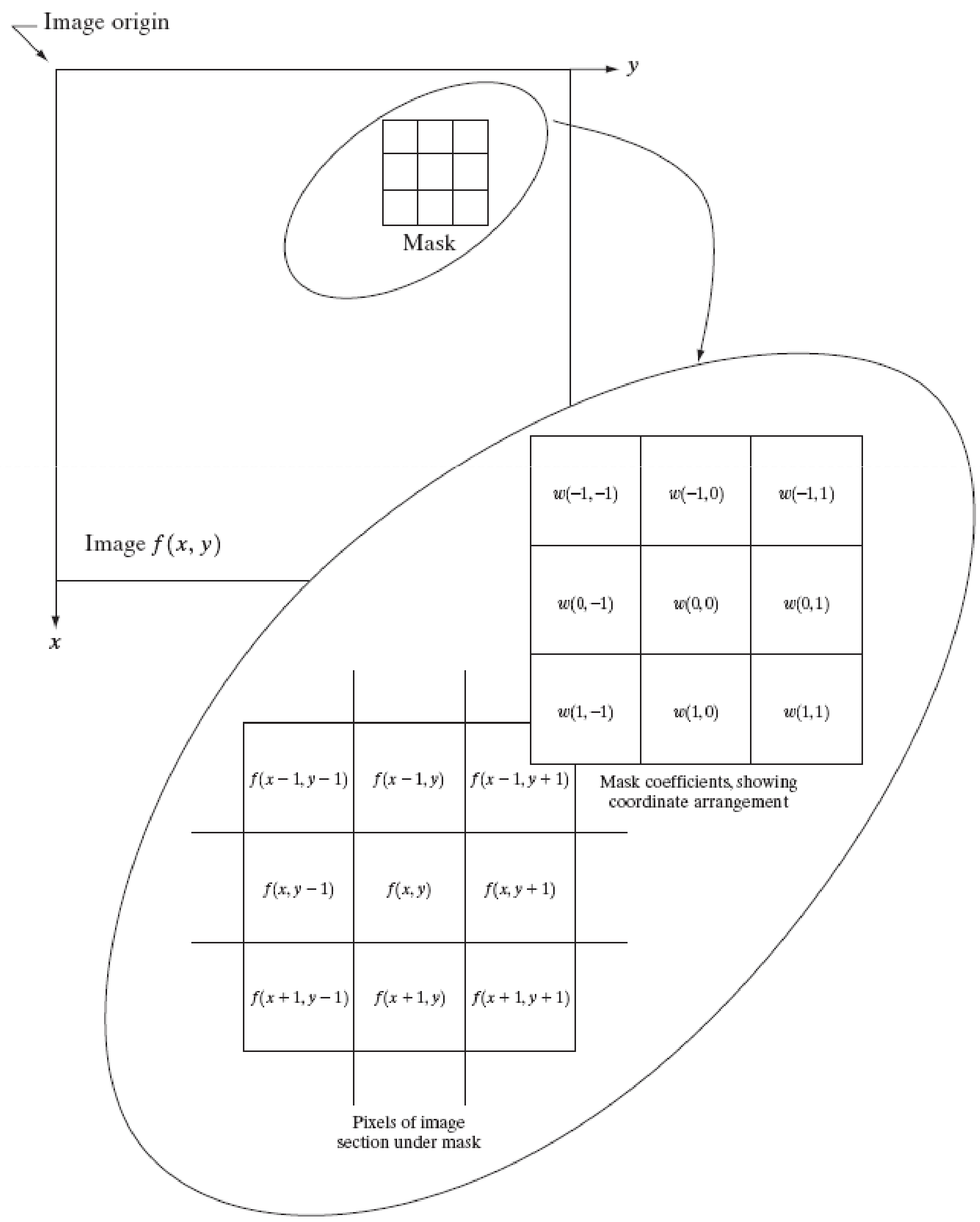


Corresponding Histograms

Spatial Filters

- To work on pixels in the neighborhood of a pixel, a sub-image is defined.
- The operation on the sub-image pixels is defined using a ***mask*** or ***filter*** with the same dimensions.
- Applying the operation to the image is referred to as convolution

Spatial Masks



Convolving Mask with Image

- Convolving mask with image is carried out by sliding the mask over the image, multiplying mask values with the pixel values falling beneath them and obtaining the sum.

$$g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t)$$

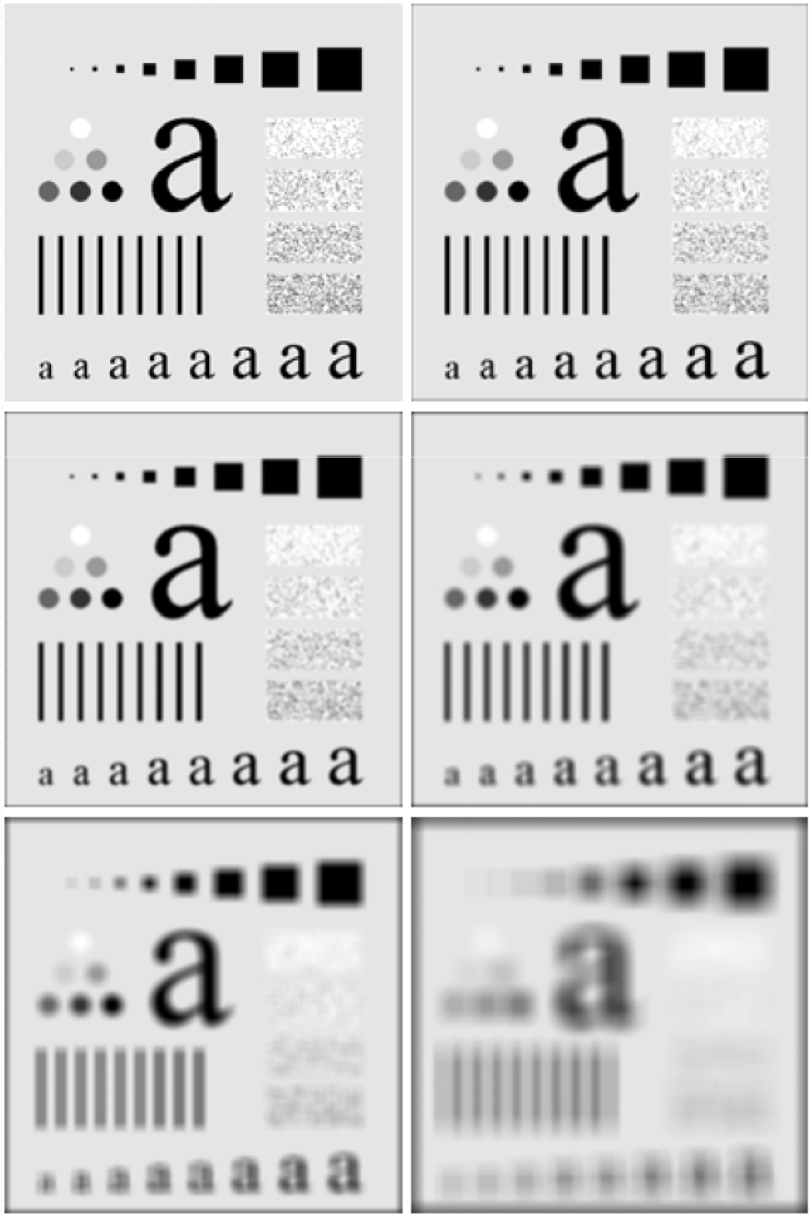
- The sum is used as the value for the position of the center of the mask over the image

Mean or Average Filter

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

$$g(x, y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t)}$$

The Effect of Average Filter

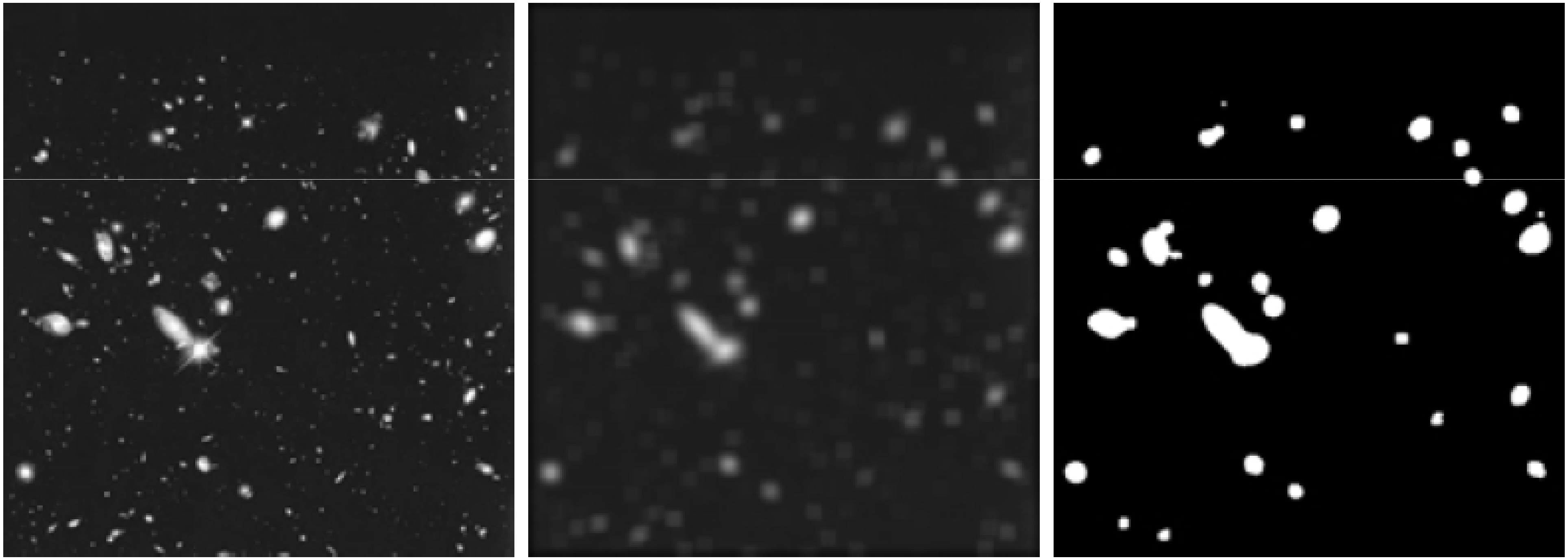


n=3, 5, 9, 15, and 35

Mean Filter in MATLAB

- I = imread(‘test.tif’);
- H = fspecial(‘average’, 3);
- I2 = imfilter(I, H);
- imshow(I), title(‘Original Image’)
- figure, imshow(I2), title(‘Filtered image’)

Mean Filter with Thresholding

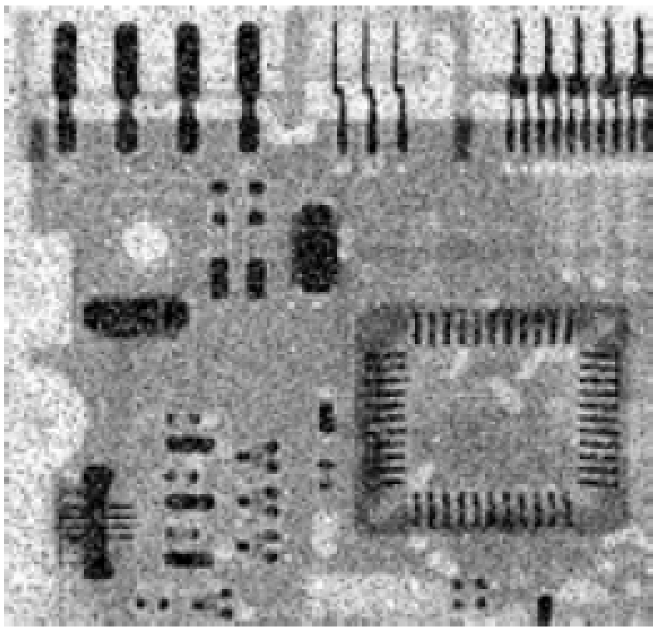


Median Filter

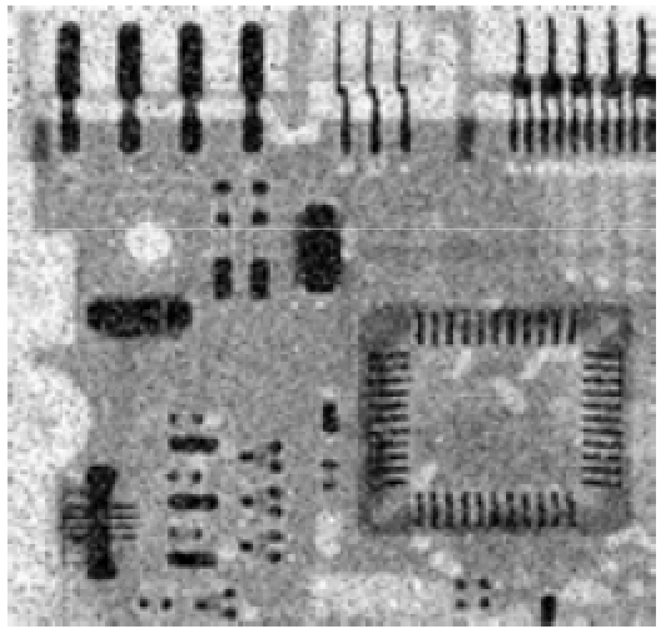
- Median filter replaces the pixel at the center of the filter with the median value of the pixels falling beneath the mask.

- Median filter does not blur the image but it rounds the corners.

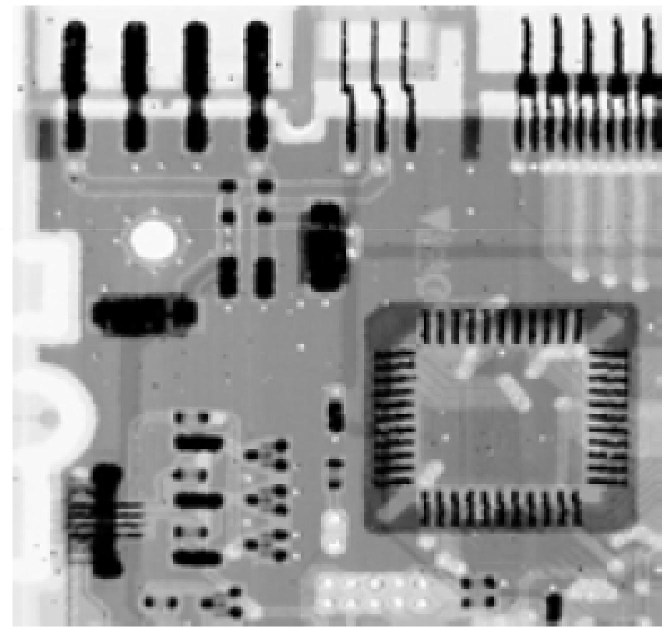
Median Filter Effect



Original Image



Mean Filtered



Median Filtered

Median Filter in MATLAB

- img = imread('test.tif');
- imSmoothed = medfilt2(img, [3 3]);

- [3 3] defines the dimension of the filter

High-pass or Sharpening Filters

- High pass filters let the high frequency content of the image pass through the filter and block the low frequency content.
- High pass filters can be modeled by first order derivative as :

$$\frac{\partial f}{\partial x} = f(x+1) - f(x).$$

- A second order derivative can also be used for extracting high frequency data

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x).$$

Use of First Derivative - Gradient

- Gradient is a vector which points in the direction of the greatest rate of increase of the scalar field, and whose magnitude is the greatest rate of change.

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

$$\begin{aligned} \nabla f &= \text{mag}(\nabla \mathbf{f}) \\ &= \left[G_x^2 + G_y^2\right]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2} \end{aligned}$$

Edge Detection Filters

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

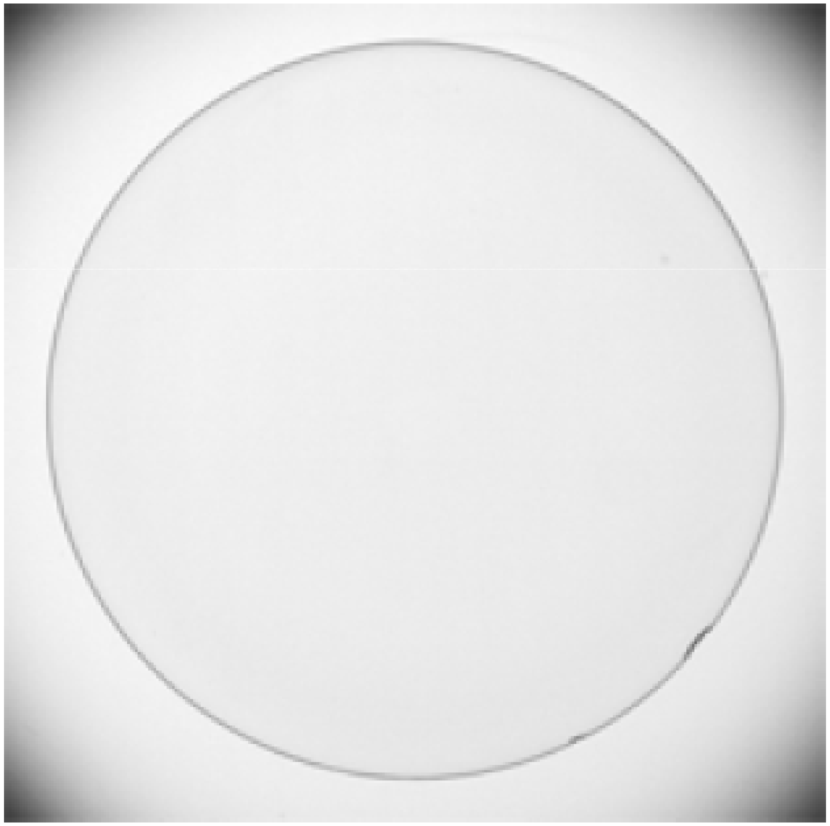
−1	0
0	1

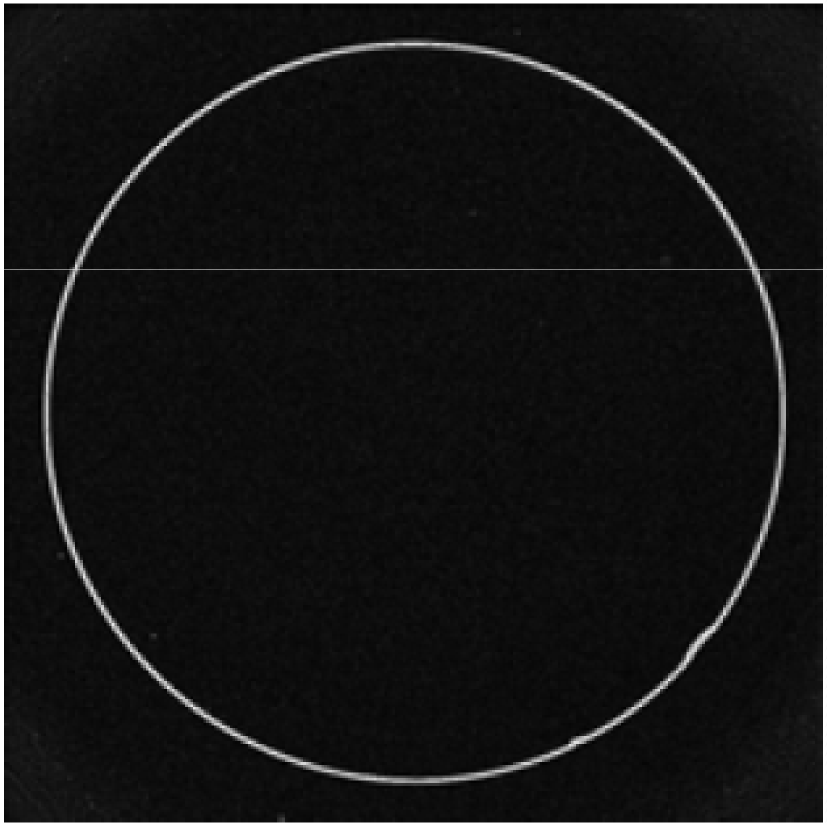
0	−1
1	0

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

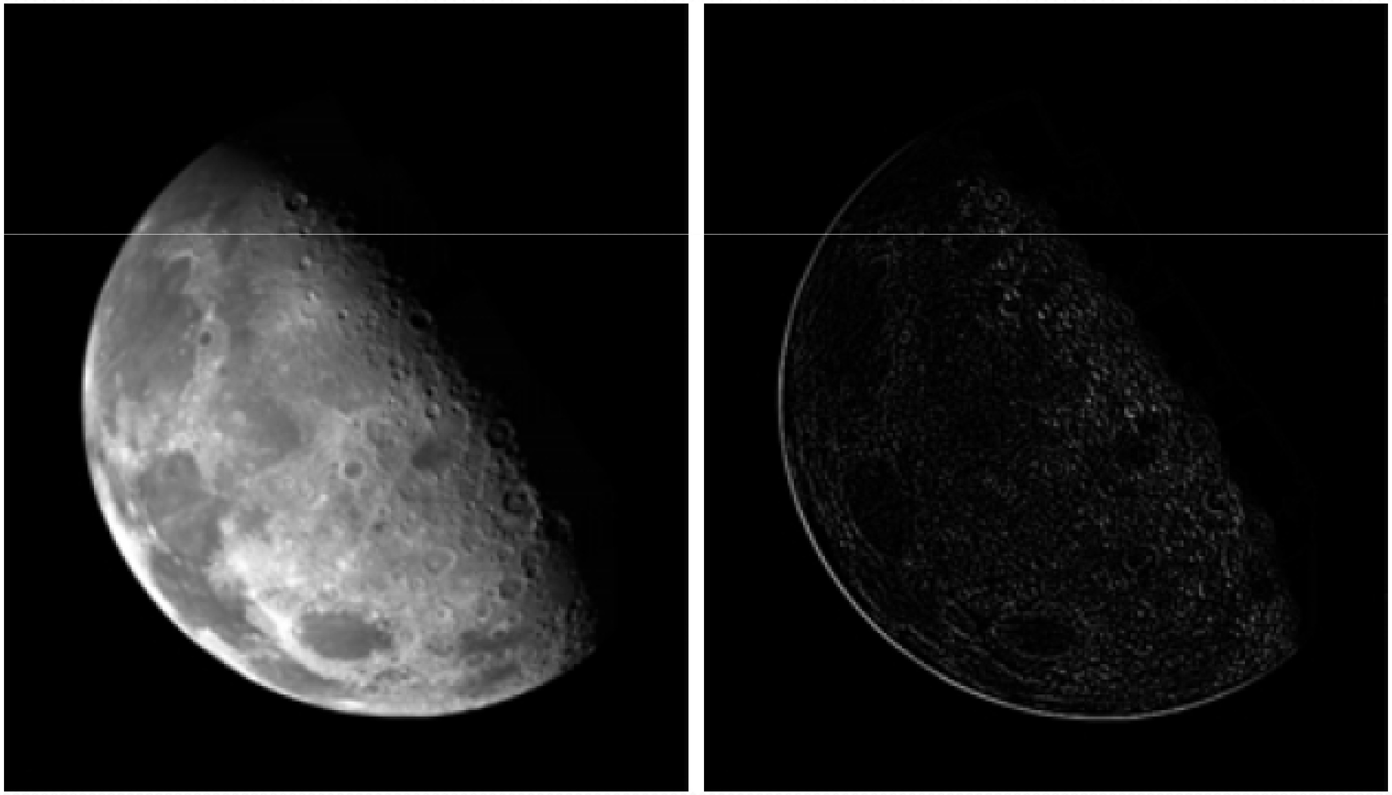
Edge Detection Example





Edge Image Example

Edge Image Example



Edge Detection in MATLAB

- Img = imread('test.tif');
- edgeImg = edge(Img, 'sobel');
- imshow(edgeImg)

Line Detection Filters

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

−1	−1	2
−1	2	−1
2	−1	−1

+45°

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

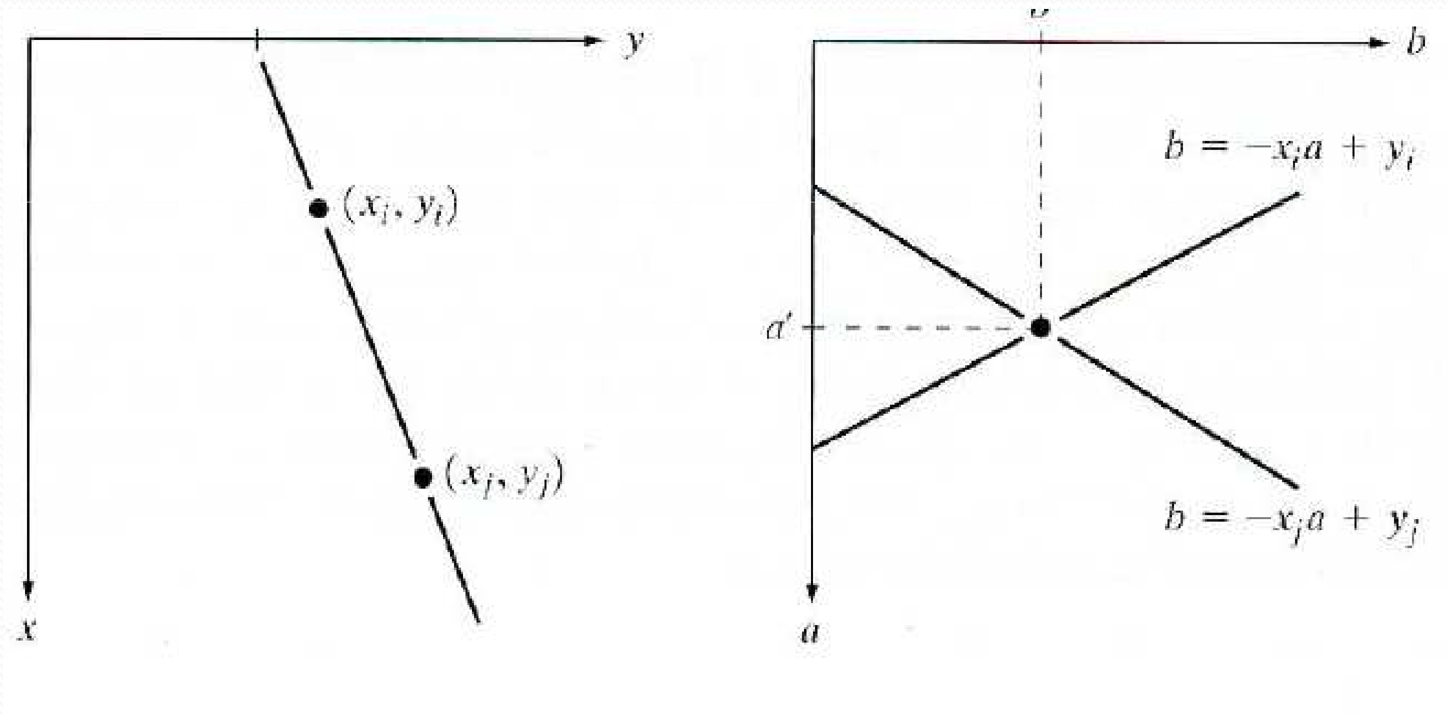
2	−1	−1
−1	2	−1
−1	−1	2

−45°

Is local Processing Adequate?

- Local processing gives information about the local neighborhood of pixels.
- The global characteristics of the pixels cannot be determined from their local characteristics.
- E.g. Line detection using gradient operator can provide the probability of a pixel being on a line but it cannot group the pixels into line segments.

Global Processing Example: Hough Transform



Line Detection using Hough Transform

- Define the line using its parametric equation $y=mx+c$
  - m is the slope of the line, c in the intersection with y axis
- Define a buffer (M, C) and initialize it to zero
- For all edge pixels Do
  - For all combination of m and c values Do
    - If the pixel satisfies $y=m_i x+c_i$ the increment the value of the corresponding buffer location (m_i , c_i)

Exercise

- Apply Hough transform to an image and draw the detected lines on the original image
- Use http://www.mathworks.com/help/toolbox/images/ref/hough.html