Marwadi
University

IMAGE PROCESSING
01CE0507

Unit - 4
Spatial Filters

Prof. Urvi Y. Bhatt
Department of Computer Engineering

Outline

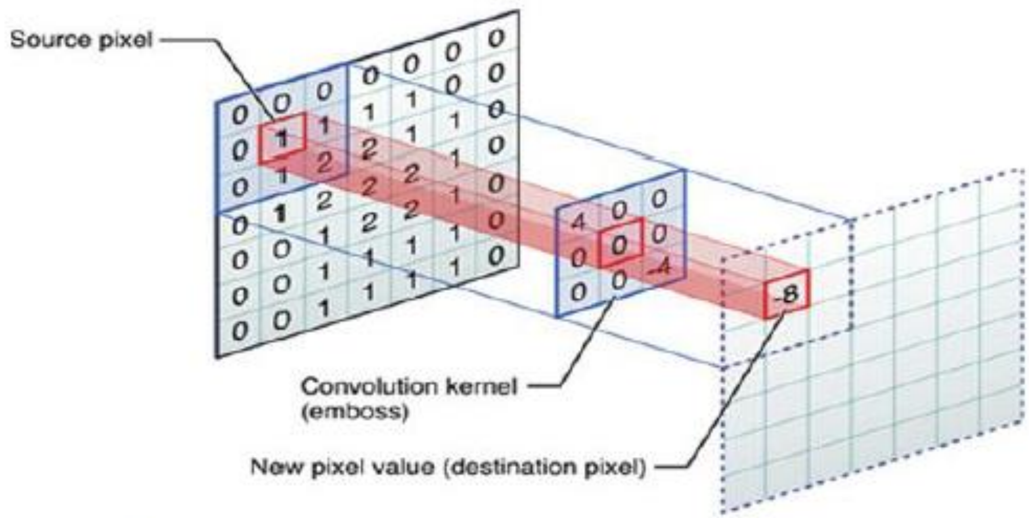
- Spatial Filtering
- Spatial Filtering techniques can be divided into two broad categories:
  - The Smoothing Spatial Filter / Low Pass Filters
  - The Sharpening Filters / High Pass Filters

Spatial Filtering

- Spatial Filtering technique is used directly on pixels of an image.
- Mask is usually considered to be added in size so that it has specific center pixel.
- This mask is moved on the image such that the center of the mask traverses all image pixels.

Process of Spatial Filtering

- The process consists of, Moving the filter mask from point to point in an image
- At each point (x, y) the response of the filter at that point is calculated
- The response is sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask also called as convolution
- So, the linear filtering process is often referred to as “Convolving a mask with an image”



a	*b*	*c*
d	*e*	*e*
f	*g*	*h*

Source Pixels

r	*s*	*t*
u	*v*	*w*
x	*y*	*z*

Kernel

$$New\ pixel = v*e + z*a + y*b + x*c + w*d + u*e + t*f + s*g + r*h$$

Convolution

- Response R of an m × n mask at any point (y), is expressed as follows

$$R = w_1 z_1 + w_2 z_2 + \cdots w_{mn} z_{mn}$$
$$= \sum_{i=1}^{mn} w_i z_i$$

- Example:For the 3×3 general mask, the response at any point (x,y) in the image is given by, $R = w_1 z_1 + w_2 z_2 + \cdots w_9 z_9$

$$= \sum_{i=1}^{9} w_i z_i$$

Correlation and Convolution

- Correlation is the process of passing the mask w by the image array f
- Convolution is the same process, except that w is rotated by 180 Degree prior to passing it by f

	f	w
Correlation	0 0 0 1 0 0 0 0	1 2 3 2 0
Convolution	0 0 0 1 0 0 0 0	0 2 3 2 1

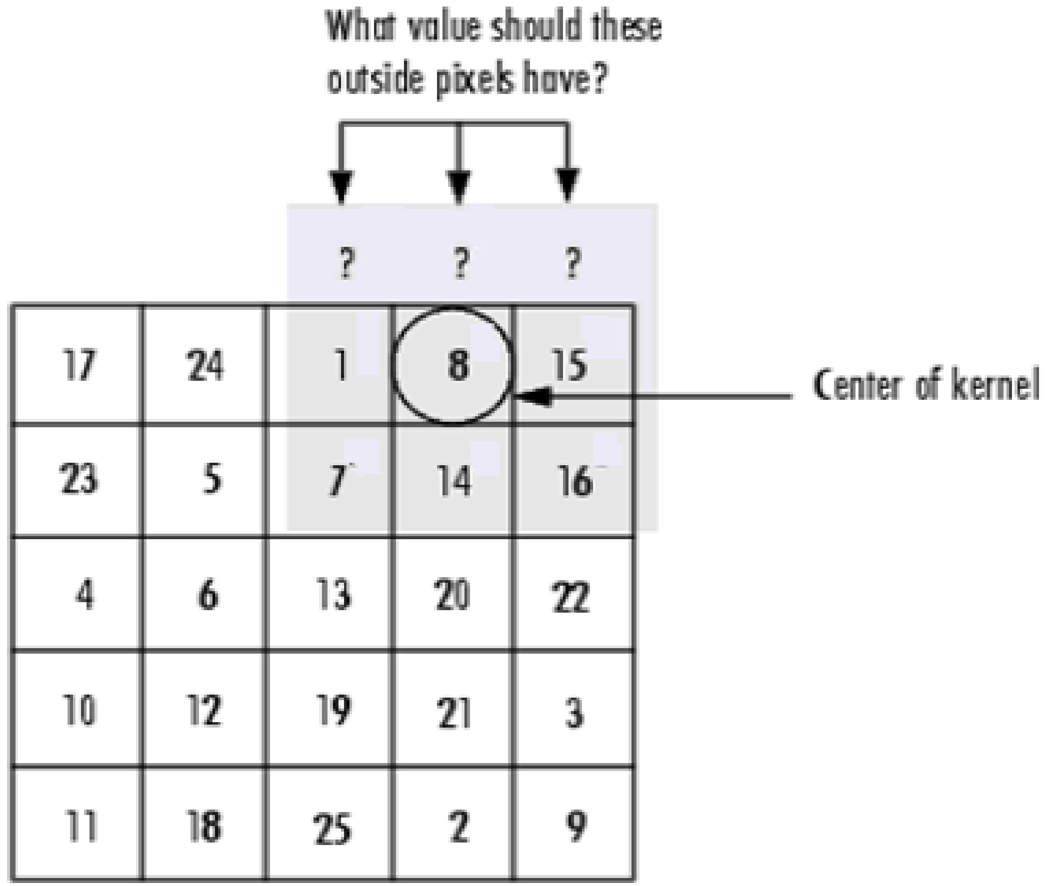
Do not add any bit

- Consider the same matrix in the calculation, do not add any extra bit.

Keeping border values unchanged

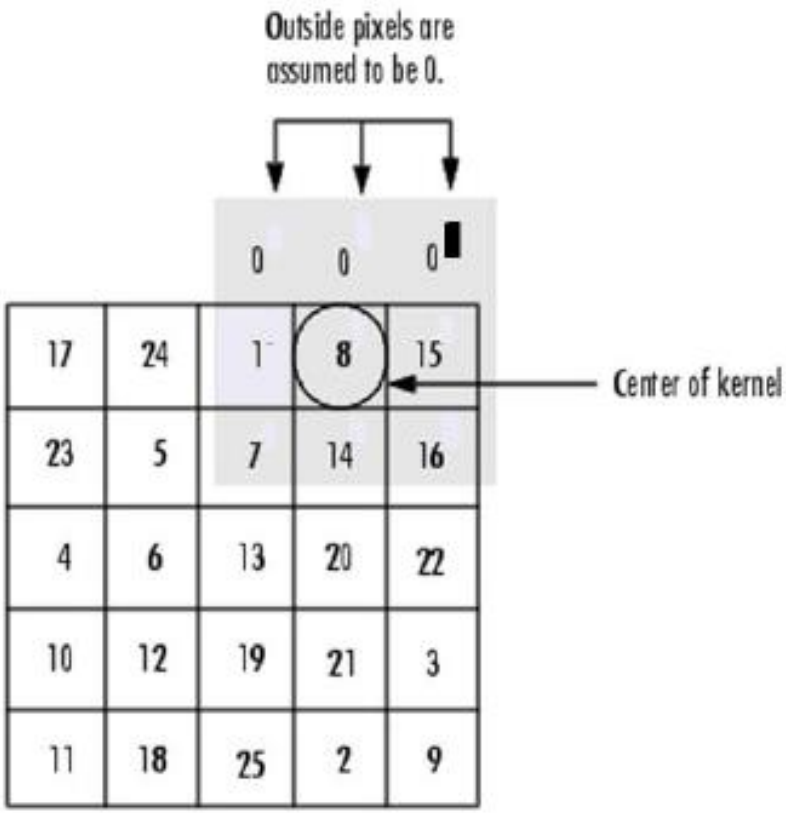
17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	8	25	2	9

What happens when the Values of the Kernel Fall Outside the Image??!



First Solution: Zero Bit padding

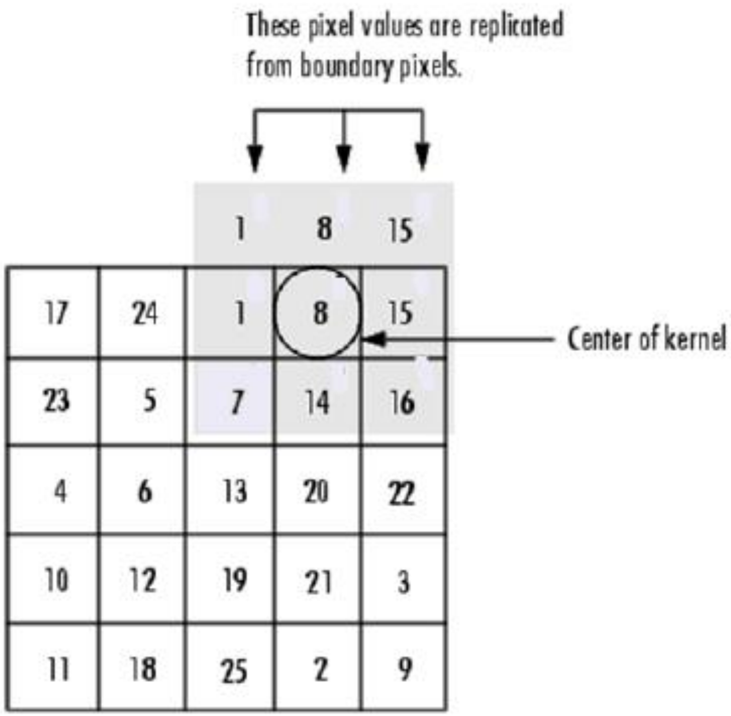
Extending border values outside with the zero value.



0	0	0	0	0	0	0
0	17	24	1	8	15	0
0	23	5	7	14	16	0
0	4	6	13	20	22	0
0	10	12	19	21	3	0
0	11	8	25	2	9	0
0	0	0	0	0	0	0

Second Solution: border padding

Extending border values outside with values at boundary



17	17	24	1	8	15	15
17	17	24	1	8	15	15
23	23	5	7	14	16	16
4	4	6	13	20	22	22
10	10	12	19	21	3	3
11	11	8	25	2	9	9
11	11	8	25	2	9	9

Smoothing Spatial Filter

- Smoothing filter is used for blurring and noise reduction in the image.
- Smoothing filter is also known as Low Pass Filters.
- Blurring is pre-processing steps for removal of small details and Noise Reduction is accomplished by blurring.

Smoothing Spatial Filter (Cont.)

- Pixel averaging in the spatial domain
  - Each pixel in the output is a weighted average of its neighbours
  - Is a convolution whose weight matrix sums to 1
- Types of Smoothing Spatial Filter:
  - Linear Filter / Mean Filter
  - Order Statistics Filter / Non-linear Filter

Linear Filter / Mean Filter

- Linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask.
- The idea is replacing the value of every pixel in an image by the average of the grey levels in the neighborhood define by the filter mask.

Linear Filter / Mean Filter (Cont.)

- Types of Mean Filter:
  - Averaging Filter / Standard Average Filter / Arithmetic Mean Filter
  - Weighted Averaging Filter / Gaussian Filter
  - Geometric Mean
  - Harmonic Mean
  - Contraharmonic Mean

Averaging Filter / Standard Average Filter/ Arithmetic Mean Filter

- It is used in reduction of the detail in image.
- All coefficients are equal.
- The response of Averaging filter is simply average of the pixes contained n neighborhood of the filter mask.

$$\frac{1}{9}\times \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Standard average filter

$$\frac{1}{25}\times \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

Averaging Filter / Standard Average Filter/ Arithmetic Mean Filter (Cont.)

8-Neighbor Mean filter

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

4-Neighbor Mean filter

0	1/5	0
1/5	1/5	1/5
0	1/5	0

1/5	0	1/5
0	1/5	0
1/5	0	1/5

Averaging Filter / Standard Average Filter/ Arithmetic Mean Filter(Cont.)

- The output of Averaging Filters is a smoothed mage with reduced “Sharp” transitions in gray levels.
- Noise and Edges consist of Sharp transitions in Gray Levels
- Thus Smoothing Filters are used for noise reduction.
- However, the have the undesirable side effect that they blur edges.

Averaging Filter / Standard Average Filter (Cont.)

110	120	90	130
91	94	98	200
90	91	99	100
82	96	85	90

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

The mask is moved from point to point in an image. At each point (x,y), the response of the filter is calculated

Standard averaging filter:

(110 +120+90+91+94+98+90+91+99)/9 =883/9 = 98.1

Averaging Filter / Standard Average Filter (Cont.)

- Apply Averaging Filter / Mean Filter. (Consider $N_4(P)$ neighbor.

110	120	90	130
91	94	98	200
90	91	99	100
82	96	85	90

The mask is moved from point to point in an image. At each point (x,y), the response of the filter is calculated

0	1/5	0
1/5	1/5	1/5
0	1/5	0

Averaging Filter / Standard Average Filter/ Arithmetic Mean Filter (Cont.)

- Apply Averaging Filter / Mean Filter. (Consider $N_D(P)$ neighbor.

110	120	90	130
91	94	98	200
90	91	99	100
82	96	85	90

The mask is moved from point to point in an image. At each point (x,y), the response of the filter is calculated

1/5	0	1/5
0	1/5	0
1/5	0	1/5

Averaging Filter / Standard Average Filter (Cont.)

2D Average filtering example using a 3 x 3 sampling window:

Keeping border values unchanged

Average = round(1+4+0+2+2+4+1+0+1)/9 = 2

Input

1	4	0	1	3	1
2	2	4	2	2	3
1	0	1	0	1	0
1	2	1	0	2	2
2	5	3	1	2	5
1	1	4	2	3	0

Output

1	4	0	1	3	1
2	2	4	2	2	3
1	0	1	0	1	0
1	2	1	0	2	2
2	5	3	1	2	5
1	1	4	2	3	0

Averaging Filter / Standard Average Filter (Cont.)

2D Average filtering example using a 3 x 3 sampling window:

Extending border values outside with values at boundary

Input

Average = round(1+4+0+1+4+0+2+2+4)/9 = 2

1	1	4	0	1	3	1	1
1	1	4	0	1	3	1	1
2	2	2	4	2	2	3	3
1	1	0	1	0	1	0	0
1	1	2	1	0	2	2	2
2	2	5	3	1	2	5	5
1	1	1	4	2	3	0	0
1	1	1	4	2	3	0	0

Output

2	2	2	2	2	2
2	2	2	2	1	2
1	2	1	1	1	2
2	2	1	1	1	2
2	2	2	2	2	2
2	2	3	3	2	2

Averaging Filter / Standard Average Filter/ Arithmetic Mean Filter (Cont.)

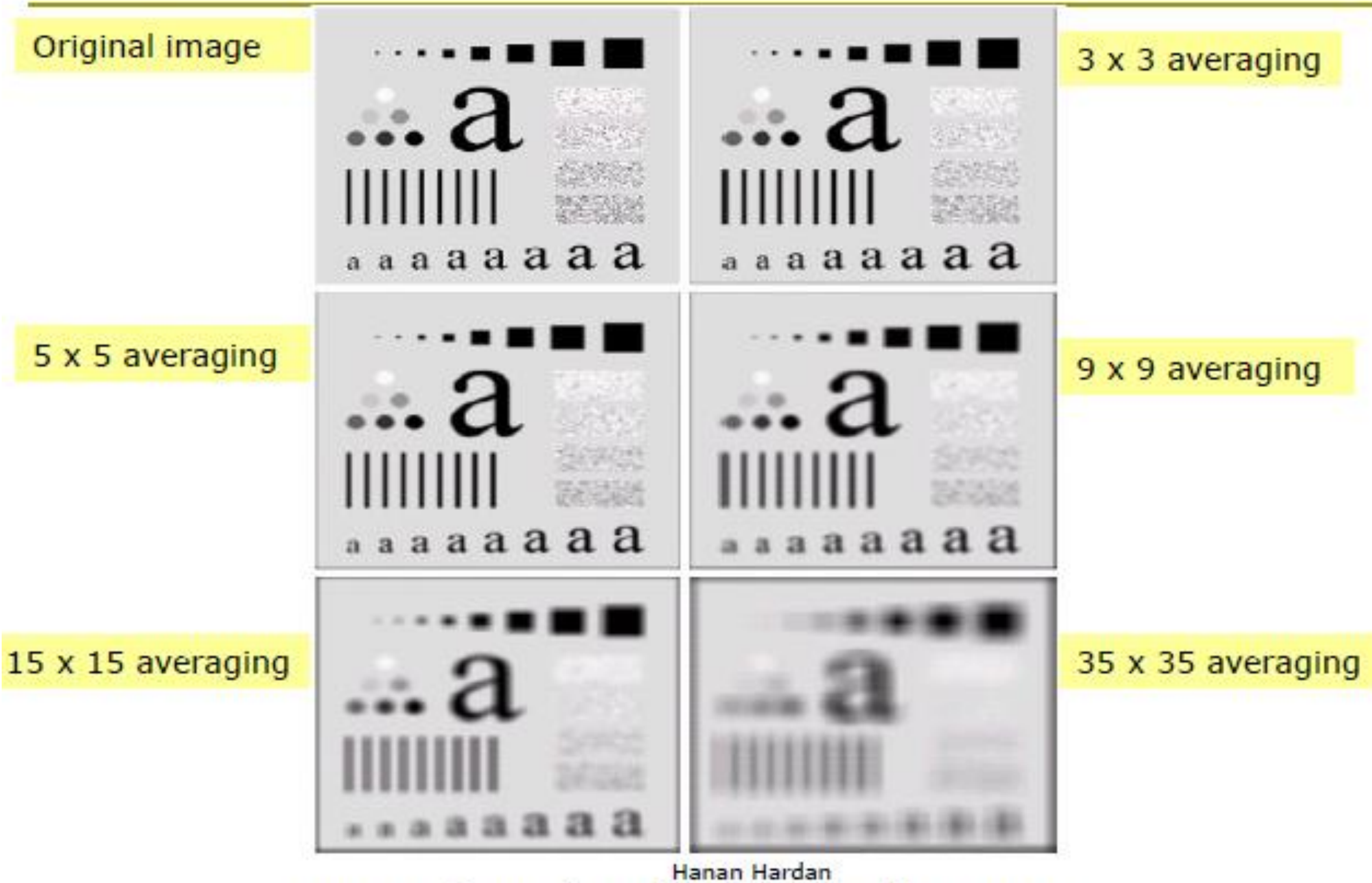
Original Image

[3x3]

[5x5]

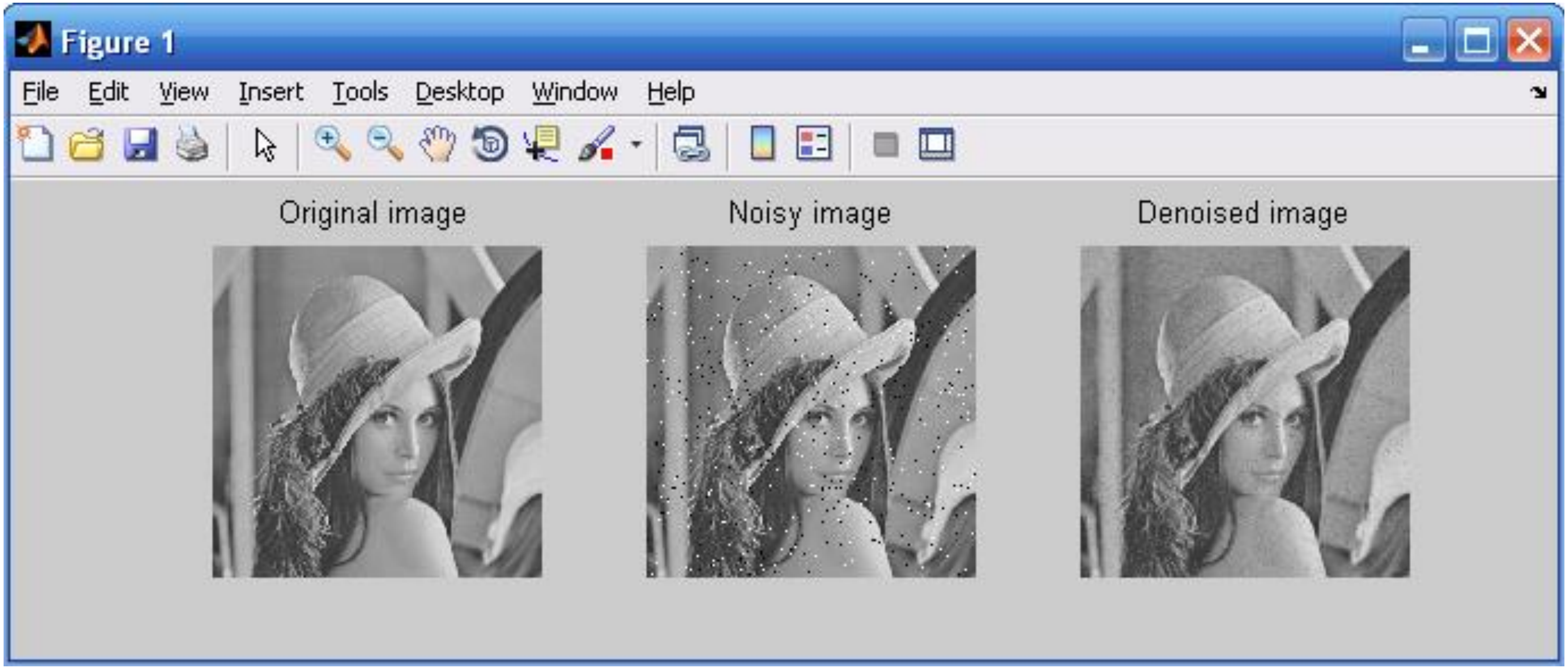
[7x7]

Averaging Filter / Standard Average Filter (Cont.)



Notice how detail begins to disappear

Averaging Filter / Standard Average Filter (Cont.)



Averaging Filter / Standard Average Filter/ Arithmetic Mean Filter (Cont.)

```
% Read Image for Noise Addition
img=imread('lena.bmp');
% Add Noise
Noi_img = imnoise(img,'salt & pepper', 0.02);
% Mask Definition
f=1/9*[1,1,1;1,1,1;1,1,1];
% Apply filter2 function
de_noi=filter2(f,Noi_img);
figure;
subplot(1,3,1);imshow(img);title('Original image')
subplot(1,3,2);imshow(Noi_img);title('Noisy image')
subplot(1,3,3);imshow(uint8(de_noi));title('Denoised image')
```

Weighted Averaging Filter / Gaussian Filter

- This mask yields a so-called weighted average, terminology used to indicate that pixels are multiplied by different coefficients, thus giving more importance (weight) to some pixels at the expense of others.
- In the mask the pixel at the center of the mask is multiplied by a higher value than any other, thus giving this pixel more importance in the calculation of the average.

Weighted Averaging Filter / Gaussian Filter (Cont.)

- In this, pixels are multiplied by different coefficients.
- Center pixel is multiplied by a higher value than average filter.

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

Weighted average filter

Weighted Averaging Filter / Gaussian Filter (Cont.)

110	120	90	130
91	94	98	200
90	91	99	100
82	96	85	90

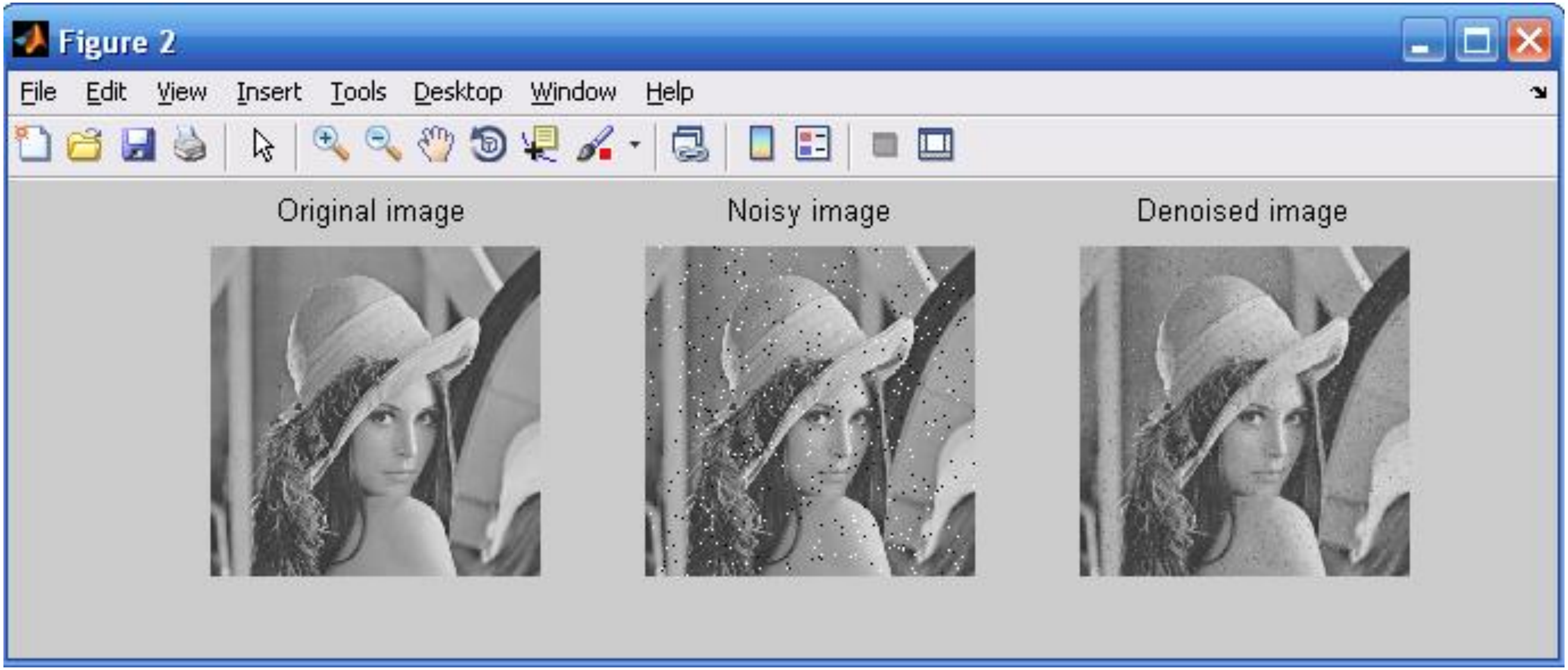
$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

The mask is moved from point to point in an image. At each point (x,y), the response of the filter is calculated

:Weighted averaging filter:

$(110 + 2 \times 120 + 90 + 2 \times 91 + 4 \times 94 + 2 \times 98 + 90 + 2 \times 91 + 99)/16 =$

Weighted Averaging Filter / Gaussian Filter (Cont.)



Weighted Averaging Filter / Gaussian Filter (Cont.)

```
% Read Image for Noise Addition
img=imread('lena.bmp');
% Add Noise
Noi_img = imnoise(img,'salt & pepper', 0.02);
% Mask Definition
f=1/16*[1,2,1;2,4,2;1,2,1];
% Apply filter2 function
de_noi=filter2(f,Noi_img);
figure;
subplot(1,3,1);imshow(img);title('Original image')
subplot(1,3,2);imshow(Noi_img);title('Noisy image')
subplot(1,3,3);imshow(uint8(de_noi));title('Denoised image')
```

Geometric Mean Filter

- The geometric mean filter is an image filtering process meant to smooth and reduce noise of an image. It is based on the mathematic geometric mean. The output image G(x,y) of a geometric mean is given by

$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t) \right]^{\frac{1}{mn}}$$

- Achieves similar smoothing to the arithmetic mean, but tends to lose less image detail

Geometric Mean Filter (Cont.)

$$G(x,y) = \left[\prod_{i,j \epsilon S} S(i,j) \right]^{\frac{1}{mn}}$$

Where S(x,y) is the original image, and the filter mask is m by n pixels.

- Each pixel of the output image at point (x,y) is given by the product of the pixels within the geometric mean mask raised to the power of 1/mn.
- For example, using a mask size of 3 by 3, pixel (x,y) in the output image will be the product of S(x,y) and all 8 of its surrounding pixels raised to the 1/9th power.

Geometric Mean Filter (Cont.)

Using the following original image with pixel (x,y) at the center:

5	16	22
6	3	18
12	3	15

Gives the result of: (5*16*22*6*3*18*12*3*15)^(1/9) = 8.77.

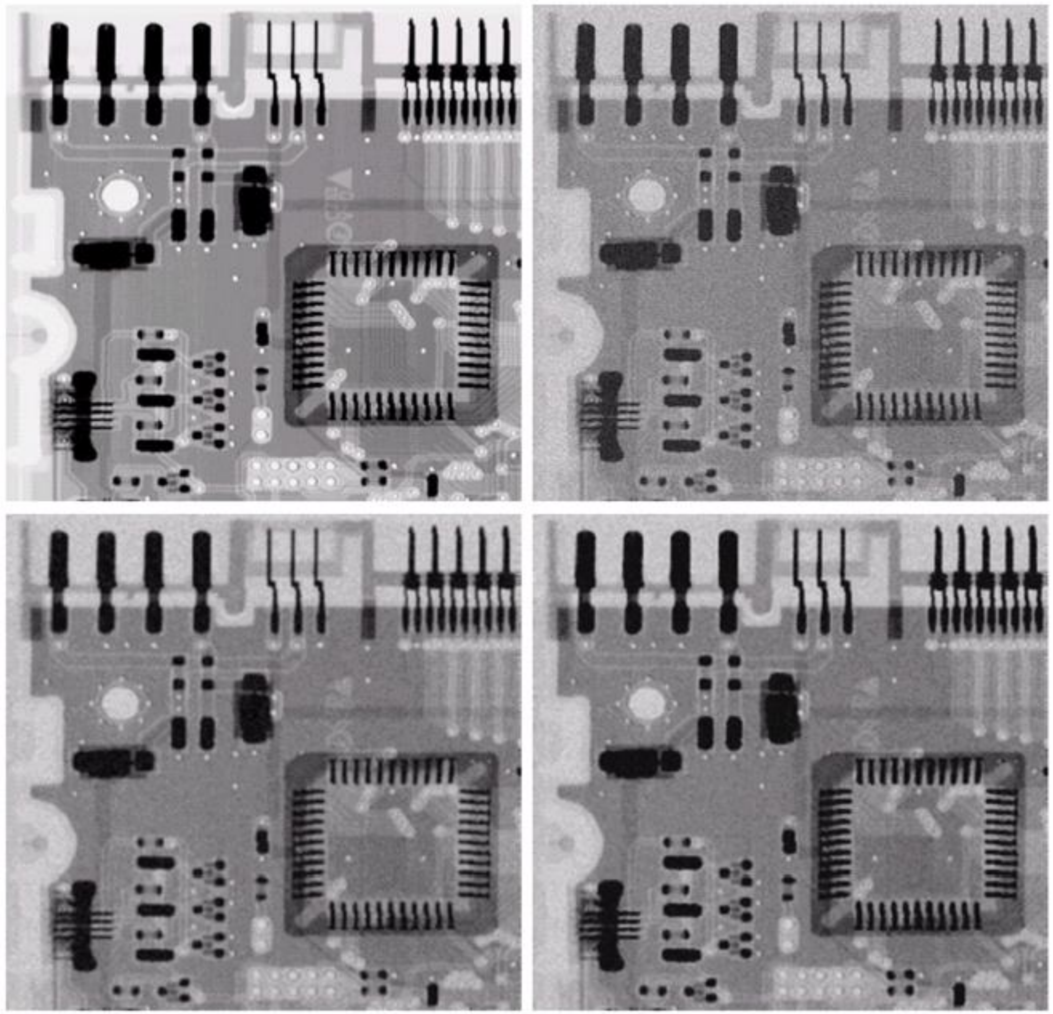
- The geometric mean filter is most widely used to filter out Gaussian noise. In general it will help smooth the image with less data loss than an arithmetic mean filter.

Geometric Mean Filter (Cont.)

- Apply Geometric Mean Filter on the below image.

110	120	90	130
91	94	98	200
90	91	99	100
82	96	85	90

Geometric Mean Filter (Cont.)



Original Image

Image Corrupted By Gaussian Noise

After A 3*3 Arithmetic Mean Filter

After A 3*3 Geometric Mean Filter

Harmonic Mean Filter

- In the harmonic mean method, the color value of each pixel is replaced with the harmonic mean of color values of the pixels in a surrounding region.
- The harmonic mean is defined as:

$$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

- Works well for salt noise, but fails for pepper noise
- Also does well for other kinds of noise such as Gaussian noise

Harmonic Mean Filter (Cont.)

- Harmonic Mean is calculated by ,

$$H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_n}}$$

5	16	22
6	3	18
12	3	15

Contraharmonic Mean Filter

- The contra harmonic mean filtering operation yields a restored image based on the expression

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

where Q is called the order of the filter.

Contraharmonic Mean Filter (Cont.)

- This filter is well suited for reducing or virtually eliminating the effects of salt-and-pepper noise.
- For positive values of Q, the filter eliminates pepper noise.
- For negative values of Q it eliminates salt noise.
- It cannot do both simultaneously.
- Note that the contra harmonic filter reduces to the arithmetic mean filter if $Q = 0$, and to the harmonic mean filter if $Q = -1$.

Contraharmonic Mean Filter (Cont.)

- The contraharmonic mean with order Q is defined as:

$$C_Q = \frac{x_1^{Q+1} + x_2^{Q+1} + \cdots + x_n^{Q+1}}{x_1^{Q} + x_2^{Q} + \cdots + x_n^{Q}}$$

Contraharmonic Mean Filter (Cont.)

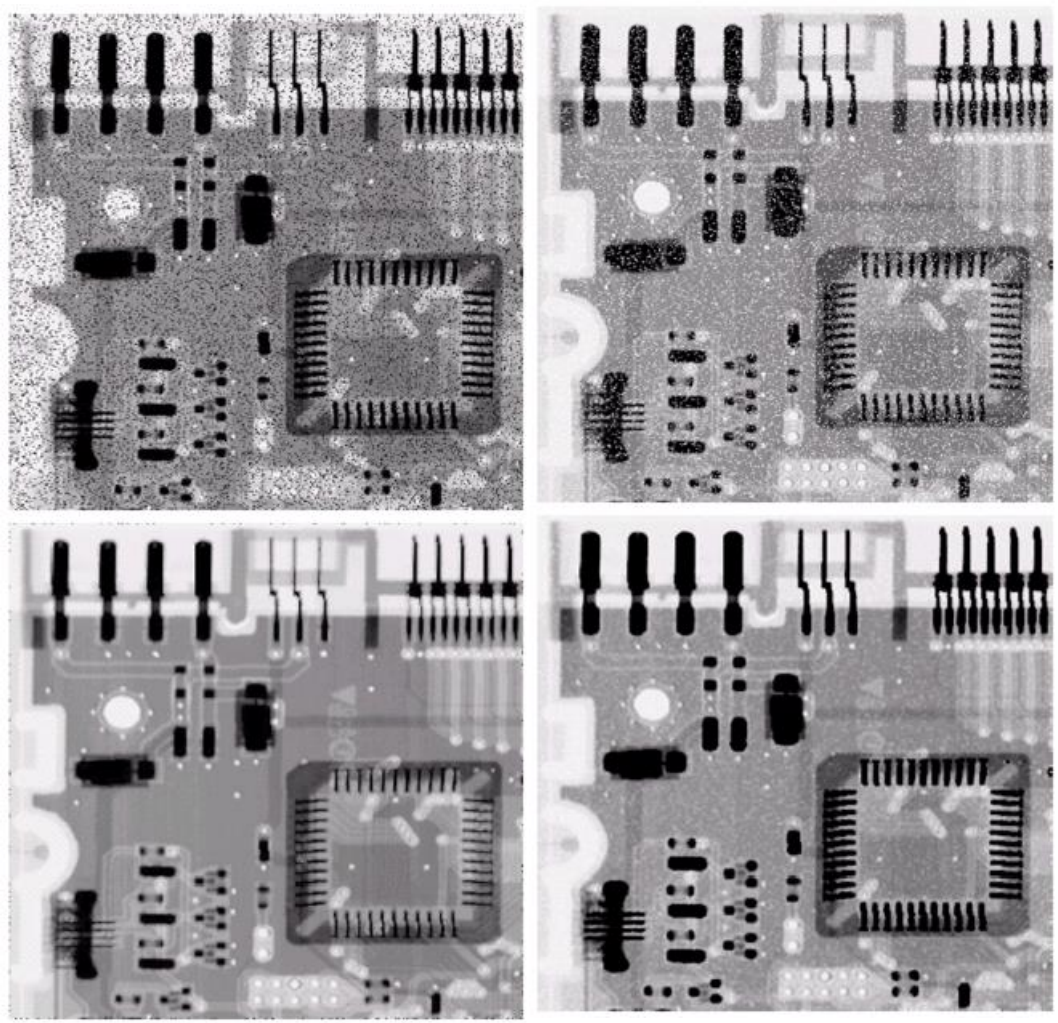


Image Corrupted By Pepper Noise

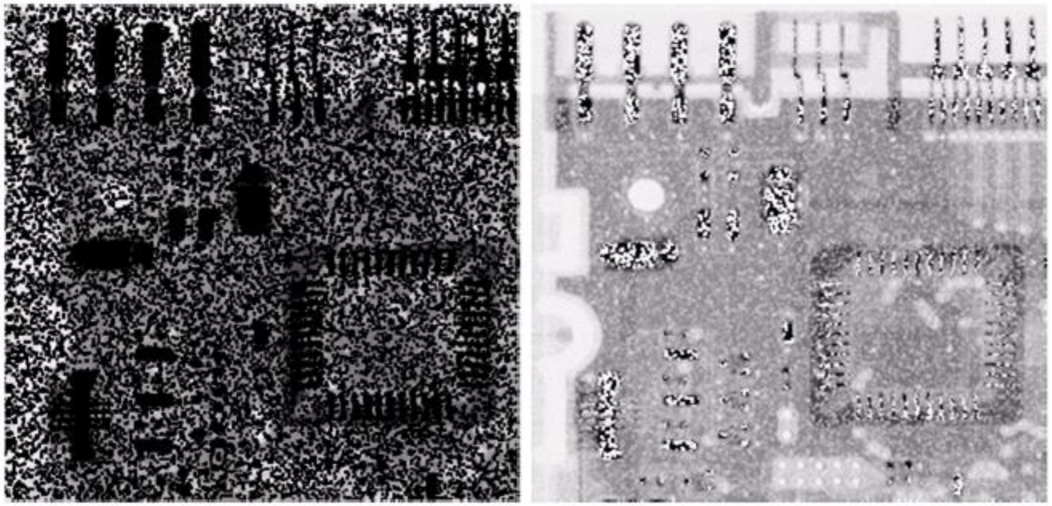
Image Corrupted By Salt Noise

Result of Filtering Above With 3*3 Contraharmonic Q=1.5

Result of Filtering Above With 3*3 Contraharmonic Q=-1.5

Contraharmonic Mean Filter (Cont.)

- Choosing the wrong value for Q when using the contraharmonic filter can have drastic results



Order Statistics Filter / Non-linear Filter

- Order statistic filters are non-linear spatial filters whose response is based on the ordering(ranking) of the pixels contained in the image area encompassed by the filter, and then replacing the value in the centre pixel with the value determined by the ranking result

Order Statistics Filter / Non-linear Filter (Cont)

Types of Order Statistics Filter / Non-linear Filter:

- Median Filtering
- Max Filtering
- Min Filtering
- Mid-point Filtering
- Alpha Trimmed Mean Filtering

Median Filtering

- The median filter is defined as the median of all pixels within a local region of an image.

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{median}\{g(s,t)\}$$

Median Filtering (Cont.)

- Replaces the value of a pixel by the median of the pixel values in the neighborhood of that pixel

110	120	90	130
91	94	98	200
90	95	99	100
82	96	85	90

becomes $\rightarrow$ $\boxed{95}$

Steps:

1. Sort the pixels in ascending order:

90,90, 91, 94, 95, 98, 99, 110, 120

2. replace the original pixel value by the median :

95

Median Filtering (Cont.)

2D Median filtering example using a 3 x 3 sampling window:

Keeping border values unchanged

Sorted: 0,0,1,1,1,2,2,4,4

Input

1	4	0	1	3	1
2	2	4	2	2	3
1	0	1	0	1	0
1	2	1	0	2	2
2	5	3	1	2	5
1	1	4	2	3	0

Output

1	4	0	1	3	1
2	1	1	1	1	3
1	1	1	1	2	0
1	1	1	1	1	2
2	2	2	2	2	5
1	1	4	2	3	0

Median Filtering (Cont.)

2D Median filtering example using a 3 x 3 sampling window:

Extending border values outside with values at boundary

Input

Sorted: 0,0,1,1,1,2,2,4,4

1	1	4	0	1	3	1	1
1	1	4	0	1	3	1	1
2	2	2	4	2	2	3	3
1	1	0	1	0	1	0	0
1	1	2	1	0	2	2	2
2	2	5	3	1	2	5	5
1	1	1	4	2	3	0	0
1	1	1	4	2	3	0	0

Output

2	2	2	2	2	2
1	1	1	1	1	1
1	1	1	1	2	2
1	1	1	1	1	2
1	2	2	2	2	2
1	2	2	3	2	2

Median Filtering (Cont.)

- The advantages of median filtering are
  - It works well for various noise types, with less blurring than linear filters of similar size
  - Odd sized neighborhoods and efficient sorts yield a computationally efficient implementation
  - Most commonly used order-statistic filter

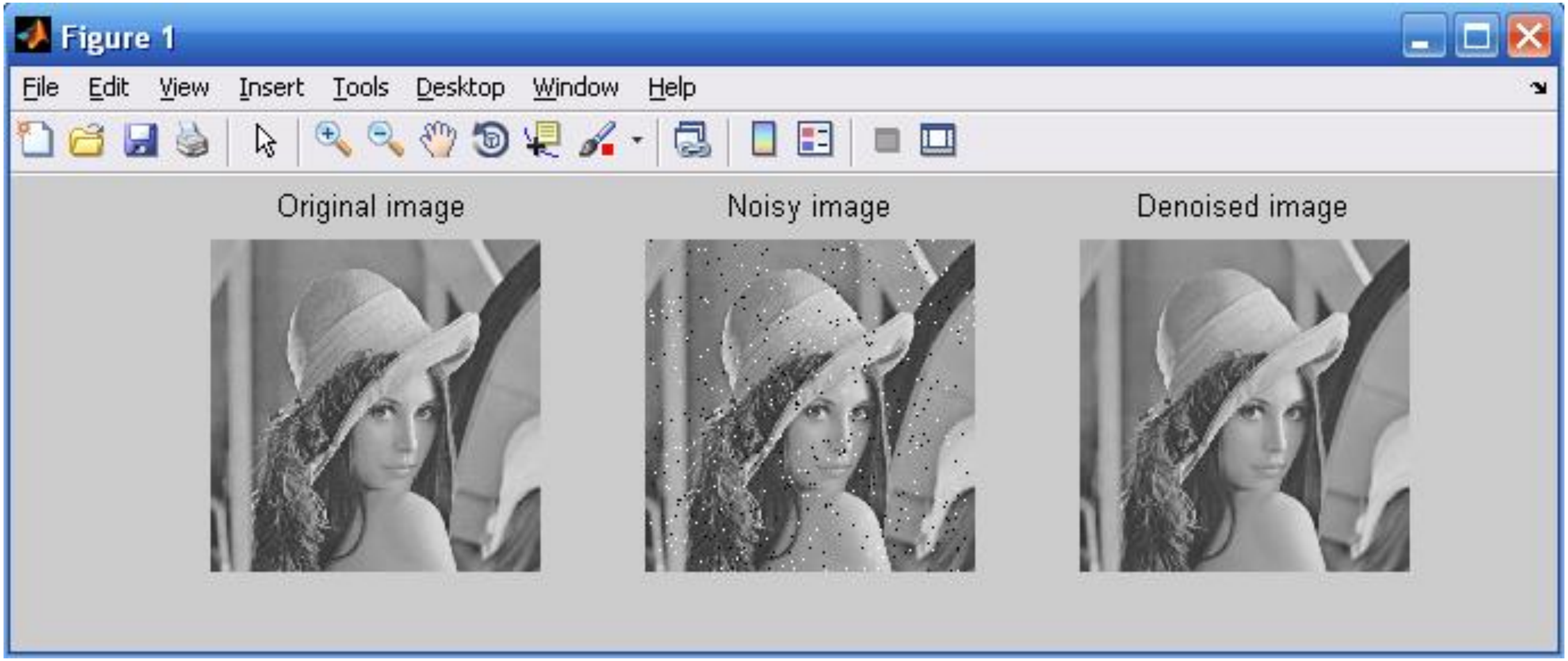
Median Filtering (Cont.)

- Very effective for removing “salt and pepper” noise

averaging

median filtering

Median Filtering (Cont.)



Median Filtering (Cont.)

```
% Read Image for Noise Addition
img=imread('lena.bmp');
% Add Noise
Noi_img = imnoise(img,'salt & pepper', 0.02);
% Apply medfilt2 function
de_noi=medfilt2(Noi_img,[3 3]);
figure;
subplot(1,3,1);imshow(img);title('Original image')
subplot(1,3,2);imshow(Noi_img);title('Noisy image')
subplot(1,3,3);imshow(uint8(de_noi));title('Denoised image')
```

Max Filtering

- The maximum filter is defined as the maximum of all pixels within a local region of an image.

$$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}} \{g(s,t)\}$$

Max Filtering (Cont.)

- The max filtering is achieved using the following equation

$$f(x,y) = \max g(s,t)$$

Find Out the Maximum from neighbor and replace it at the middle position.

110	120	90	130
91	94	98	200
90	91	99	100
82	96	85	90

Max Filtering (Cont.)

- The advantages of Max filtering are
  - Max filters are Useful for finding the brightest points in an image
  - They also tend to reduce pepper noise (i.e. dark pixel values)

Max Filtering (Cont.)

c) Result of maximum filtering image 9.3-2c; mask size=5x5

d) Result of maximum filtering image 9.3-2c; mask size=9x9

Min Filtering

- The minimum filter is defined as the minimum of all pixels within a local region of an image.

$$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\}$$

Min Filtering (Cont.)

- The min filtering is achieved using the following equation

$$f(x,y) = \min g(s,t)$$

Find Out the Minimum from neighbor and replace it at the middle position.

110	120	90	130
91	94	98	200
90	91	99	100
82	96	85	90

Min Filtering (Cont.)

- The advantages of Min filtering are
  - They also tend to reduce Salt noise (i.e. dark pixel values)

Min Filtering (Cont.)

a) Result of minimum filtering image 9.3-2a; mask size=5x5

b) Result of minimum filtering image 9.3-2a; mask size=9x9

Mid-Point Filtering

- In Mid-Point Filtering replaces the value of a pixel by the midpoint between the maximum and minimum pixels in a neighborhood
- Mid-Point filters are very useful for removing randomly distributed noise like Gaussian noise.

$$\hat{f}(x,y) = \frac{1}{2}\left[\max_{(s,t)\in S_{xy}}\{g(s,t)\} + \min_{(s,t)\in S_{xy}}\{g(s,t)\}\right]$$

Mid-Point Filtering (Cont.)

- Find Out the Minimum , Minimum – then find Mid and replace it at the middle position.

110	120	90	130
91	94	98	200
90	91	99	100
82	96	85	90

Mid-Point Filtering (Cont.)

a) Image with gaussian noise, variance = 300, mean = 0

b) Result of midpoint filter mask size = 3

c) Image with uniform noise, variance = 300, mean = 0

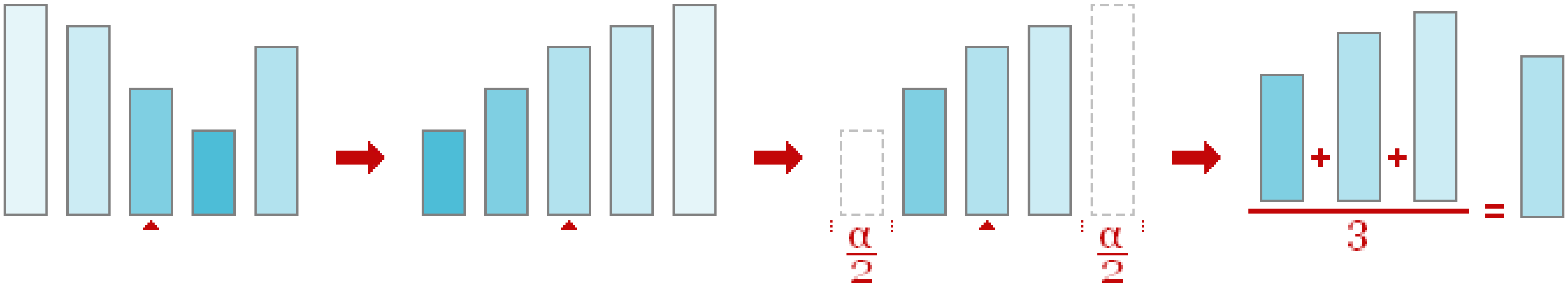
d) Result of midpoint filter mask size = 3

Alpha Trimmed Mean Filtering

- Alpha-trimmed mean filter is windowed filter of nonlinear class, by its nature is hybrid of the mean and median filters.
- The basic idea behind filter is for any element of the signal (image) look at its neighborhood, discard the most atypical elements and calculate mean value using the rest of them.
- Alpha you can see in the name of the filter is indeed parameter responsible for the number of trimmed elements.

Alpha Trimmed Mean Filtering (Cont.)

- The basic idea here is to order elements, discard elements at the beginning and at the end of the got ordered set and then calculate average value using the rest.
- For instance, let us calculate alpha-trimmed mean for the case, depicted in fig. 1.



Alpha Trimmed Mean Filtering (Cont.)

- Alpha-trimmed mean filter algorithm:
  1) Place a window over element
  2) Pick up elements
  3) Order elements
  4) Discard elements at the beginning and at the end of the got ordered set
  5) Take an average — sum up the remaining elements and divide the sum by their number

Comparison Between Filters

Parameters	Average Filter	Weighted Filter	Median Filter
Noise Reduction	Reduces Noise but it introduces blurring effect at edges.	Blurring effect is less as compared with Average filter	Blurring effect is less as compared with Average filter
Percentage of noise Reduction	100% noise Not Reduced	100% noise Not Reduced	Almost 100% noise reduced.
Size of Filter	As we increase the size of the filter mask, Noise reduces but blurring effect increases.	As we increase the size of the filter mask, Noise reduces but blurring effect increases.	As we increase the size of the filter mask, 100% of Noise reduces but blurring effect at edges increases.
Mask	1/9x[1,1,1;1,1,1;1,1,1]	1/16x[1,2,1;2,4,4;1,2,1]	Pixel value is replaced by median value of neighborhood.
MATLAB Function	filter2(mask,Noisy_img)	filter2(mask,Noisy_img)	medfilt2(Noisy_img,[3 3])

To Compare all Filters

- https://www.digimizer.com/manual/m-image-filters.php


Thank you