Digital Image Processing
CS390S

Feng Jiang
2018 Spring

METROPOLITAN STATE UNIVERSITY
OF DENVER

Image enhancement

Enhancement Techniques

- Spatial
  Operates on pixels
  Point wise & filtering
- Frequency Domain
  Operates on FT of Image

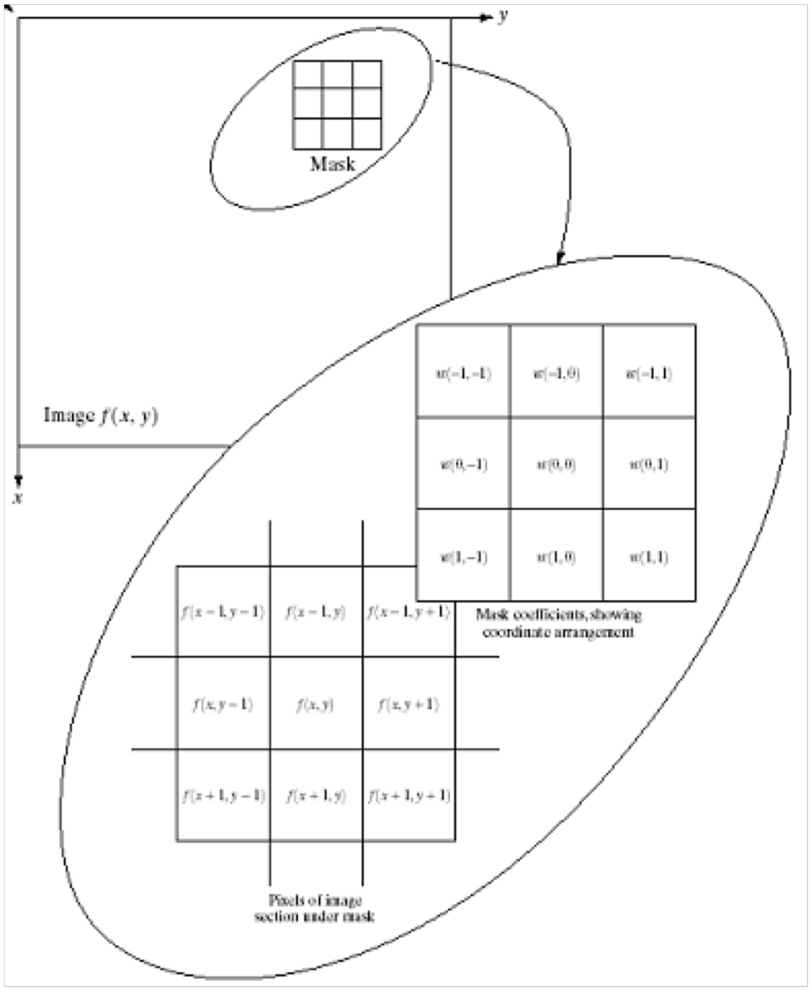
Spatial Filtering

- Use of spatial masks for image processing (spatial filters)

- Linear and nonlinear filters

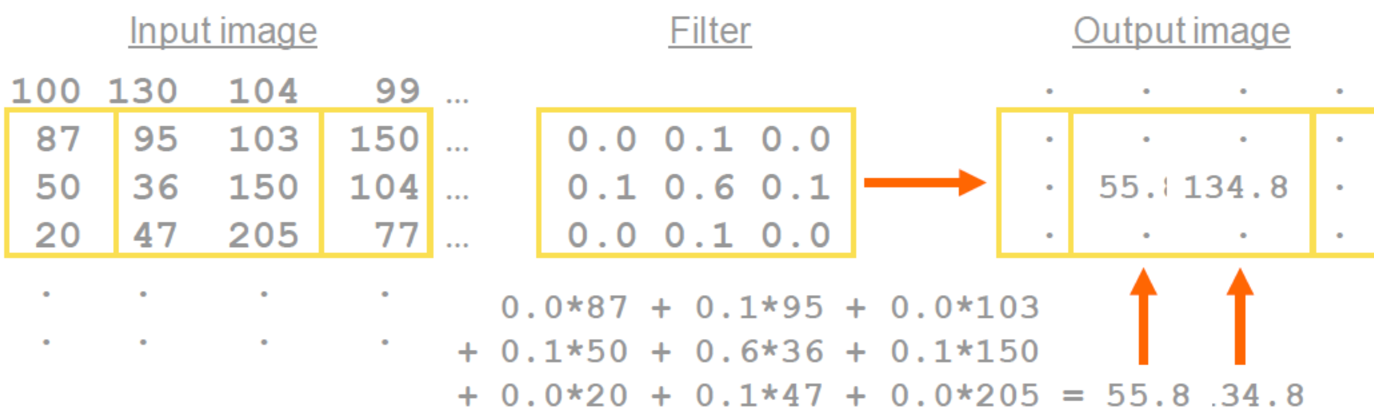
Spatial Filtering

Correlation
Convolution



Spatial Filtering

- Operation on image neighborhood and small …
  - “mask”, “filter”, “stencil”, “kernel”
- Linear operations within a moving window



Spatial Filtering

Correlation

- 1D $$g(x) = \sum_{s=-a}^{a} w(s) f(x+s)$$
- 2D $$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

$$w(s,t) = \boxed{\begin{array}{ccccc} w(-a,-b) & \cdots & & \cdots & w(a,-b) \\ \vdots & & & & \vdots \\ & \cdots & w(0,0) & \cdots & \\ \vdots & & & & \vdots \\ w(-a,b) & \cdots & & \cdots & w(a,b) \end{array}}$$

Spatial Filtering

convolution

- Discrete

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

- Continuous

$$g(x,y) = w(x,y) * f(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} w(s,t) f(x-s, y-t) ds dt$$

- Same as cross correlation <u>with kernel transposed</u> around each axis
- The two operations (correlation and convolution) are the same if the kernel is symmetric about axes

$$g = w \circ f = w^* * f$$

w^* reflection of w

Spatial Filtering

Correlation properties and filter design

- Normalize
- Sums to one

- Symmetry
- Left, right, up, down
- Rotational

- Shift invariant

$$g = w \circ f \qquad g(x,y) = w(x,y) \circ f(x,y)$$

$$w(x,y) \circ f(x-x_0, y-y_0) = \sum_{s=-\infty}^{\infty} \sum_{t=-\infty}^{\infty} w(s,t) f(x-x_0+s, y-y_0+t) = g(x-x_0, y-y_0)$$

- Linear $w \circ (\alpha e + \beta f) = \alpha w \circ e + \beta w \circ f$

Compact notation

$$C_{wf} = w \circ f$$

- Special case: auto correlation

FIGURE 2.14 Illustration of one-dimensional correlation and convolution.

Correlation		Convolution	
	Origin f w	Origin f w rotated 180°	
(a)	0 0 0 1 0 0 0 0 1 2 3 2 0	0 0 0 1 0 0 0 0 0 2 3 2 1	(i)
(b)	0 0 0 1 0 0 0 0 1 2 3 2 0 Starting position alignment	0 0 0 1 0 0 0 0 0 2 3 2 1	(j)
(c)	Zero padding 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 0	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 2 3 2 1	(k)
(d)	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 0 Position after one shift	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 2 3 2 1	(l)
(e)	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 0 Position after four shifts	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 2 3 2 1	(m)
(f)	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 0 Final position	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 2 3 2 1	(n)
(g)	`'full'` correlation result 0 0 0 0 2 3 2 1 0 0 0 0	`'full'` convolution result 0 0 0 1 2 3 2 0 0 0 0 0	(o)
(h)	`'same'` correlation result 0 0 2 3 2 1 0 0	`'same'` convolution result 0 1 2 3 2 0 0 0	(p)

Spatial Filtering

Correlation
Convolution

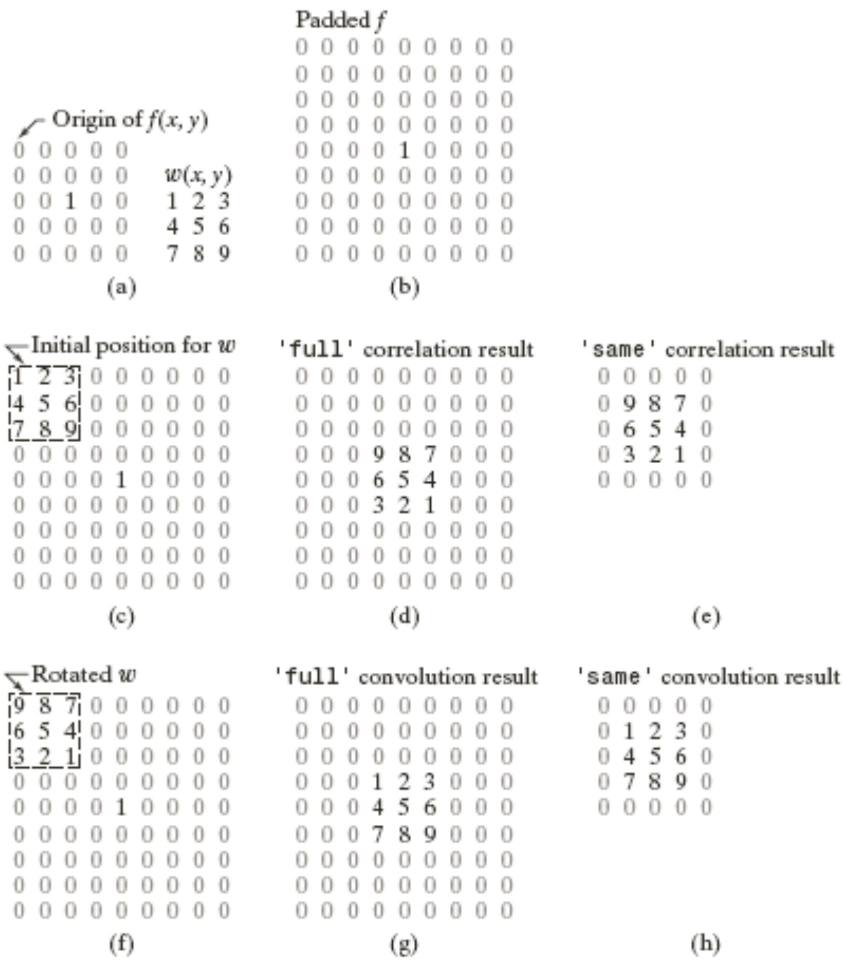


FIGURE 2.15 Illustration of two-dimensional correlation and convolution. The 0s are shown in gray to simplify viewing.

http://www.cs.umd.edu/~djacobs/CMSC426/Convolution.pdf

Spatial Filtering

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

$a=(m-1)/2$ and $b=(n-1)/2$,
m x n (odd numbers)

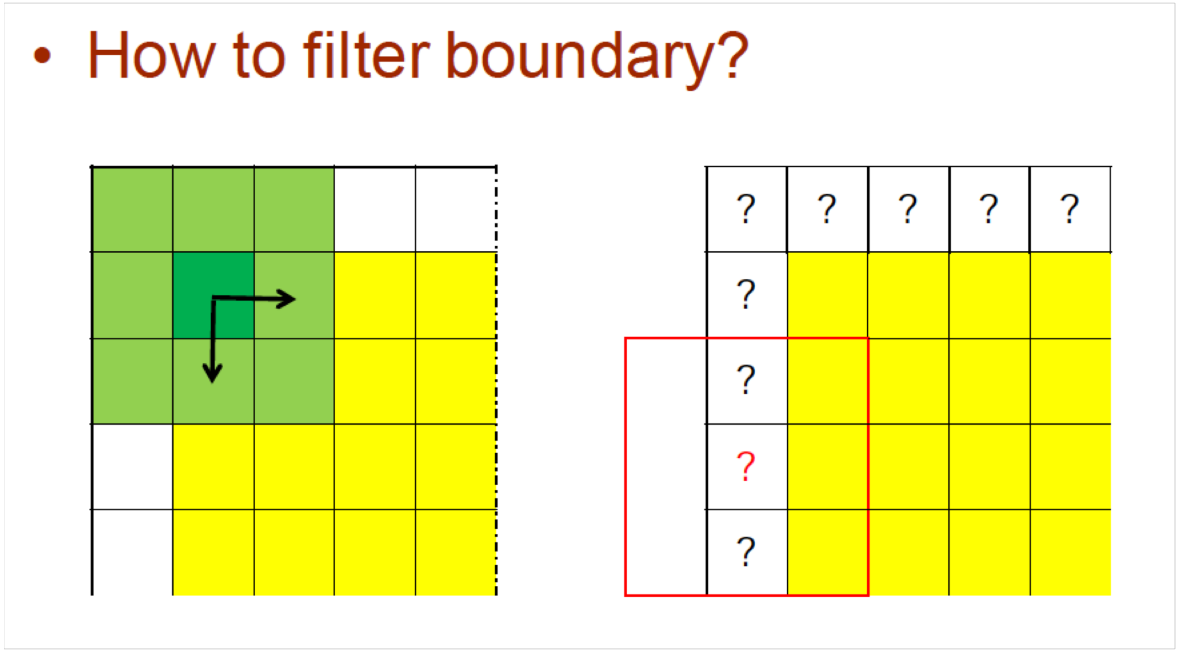
For x=0,1,…,M-1 and y=0,1,…,N-1

The basic approach is to sum products between the mask coefficients and the intensities of the pixels under the mask at a specific location in the image:

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9$$ (for a 3 x 3 filter)

Spatial Filtering

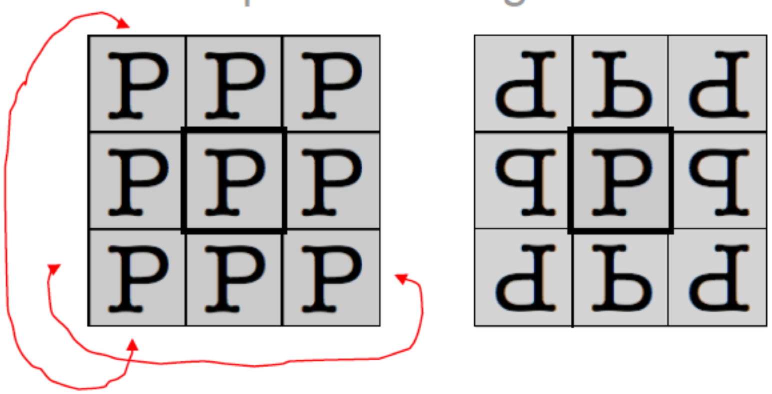
Correlation



Spatial Filtering

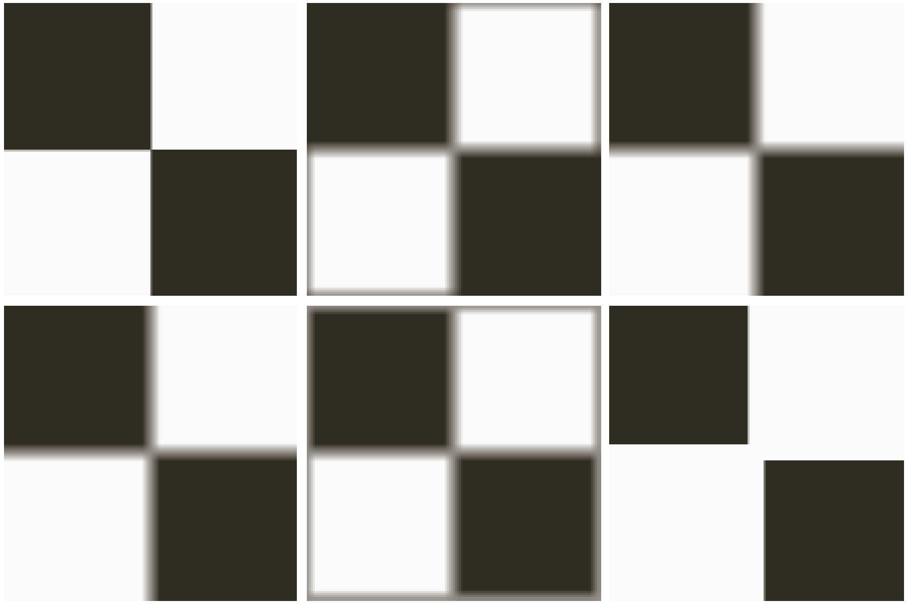
Correlation: Technical Details

- Boundary conditions
  - Boundary not filtered (keep it 0)
  - Pad image with amount (a,b)
    - Constant value or repeat edge values
  - Cyclical boundary conditions
    - Wrap or mirroring



Spatial Filtering

Correlation



a b c
d e f

FIGURE 2.16
(a) Original image. (b) Result of using `imfilter` with default zero padding. (c) Result with the `'replicate'` option. (d) Result with the `'symmetric'` option. (e) Result with the `'circular'` option. (f) Result of converting the original image to class `uint8` and then filtering with the `'replicate'` option. A filter of size 31×31 with all 1s was used throughout.

W=ones(31);
G = imfilter(f,w)
G = imfilter(f,w,'replicate')
G = imfilter(f,w,'symmetric')
G = imfilter(f,w,'circular')

Spatial Filtering

Correlation

Options	Description
Filtering Mode	
`'corr'`	Filtering is done using correlation (see Figs. 2.14 and 2.15). This is the default.
`'conv'`	Filtering is done using convolution (see Figs. 2.14 and 2.15).
Boundary Options	
`P`	The boundaries of the input image are extended by padding with a value, `P` (written without quotes). This is the default, with value 0.
`'replicate'`	The size of the image is extended by replicating the values in its outer border.
`'symmetric'`	The size of the image is extended by mirror-reflecting it across its border.
`'circular'`	The size of the image is extended by treating the image as one period a 2-D periodic function.
Size Options	
`'full'`	The output is of the same size as the extended (padded) image (see Figs. 2.14 and 2.15).
`'same'`	The output is of the same size as the input. This is achieved by limiting the excursions of the center of the filter mask to points contained in the original image (see Figs. 2.14 and 2.15). This is the default.

TABLE 2.3 Options for function `imfilter`.

W=ones(31);
G = imfilter(f,w)
G = imfilter(f,w,’replicate’)
G = imfilter(f,w,’symmetric’)
G = imfilter(f,w,’circular’)

Spatial Filtering

Examples 1

$$\begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{matrix}$$

$$1/9 * \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

Examples 2

$$\frac{1}{9} * \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

$$\frac{1}{25} * \begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{matrix}$$

Spatial Filtering

Smoothing and Noise

Noisy image

5x5 box filter

General Spatial Filter

FIGURE 3.33 Another representation of a general 3×3 spatial filter mask.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

General Spatial Filter

Most popular: average filter

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

a b

FIGURE 3.34 Two 3×3 smoothing (averaging) filter masks. The constant multipli er in front of each mask is equal to the sum of the values of its coefficients, as is required to compute an average.

Spatial Filtering Application
image smoothing

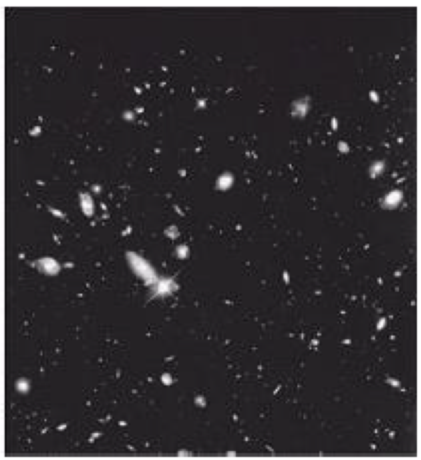
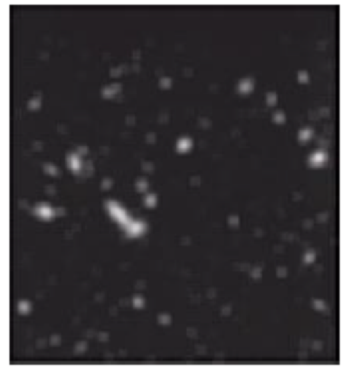


Image from the
Hubble Space Telescope



After 15X15
averaging mask

After
Thresholding

Demo

Spatial Filtering Application
image smoothing

FIGURE 3.33 (a) Original image, of size 500×500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their intensity levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50×120 pixels.

a b
c d
e f

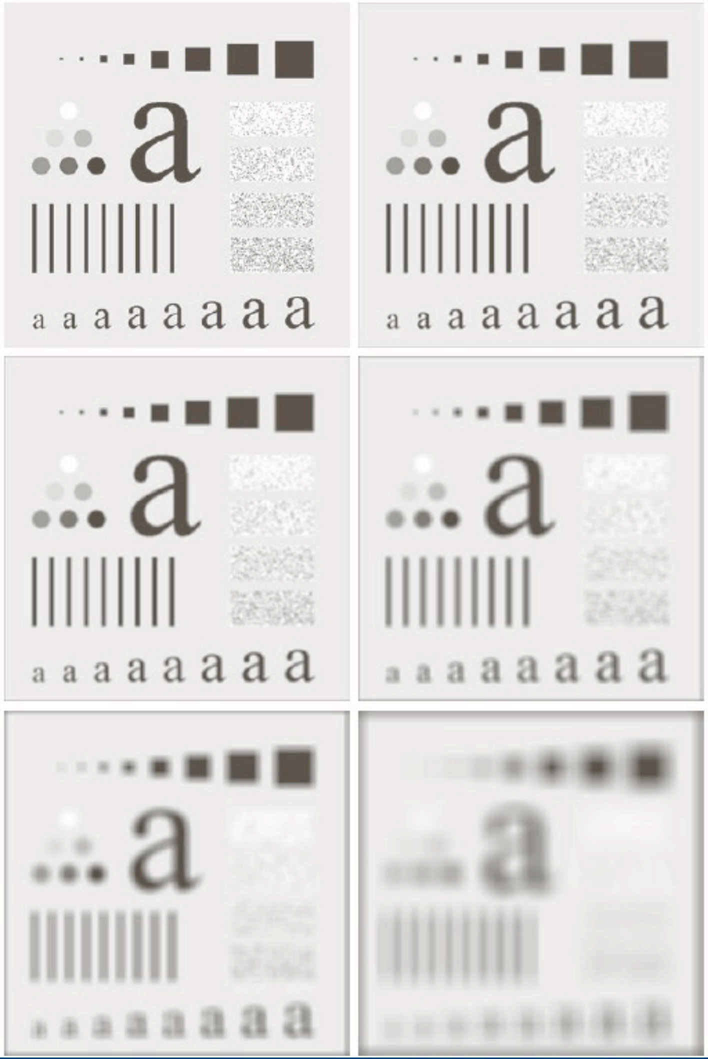


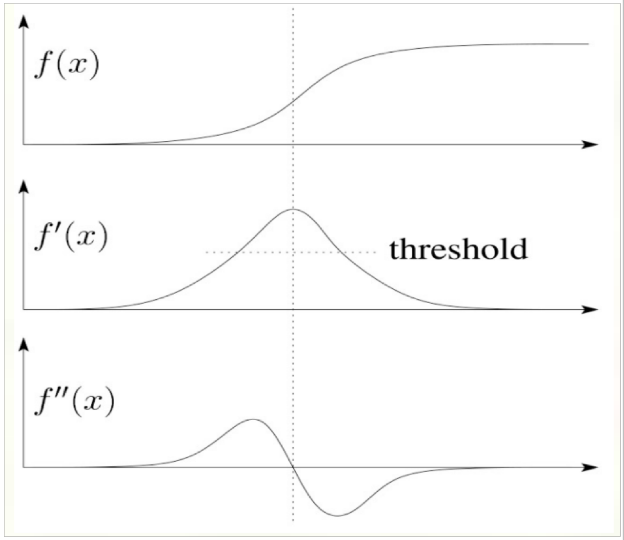
Image Enhancement in Spatial Domain

- Image Histogram
- Image Negative
- Contrast Stretching
- Power-Law Transformation
- Histogram Equalization
- Local Enhancement
- Image Subtraction
- Image Averaging
- Image Smoothing
- Image Sharpening

Derivatives

- First derivative "gradient"

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



- Second derivative

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

Derivatives

- First derivative "gradient"

$$i)\ \left[\mathbf{h}_x\right] = \left[\mathbf{h}_y\right]^t = \begin{bmatrix} 1 & -1 \end{bmatrix}$$

$$ii)\ \left[\mathbf{h}_x\right] = \left[\mathbf{h}_y\right]^t = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

- Second derivative

$$\Delta = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- Like derivative

−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

Observations

- 1st order derivatives produce **thicker** edges in an image
- 2nd order derivatives have stronger response to **fine** detail
- 1st order derivatives have stronger response to a gray lever step
- 2nd order derivatives produce a double response at step changes in gray level

http://www.mif.vu.lt/atpazinimas/dip/FIP/fip-Derivati.html

Spatial Filtering

Derivative Example

$$\begin{matrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{matrix}$$

$$\begin{matrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{matrix}$$

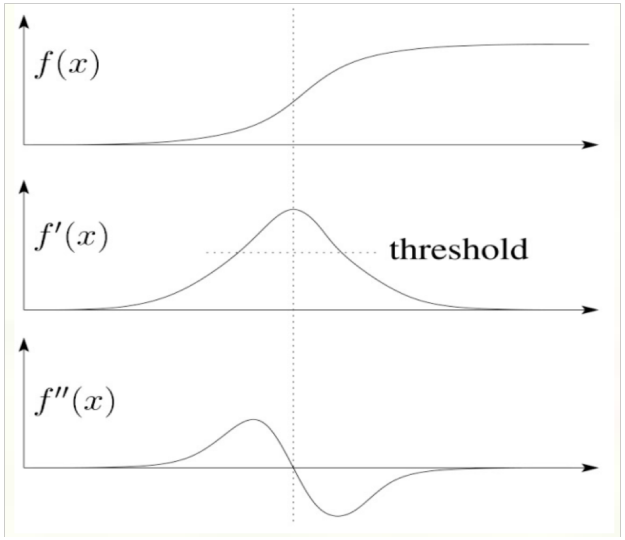
Spatial Filtering -Laplacian filter

An equivalent measure of the second derivative in 2D is the **Laplacian**:

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Using the same arguments we used to compute the gradient filters, we can derive a Laplacian filter to be:

$$\Delta = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



http://www.mif.vu.lt/atpazinimas/dip/FIP/fip-Derivati.html

Spatial Filtering- Laplacian filter

a b
c d

(a)

0	1	0
1	−4	1
0	1	0

(b)

1	1	1
1	−8	1
1	1	1

(c)

0	−1	0
−1	4	−1
0	−1	0

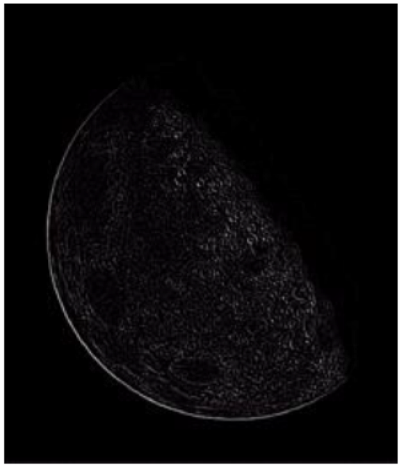
(d)

−1	−1	−1
−1	8	−1
−1	−1	−1

Spatial Filtering

Image Sharpening

Image of the North Pole of the moon



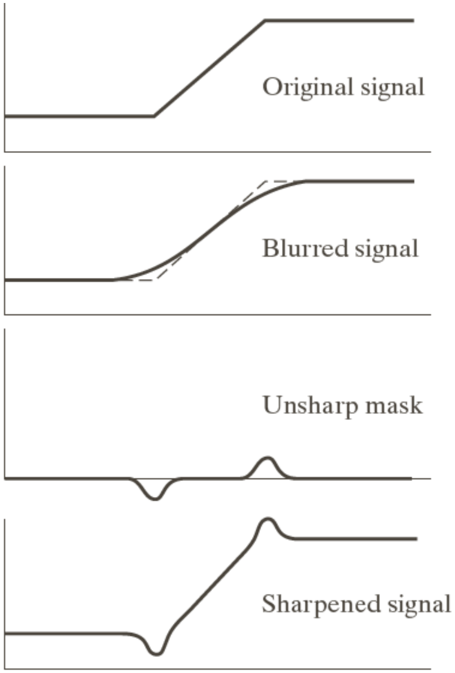
Laplacian-filtered image using (b)

Image enhanced

Demo

e.g. basicSpatialFilter.m

Image sharpening



a
b
c
d

FIGURE 3.39 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).

General way of image sharpening

Image sharpening- “Sobel Filter”

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

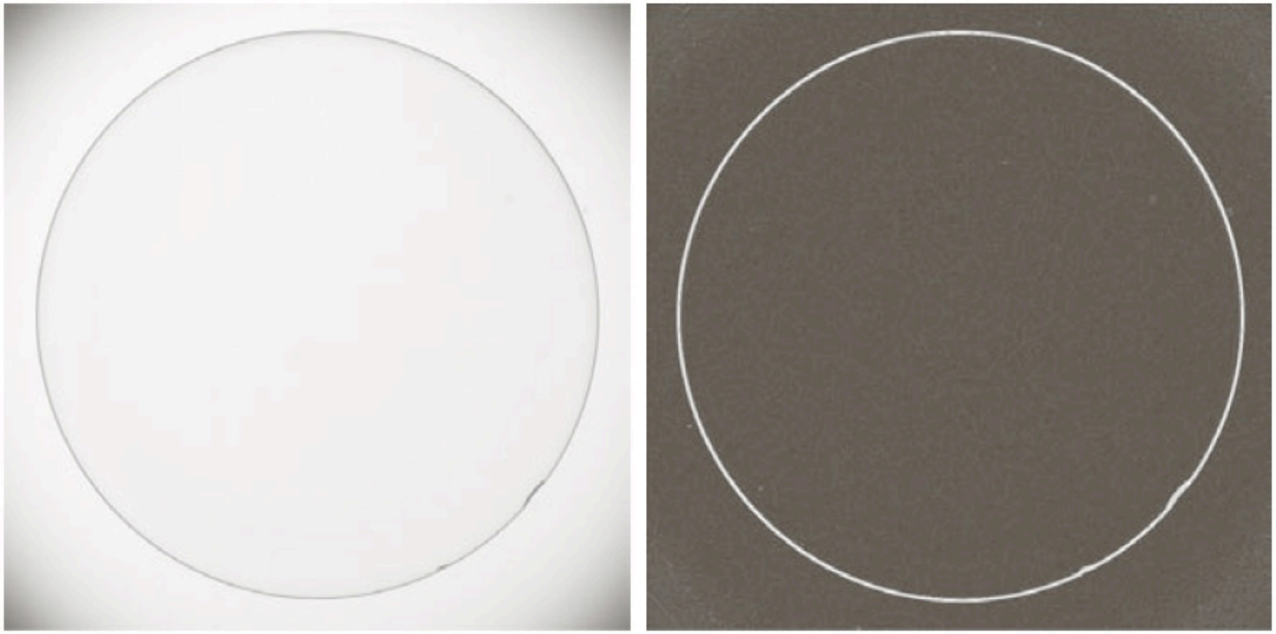
−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

a
b c
d e

FIGURE 3.41 A 3×3 region of an image (the zs are intensity values). (b)–(c) Roberts cross gradient operators. (d)–(e) Sobel operators. All the mask coefficients sum to zero, as expected of a derivative operator.

Image sharpening- “Sobel Filter”



a b

FIGURE 3.42 (a) Optical image of contact lens (note defects on the boundary at 4 and 5 o’clock). (b) Sobel gradient. (Original image courtesy of Pete Sites, Perceptics Corporation.)

Image sharpening- “Sobel Filter”

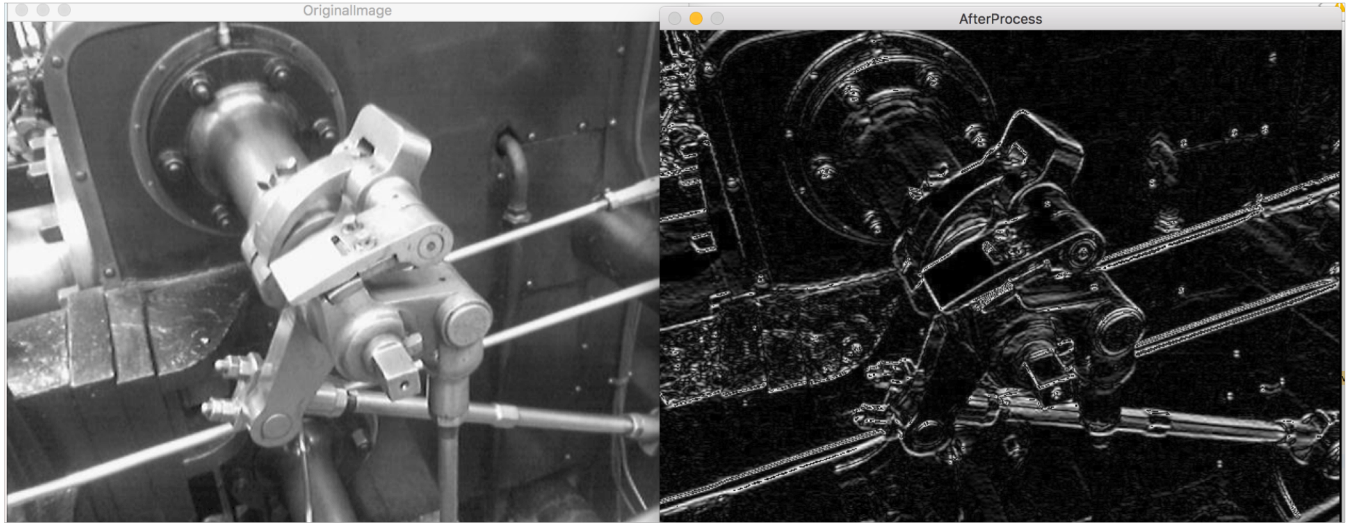
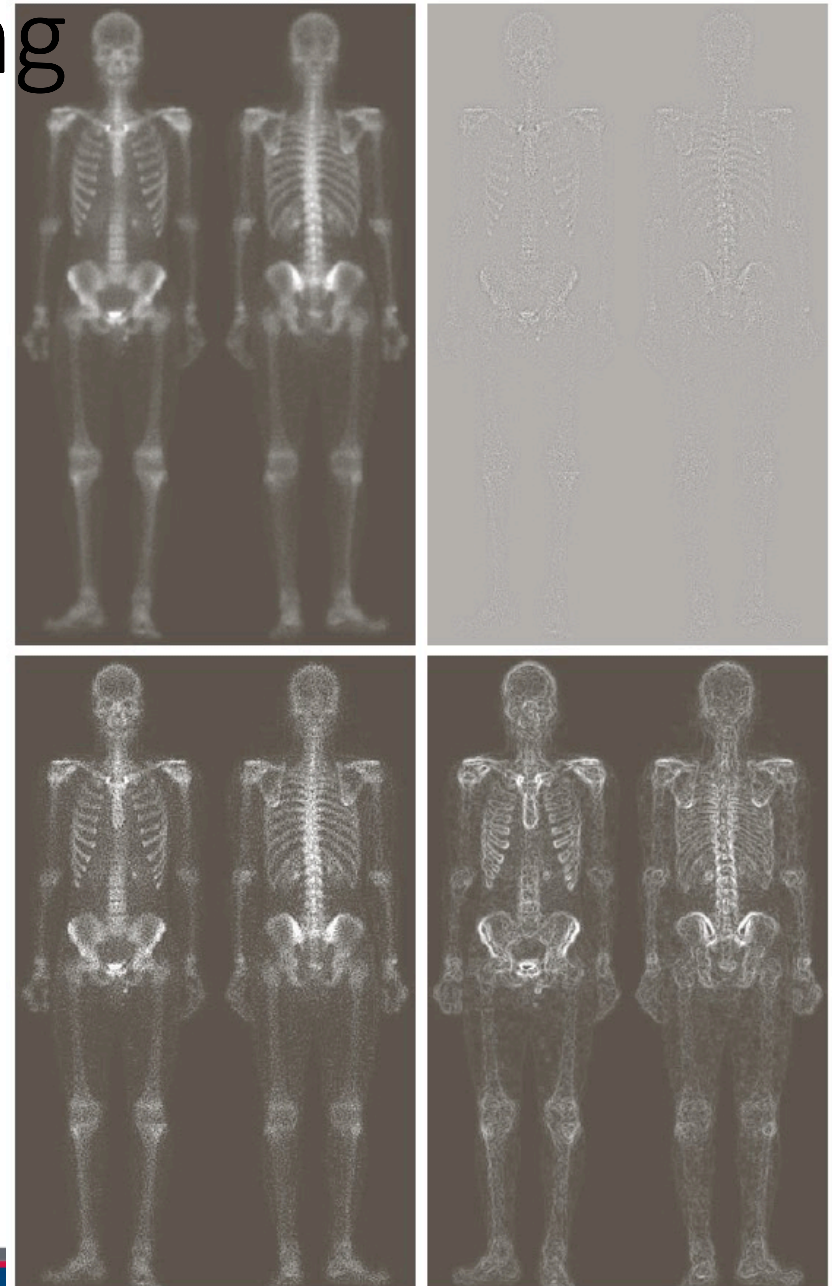


Image sharpening

a b
c d

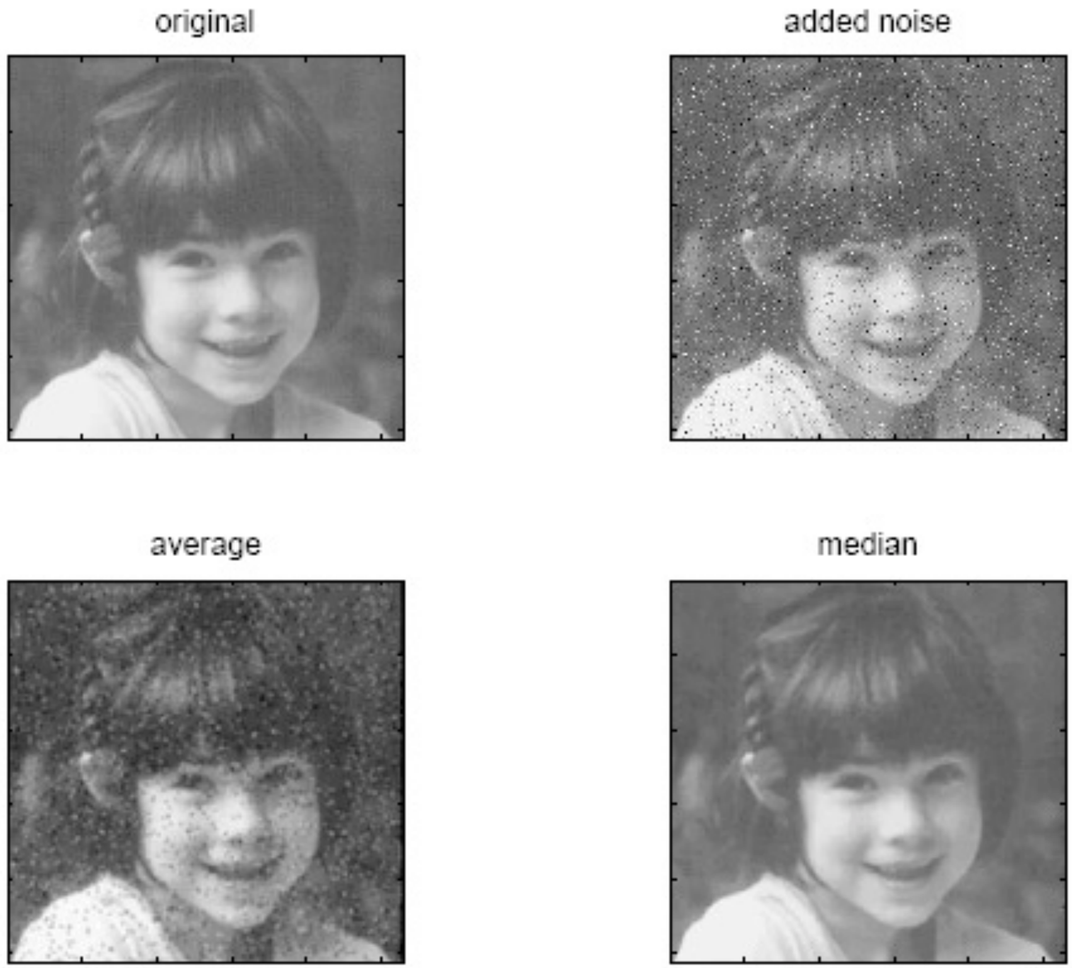
FIGURE 3.43
(a) Image of whole body bone scan.
(b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b).
(d) Sobel gradient of (a).



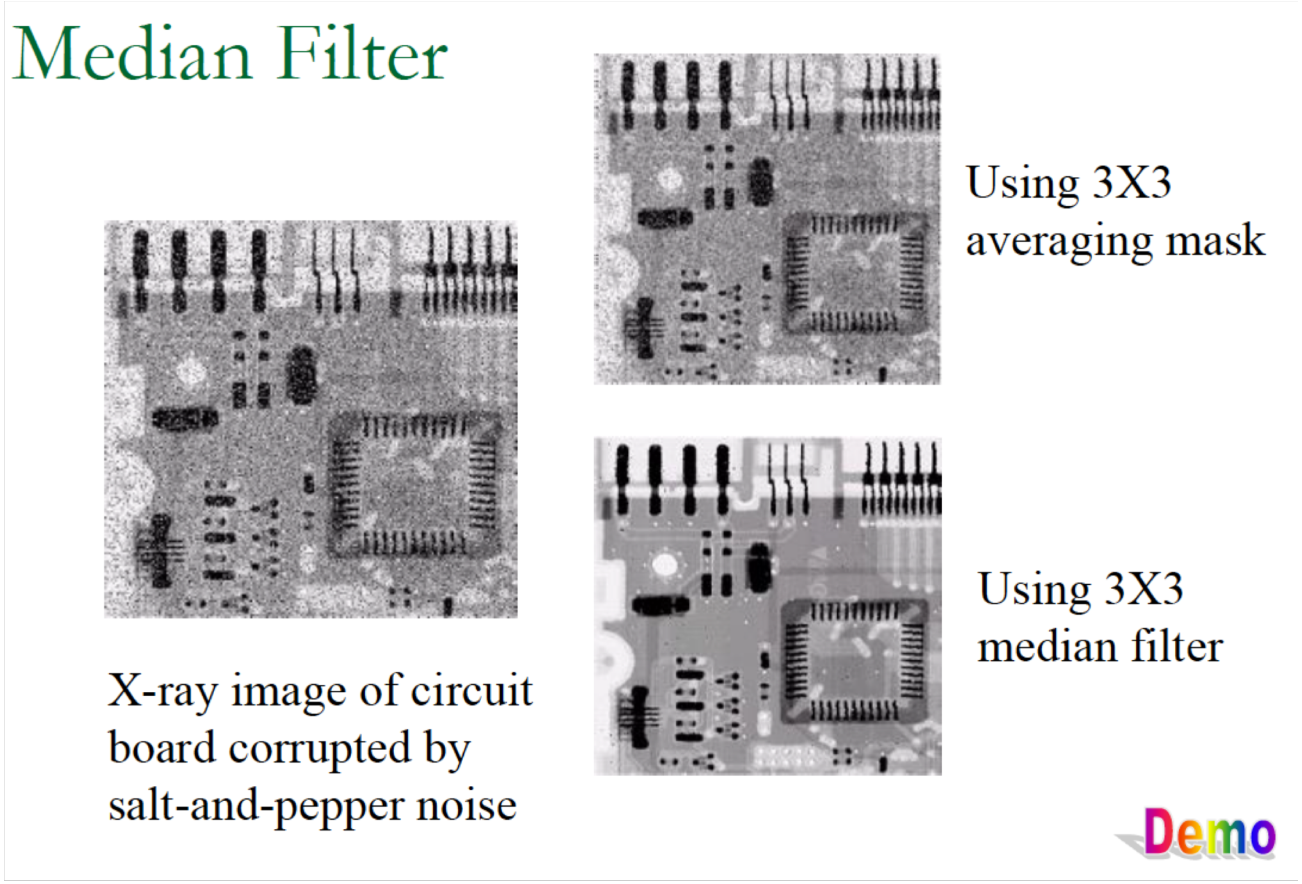
Non-linear Filter

- Median filtering (nonlinear)
  - Used primarily for noise reduction (eliminates isolated spikes)
  - The gray level of each pixel is replaced by the median of the gray levels in the neighborhood of that pixel (instead of by the average as before).

Median Filter



Median Filter



Median Filter

Median Filtering

Median 3x3 Median 5x5

Median Filter

Median Filtering

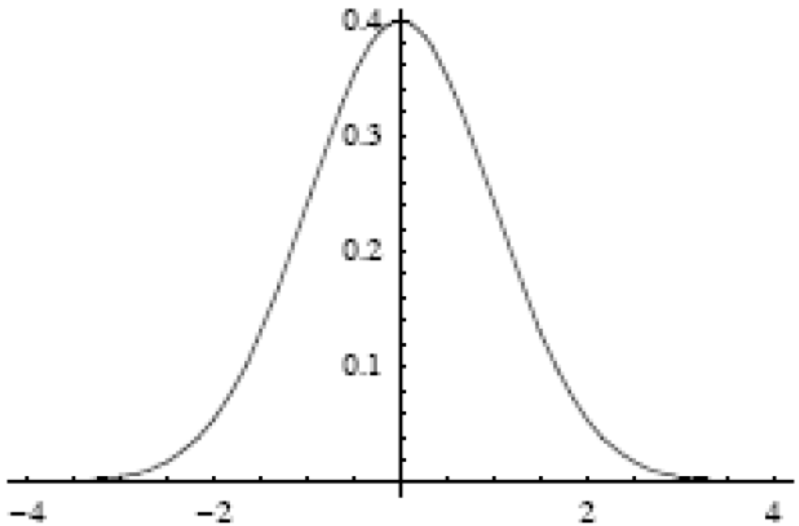
- Iterate

Median 3x3

2x Median 3x3

Other spatial filters-Gaussian

$$\sigma = 1;\ \mathrm{Plot}\left[\frac{1}{\sqrt{2\,\pi\,\sigma^2}}\,E^{-\frac{x^2}{2\,\sigma^2}},\ \{x, -4, 4\},\ \mathrm{ImageSize} \to\right.$$



Other spatial filters-Gaussian

$$G_{1D}(x;\sigma) = \frac{1}{\sqrt{2\pi}\,\sigma}\, e^{-\frac{x^2}{2\sigma^2}},\; G_{2D}(x, y; \sigma) = \frac{1}{2\pi\sigma^2}\, e^{-\frac{x^2+y^2}{2\sigma^2}},\; G_{ND}(\vec{x}; \sigma) = \frac{1}{\left(\sqrt{2\pi}\,\sigma\right)^N}\, e^{-\frac{|\vec{x}|^2}{2\sigma^2}}$$

Normalization to 1.0

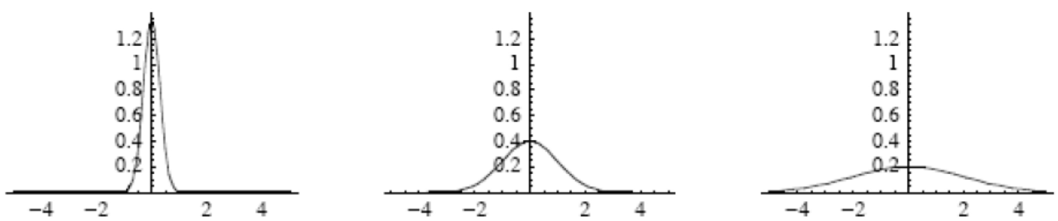


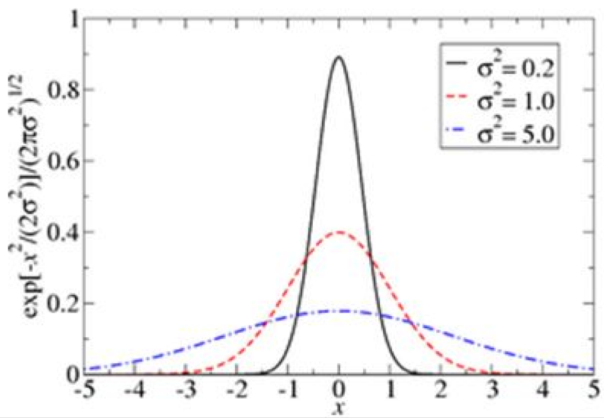
Figure 3.2 The Gaussian function at scales $\sigma = .3$, $\sigma = 1$ and $\sigma = 2$. The kernel is normalized, so the total area under the curve is always unity.

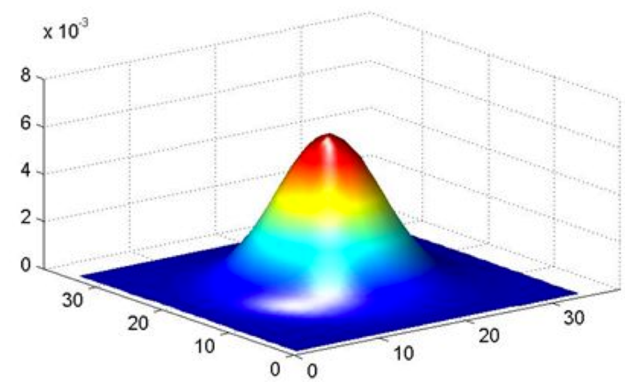
Other spatial filters-Gaussian

Gaussian Filter

- 1D Gaussian filter: $f(x) = e^{-\frac{x^2}{2\sigma^2}}$

- 2D Gaussian filter: $f(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}}$





Other spatial filters-Gaussian

1	2	1
2	4	2
1	2	1

2	7	12	7	2
7	31	52	31	7
12	52	127	52	12
7	31	52	31	7
2	7	12	7	2

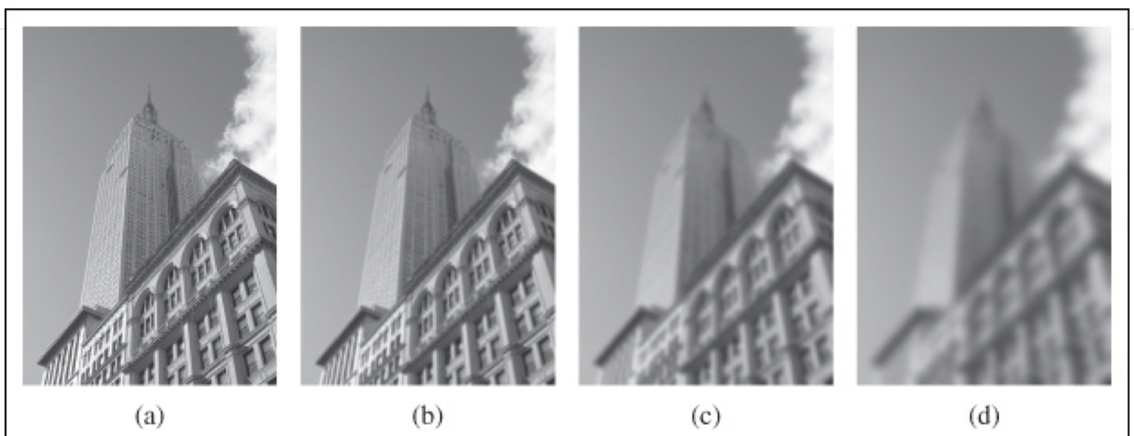
1	1	2	2	2	1	1
1	3	4	5	4	3	1
2	4	7	8	7	4	2
2	5	8	10	8	5	2
2	4	7	8	7	4	2
1	31	4	5	4	3	1
1	1	2	2	2	1	1

An integer valued 5 by 5 convolution kernel approximating a Gaussian with a σ of 1 is shown to the right,

$\frac{1}{273}$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

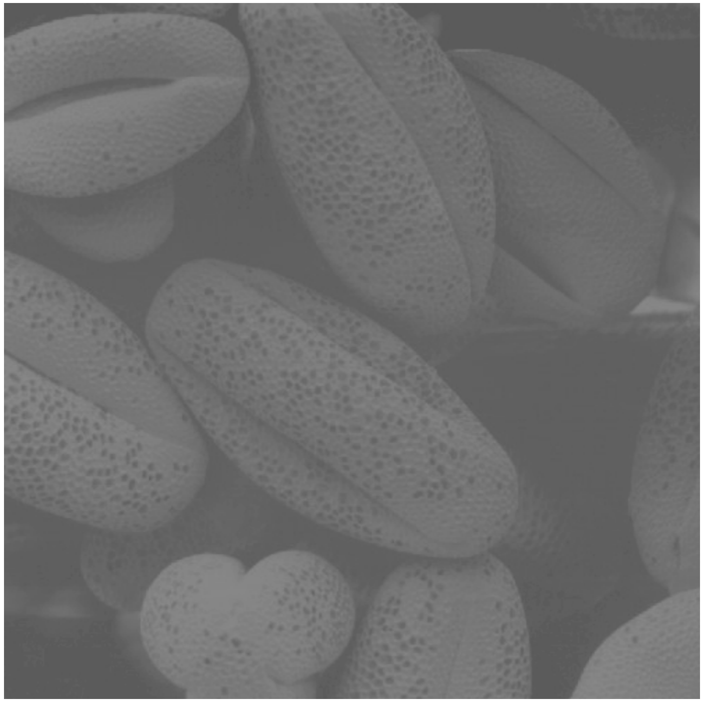
Other spatial filters-Gaussian



with $\sigma = 2$; (c) with $\sigma = 5$; (d) with $\sigma = 10$.

Exercise

- 1. Contrast stretch



$$s = \frac{r - r_{\min}}{r_{\max} - r_{\min}} \times 255$$

Exercise

- 2. Sobel filter

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

- Example: Robert filter

```
R1 = [-1 0
      0 1];

R2 = [0 -1
      1 0];

im2 = filter2(R1, im,'same');
im3 = filter2(R2, im, 'same');
im4 = sqrt((im2.^2+im3.^2)/2);
```

Exercise

- 3. median and averaging (mean) filter

```
%For median
clear all;
close all;

im = imread('circuit2.jpg');
im = double(im);
figure,
imagesc(im);
colormap(gray(256));
axis image;
axis off;

L = [1 1  1
     1 1 1
     1 1  1]/9;

----------------?
```

```
im2 = imfilter(im,L,'same');

figure,
% subplot(3,1,1)
imagesc(im2);
colormap(gray(256));
title('mean');
axis image;
axis off;

% subplot(3,1,2)
figure,
imagesc(abs(im3));
colormap(gray(256));
axis image;
axis off;
```

Matlab tools

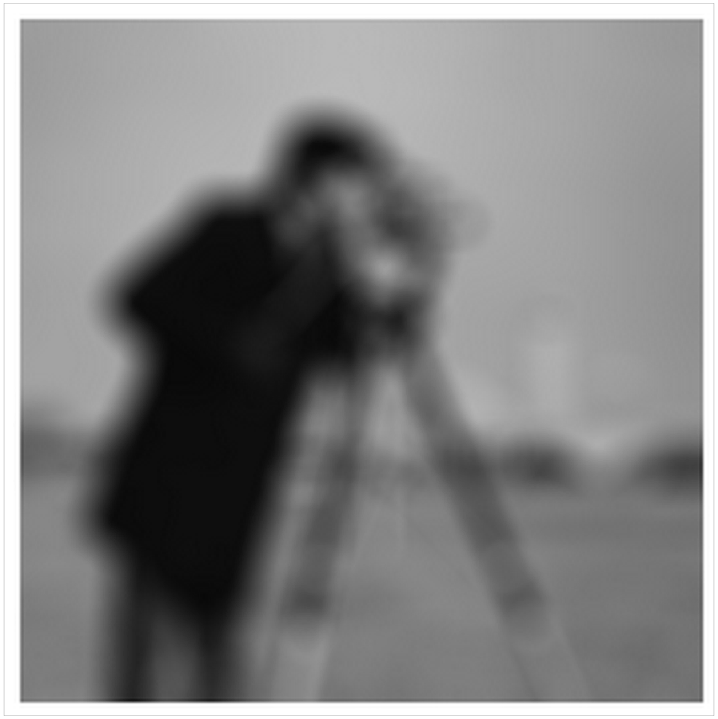
`h = fspecial(type)` creates a two-dimensional filter `h` of the specified `type`. `fspecial` returns `h` as a correlation kernel, which is the appropriate form to use with `imfilter`. *`type`* is a string having one of these values.

Value	Description
`average`	Averaging filter
`disk`	Circular averaging filter (pillbox)
`gaussian`	Gaussian lowpass filter
`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

Log: https://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm
e.g. BasicSpatialFilter2.m

Matlab tools

```
H = fspecial('disk',10);
Blurred=imfilter(I,H,'replicate');
imshow(blurred);
```



Exercise – find the biggest circle

circle.jpg on blackboard

Exercise – find the biggest circle

circle.jpg on blackboard

```
X =

    18     3     1    11
     8    10    11     3
     9    14     6     1
     4     3    15    21
```

```
[row,col] = find(X>0 & X<10,3)
```

```
row =

     2
     3
     4


col =

     1
     1
     1
```

Assignment 2 – find the pupil boundary