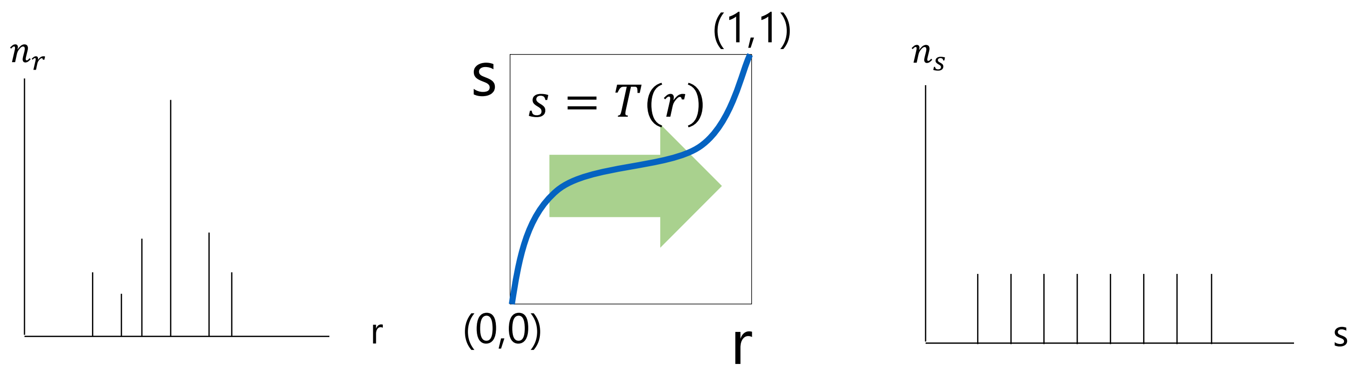
Image Processing

Region-wise Operation

Yeong Jun Koh
Department of Computer Science & Engineering
Chungnam National University

Histogram Equalization

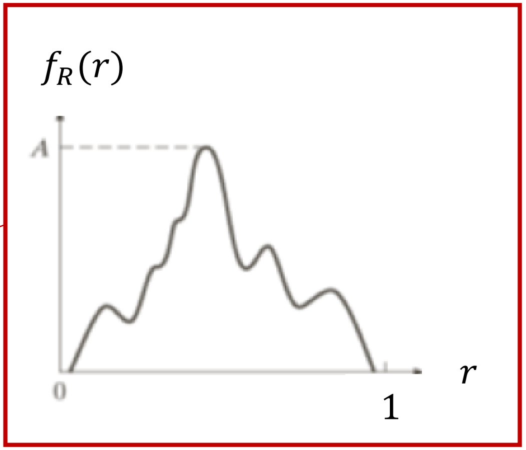
- Goal
  - Enhance an input image to have the gray level distribution, which is as uniform as possible

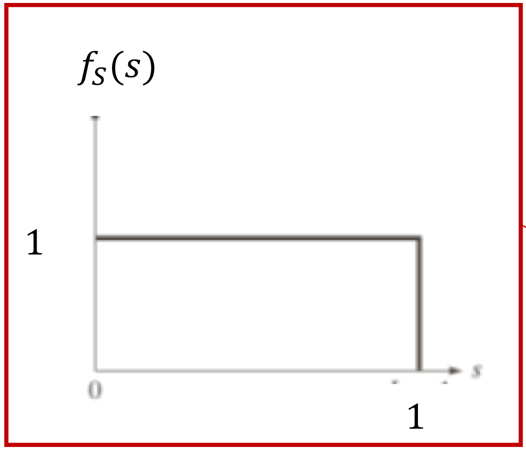


Histogram Equalization

- Continuous case
  - Regard histograms as probability density function (PDF)
  - Normalized r to [0, 1]

which we have





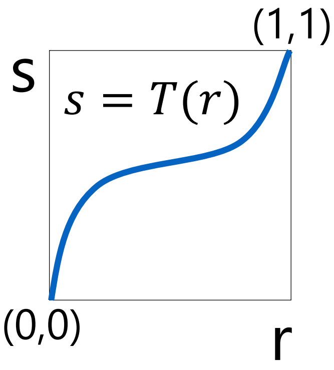
which we want

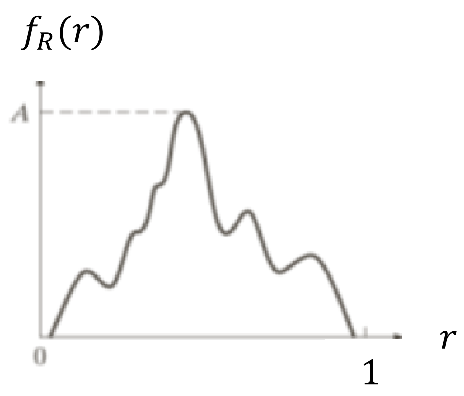
Histogram Equalization

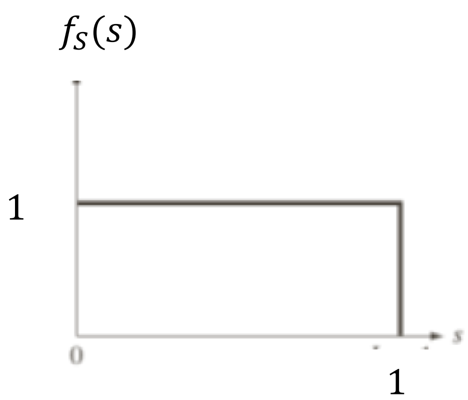
- Continuous case
  - $f_S(s) = 1$
  - $f_S(s)ds = f_R(r)dr$

$$f_R(r) = f_S(s)\frac{ds}{dr} = f_S(s)\frac{dT(r)}{dr} = T'(r)$$

$$T(r) = \int_{-\infty}^{r} f_R(t)dt = \int_{0}^{r} f_R(t)dt$$







Histogram Equalization

- Continuous case $$s = T(r) = CDF(r) = \int_0^r f_R(t)dt$$
- Discrete approximation $$s_k = T(r_k) = \sum_{j=0}^{k} f_R(r_j) = \sum_{j=0}^{k} \frac{n_j}{N} \quad , N = \sum n_r$$
  - Normalize histogram

input gray level k	0	1	2	3	4	5	6	7
normalized input r_k	0	1/7	2/7	3/7	4/7	5/7	6/7	1
histogram n_k	1	3	2	7	8	3	0	1
normalized histogram n_k/N	1/25	3/25	2/25	7/25	8/25	3/25	0	1/25
normalized output s_k	1/25	4/25	6/25	13/25	21/25	24/25	24/25	1
denormalized output $o_k = s_k \times 7$	7/25	28/25	42/25	91/25	147/25	168/25	168/25	7
output gray level floor(o_k)	0	1	1	3	5	6	6	7
m	0	1	2	3	4	5	6	7
output histogram n_m	1	5	0	7	0	8	3	1

Region-wise Operation

- Neighborhood in image

- Fundamentals to spatial filtering
  - Low-pass, high-pass filters
  - Gaussian, Laplacian filters

Spatial Filtering

- Spatial filtering is performed for each pixel (i, j) using neighborhoods of (i, j)
  - $I(i, j)$: input image $G(i, j)$: output image
  - $m(s, t)$: filtering kernel (mask, window, or template)

$$G(i, j) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} m(s, t) I(i + s, j + t)$$

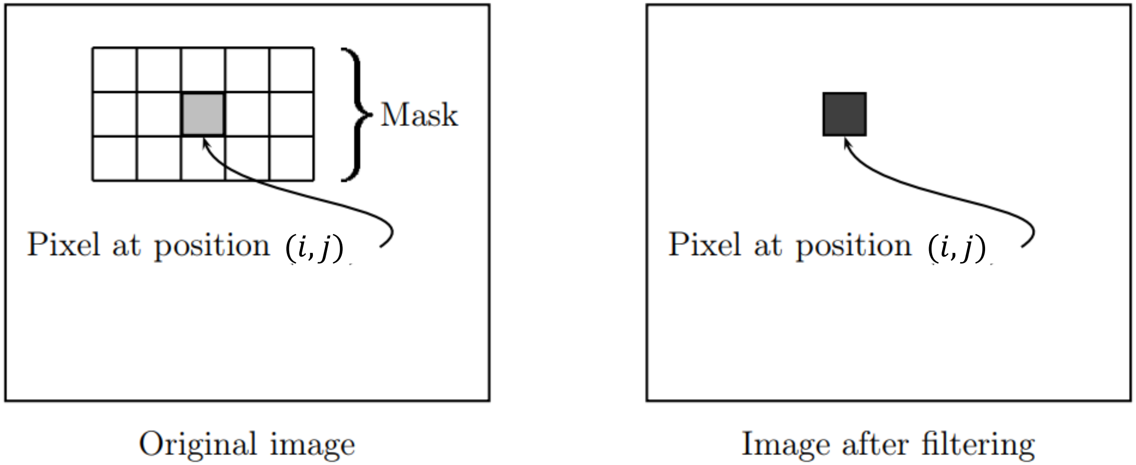


Figure 3.1: Using a spatial mask on an image

Filtering = SUM (Mask .× Neighborhood)

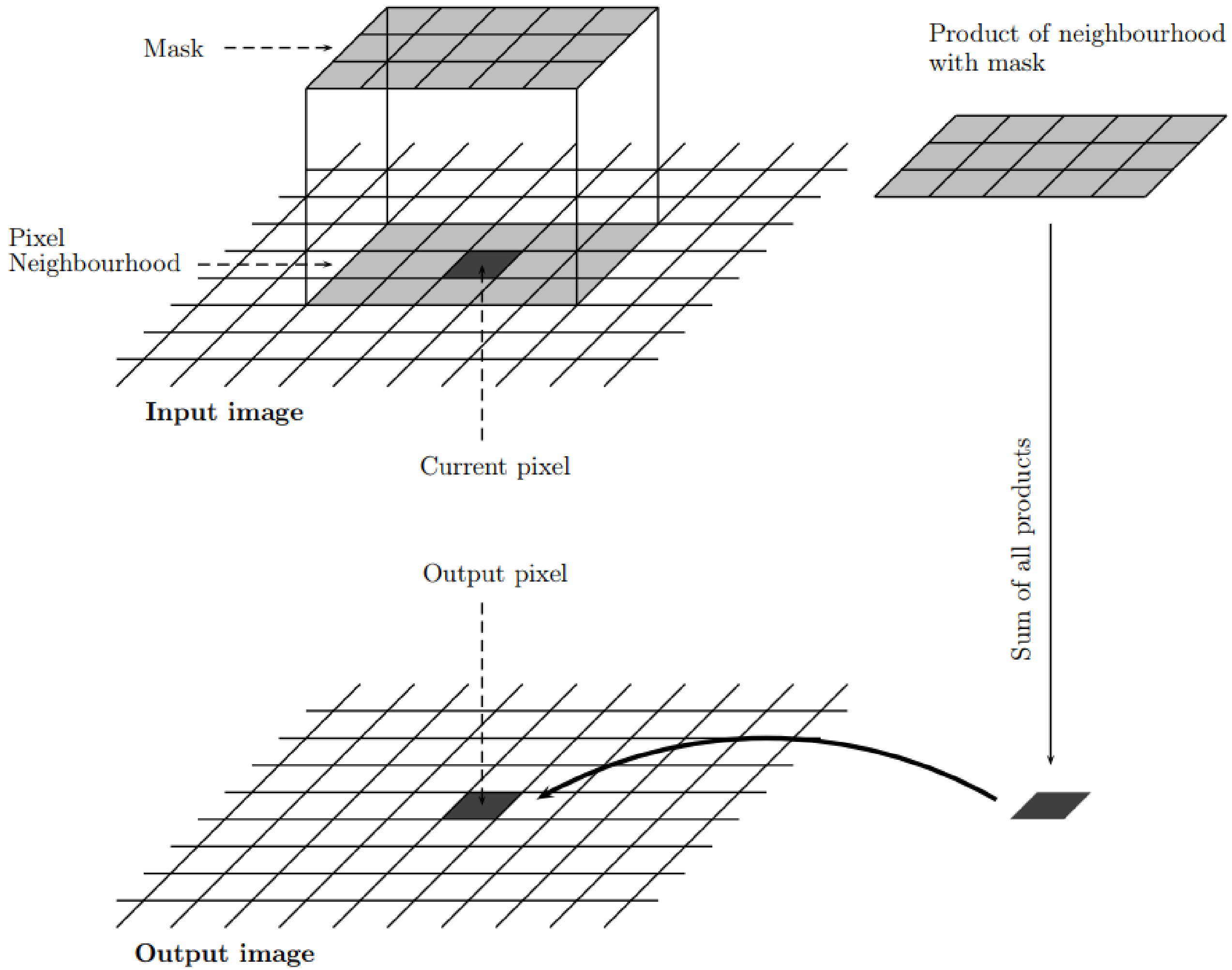


Figure 3.2: Performing spatial filtering

Filtering = SUM (Mask .× Neighborhood)

$$G(i,j) = \sum \left(M \;.\!\times\; P \right)$$

$m(-1,-2)$	$m(-1,-1)$	$m(-1,0)$	$m(-1,1)$	$m(-1,2)$
$m(0,-2)$	$m(0,-1)$	$m(0,0)$	$m(0,1)$	$m(0,2)$
$m(1,-2)$	$m(1,-1)$	$m(1,0)$	$m(1,1)$	$m(1,2)$

.×

$p(i-1,j-2)$	$p(i-1,j-1)$	$p(i-1,j)$	$p(i-1,j+1)$	$p(i-1,j+2)$
$p(i,j-2)$	$p(i,j-1)$	$p(i,j)$	$p(i,j+1)$	$p(i,j+2)$
$p(i+1,j-2)$	$p(i+1,j-1)$	$p(i+1,j)$	$p(i+1,j+1)$	$p(i+1,j+2)$

Question

$$\sum_{s=-a}^{a} \sum_{t=-b}^{b} m(s,t) = ?$$

Spatial Filtering

- Spatial filtering is performed for each pixel (i, j) using neighborhoods of (i, j)
  - $I(i, j)$: input image $G(i, j)$: output image
  - $m(s, t)$: filtering kernel (mask, window, or template)

$$G(i,j) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} m(s,t) I(i+s, j+t)$$

- The filter is defined according to what kind of $m(s, t)$ is used
- The filter serves as an essential building block for many application
  - Blurring, sharpening, image restoration, and so on
  - Filter kernel should be defined depending on applications

Filtering = SUM (Mask .× Neighborhood)

- Masking with a mask m of size $(2a+1) \times (2b+1)$

$$G(i,j) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} m(s,t) I(i+s, j+t)$$

- Convolving with a filter h of size $(2a+1) \times (2b+1)$

$$G'(i,j) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} h(s,t) I(i-s, j-t)$$

- Note that $G(i,j) = G'(i,j)$ if $m(s,t) = h(-s,-t)$

Masking with

a	b	c
d	e	f
g	h	i

=

Convolving with

i	h	g
f	e	d
c	b	a

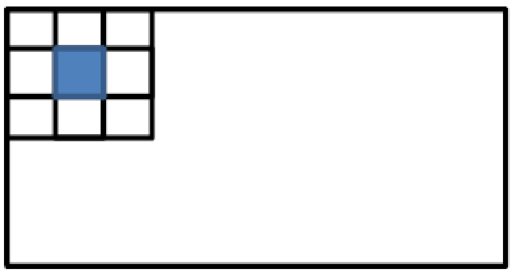
Exceptional Process on Image Boundary

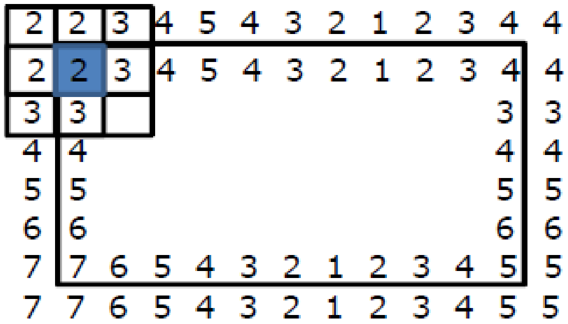
- Boundary problem
  - Limit the excursion of the center of the mask, so that the mask is fully contained within the image
  - Output image is smaller than input image
- How to process pixels at image boundary?

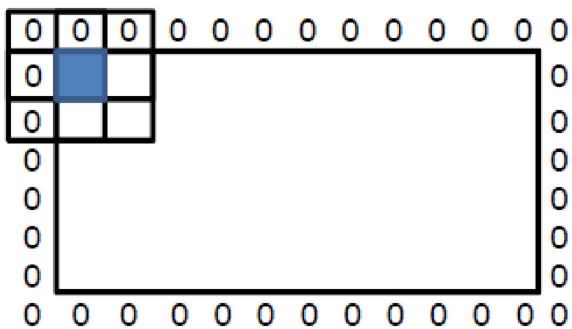
Exceptional Process on Image Boundary

- Boundary problem

sk







Or any other ways?

Exceptional Process on Image Boundary

- Zero padding
- Repetition
- Mirroring
- etc

0	0	0	0	0
0	0	0	0	0
0	0	a	b	c
0	0	d	e	f
0	0	g	h	i

Zero padding

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

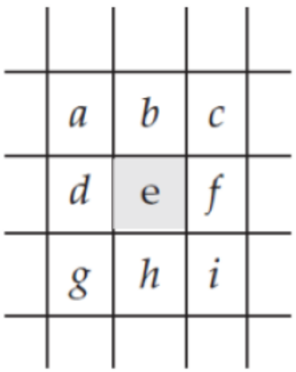
Repetition

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

Mirroring

Image Averaging

- Uniform mean filtering
  - The simplest low-pass filter



$$\longrightarrow \frac{1}{9}(a+b+c+d+e+f+g+h+i)$$

Filter kernel

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

Image Averaging

(a) Original image

(b) Average filtering

(c) Using a 9×9 filter

(d) Using a 25×25 filter

Figure 3.4: Average filtering

Smoothing Spatial Filters

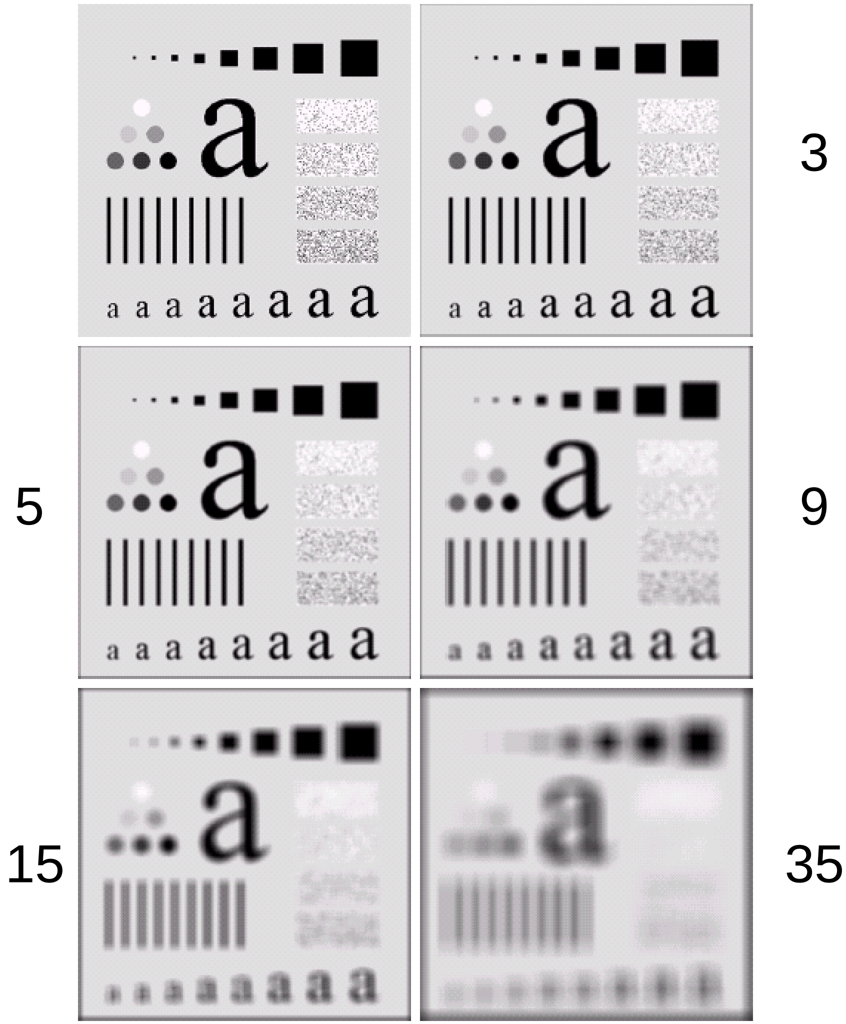
- Averaging filter and weighted averaging filter

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- Blends with adjacent pixel values
- Blurring
  - Removal of small details before large object extraction
  - Bridging of small gaps in lines or curves
  - Reduction of sharp transitions in gray levels
    - Advantage: noise reduction
    - Disadvantage: edge blurring

Smoothing Spatial Filters

- Losing edges
- Reducing noises
- Removing small objects



Smoothing Spatial Filters

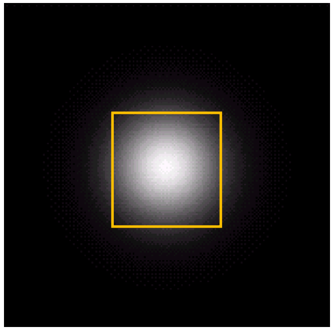
- Low-pass filter
  - Uniform averaging filter
  - Weighted averaging filter
  - Gaussian filter

Gaussian Filters

- Gaussian distribution
  - One of the most commonly used parametric models

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{(x-\mu_x)^2}{2\sigma_x{}^2} - \frac{(y-\mu_y)^2}{2\sigma_y{}^2}\right)$$



2d Gaussians

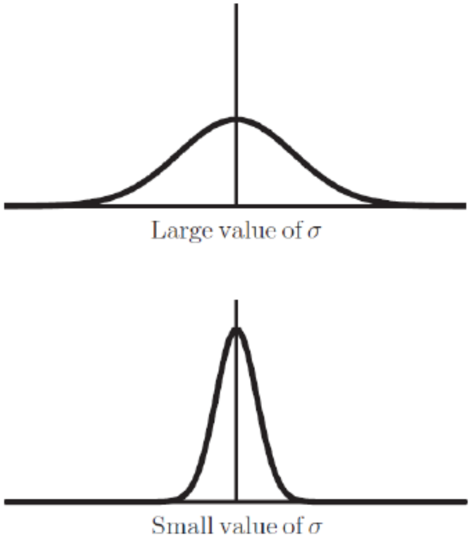


FIGURE 5.7 *One-dimensional Gaussians.*

Gaussian Filters

- Gaussian filter's advantage
  - It considers spatial distance within neighborhoods

High-pass filter

- High-pass filter
  - High frequencies: details, edges and noise
  - Estimate intensity change
  - Image sharpening

Sharpening Spatial Filters

- Highlight fine detail
  - Enhance detail that has been blurred

- Difference operator
  - cf. summation operator for smoothing
  - Derivative in digital domain

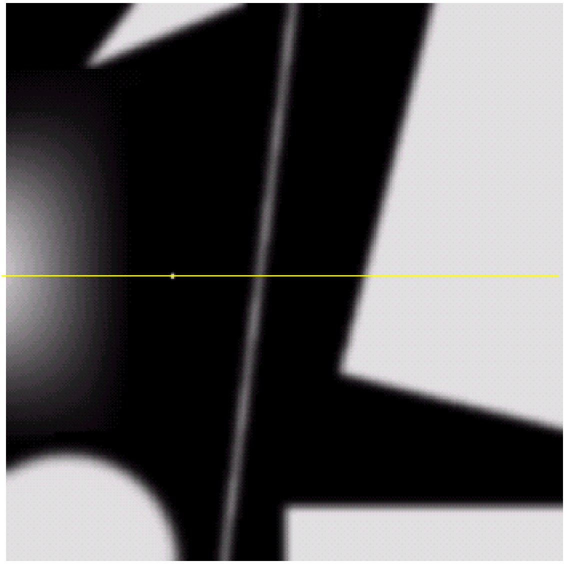
- 1st-order derivative (1D case)

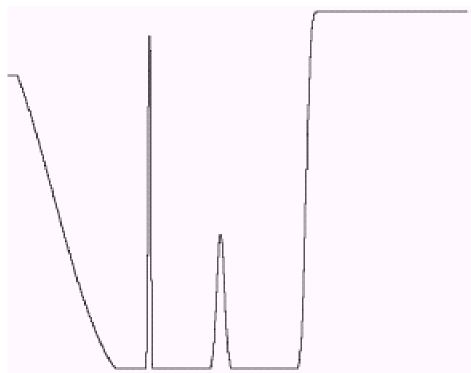
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- 2nd-order derivative

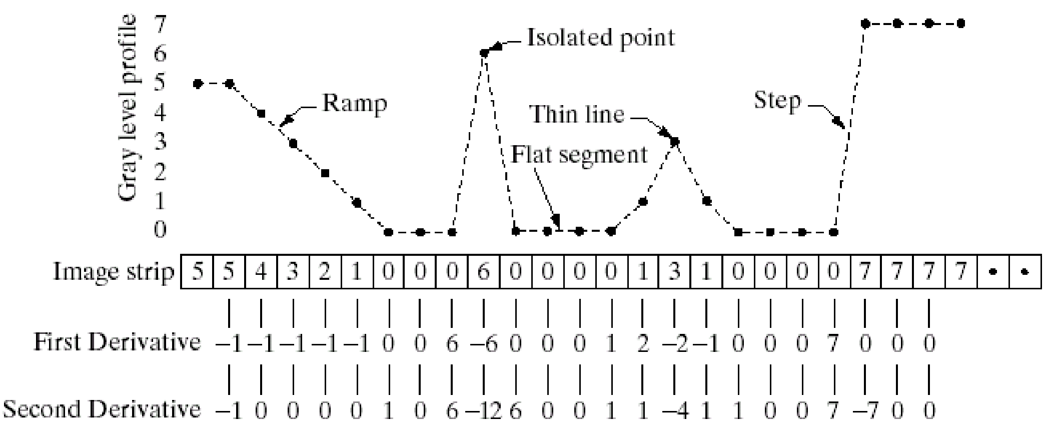
$$\frac{\partial^2 f}{\partial x^2} = f(x+1) - f(x) - [f(x) - f(x-1)] = f(x+1) + f(x-1) - 2f(x)$$

Sharpening Spatial Filters





- 1st-order derivative generates thicker edges
- 2nd-order derivative has a stronger response to fine detail



Laplacian Filter

- Use of second derivatives for enhancement

Laplacian Filtering output $L(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$

- Negative definition

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

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

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

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y).$$

0	1	0
1	-4	1
0	1	0

- Positive definition

$$\nabla^2 f = -[f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] + 4f(x, y).$$

Laplacian Filter

- Laplacian mask
  - Laplacian operator indicates how brighter the current pixel is than the neighborhood

0	1	0
1	-4	1
0	1	0

or

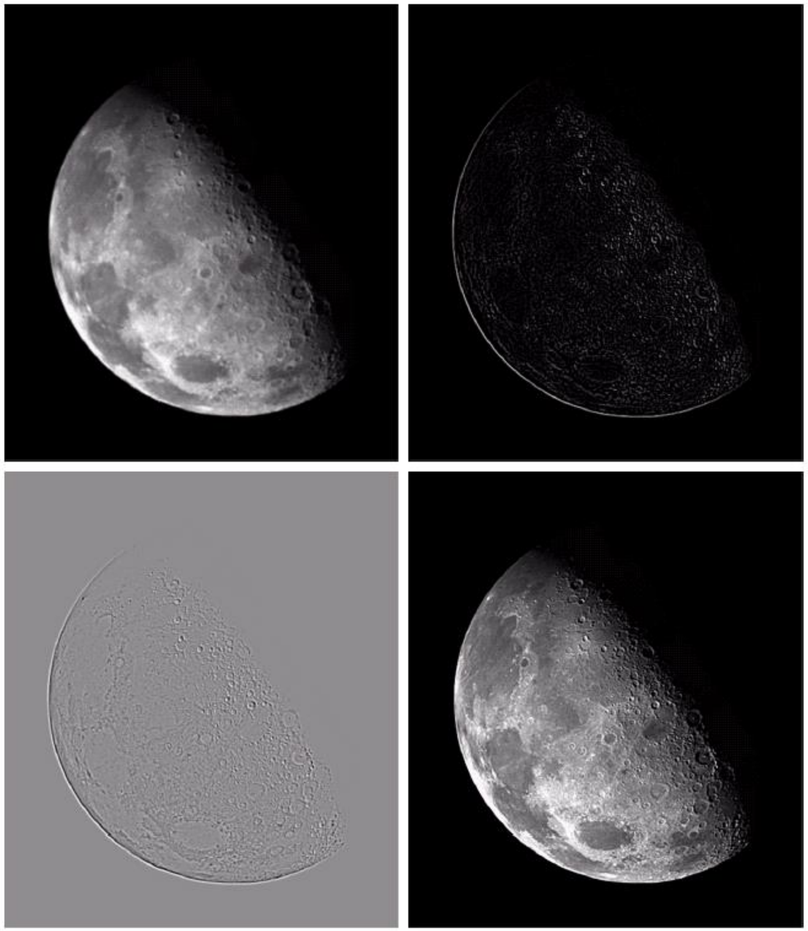
1	1	1
1	-8	1
1	1	1

More general form

$\frac{\alpha}{1+\alpha}$	$\frac{1-\alpha}{1+\alpha}$	$\frac{\alpha}{1+\alpha}$
$\frac{1-\alpha}{1+\alpha}$	$\frac{-4}{1+\alpha}$	$\frac{1-\alpha}{1+\alpha}$
$\frac{\alpha}{1+\alpha}$	$\frac{1-\alpha}{1+\alpha}$	$\frac{\alpha}{1+\alpha}$

Laplacian Filter

$f(x, y)$ $\nabla^2 f(x, y)$



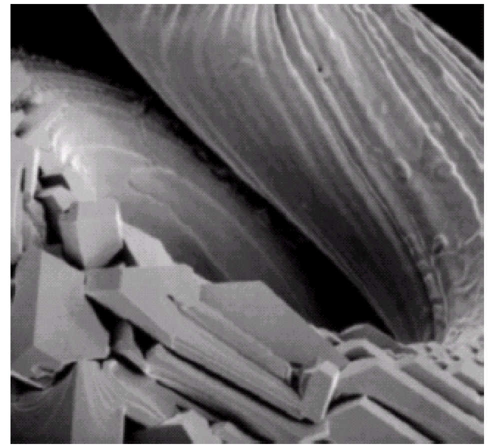
$f(x, y) + \nabla^2 f(x, y)$

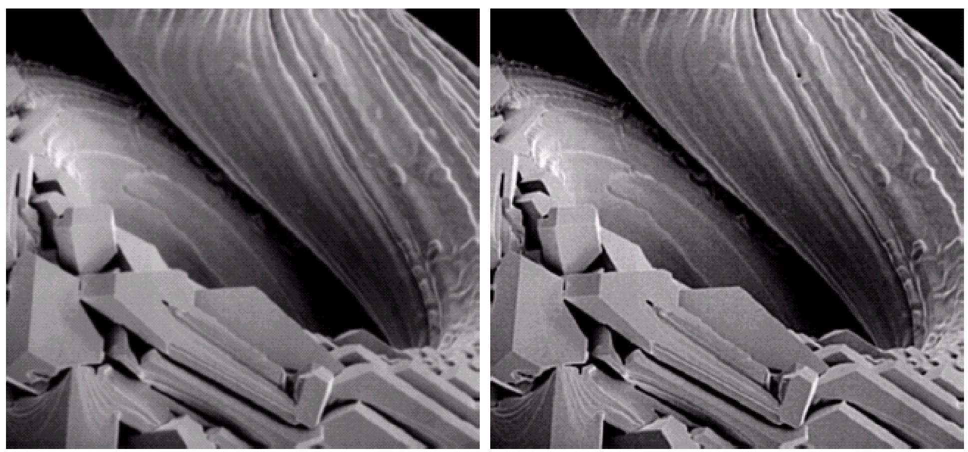
Enhancing details
Frequently used sharpening filter

Laplacian Filter

0	−1	0
−1	5	−1
0	−1	0

−1	−1	−1
−1	9	−1
−1	−1	−1





a b c
d e

FIGURE 3.41 (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Sobel Filter

- Use of first derivatives (directional gradient)
  - Gradient

$$\nabla \mathbf{f} = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]^T$$

  - Magnitude of gradient

$$\nabla f = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \approx \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|$$

  - Sobel mask

$$M(x, y) = |S_x * f| + |S_y * f|$$

S_x

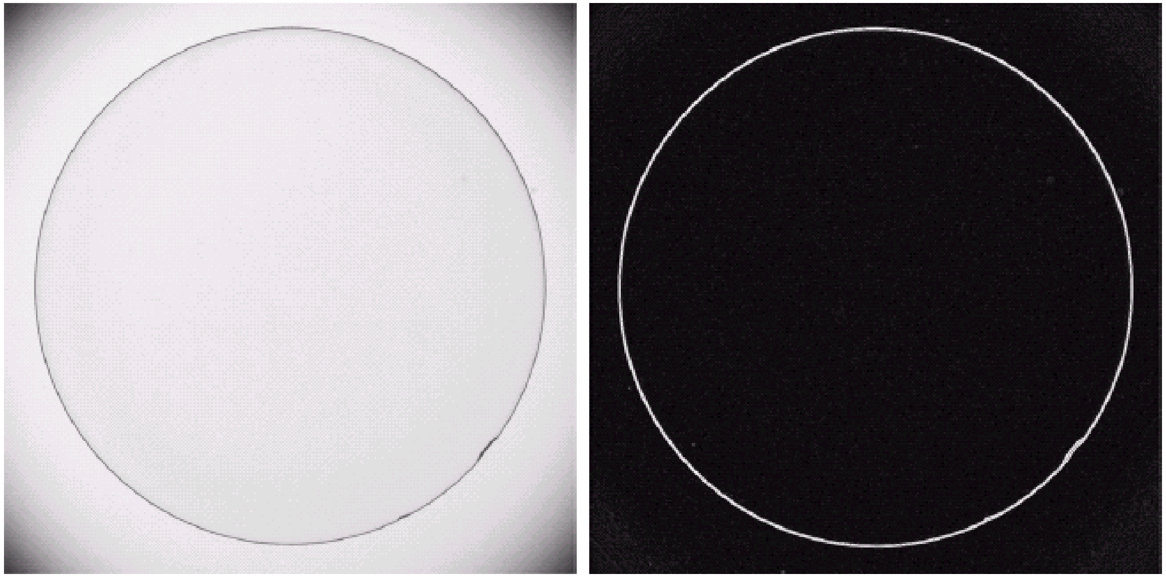
-1	0	1
-2	0	2
-1	0	1

S_y

-1	-2	-1
0	0	0
1	2	1

Sobel Filter

f ∇f



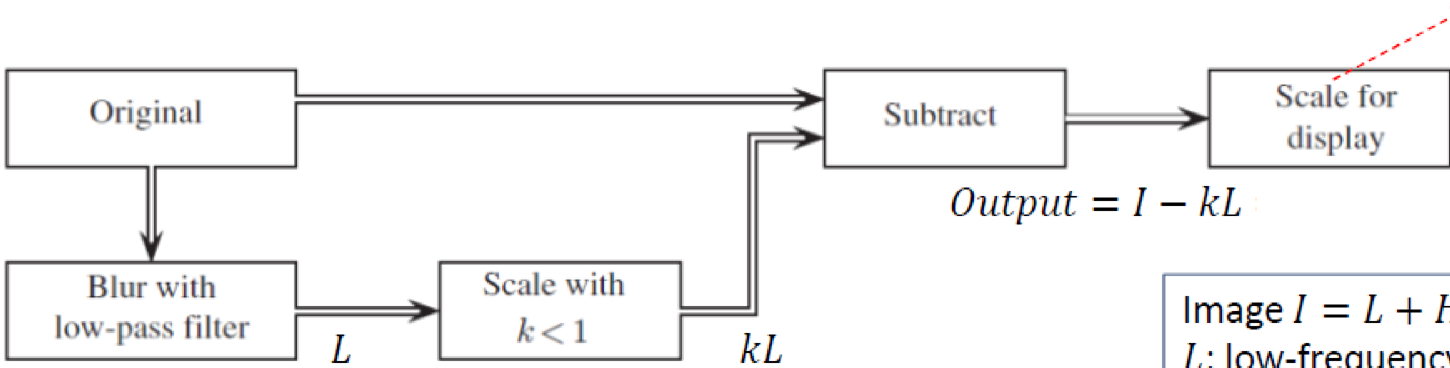
a b

FIGURE 3.45 Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Mr. Pete Sites, Perceptics Corporation.)

Edge Sharpening

- Unsharp Masking

Scaling matters!



$Output = I - kL$

FIGURE 5.10 *Schema for unsharp masking.*

Image $I = L + H$
L: low-frequency component
H: high-frequency component

Unsharp Masking

$$f = I - kL$$

$$f = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - k \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

Scaling

$$f = \frac{1}{1-k} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{k}{1-k} \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

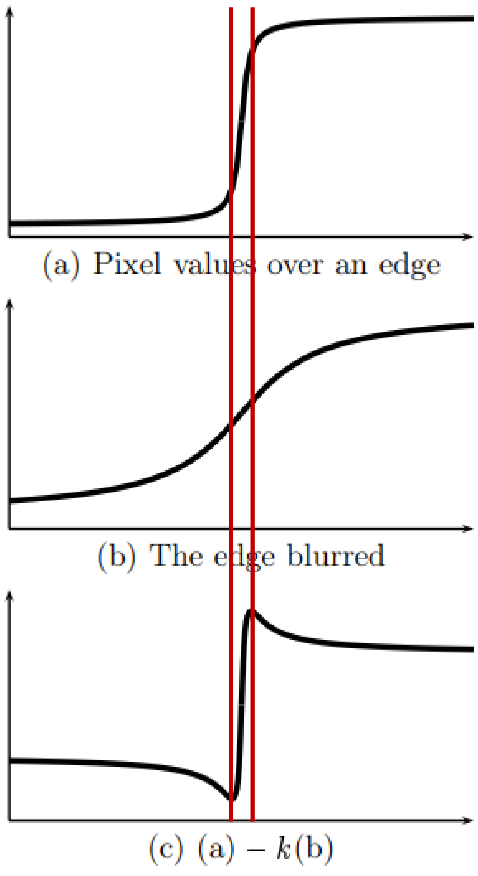


Figure 3.9: Unsharp masking

Unsharp Masking

(a) Original image

(b) The image after unsharp masking

Figure 3.8: An example of unsharp masking

Unsharp Masking

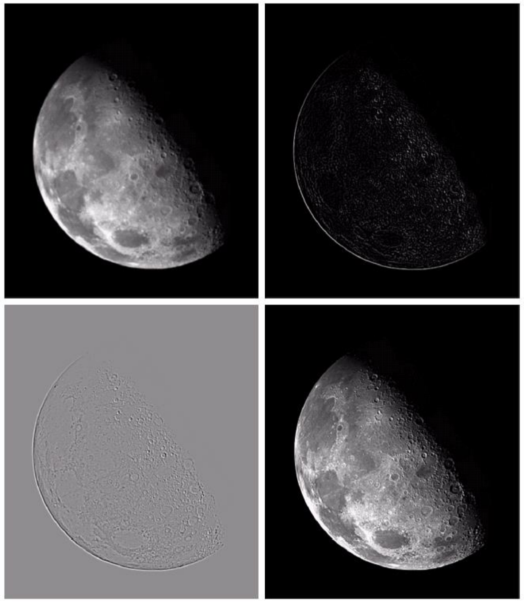
- High-boost filtering

$$F_{hb} = AI - L$$
$$= (A-1)I + I - L = (A-1)I + H$$

- If we use the Laplacian

$$H = I + \nabla^2 f$$

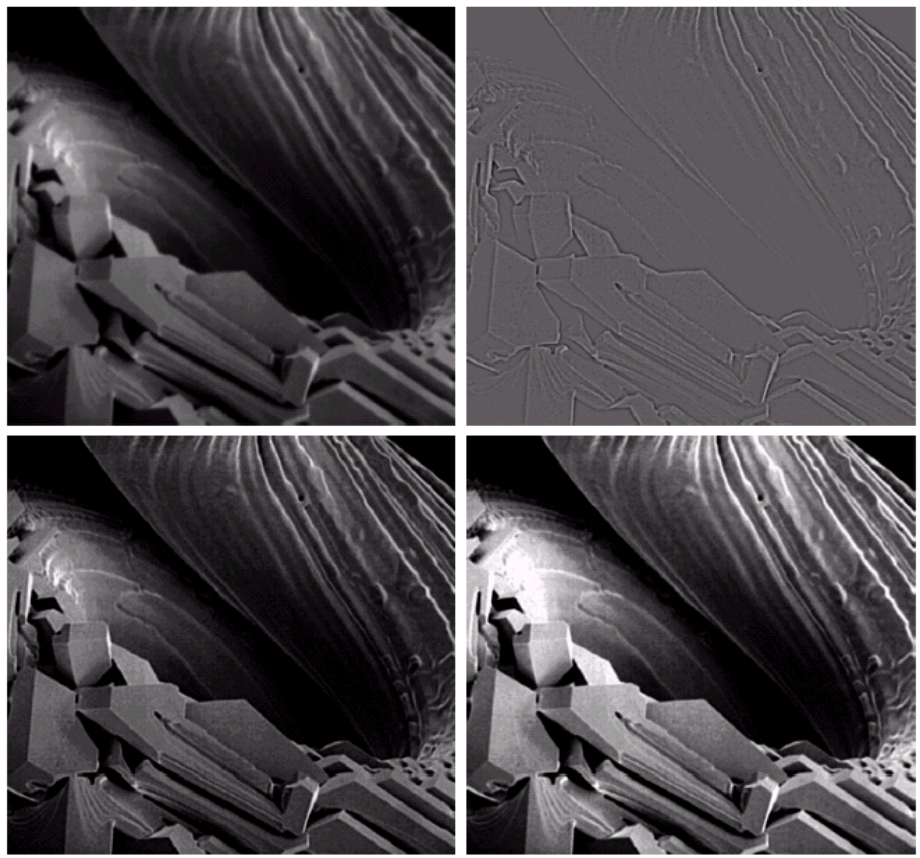
$$F_{hb} = AI + \nabla^2 f$$



High-boost Filtering Using Laplacian

- Input image is darker than desired

$$F_{hb} = AI + \nabla^2 f$$



a b
c d

FIGURE 3.43 (a) Same as Fig. 3.41(c), but darker. (a) Laplacian of (a) computed with the mask in Fig. 3.42(b) using $A = 0$. (c) Laplacian enhanced image using the mask in Fig. 3.42(b) with $A = 1$. (d) Same as (c), but using $A = 1.7$.

Order-Statistics Filter

- Sort the gray levels of the neighborhood
  - (0, 1, 2, 2, 3, 4, 5, 6, 6)

    min median max
- Min filter
  - Replace the center pixel with the minimum gray level (0)
- Max filter
  - Replace the center pixel with the maximum gray level (6)
- Median filter
  - Replace the center pixel with the median (3)
  - Excellent suppression of salt-and-pepper noises without blurring

6	4	6
2	1	3
2	5	0

3x3 averaging filter 3x3 median filter