

# Image Enhancement

Fall 2024

Yi-Ting Chen

54-59 !!



What do you want to be?

Could we bring values to others?

# Yi-Ting Chen

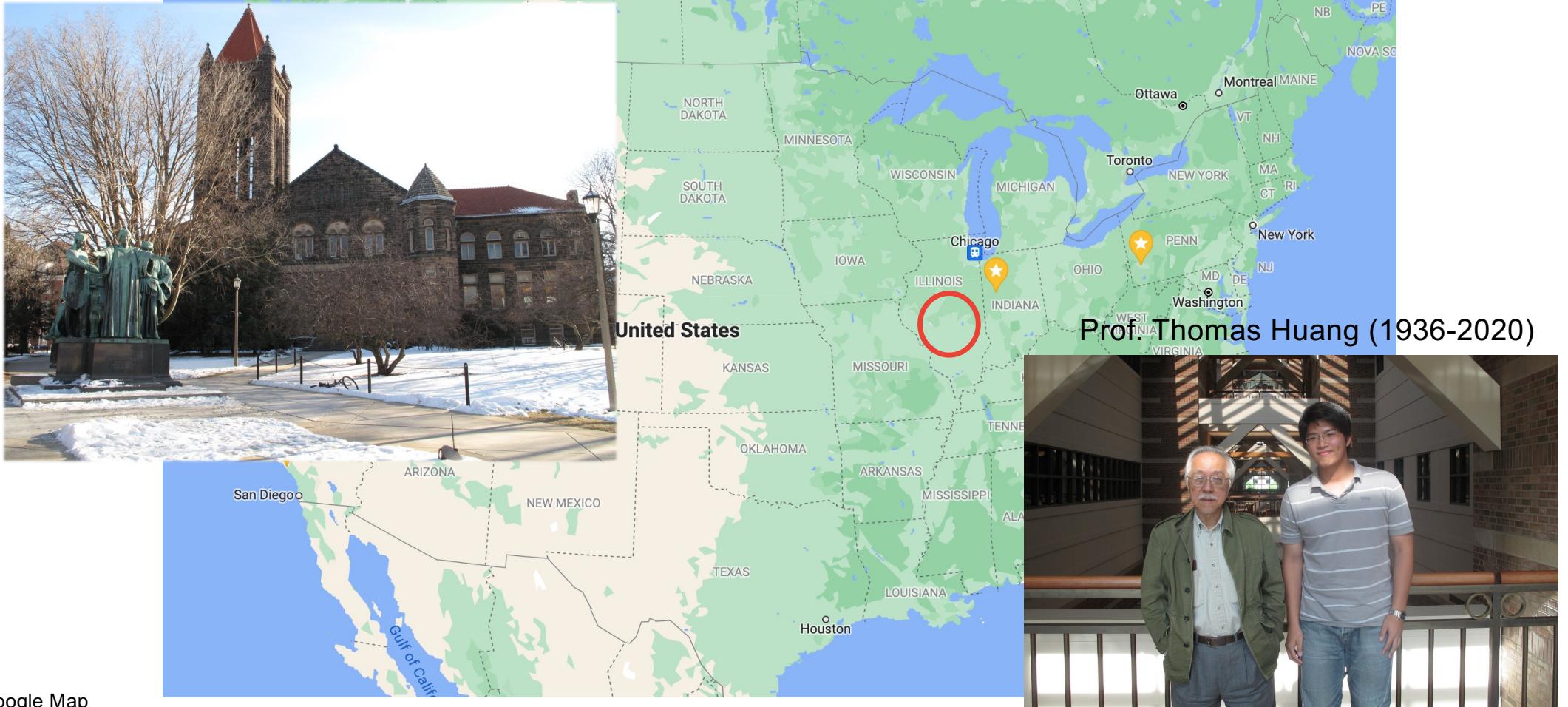
- NCTU EE B.S., Taiwan (2005-2009)
- UIUC ECE Exchange Student, USA (2009.01-06)
- Purdue University ECE Ph.D., USA (2010-2015)
- HP Inc. Imaging Science Intern, USA (2011.06-08)
- UC Merced, Computer Vision Researcher, USA (2014-2015)
- Google Inc. Visiting Scholar, USA (2015.01-04)
- Honda Research Institute USA, Senior Scientist (2015-2020)
- NYCU CS Assistant Professor, Taiwan (2021-Present)
- Director of HCIS Lab, Taiwan (2021-Present)



# Lesson 1

# Do the homework

# University of Illinois at Urbana-Champaign



# First “Intense” Research Experience



Two seniors met at UIUC

## A STUDY OF MULTI-VIEW GENDER RECOGNITION ON A LARGE DATABASE

*Yi-Ting Chen, Zhen Li, Thomas S. Huang*

University of Illinois at Urbana-Champaign  
Beckman Institute

### ABSTRACT

In this paper, we conduct a systematic study of multi-view gender recognition on a large database. We focus on the basic methodologies of dealing with view point variations, and investigated into their performance on gender recognition problems. Experimental evaluation is conducted on a large multi-view database consisting of 23,136 facial images of 241 subjects under 32 different views and 3 different illuminations. We also introduce a Pyramid-SVM approach for facial recognition tasks. As an extension to the conventional “holistic features + SVM” framework, our proposed approach manages to preserve local information of face images originally discarded in the holistic features. Experimental results demonstrate that the Pyramid-SVM approach significantly outperforms the traditional SVM classifier.

*Index Terms*— Multi-View, Gender Recognition, Pyramid-SVM

### 1. INTRODUCTION

Although a lot of studies have been conducted on facial gender recognition, most of them involves only frontal or near frontal faces (except [1]). This is far from reality, though. In real-life, facial images captured from the videos, especially in surveillance or ECRM (Electronic Customer Relationship Management) scenarios, have arbitrary illumination, imaging noises, sometimes non-neutral emotions and most importantly, the faces are often captured in a non-frontal view. The current algorithms fail to provide a solution for gender identification in these real-world situations. One major difficulty of conducting such kind of research is the lack of a large multi-view database that contains subjects under all kinds of poses (and even with illuminations or expressions variation).

In this paper, we conduct a systematic study of multi-view gender recognition problem on a large database. Especially, we focus on the methodology of dealing with view point variations, which is rarely discussed in the literature. Experiments are based on the private YAS database, which consists of 23,136 facial images of 241 subjects under 32 different views and 3 different illuminations. Because of our primary interest in the issue of view point variation, and also because face images are well aligned in the YAS database, we simply

After working on the gender recognition problem, I thought I knew what pattern recognition is!

But I really don't...

# Graduate School Application (PhD)

- In 2009 (Financial crisis)
- 11 programs based on US News Ranking
- Research Interest
  - Image processing and pattern recognition
  - I wrote them because I have some research experiences
- However, there are lots of faculties working on these two keywords!



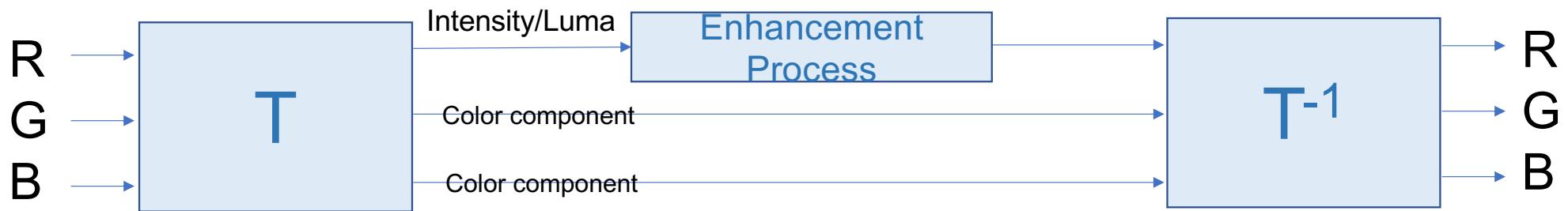
# Takeaway

- Before applying to a program (MS or PhD), please try your best to understand what they are working on!
  - Check out websites
  - Chat with seniors
  - Read papers
  - Ask faculties

# Topics

- Contrast Enhancement
- Sharpness Enhancement
- Noise Suppression

# Enhancement of Color Images

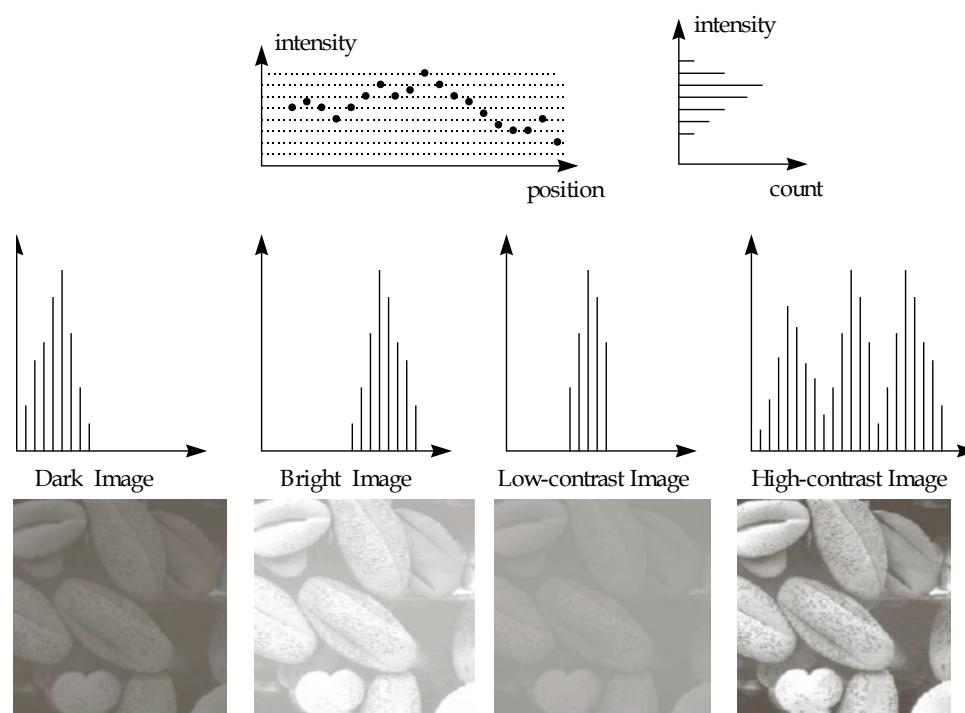


# Topics to be discussed in Image Enhancement

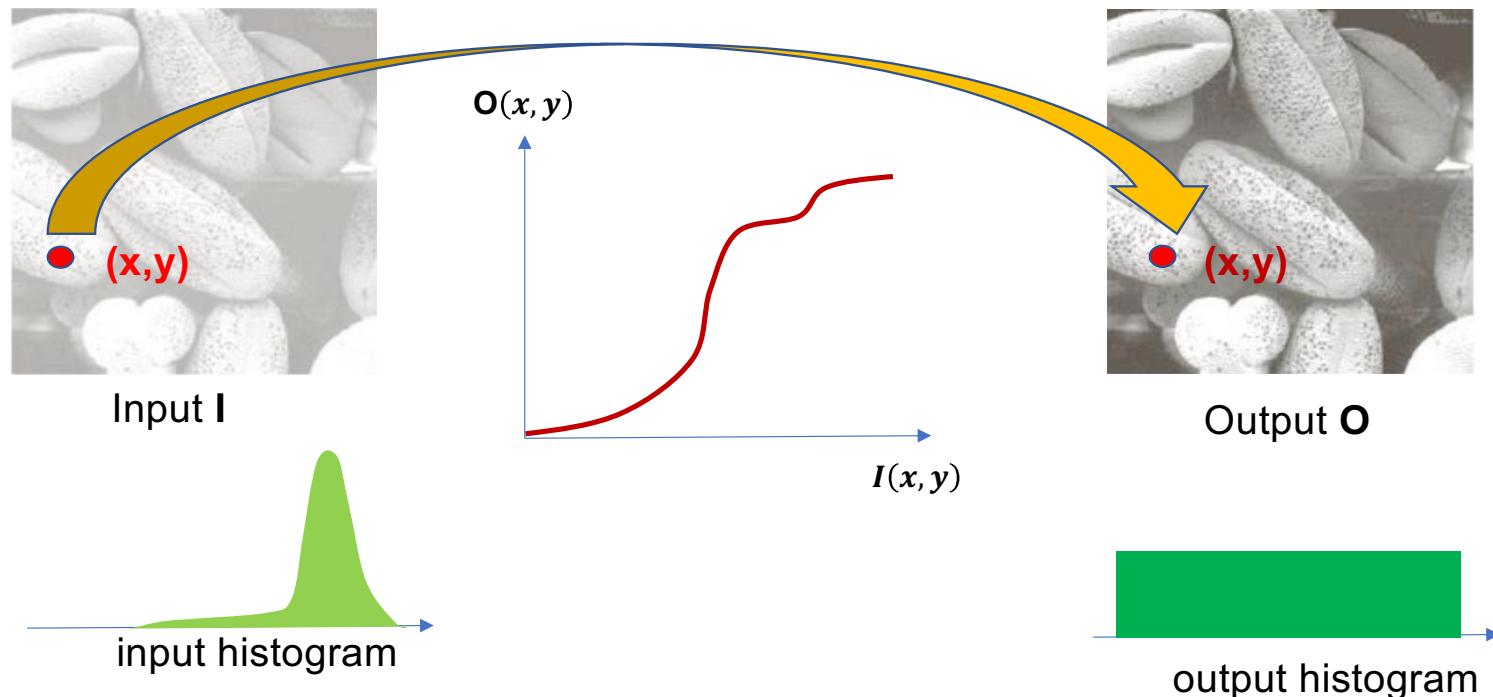
- Intensity domain
  - Histogram
- Spatial domain
  - Linear filtering
  - Nonlinear filtering
- Frequency domain
  - High-pass filtering
  - Band-pass filtering
  - Low-pass filtering

# Intensity-domain Processing

# Histogram

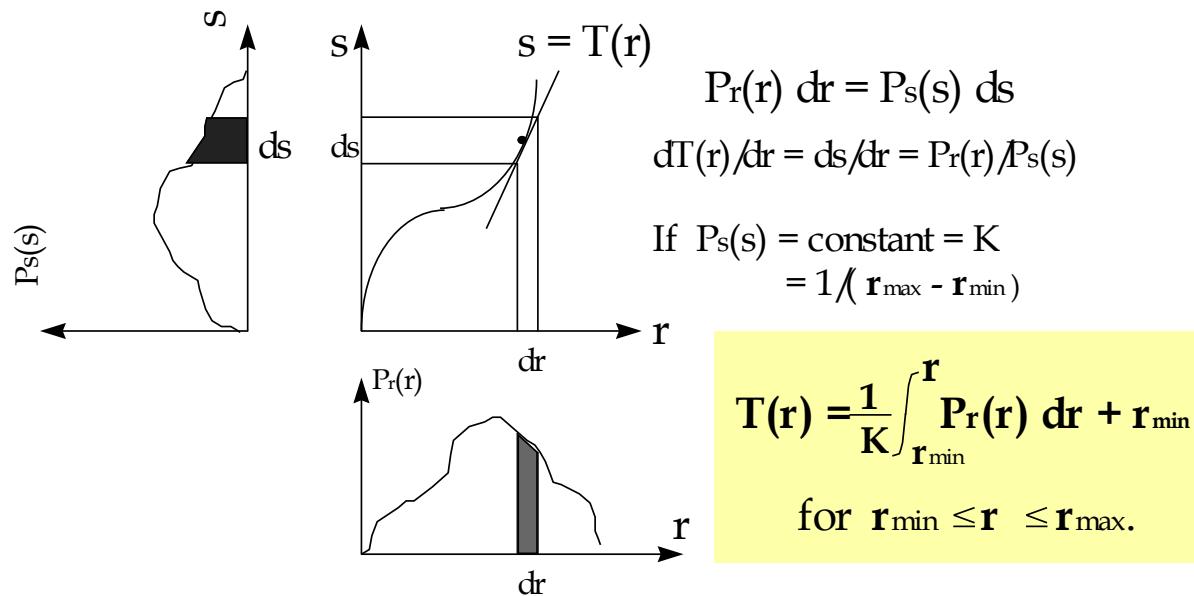


# Histogram Equalization (1/3)

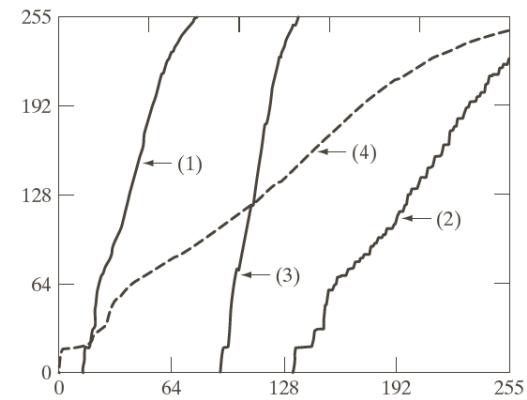
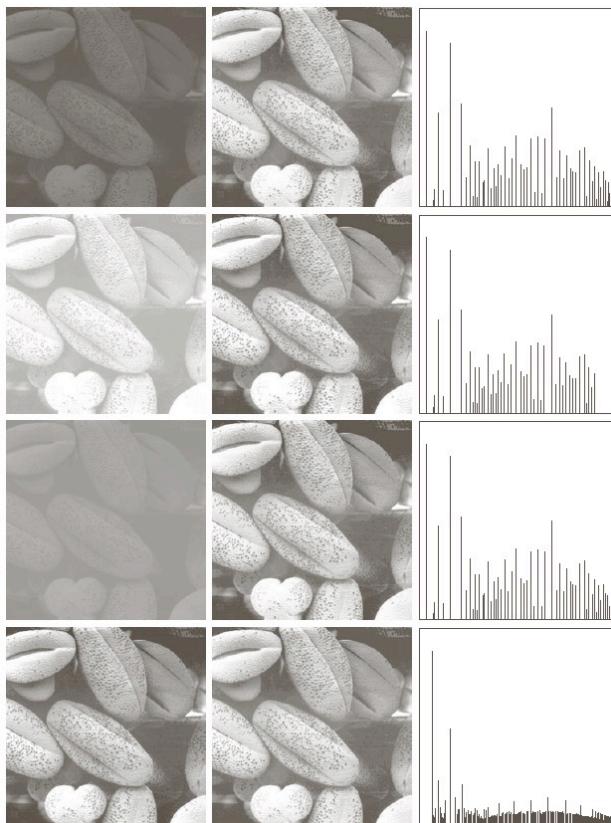


# Histogram Equalization (2/3)

- (a)  $T(r)$  is a monotonic<sup>†</sup> increasing function in the interval  $0 \leq r \leq L - 1$ ; and
- (b)  $0 \leq T(r) \leq L - 1$  for  $0 \leq r \leq L - 1$ .

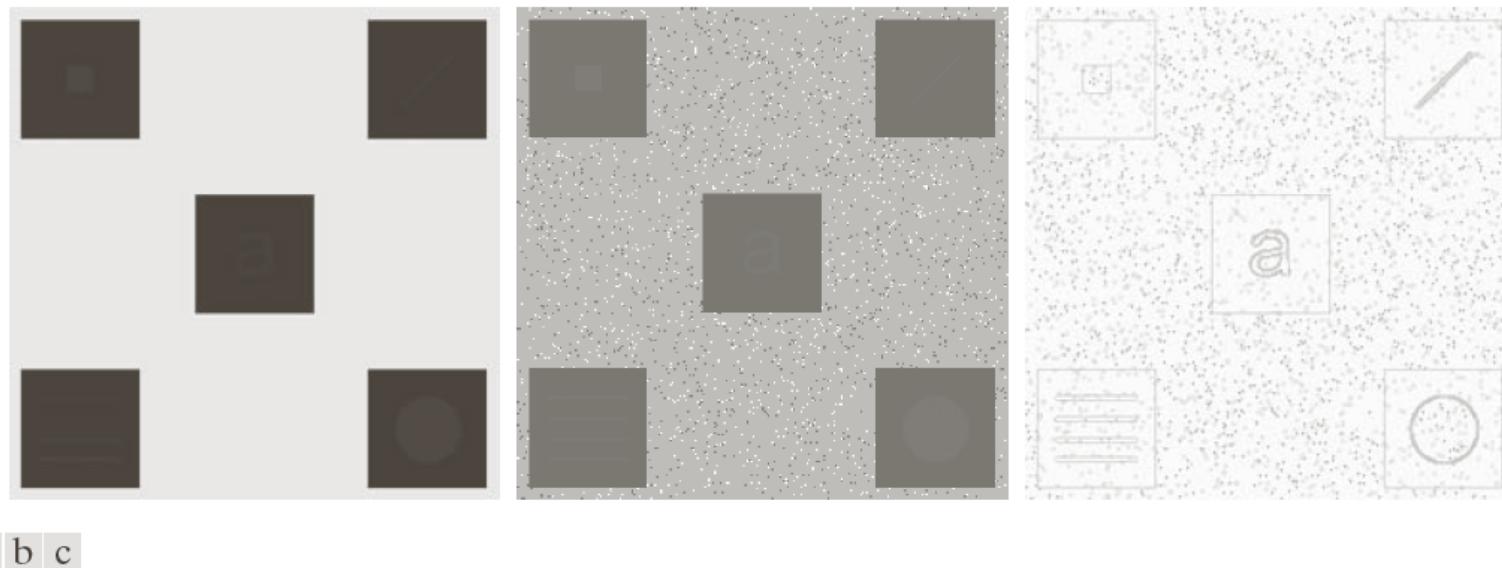


# Histogram Equalization (3/3)



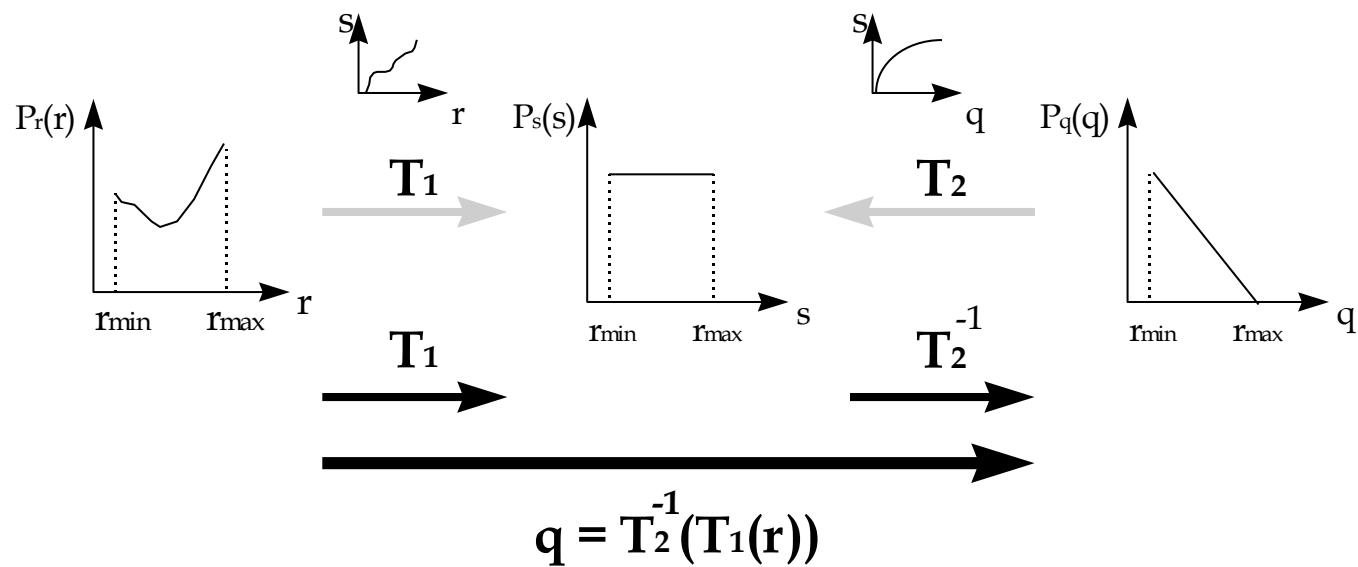
**FIGURE 3.21**  
Transformation functions for histogram equalization. Transformations (1) through (4) were obtained from the histograms of the images (from top to bottom) in the left column of Fig. 3.20 using Eq. (3.3-8).

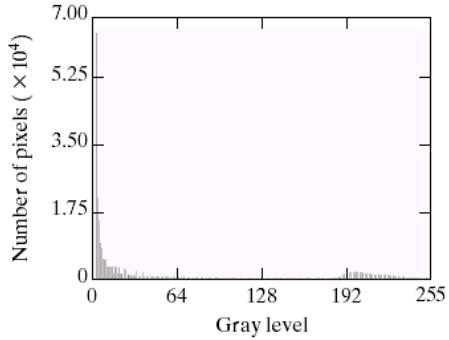
# Local Histogram Equalization



**FIGURE 3.26** (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size  $3 \times 3$ .

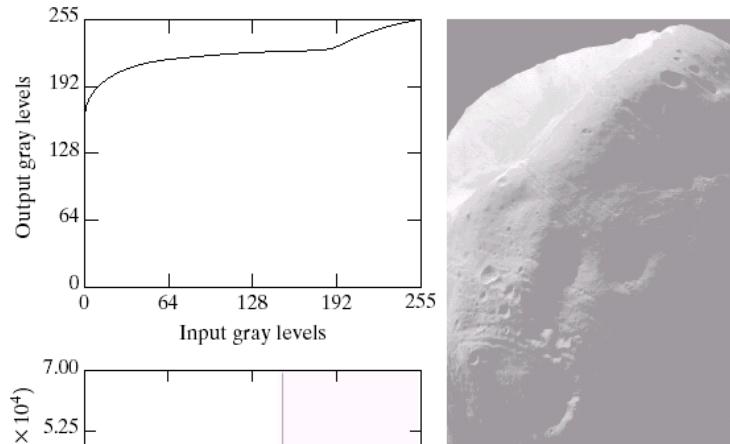
# Histogram Matching





a b

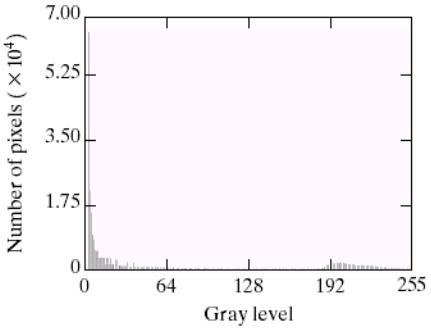
**FIGURE 3.20** (a) Image of the Mars moon Photos taken by NASA's *Mars Global Surveyor*. (b) Histogram. (Original image courtesy of NASA.)



a  
b  
c

**FIGURE 3.21**  
(a) Transformation function for histogram equalization.  
(b) Histogram-equalized image (note the washed-out appearance).  
(c) Histogram of (b).

### Histogram Equalization

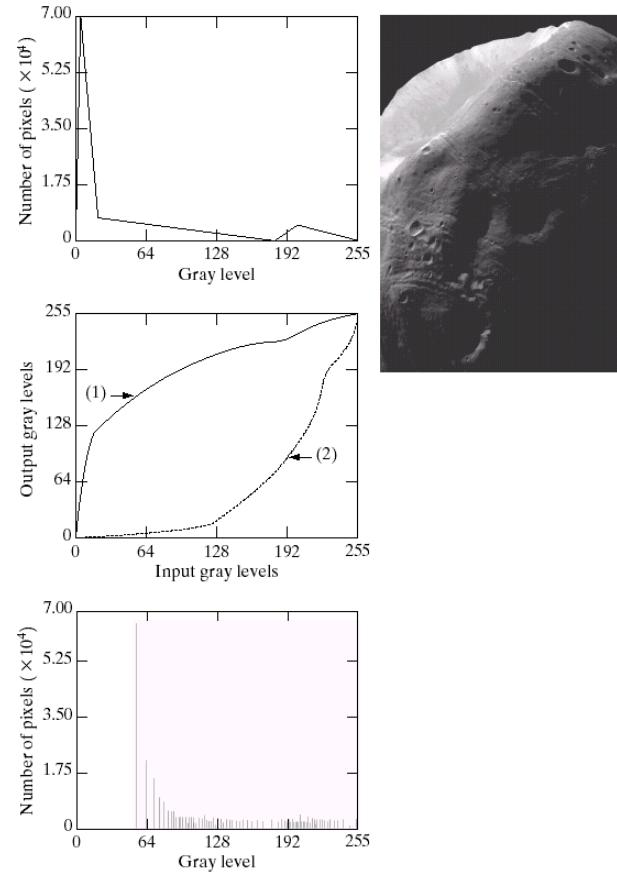


a b

**FIGURE 3.20** (a) Image of the Mars moon Photos taken by NASA's *Mars Global Surveyor*. (b) Histogram. (Original image courtesy of NASA.)

a  
b  
d

**FIGURE 3.22**  
 (a) Specified histogram.  
 (b) Curve (1) is from Eq. (3.3-14), using the histogram in (a); curve (2) was obtained using the iterative procedure in Eq. (3.3-17).  
 (c) Enhanced image using mappings from curve (2).  
 (d) Histogram of (c).



## Histogram Matching

# Spatial-domain Processing

# Topics

- Image Smoothing
- Sharpness Enhancement
- Contrast Enhancement

# Image Noise



$$I(x, y) = S(x, y) + N(x, y)$$

signal      noise

Typically, we assume

- Image noise has zero mean.
- Image noise has the same variance at different pixels.
- image noise at different pixels are uncorrelated.

Ref: [https://en.wikipedia.org/wiki/Image\\_noise](https://en.wikipedia.org/wiki/Image_noise)

# Image Noise



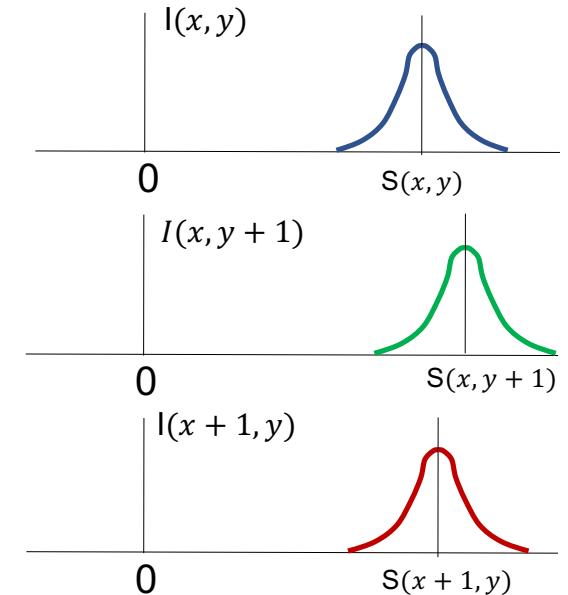
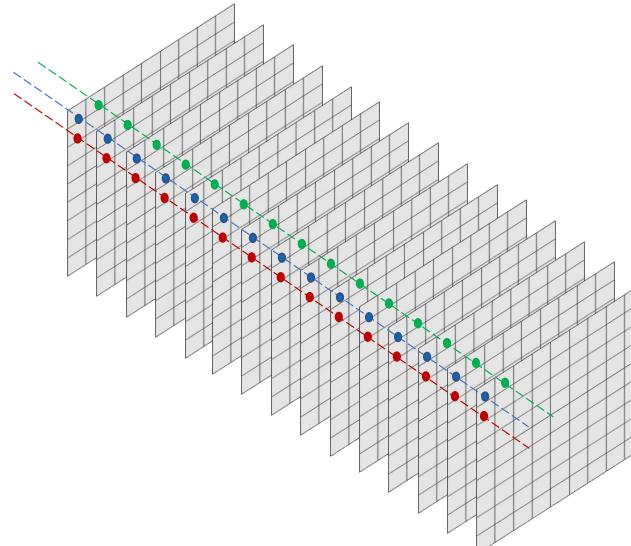
$$I(x, y) = S(x, y) + N(x, y)$$

signal      noise

$$\begin{aligned} E[I(x, y)] &= E[S(x, y) + N(x, y)] \\ &= S(x, y) + E[N(x, y)] = S(x, y) \end{aligned}$$

- Image noise has zero mean.
- Image noise has the same variance at different pixels.

$$I(x, y) - E[I(x, y)] = I(x, y) - S(x, y) = N(x, y)$$



# Image Noise

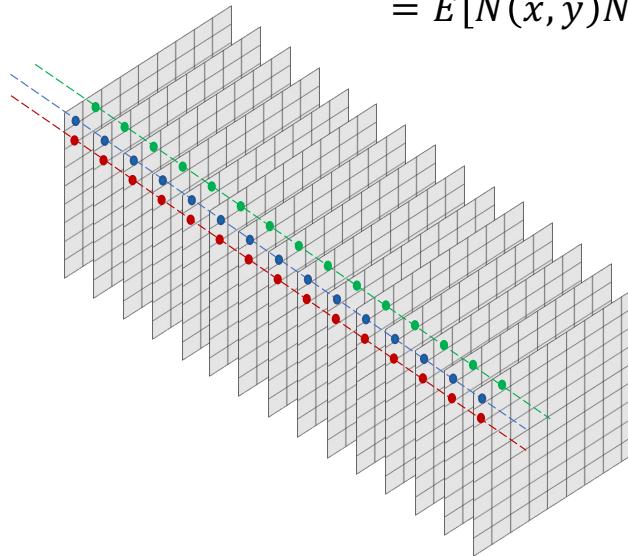


$$I(x, y) = S(x, y) + N(x, y)$$

signal      noise

- image noise at different pixels are uncorrelated.

$$\begin{aligned} & E[(I(x, y) - E[I(x, y)])(I(x', y') - E[I(x', y')])] \\ &= E[N(x, y)N(x', y')] = E[N(x, y)]E[N(x', y')] = 0 \end{aligned}$$



# Smoothing Filtering

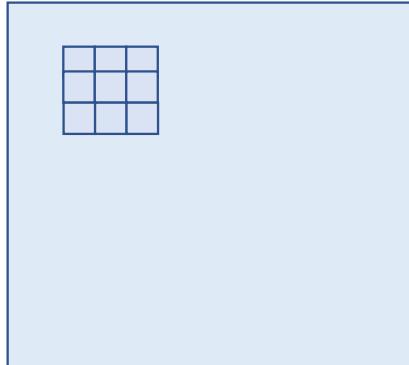
Purpose: blurring & noise reduction



**Linear Filter: Low-pass spatial filter**

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

# Local Averaging



$$\hat{I}(x, y) = \frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 I(x + i, y + j) \quad \text{where } I(x, y) = S(x, y) + N(x, y)$$

signal   noise

$$E[\hat{I}(x, y)] = \frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 S(x + i, y + j) + E\left[\frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 N(x + i, y + j)\right]$$

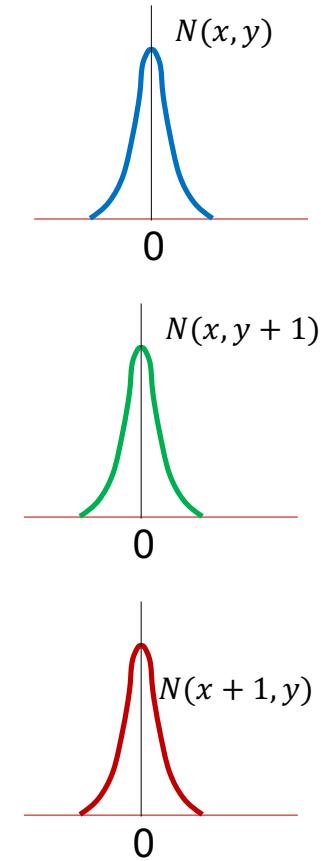
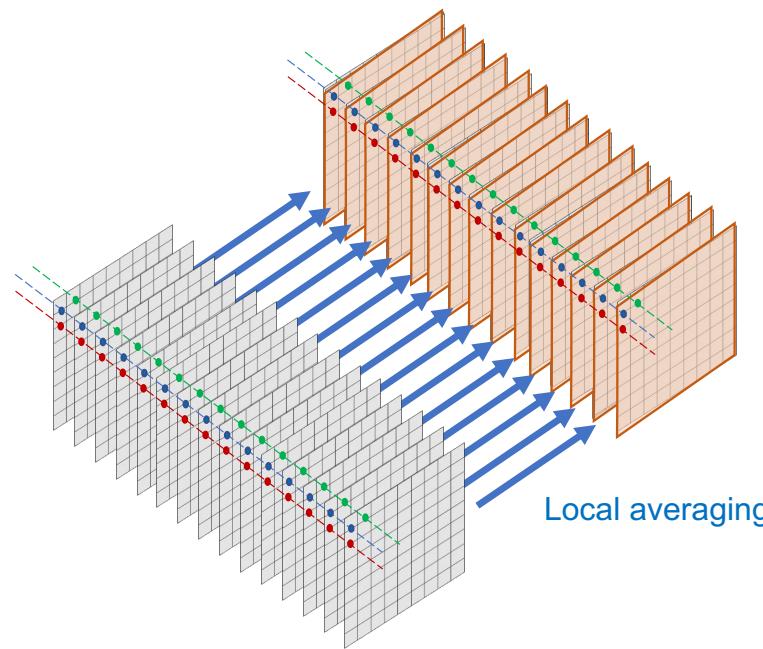
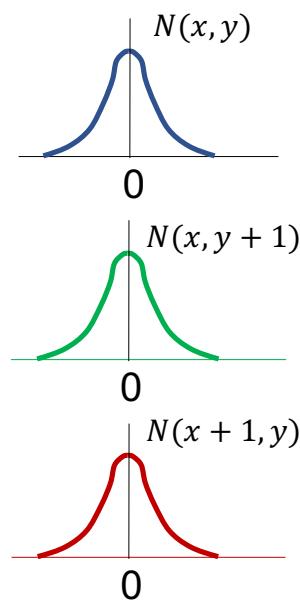
$$= \frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 S(x + i, y + j)$$

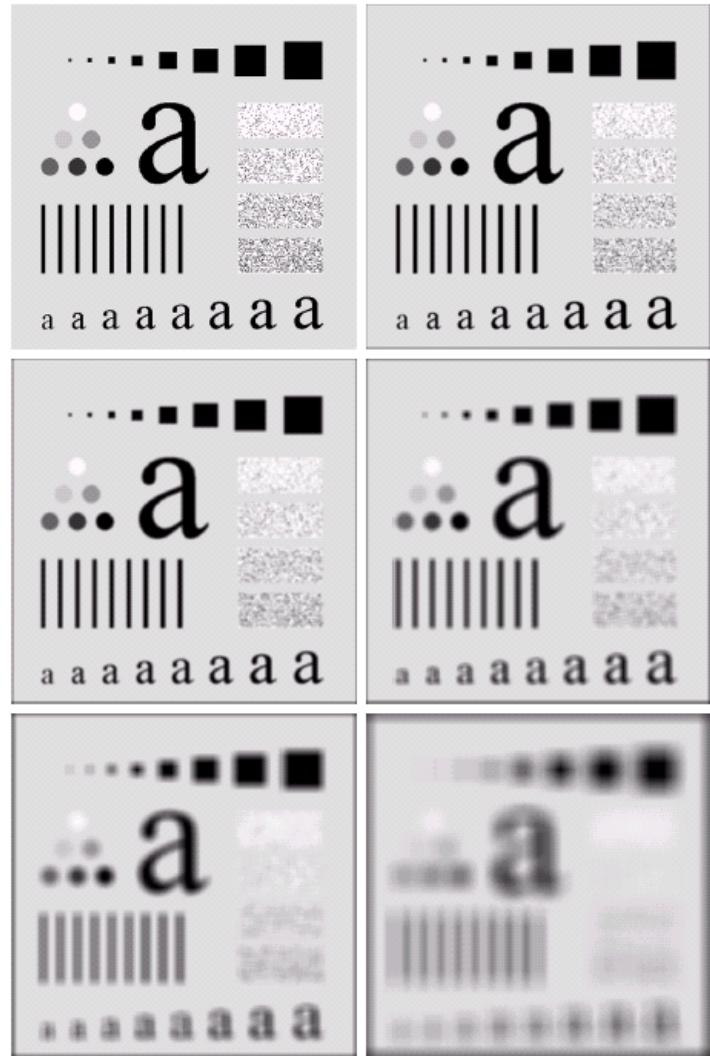
$$\begin{aligned}\text{Var}(X) &= E[(X - E[X])^2] \\ &= E[X^2 - 2X E[X] + (E[X])^2] \\ &= E[X^2] - 2 E[X] E[X] + (E[X])^2 \\ &= E[X^2] - (E[X])^2\end{aligned}$$

$$\text{Var}[\hat{I}(x, y)] = E[(\hat{I}(x, y) - E(\hat{I}(x, y))^2] = E\left[\left\{\frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 N(x + i, y + j)\right\}^2\right]$$

$$= \frac{1}{81} \left\{ \sum_{i=-1}^1 \sum_{j=-1}^1 E[(N(x + i, y + j))^2] + \text{cross terms} \right\} = \frac{1}{81} \left\{ \sum_{i=-1}^1 \sum_{j=-1}^1 \text{Var}[N(x + i, y + j)] \right\} = \frac{1}{9} \text{Var}[N(x, y)]$$

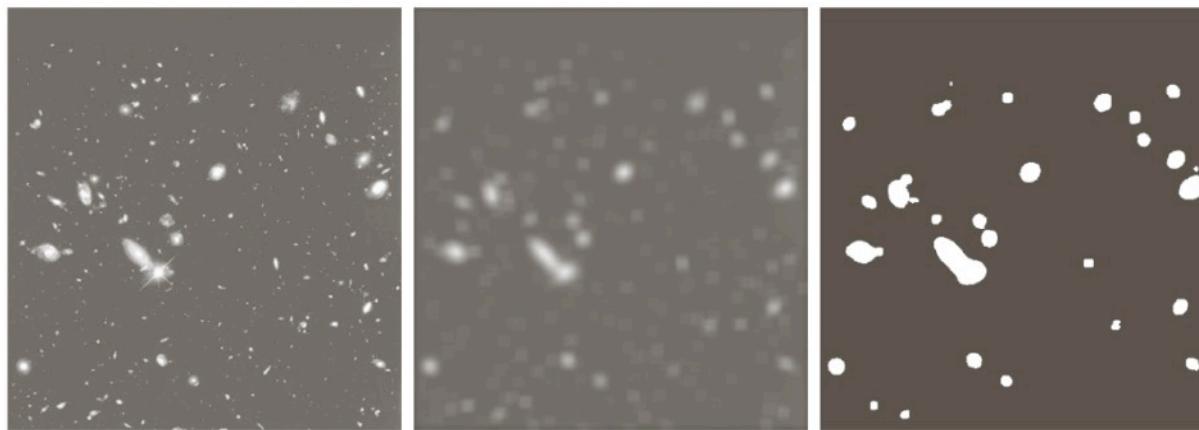
# Local Averaging





**FIGURE 3.33** (a) Original image, of size  $500 \times 500$  pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes  $m = 3, 5, 9, 15, 25, 35$ , and  $55$ , 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  $50 \times 120$  pixels.

a b  
c d  
e f



a b c

**FIGURE 3.34** (a) Image of size  $528 \times 485$  pixels from the Hubble Space Telescope. (b) Image filtered with a  $15 \times 15$  averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

# Noise Models (1/6)

- **Gaussian Noise**

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2}$$

Remark: such as electronic circuit noise and sensor noise due to poor illumination and/or high temperature.

- **Rayleigh Noise**

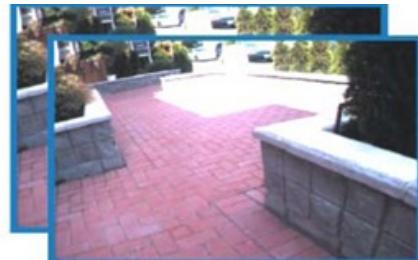
$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases}$$

$$\mu = a + \sqrt{\pi b / 4}$$

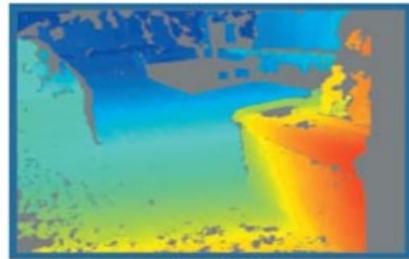
$$\sigma^2 = \frac{b(4 - \pi)}{4}$$

Remark: such as noise in range imaging.

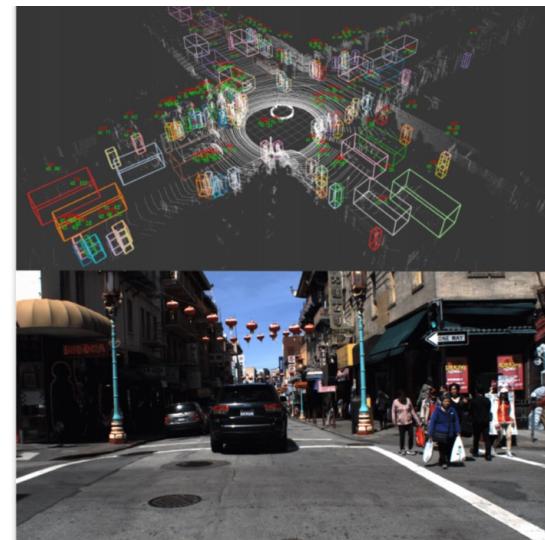
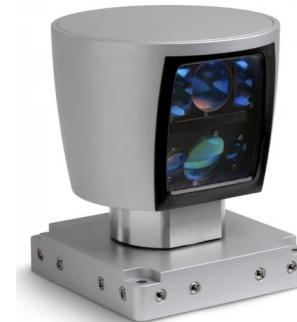
# How do computers perceive depth?



(a) Raw Images



(b) Depth Image



# Velodyne LiDAR



**Ultra Puck**



**Puck**



**Puck LITE**



**Puck HI-Res**



**Alpha Prime**

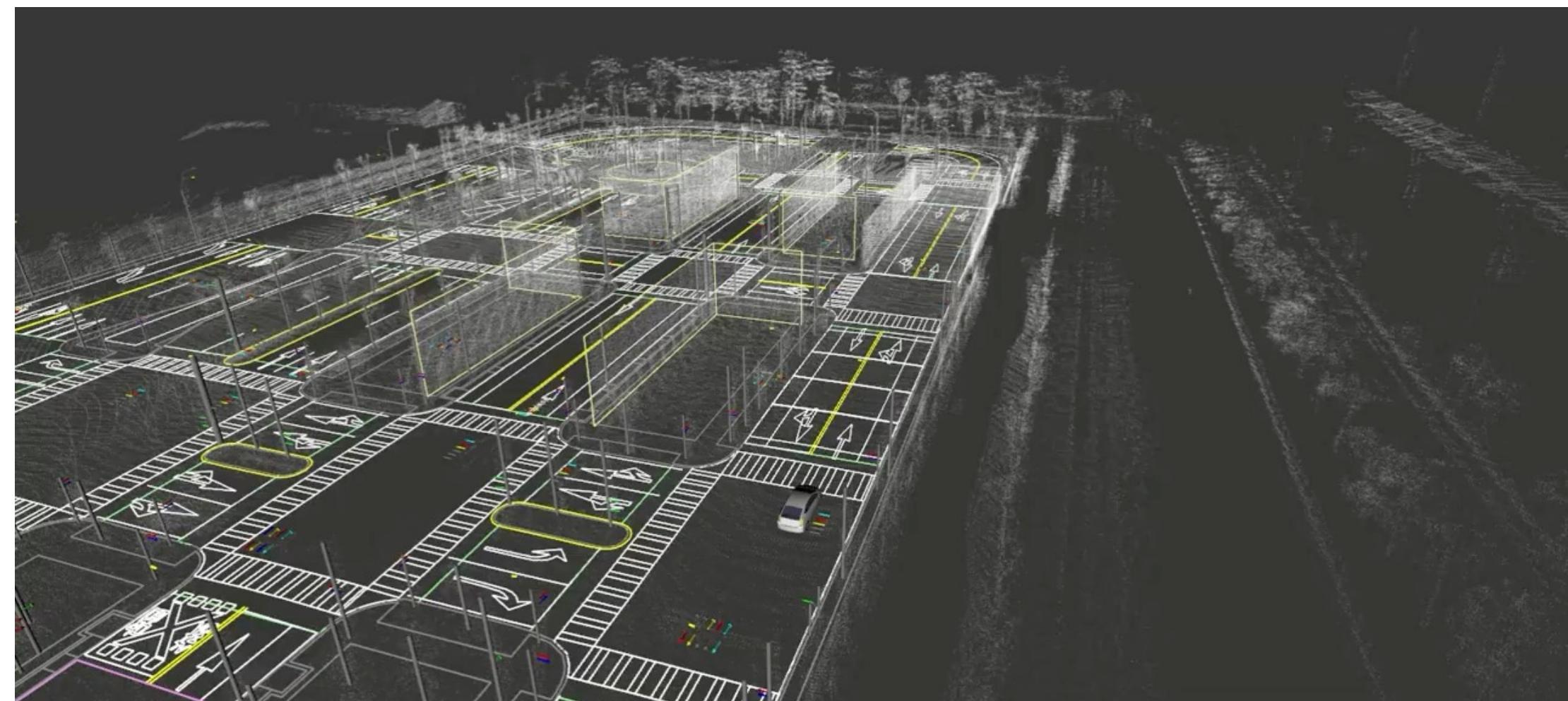


**HDL-32E**



**HDL-64E**

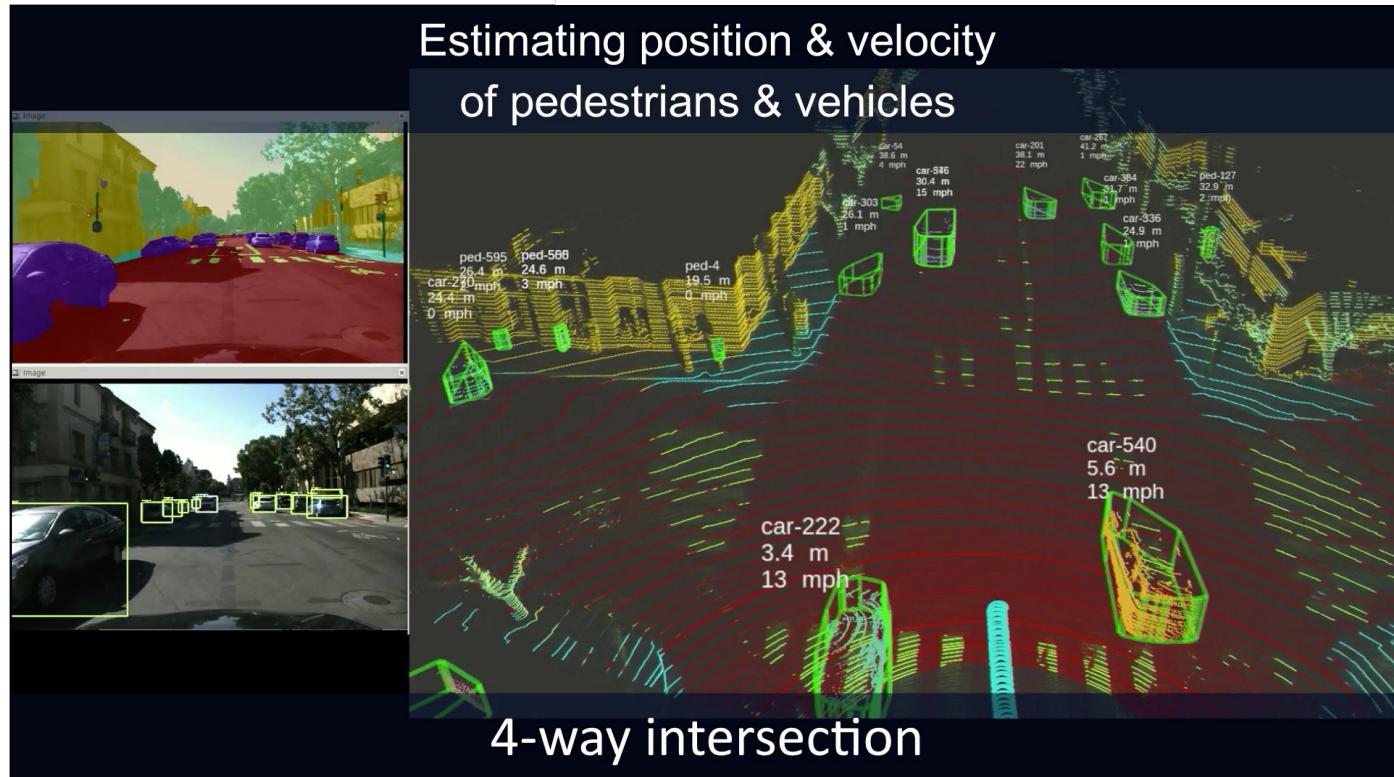




# Perception

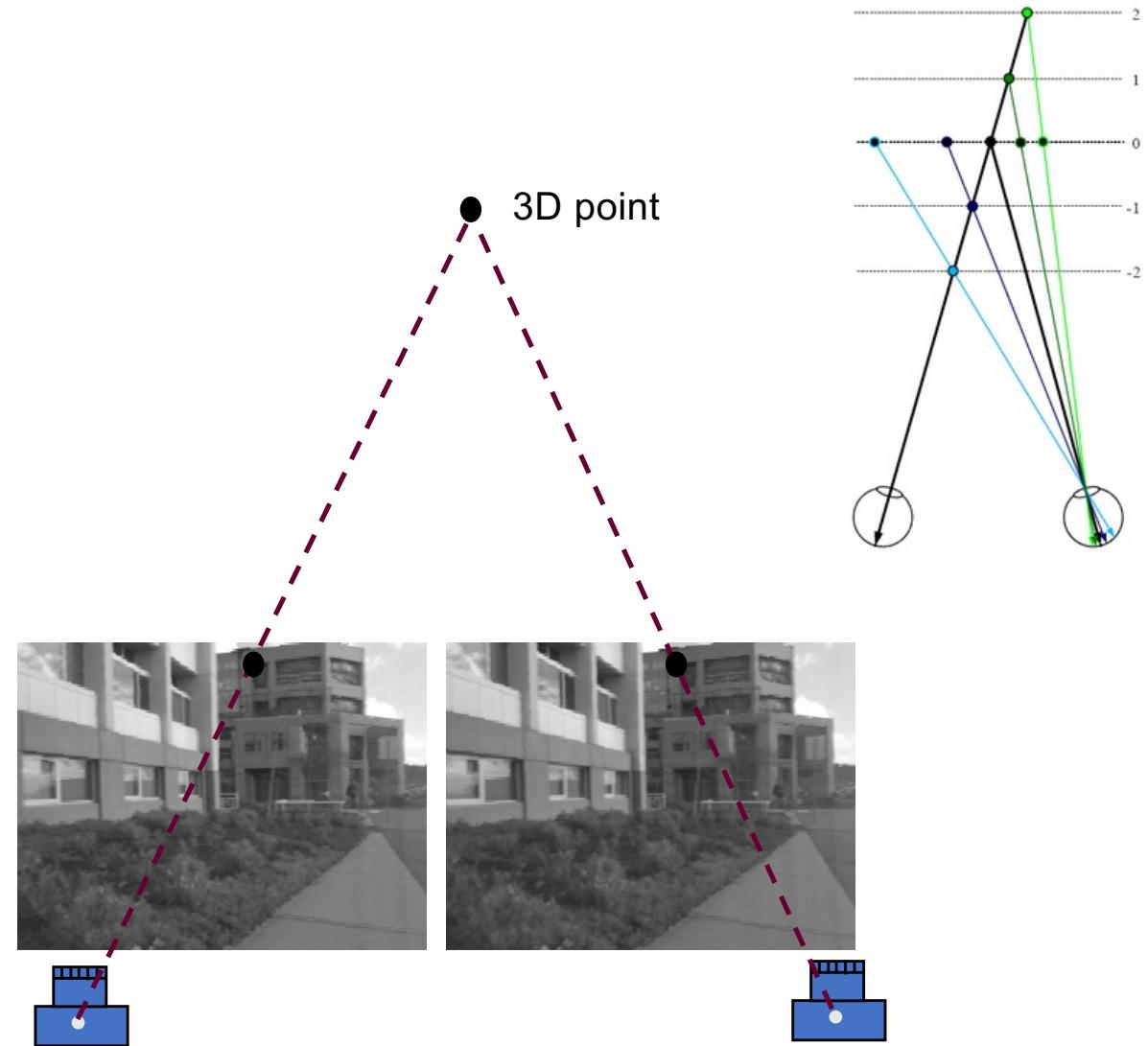


**State estimation:**  
Position (detection)  
Velocity (tracking)



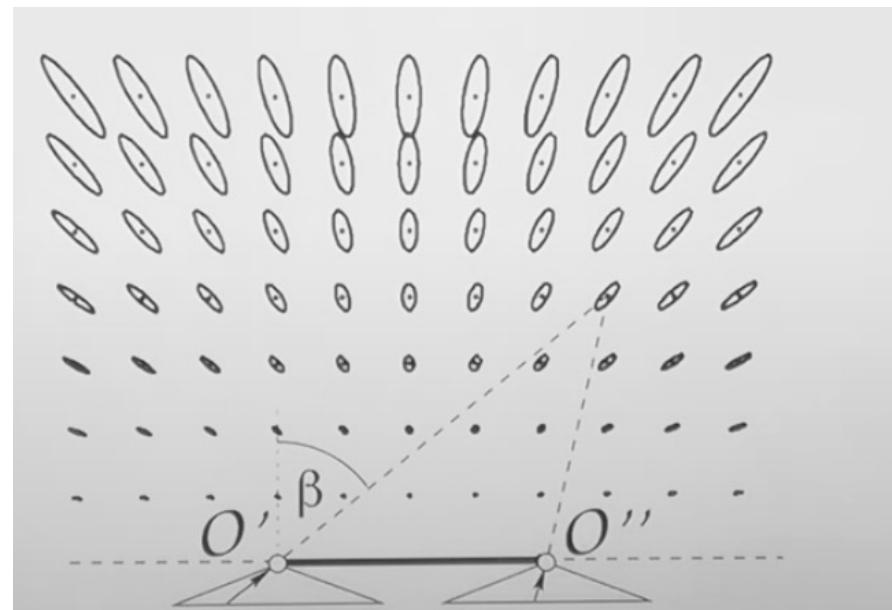
# Stereo Camera

- As camera is shifted (viewpoint changed):
  - 3D points are projected to different 2D locations
- 2D shifts
  - stereo disparity





An example video captured with Intel D435



# Noise Models (2/6)

- **Erlang (Gamma) Noise**

$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$
$$\mu = \frac{b}{a}$$
$$\sigma^2 = \frac{b}{a^2}$$

Remark: such as noise in laser imaging.

- **Exponential Noise (a special case of the Erlang pdf)**

$$p(z) = \begin{cases} ae^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$
$$\mu = \frac{1}{a}$$
$$\sigma^2 = \frac{1}{a^2}$$

Remark: such as noise in laser imaging.

# Noise Models (3/6)

- **Uniform Noise**

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$$
$$\mu = \frac{a+b}{2}$$
$$\sigma^2 = \frac{(b-a)^2}{12}$$

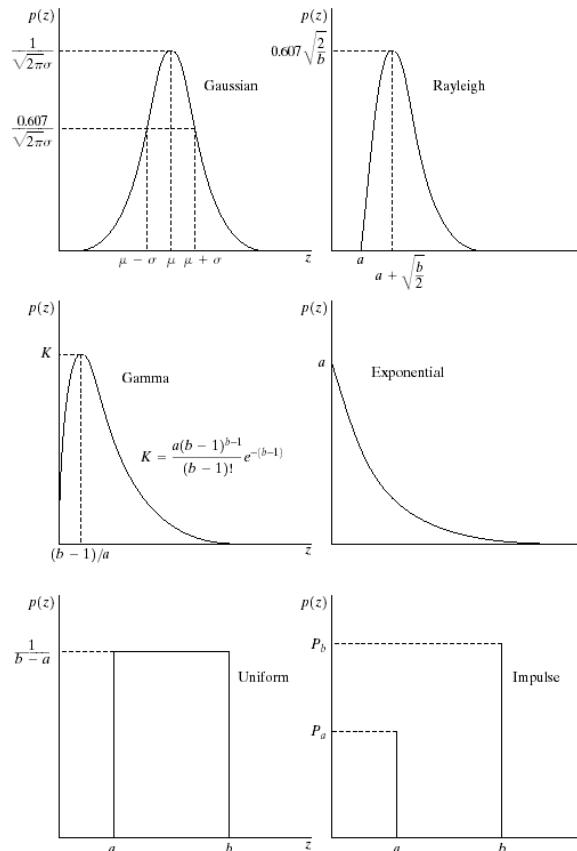
Remark: useful as the basis for numerous random number generation.

- **Impulse (Salt-and-Pepper; Shot; Spike) Noise**

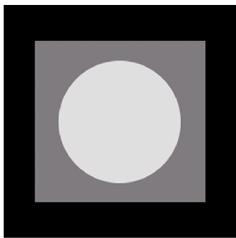
$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$

Remark: found in situations where quick transients, such as faulty switching, take place during imaging.

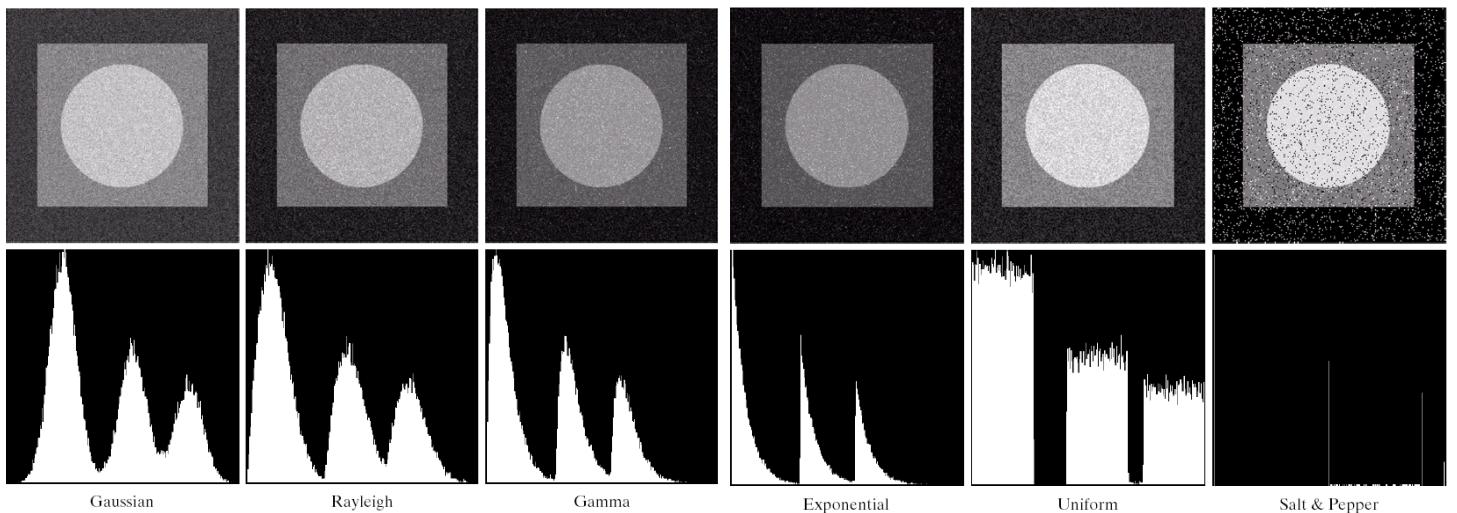
# Noise Models (4/6)



# Noise Models (5/6)



**FIGURE 5.3** Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.



**FIGURE 5.4** Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

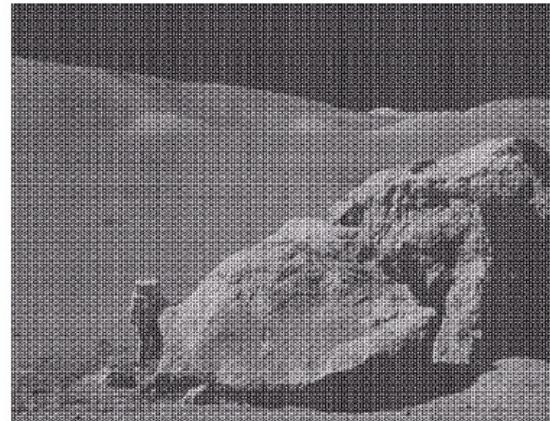
**FIGURE 5.4 (Continued)** Images and histograms resulting from adding exponential, uniform, and impulse noise to the image in Fig. 5.3.

# Noise Models (6/6)

- **Periodic Noise**

a  
b

**FIGURE 5.5**  
(a) Image corrupted by sinusoidal noise.  
(b) Spectrum (each pair of conjugate impulses corresponds to one sine wave). (Original image courtesy of NASA.)



# Smoothing Filtering

Purpose: blurring & noise reduction

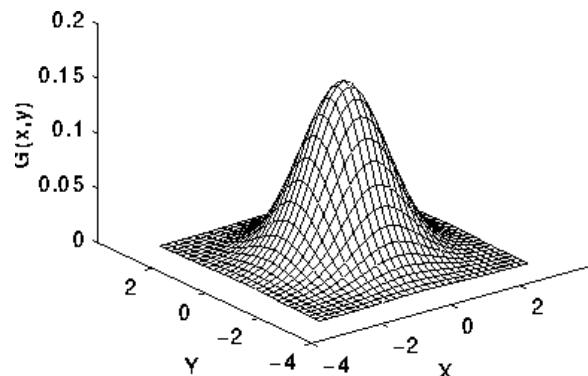


**Linear Filter: Low-pass spatial filter**

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

# Gaussian Smoothing

$$h(x, y) = \frac{1}{\sqrt{2\pi\sigma_x^2}\sqrt{2\pi\sigma_y^2}} e^{-(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2})}$$



$\frac{1}{273}$

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

$\sigma = 1$

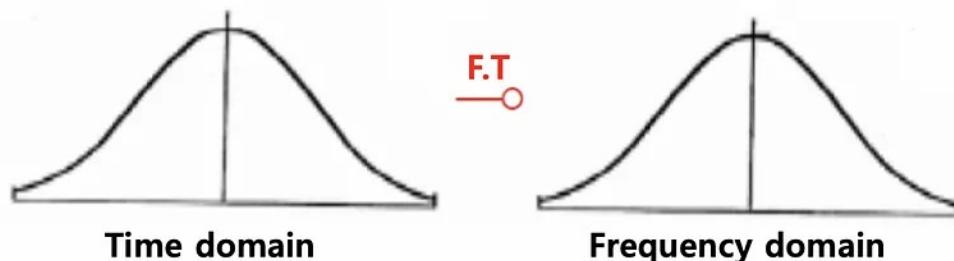
Ref: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm>

- Gaussian kernel is separable and symmetric



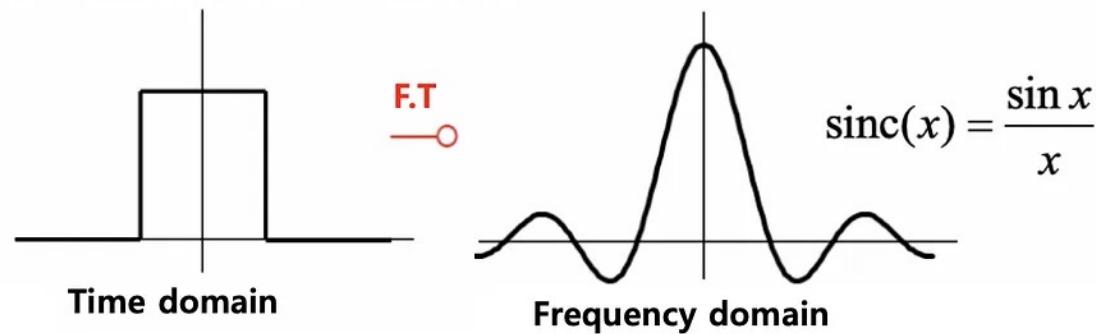
[https://en.wikipedia.org/wiki/Gaussian\\_blur](https://en.wikipedia.org/wiki/Gaussian_blur)

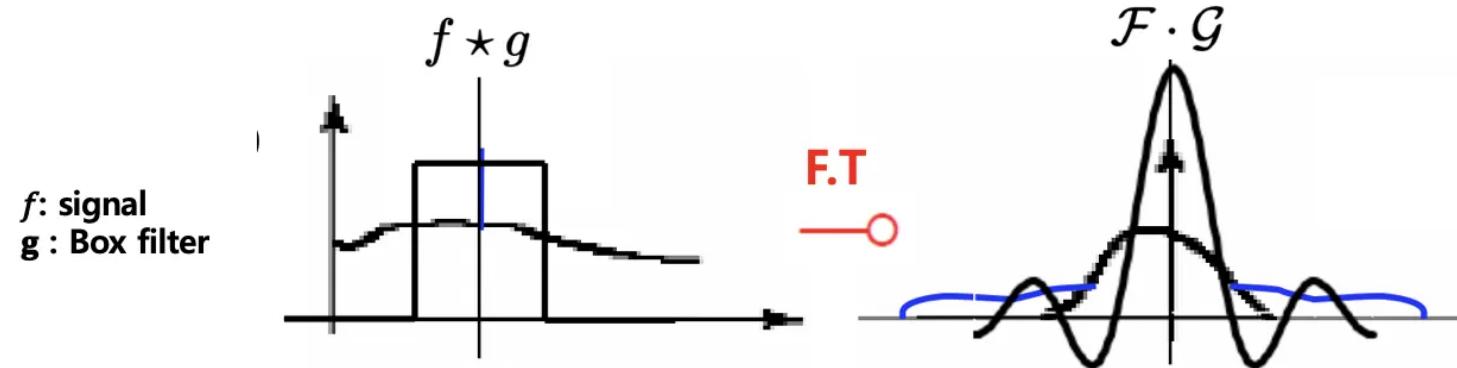
- A Gaussian transforms to a Gaussian



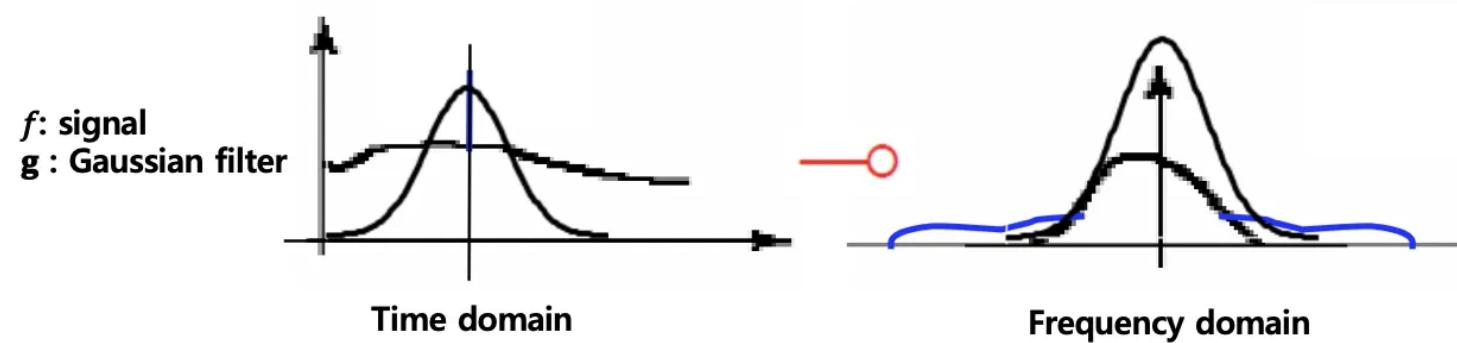
All of this is  
symmetric!

- A box filter transforms to a sinc





Blue line: noise pixel (left) and its corresponding frequency components (right) introduced by the noise pixel



# Order-Statistics Filters (1/2)

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

Remark: work well for both bipolar and unipolar impulse noise.

- **Max and Min Filters**  $\hat{f}(x, y) = \underset{(s,t) \in S_{xy}}{\max}\{g(s, t)\}$        $\hat{f}(x, y) = \underset{(s,t) \in S_{xy}}{\min}\{g(s, t)\}$

Remark: Max filter works well for pepper noise.

Min filter works well for salt noise.

- **Midpoint Filter**  $\hat{f}(x, y) = \frac{1}{2}[\underset{(s,t) \in S_{xy}}{\max}\{g(s, t)\} + \underset{(s,t) \in S_{xy}}{\min}\{g(s, t)\}]$

Remark: work well for Gaussian noise and uniform noise.

# Order-Statistics Filters (2/2)

## Adaptive, Local Noise Reduction Filter

$$\hat{f}(x, y) = g(x, y) - \frac{\sigma_\eta^2}{\sigma_{S_{xy}}^2} [g(x, y) - \bar{z}_{S_{xy}}]$$

$S_{xy}$ : a neighborhood centered at  $(x, y)$

$g(x, y)$ : the value of the noisy image at  $(x, y)$

$\sigma_\eta^2$ : the variance of the noise

$\bar{z}_{S_{xy}}$ : the local average intensity of the pixels in  $S_{xy}$

$\sigma_{S_{xy}}^2$ : the local variance of the pixels in  $S_{xy}$

# Bilateral Filter (1/4)

- Proposed by C. Tomasi and R. Manduchi, 1998.
- Based on geometric closeness and photometric similarity.

**Linear filter**

$$h(\mathbf{x}) = k_d^{-1}(\mathbf{x}) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) c(\xi, \mathbf{x}) d\xi$$

where  $k_d(\mathbf{x}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\xi, \mathbf{x}) d\xi$

$\mathbf{f}(\mathbf{x})$ : original image

$c(\xi, \mathbf{x})$ : measure the *geometric* closeness between  $\mathbf{x}$  and a nearby point  $\xi$

# Bilateral Filter (2/4)

## Bilateral filter

$$h(\mathbf{x}) = k^{-1}(\mathbf{x}) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{f}(\xi) c(\xi, \mathbf{x}) s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x})) d\xi$$

$$\text{where } k(\mathbf{x}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\xi, \mathbf{x}) s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x})) d\xi$$

$s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x}))$ : measure the *photometric* similarity between the pixel at  $\mathbf{x}$  and that of a nearby point  $\xi$ .

# Bilateral Filter (3/4)

Example

$$c(\xi, \mathbf{x}) = \exp\left\{-\frac{1}{2}\left(\frac{d(\xi, \mathbf{x})}{\sigma_d}\right)^2\right\} \quad \text{where } d(\xi, \mathbf{x}) = \|\xi - \mathbf{x}\|$$

$$s(\xi, \mathbf{x}) = \exp\left\{-\frac{1}{2}\left(\frac{\delta(f(\xi), f(\mathbf{x}))}{\sigma_r}\right)^2\right\} \quad \text{where } \delta(\phi, \mathbf{f}) = \|\phi - \mathbf{f}\|$$

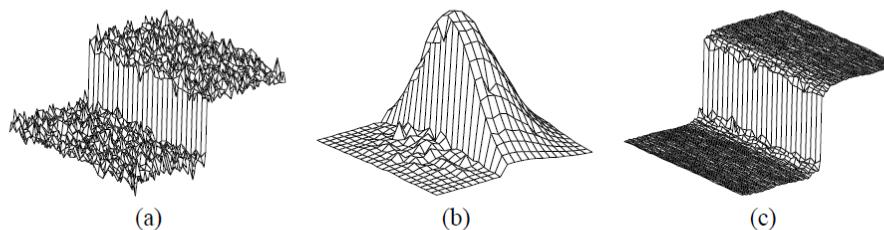
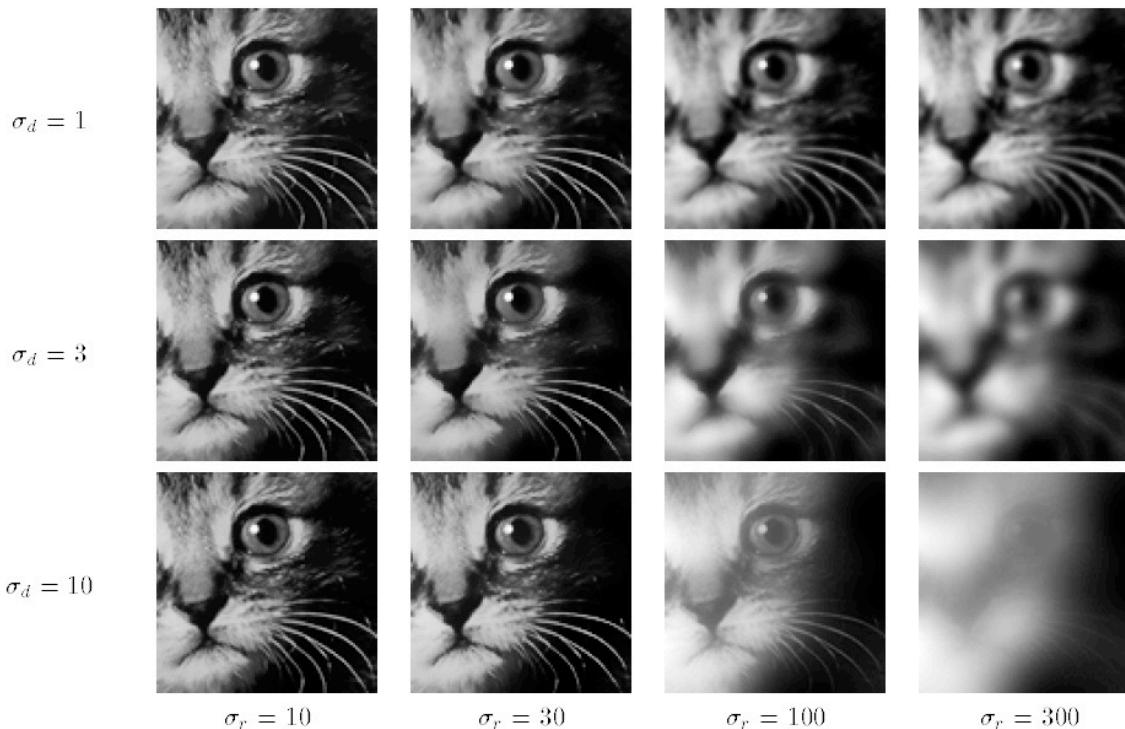


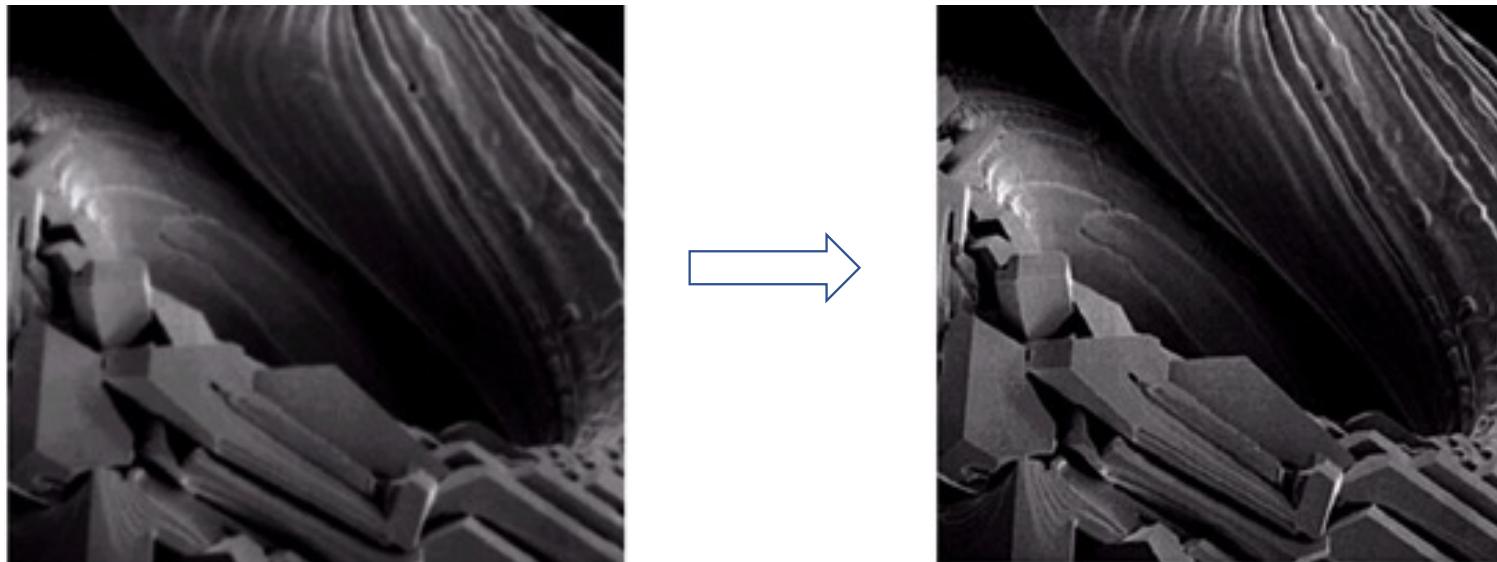
Figure 1: (a) A 100-gray-level step perturbed by Gaussian noise with  $\sigma = 10$  gray levels. (b) Combined similarity weights  $c(\xi, \mathbf{x})s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x}))$  for a  $23 \times 23$  neighborhood centered two pixels to the right of the step in (a). The range component effectively suppresses the pixels on the dark side. (c) The step in (a) after bilateral filtering with  $\sigma_r = 50$  gray levels and  $\sigma_d = 5$  pixels.

# Bilateral Filter (4/4)



# Sharpness Enhancement

- Purpose: highlight or enhance fine detail.

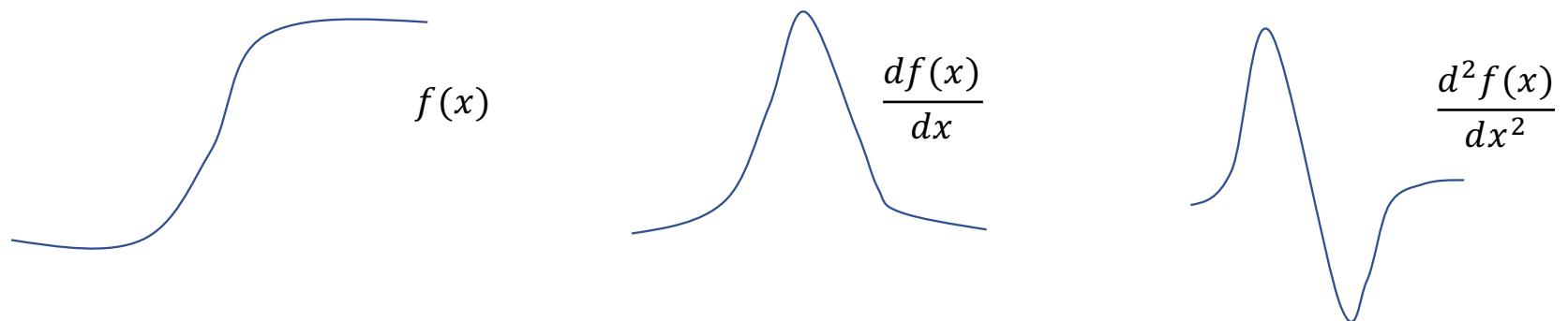


# Sharpness Enhancement

- Typically used measurement:

$$\text{1st-order derivative } \frac{\partial f}{\partial x} = f(x+1) - f(x)$$

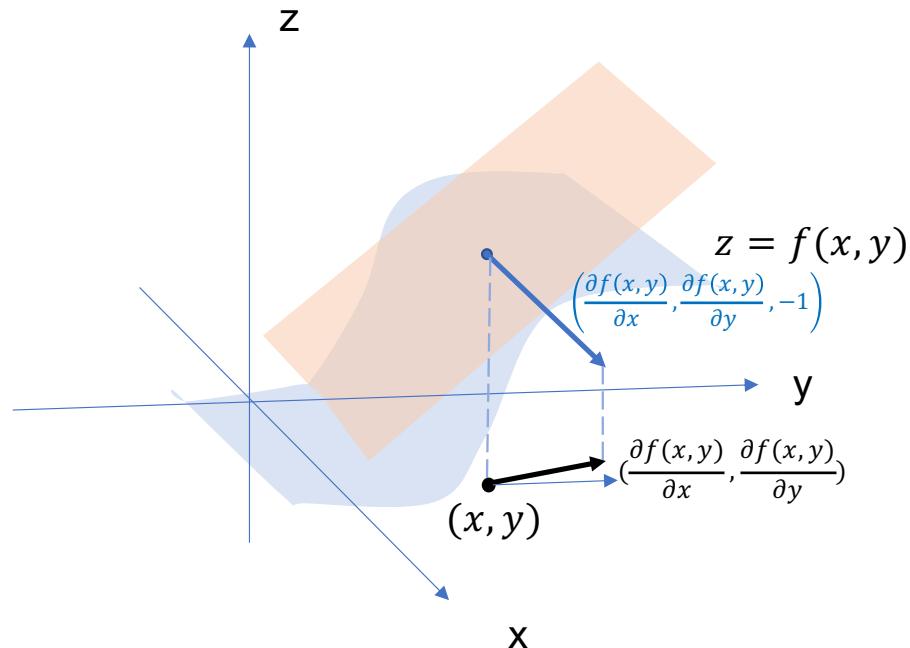
$$\text{2nd-order derivative } \frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$



# 1<sup>st</sup> Derivative

$$\nabla f(x, y) = \left( \frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right)$$

**Gradient**



## Roberts

1	0
0	-1

0	1
-1	0

## Prewitt

1	1	1
0	0	0
-1	-1	-1

-1	0	1
-1	0	1
-1	0	1

## Sobel

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

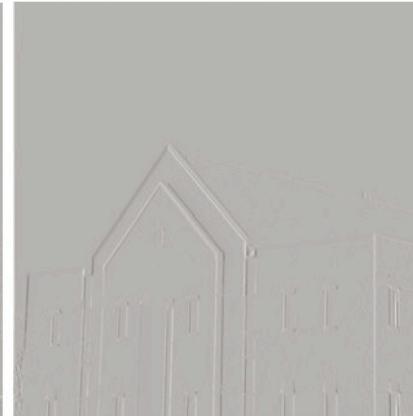
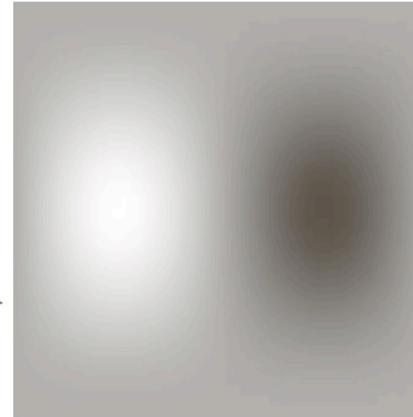
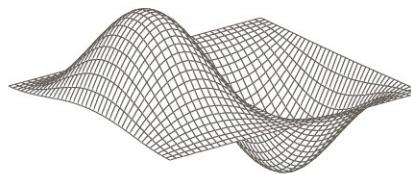
-1	0	1
-2	0	2
-1	0	1

# Sobel Operator

a b  
c d

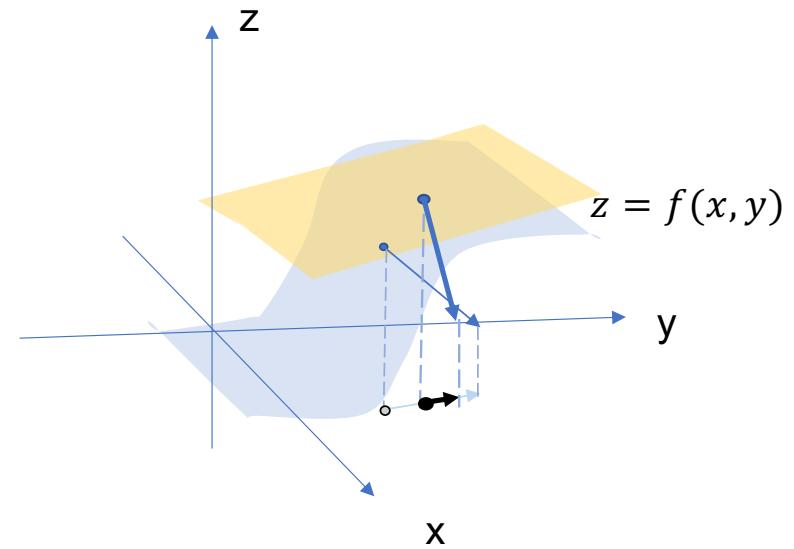
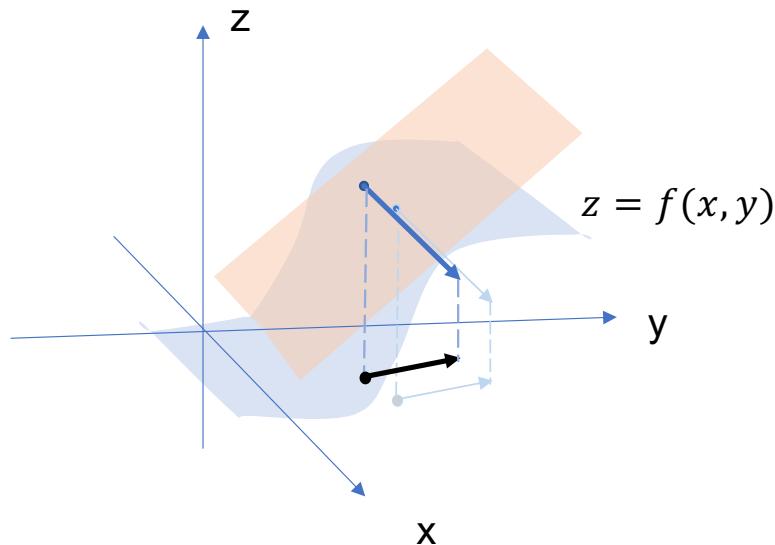
**FIGURE 4.39**  
(a) A spatial mask and perspective plot of its corresponding frequency domain filter. (b) Filter shown as an image. (c) Result of filtering Fig. 4.38(a) in the frequency domain with the filter in (b). (d) Result of filtering the same image with the spatial filter in (a). The results are identical.

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



## 2<sup>nd</sup> Derivatives

$\frac{\partial^2 f(x,y)}{\partial \vec{n}^2}$  varies for different  $\vec{n}$



# 2<sup>nd</sup>-Derivatives

- **Laplacian Filter**

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

0	1	0	1	1	1
1	-4	1	1	-8	1
0	1	0	1	1	1
0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

a b  
c d

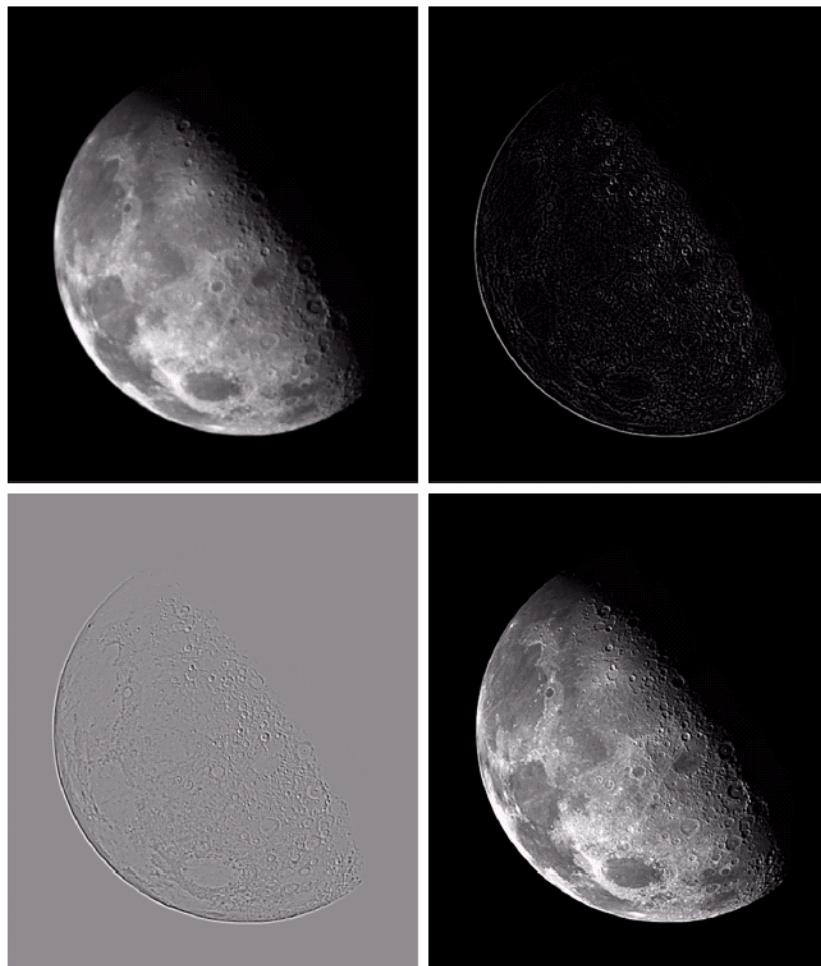
**FIGURE 3.39**

- (a) Filter mask used to implement the digital Laplacian, as defined in Eq. (3.7-4).  
(b) Mask used to implement an extension of this equation that includes the diagonal neighbors. (c) and (d) Two other implementations of the Laplacian.

a b  
c d

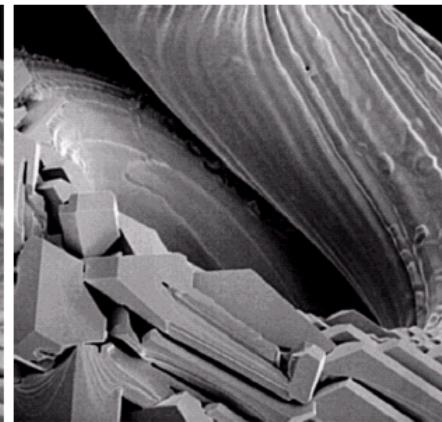
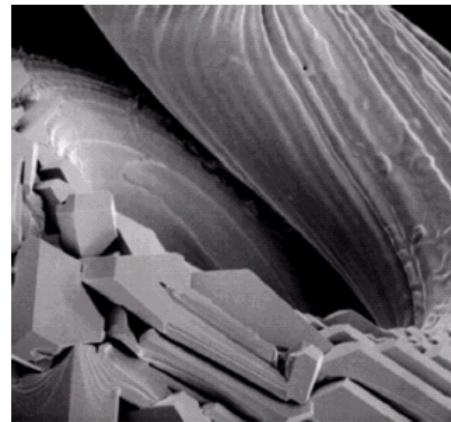
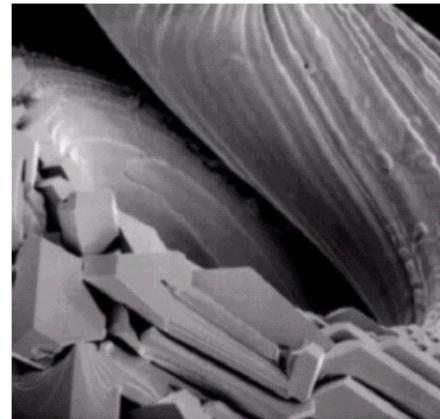
**FIGURE 3.40**

- (a) Image of the North Pole of the moon.  
(b) Laplacian-filtered image.  
(c) Laplacian image scaled for display purposes.  
(d) Image enhanced by using Eq. (3.7-5).  
(Original image courtesy of NASA.)



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.)

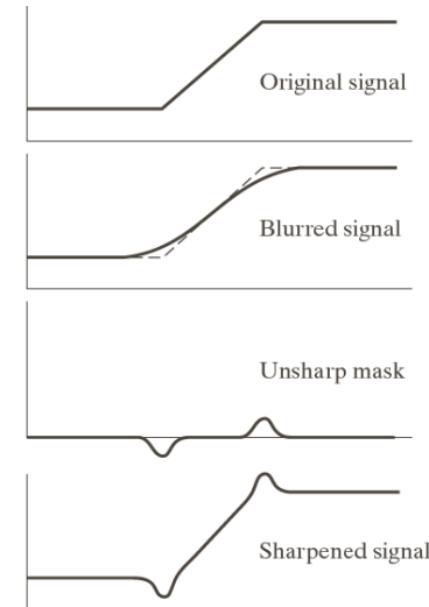
# Unsharp Masking & High-Boosting Filtering

$$g_{mask}(x, y) = f(x, y) - \bar{f}(x, y)$$

$\bar{f}(x, y)$ : a blurred version of  $f(x, y)$

$$g(x, y) = f(x, y) + k \cdot g_{mask}(x, y)$$

$k = 1$  : Unsharp Masking  
 $k > 1$  : High-Boost Filtering



# Gaussian Smoothing + Differentiation

$$\frac{\partial}{\partial x} (f(x, y) * G(x, y)) = f(x, y) * \left( \frac{\partial G(x, y)}{\partial x} \right)$$

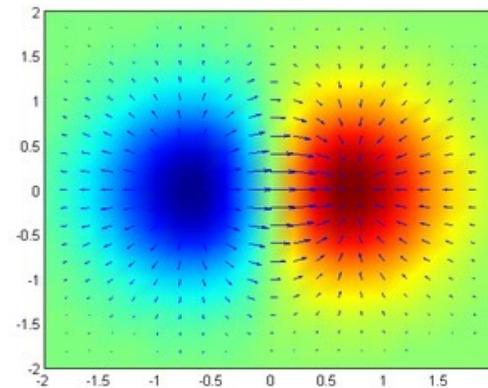
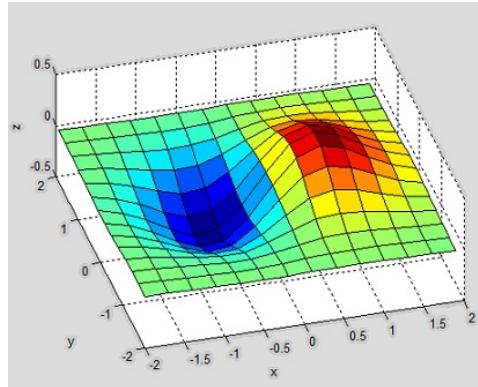
$$\frac{\partial}{\partial y} (f(x, y) * G(x, y)) = f(x, y) * \left( \frac{\partial G(x, y)}{\partial y} \right)$$

$$\nabla^2 (f(x, y) * G(x, y)) = f(x, y) * \nabla^2 G(x, y)$$

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

# First Derivatives (Gradient)

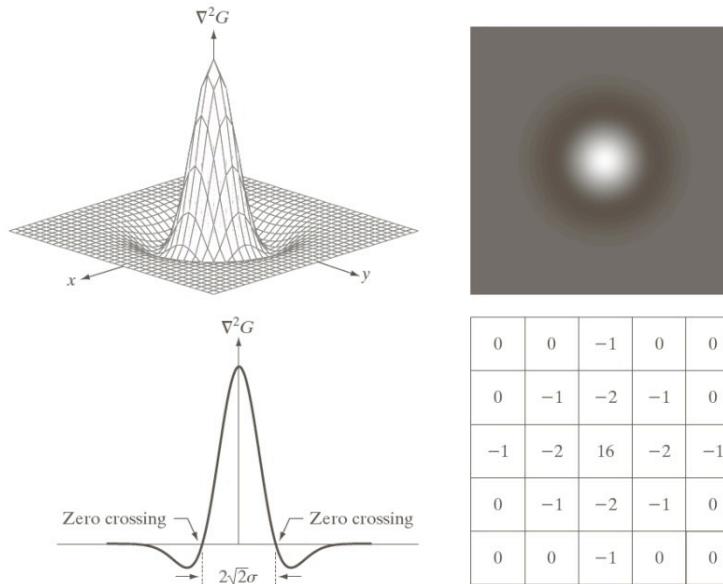
Gradient of  $f$  at  $(x,y)$ :  $\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]^T$



[https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/12954/versions/7/previews/googleearth/html/ge\\_quiver.html](https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/12954/versions/7/previews/googleearth/html/ge_quiver.html)  
<https://en.wikipedia.org/wiki/Gradient>

# LOG (Laplacian of Gaussian) Operator

$$\nabla^2 G(x, y) = \frac{-1}{\pi\sigma^4} \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$



- **DOG ( Difference of Gaussian ) Operator**

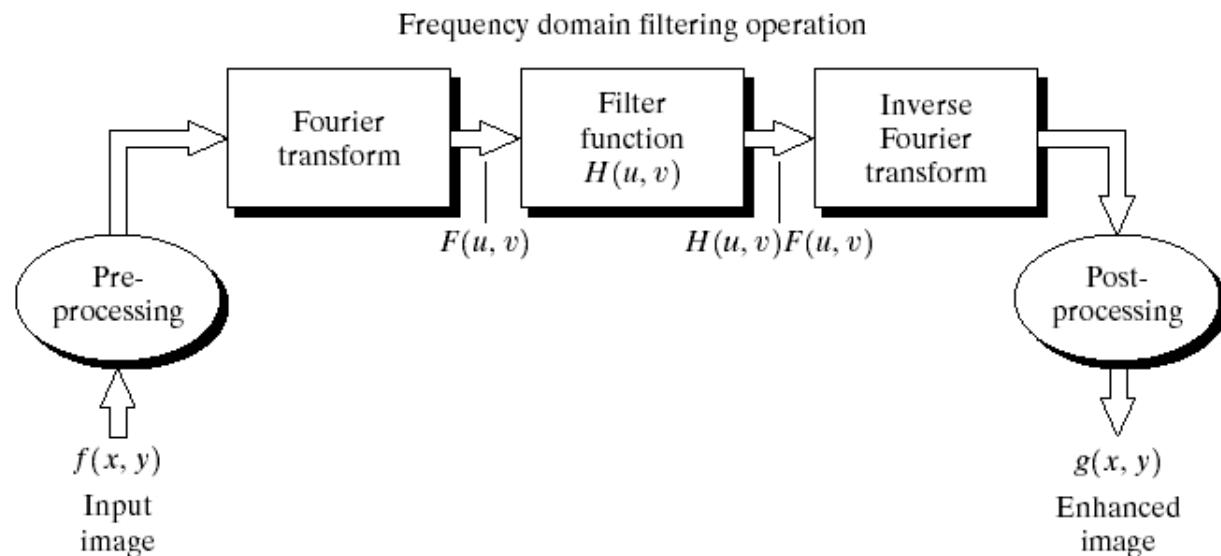
Approximate  $\nabla^2 G(x, y) = \frac{-1}{\pi\sigma^4} \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) e^{-\frac{x^2 + y^2}{2\sigma^2}}$

with  $h(x, y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2 + y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2 + y^2}{2\sigma_2^2}}$

where  $\frac{\sigma_2}{\sigma_1} \approx 1.6$       and       $\sigma^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 - \sigma_2^2} \ln\left[\frac{\sigma_1^2}{\sigma_2^2}\right]$

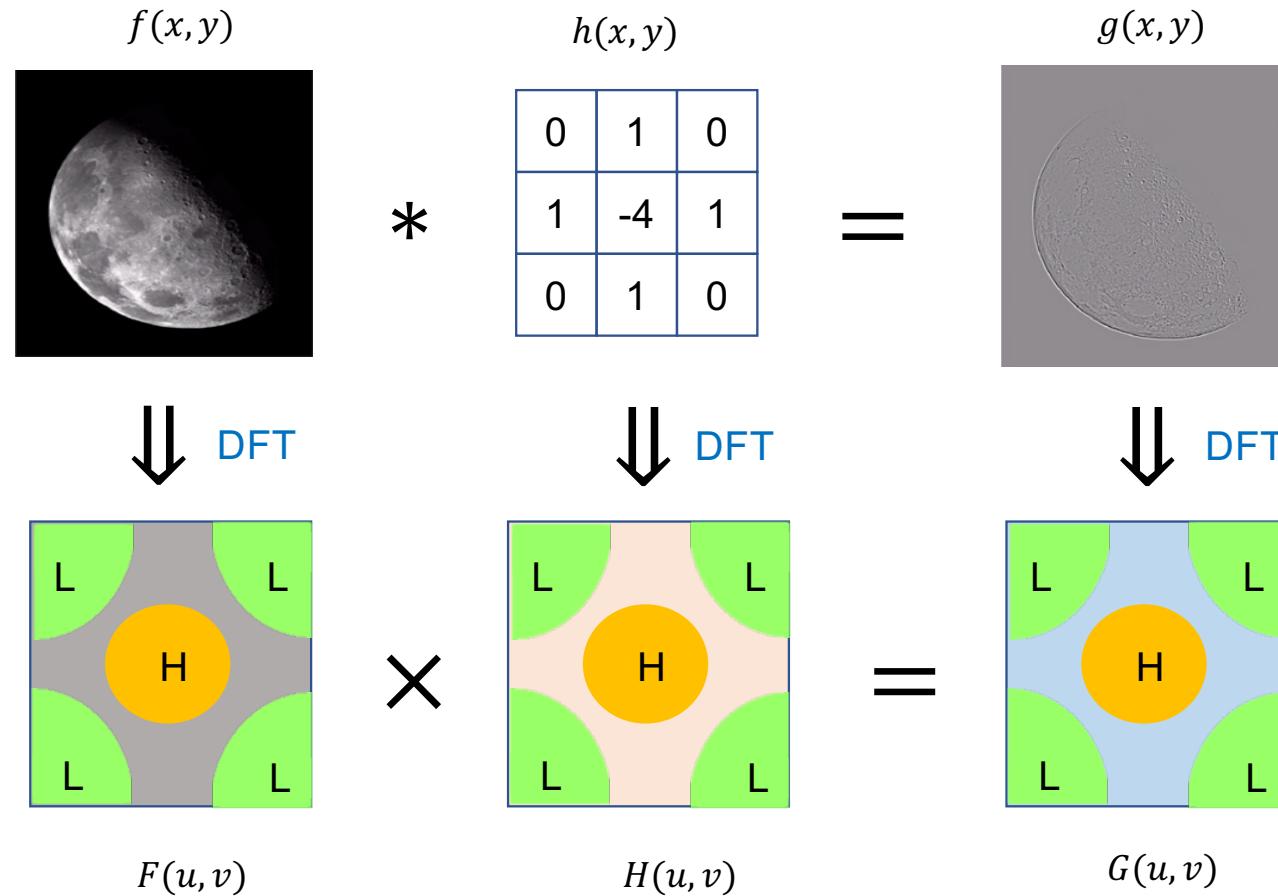
# Frequency-domain Processing

# Frequency-domain Filtering

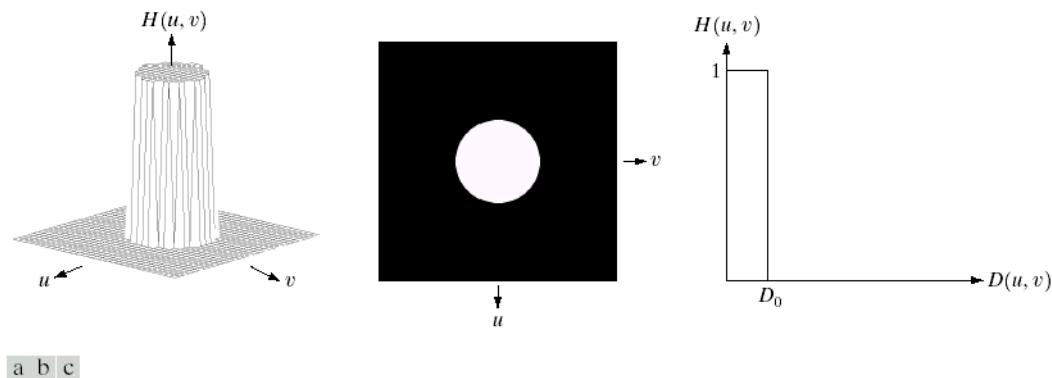


**FIGURE 4.5** Basic steps for filtering in the frequency domain.

# Linear Filtering in Frequency Domain

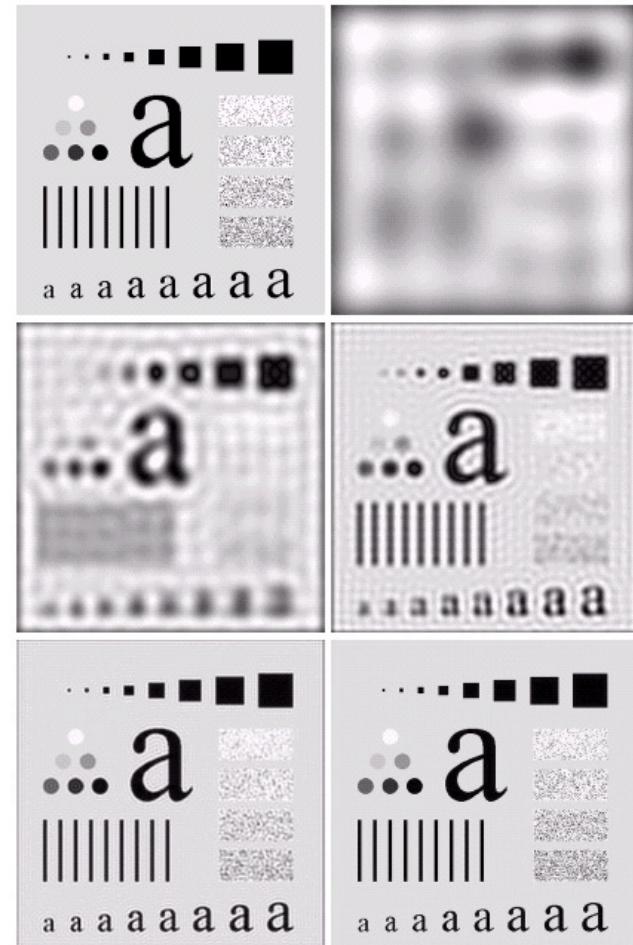


# Ideal Low-pass Filters

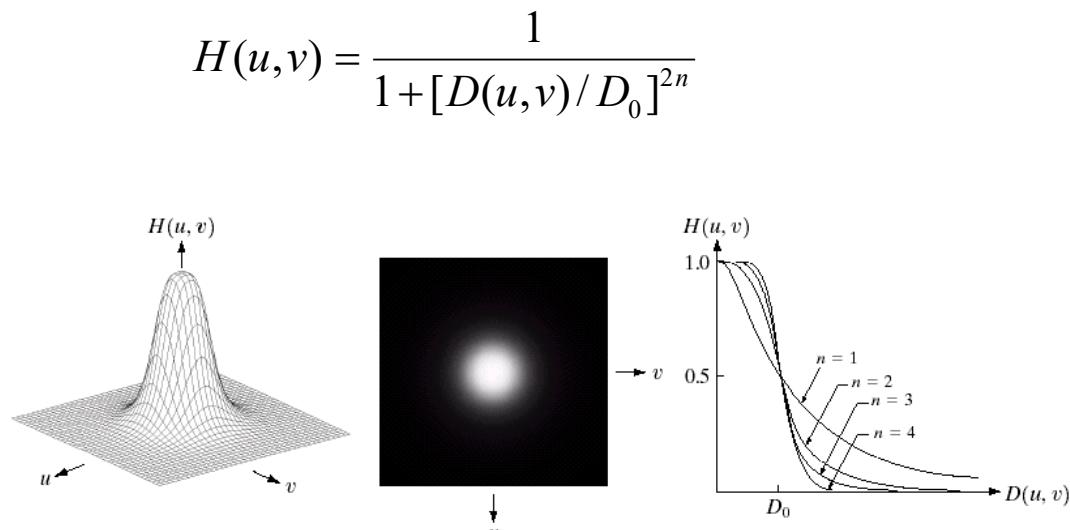


a b c

**FIGURE 4.10** (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

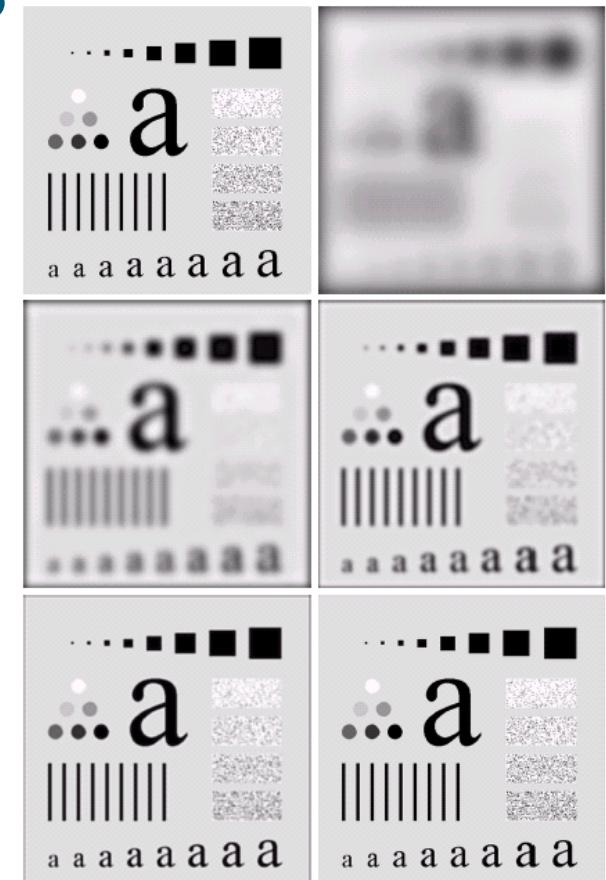


# Butterworth Low-pass Filters



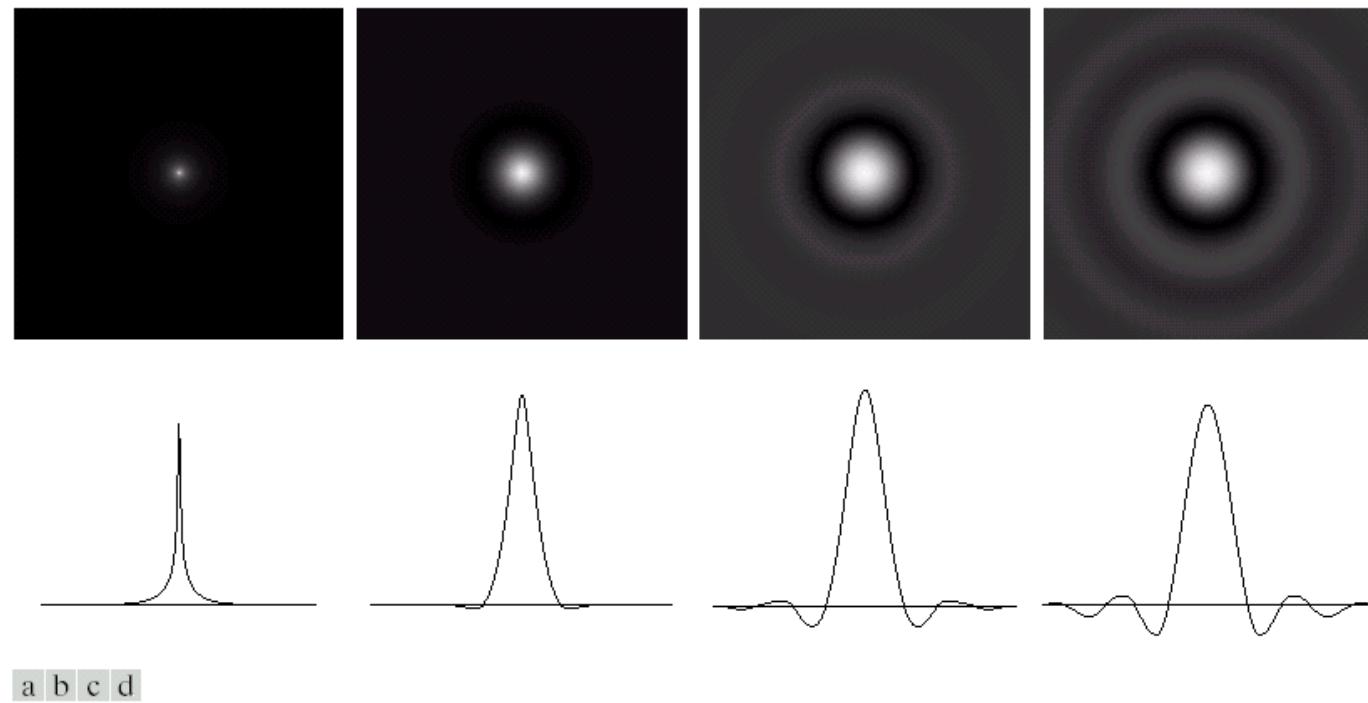
a b c

**FIGURE 4.14** (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.



a b  
c d  
e f

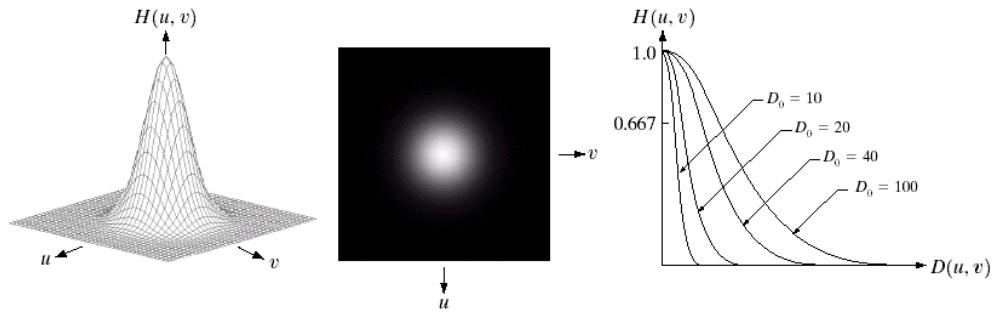
**FIGURE 4.15** (a) Original image. (b)-(f) Results of filtering with BLPFs of order 2, with cutoff frequencies at radii of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Fig. 4.12.



**FIGURE 4.16** (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

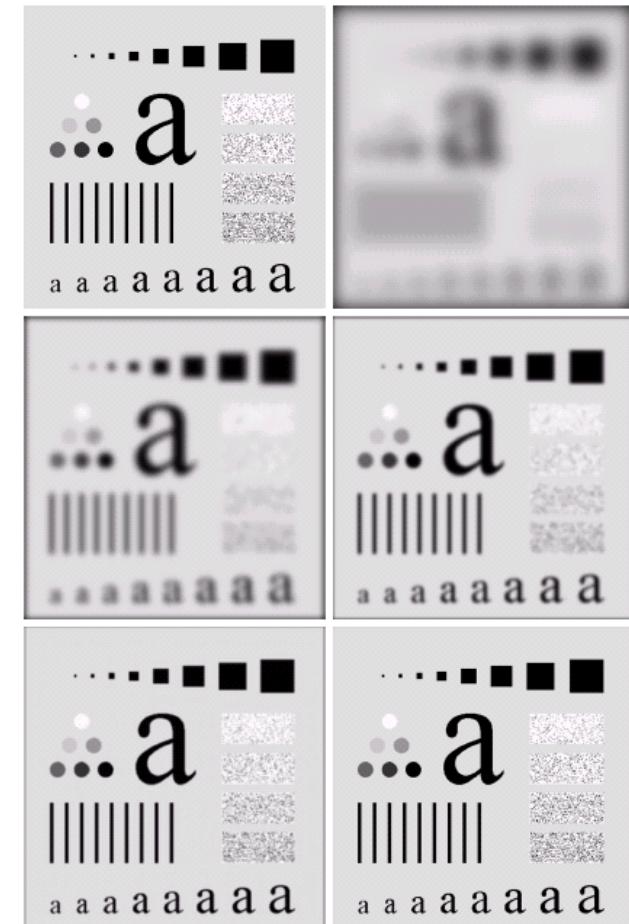
# Gaussian Low-pass Filters

$$H(u, v) = e^{-D^2(u, v) / 2\sigma^2}$$



a b c

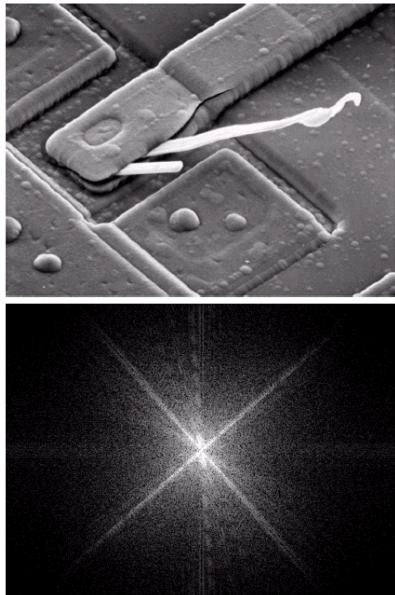
**FIGURE 4.17** (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of  $D_0$ .



**FIGURE 4.18** (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15.

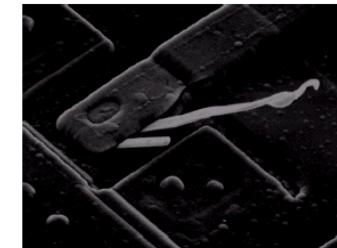
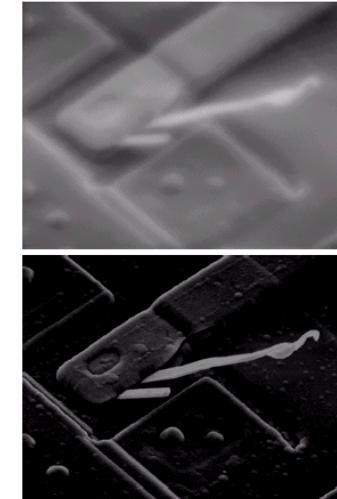
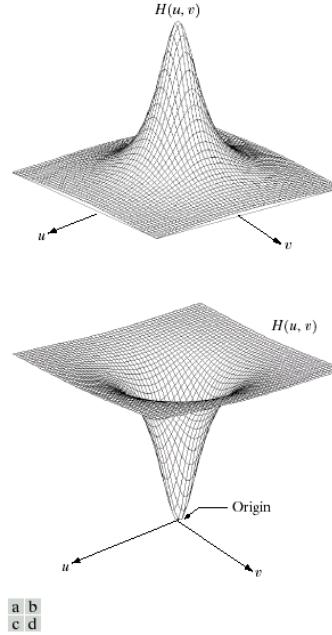
a b  
c d  
e f

# Sharpness Enhancement



a  
b

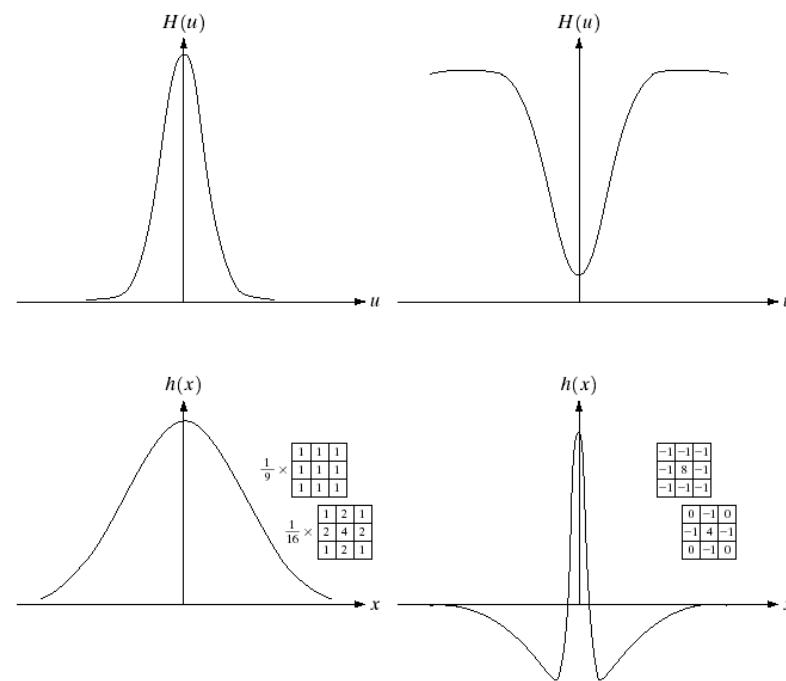
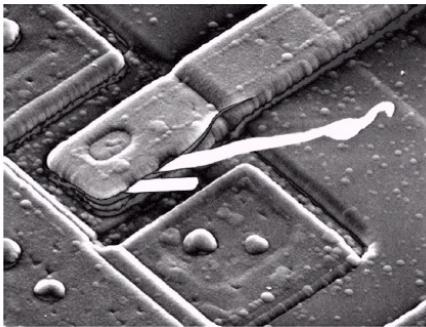
**FIGURE 4.4**  
(a) SEM image of a damaged integrated circuit.  
(b) Fourier spectrum of (a).  
(Original image courtesy of Dr. J. M. Hudak, Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada.)



a  
b  
c  
d

**FIGURE 4.7** (a) A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image in Fig. 4.4(a).  
(c) A two-dimensional highpass filter function. (d) Result of highpass filtering the image in Fig. 4.4(a).

**FIGURE 4.8**  
Result of highpass filtering the image in Fig. 4.4(a) with the filter in Fig. 4.7(c), modified by adding a constant of one-half the filter height to the filter function. Compare with Fig. 4.4(a).

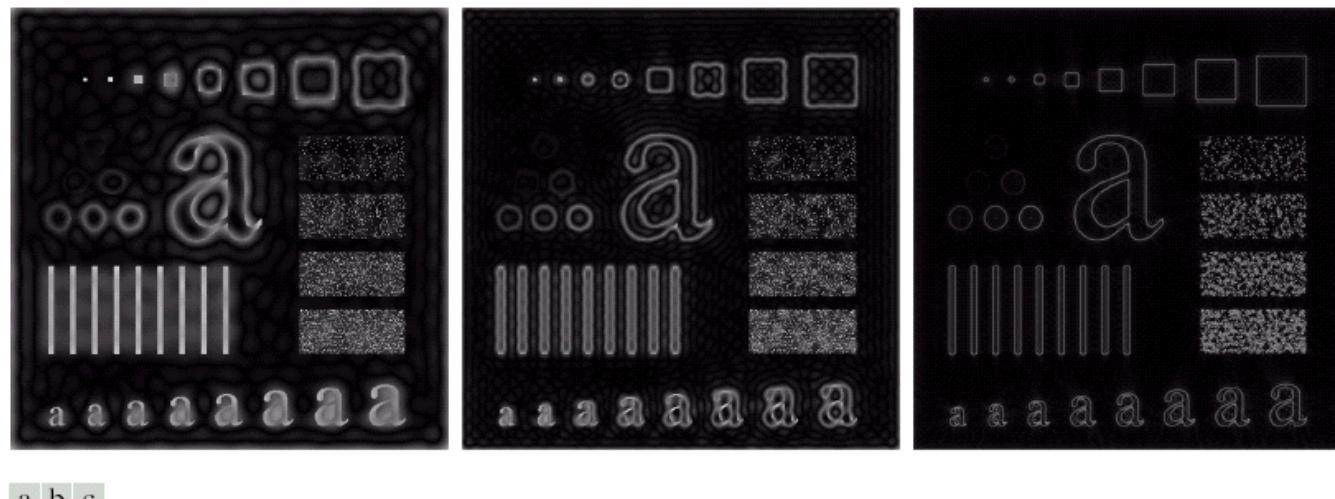


a	b
c	d

**FIGURE 4.9**  
(a) Gaussian frequency domain lowpass filter.  
(b) Gaussian frequency domain highpass filter.  
(c) Corresponding lowpass spatial filter.  
(d) Corresponding highpass spatial filter. The masks shown are used in Chapter 3 for lowpass and highpass filtering.

# Ideal High-pass Filter

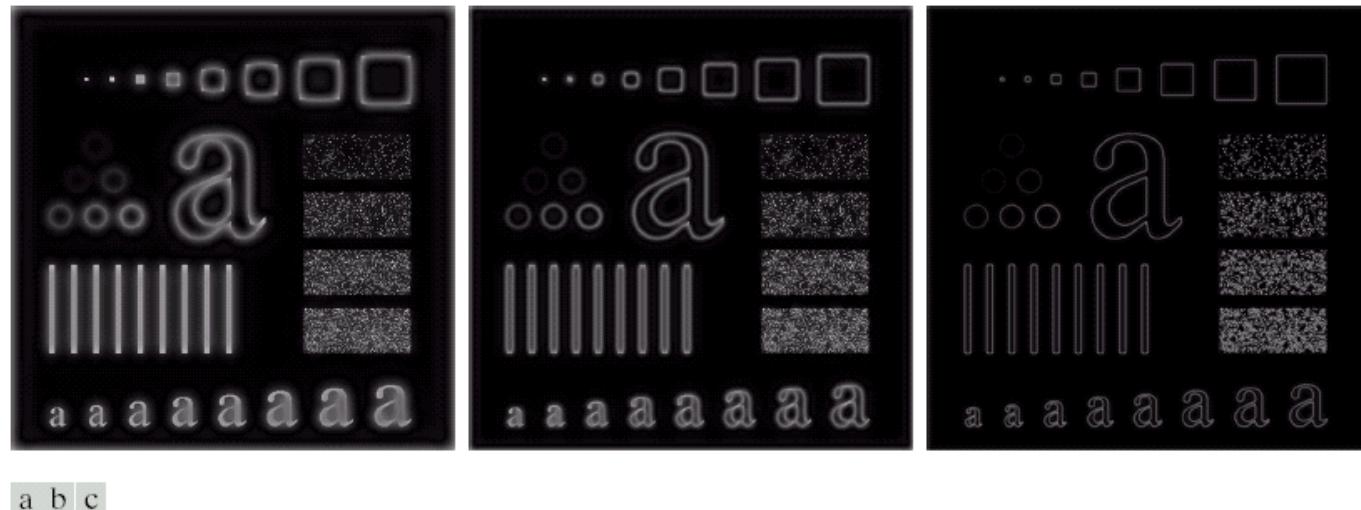
$$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) > D_0 \end{cases}$$



**FIGURE 4.24** Results of ideal highpass filtering the image in Fig. 4.11(a) with  $D_0 = 15, 30$ , and  $80$ , respectively. Problems with ringing are quite evident in (a) and (b).

# Butterworth High-pass Filter

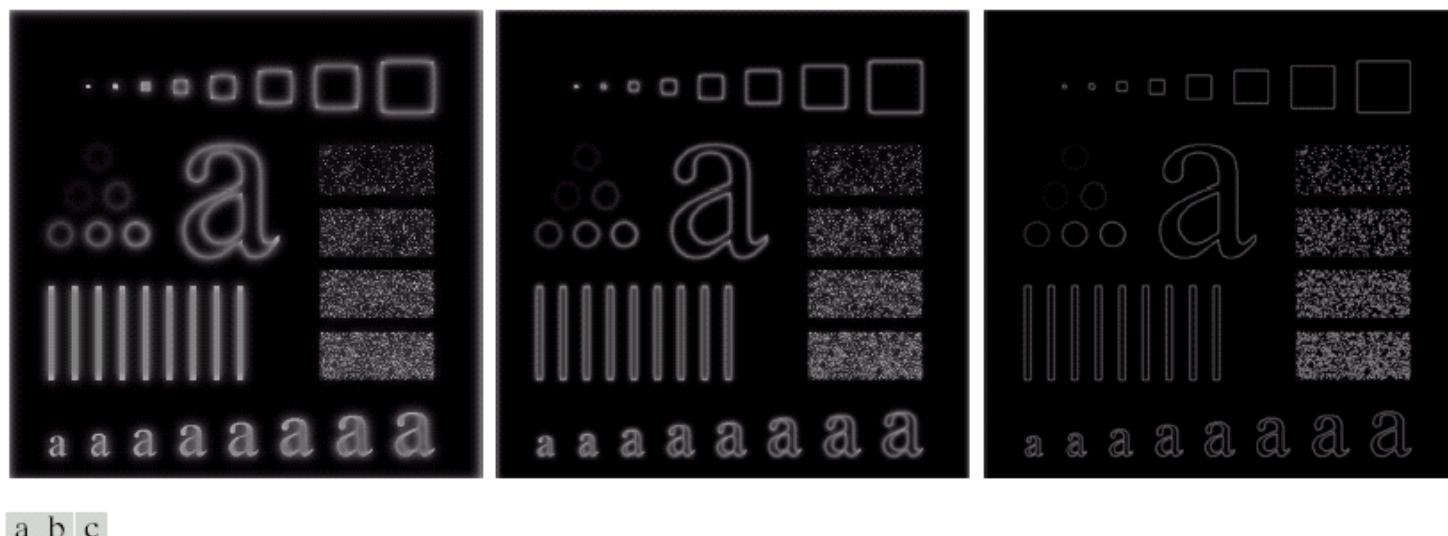
$$H(u, v) = \frac{1}{1 + [D_0/D(u, v)]^{2n}}$$



**FIGURE 4.25** Results of highpass filtering the image in Fig. 4.11(a) using a BHPF of order 2 with  $D_0 = 15$ , 30, and 80, respectively. These results are much smoother than those obtained with an ILPF.

# Gaussian High-pass Filter

$$H(u, v) = 1 - e^{-D^2(u, v)/2D_0^2}$$



**FIGURE 4.26** Results of highpass filtering the image of Fig. 4.11(a) using a GHPF of order 2 with  $D_0 = 15$ , 30, and 80, respectively. Compare with Figs. 4.24 and 4.25.

# Unsharp Masking, High-Boosting Filtering

$$g_{mask}(x, y) = f(x, y) - f_{lp}(x, y) \Leftrightarrow H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

$$g(x, y) = f(x, y) + k g_{mask}(x, y) \Leftrightarrow H_{hb}(u, v) = 1 + k H_{hp}(u, v)$$

$$g(x, y) = f(x, y) + k g_{mask}(x, y) \Leftrightarrow H_{hb}(u, v) = 1 + k H_{hp}(u, v)$$



original



blurred image



result

(Ref: [http://www.astropix.com/HTML/J\\_DIGIT/USM.HTM](http://www.astropix.com/HTML/J_DIGIT/USM.HTM))

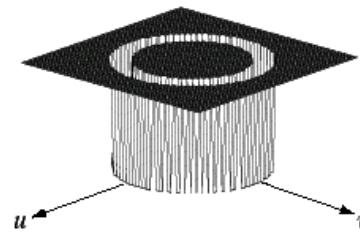
# Bandreject Filter

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) < D_0 - \frac{W}{2} \\ 0 & \text{if } D_0 - \frac{W}{2} \leq D(u, v) \leq D_0 + \frac{W}{2} \\ 1 & \text{if } D(u, v) > D_0 + \frac{W}{2} \end{cases}$$

$D(u, v)$ : distance from the origin.

- **Butterworth**

$$H(u,v) = \frac{1}{1 + \left[ \frac{D(u,v)W}{D^2(u,v) - D_0^2} \right]^{2n}}$$

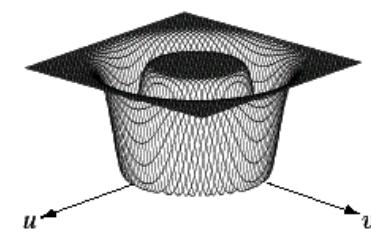
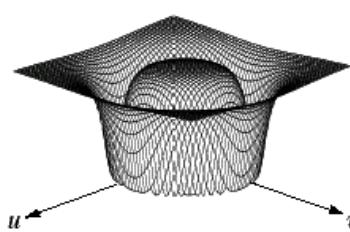


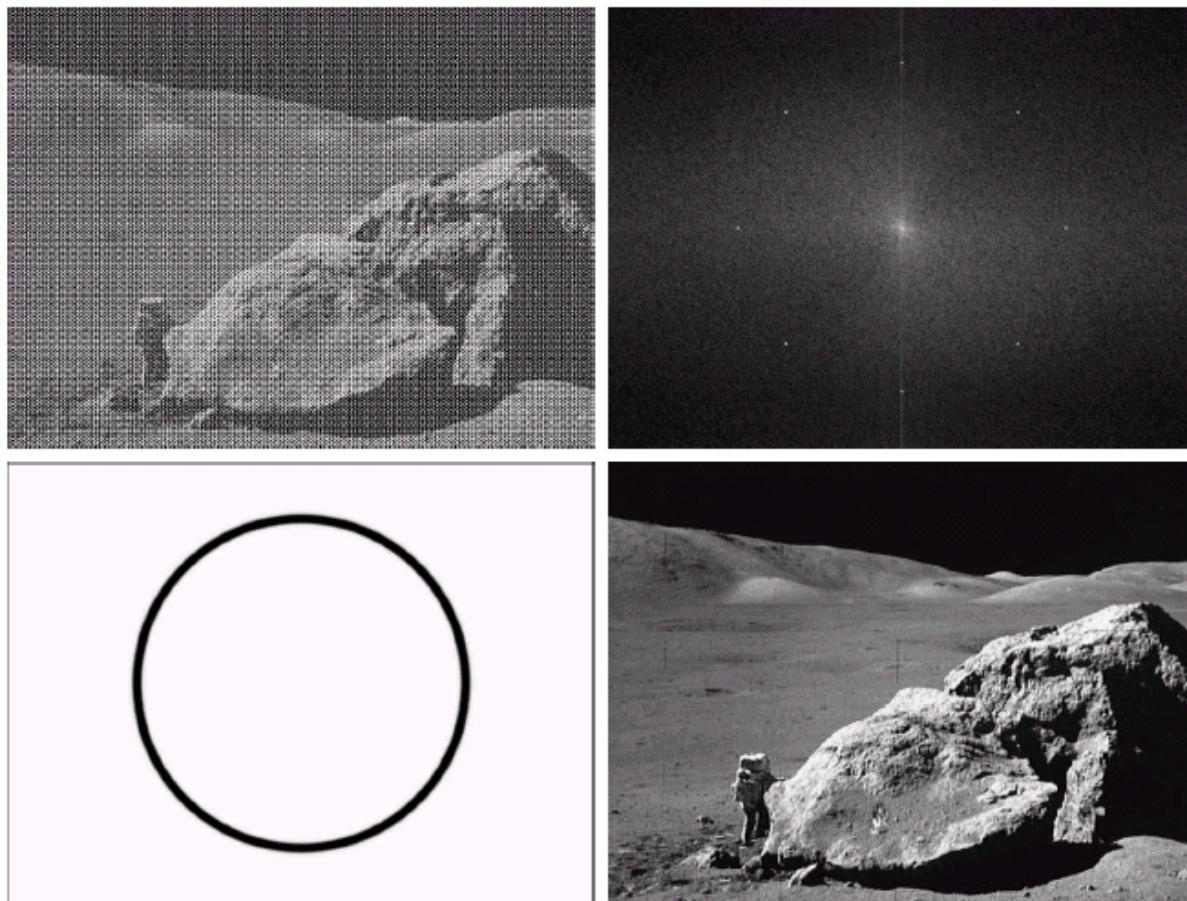
a b c

**FIGURE 5.15** From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

- **Gaussian**

$$H(u,v) = 1 - e^{-\frac{1}{2} \left[ \frac{D^2(u,v) - D_0^2}{D(u,v)W} \right]^2}$$





a b  
c d

**FIGURE 5.16**  
(a) Image corrupted by sinusoidal noise.  
(b) Spectrum of (a).  
(c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)

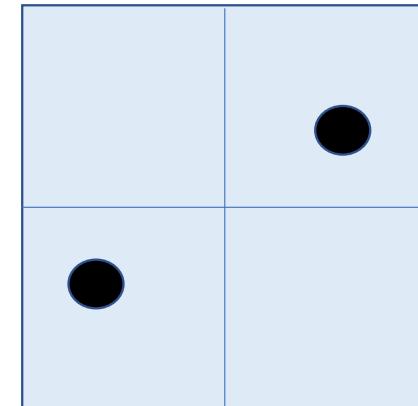
# Notch Filter

- **Ideal notch reject filter**

$$H(u,v) = \begin{cases} 0 & \text{if } D_1(u,v) \leq D_0 \text{ or } D_2(u,v) \leq D_0 \\ 1 & \text{otherwise} \end{cases}$$

$$D_1(u,v) = [(u - M/2 - u_0)^2 + (v - N/2 - v_0)^2]^{1/2}$$

$$D_2(u,v) = [(u - M/2 + u_0)^2 + (v - N/2 + v_0)^2]^{1/2}$$

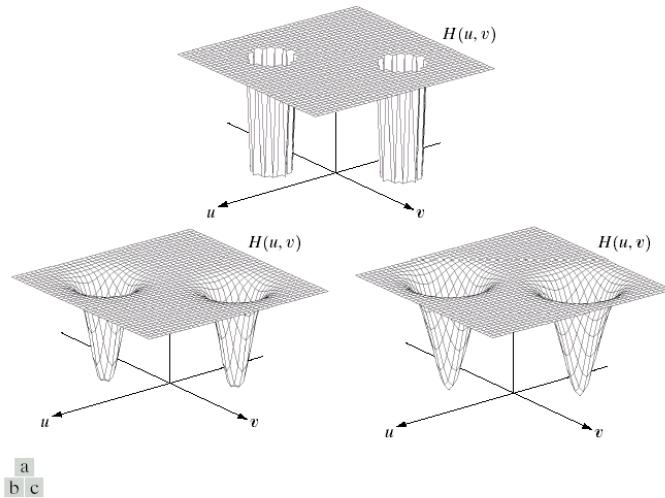


- **Butterworth notch reject filter**

$$H(u,v) = \frac{1}{1 + \left[ \frac{D_0^2}{D_1(u,v)D_2(u,v)} \right]^n}$$

- **Gaussian notch reject filter**

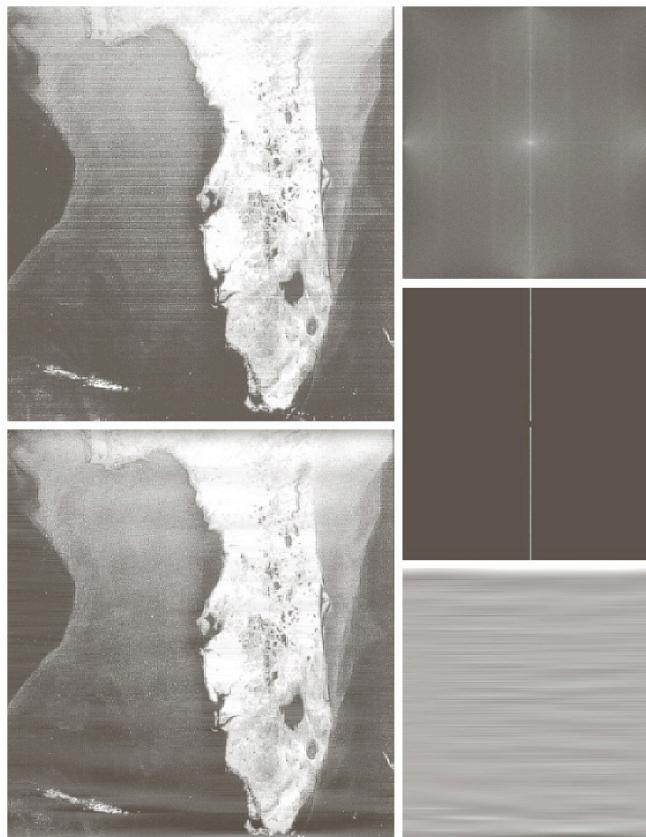
$$H(u,v) = 1 - e^{-\frac{1}{2} \left[ \frac{D_1(u,v)D_2(u,v)}{D_0^2} \right]^2}$$



**FIGURE 5.18** Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

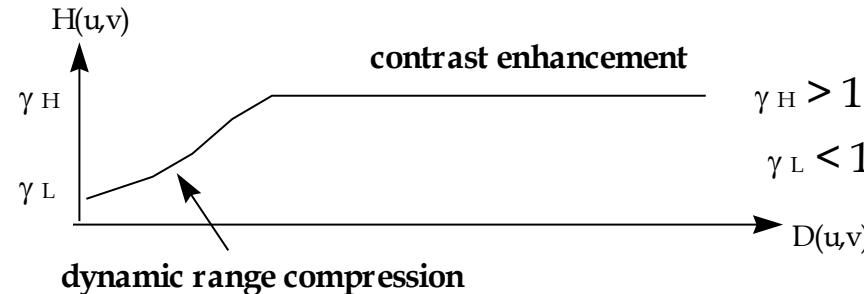
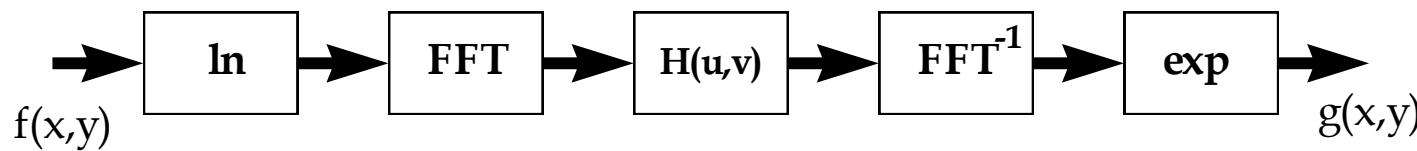
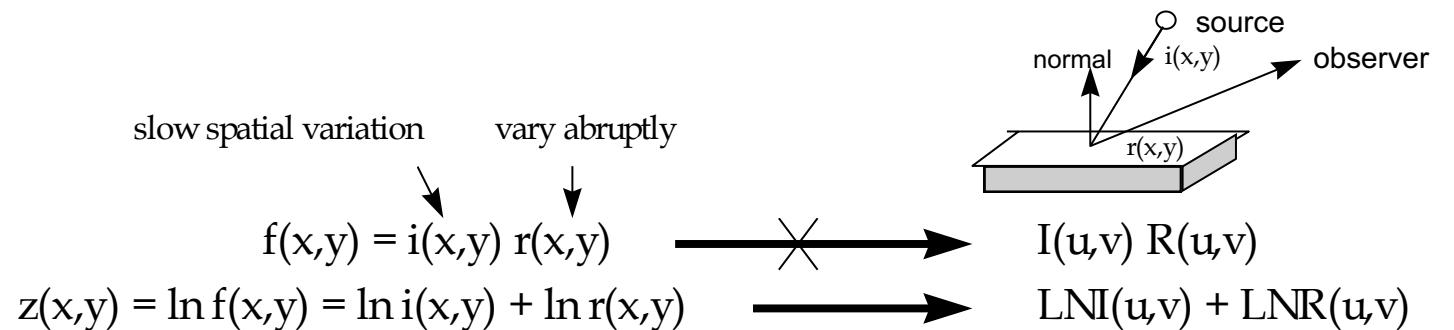
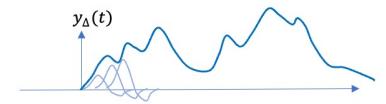
# Notch Pass Filter

$$H_{np}(u, v) = 1 - H_{nr}(u, v)$$



**FIGURE 5.19**  
(a) Satellite image of Florida and the Gulf of Mexico showing horizontal scan lines.  
(b) Spectrum. (c) Notch pass filter superimposed on (b). (d) Spatial noise pattern. (e) Result of notch reject filtering.  
(Original image courtesy of NOAA.)

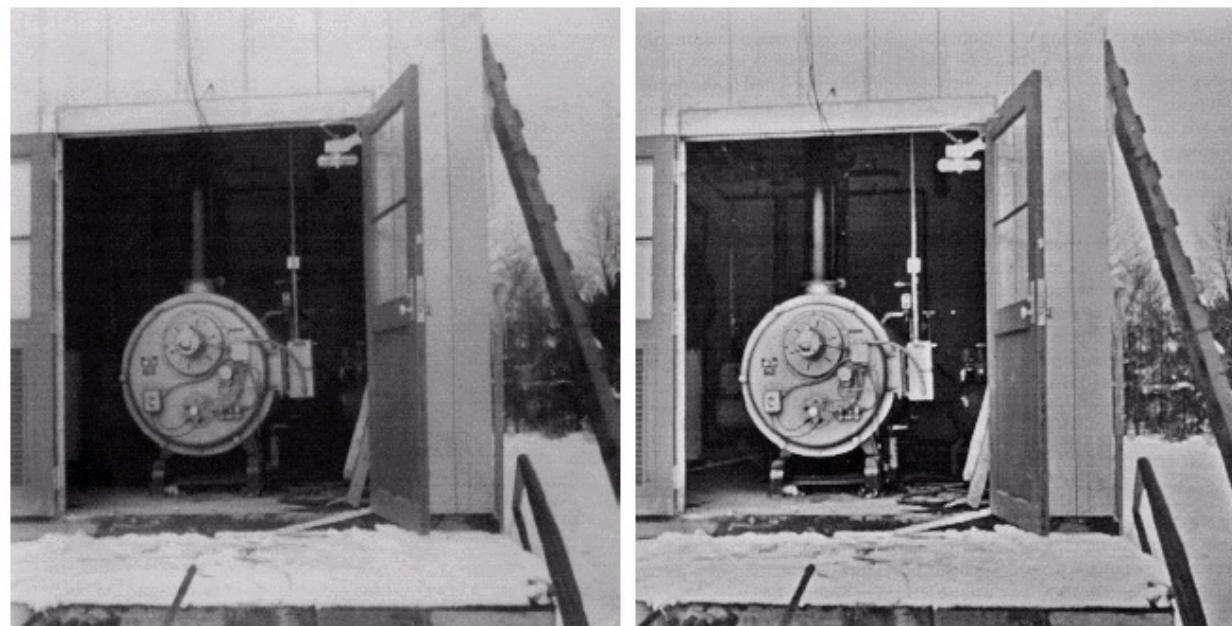
# Homomorphic Filter



$$\begin{aligned} y_\Delta(t) &= T\left\{\sum_{n=-\infty}^{\infty} x(n\Delta)\delta_\Delta(t - n\Delta)\Delta\right\} \\ &= \sum_{n=-\infty}^{\infty} x(n\Delta)T\{\delta_\Delta(t - n\Delta)\}\Delta \\ &= \sum_{n=-\infty}^{\infty} x(n\Delta)h_\Delta(t - n\Delta)\Delta \end{aligned}$$

a b

**FIGURE 4.33**  
(a) Original  
image. (b) Image  
processed by  
homomorphic  
filtering (note  
details inside  
shelter).  
(Stockham.)





Original image



Processed image

(Ref: <http://www.vision.ee.ethz.ch/~pcattin/SIP/5-Enhancement.html> )