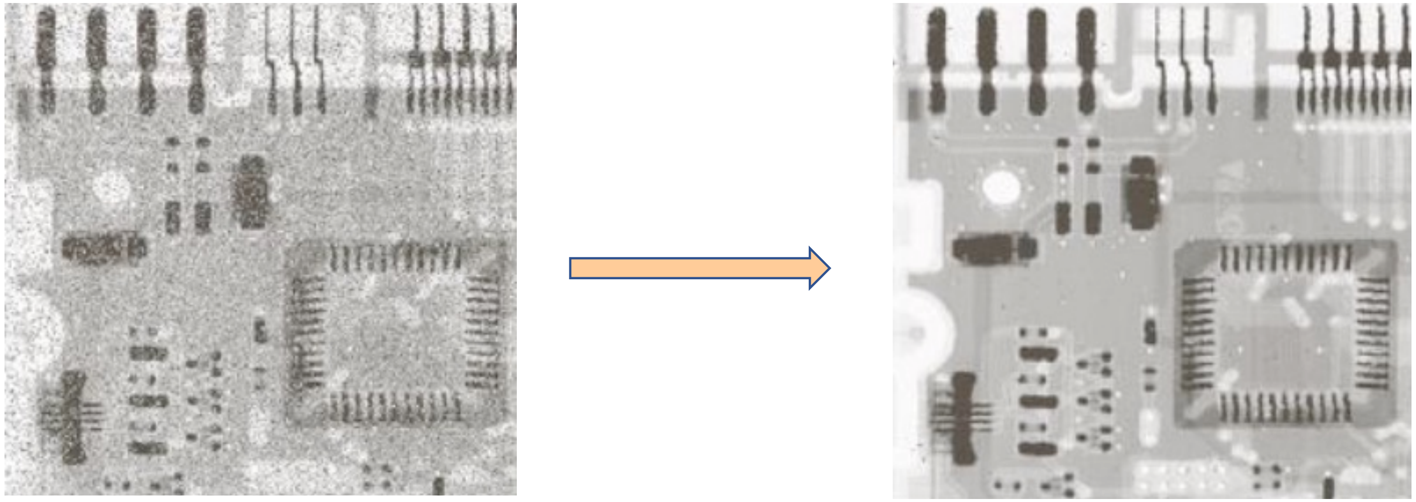


Image Enhancement

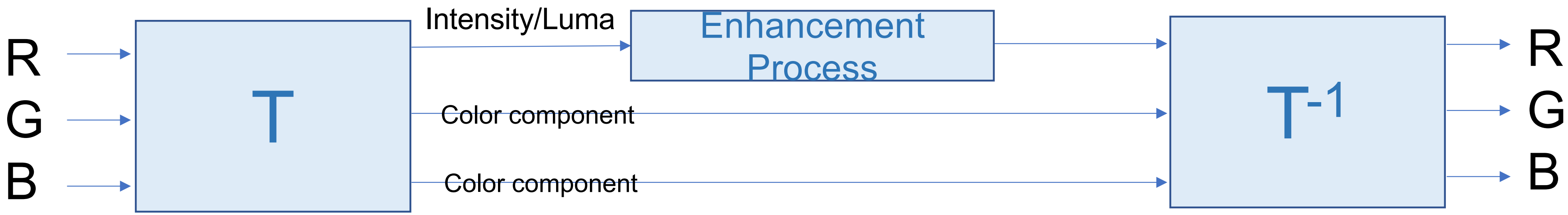
Fall 2024

Yi-Ting Chen

Topics

- Contrast Enhancement
- Sharpness Enhancement
- Noise Suppression

Enhancement of Color Images

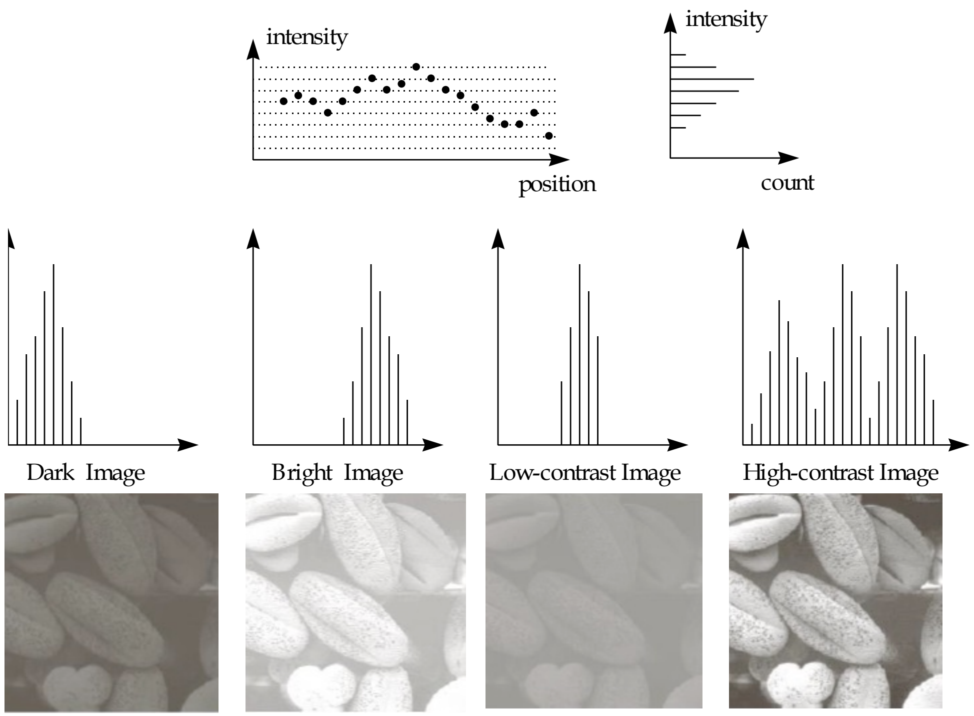


Topics to be discussed in Image Enhancement

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

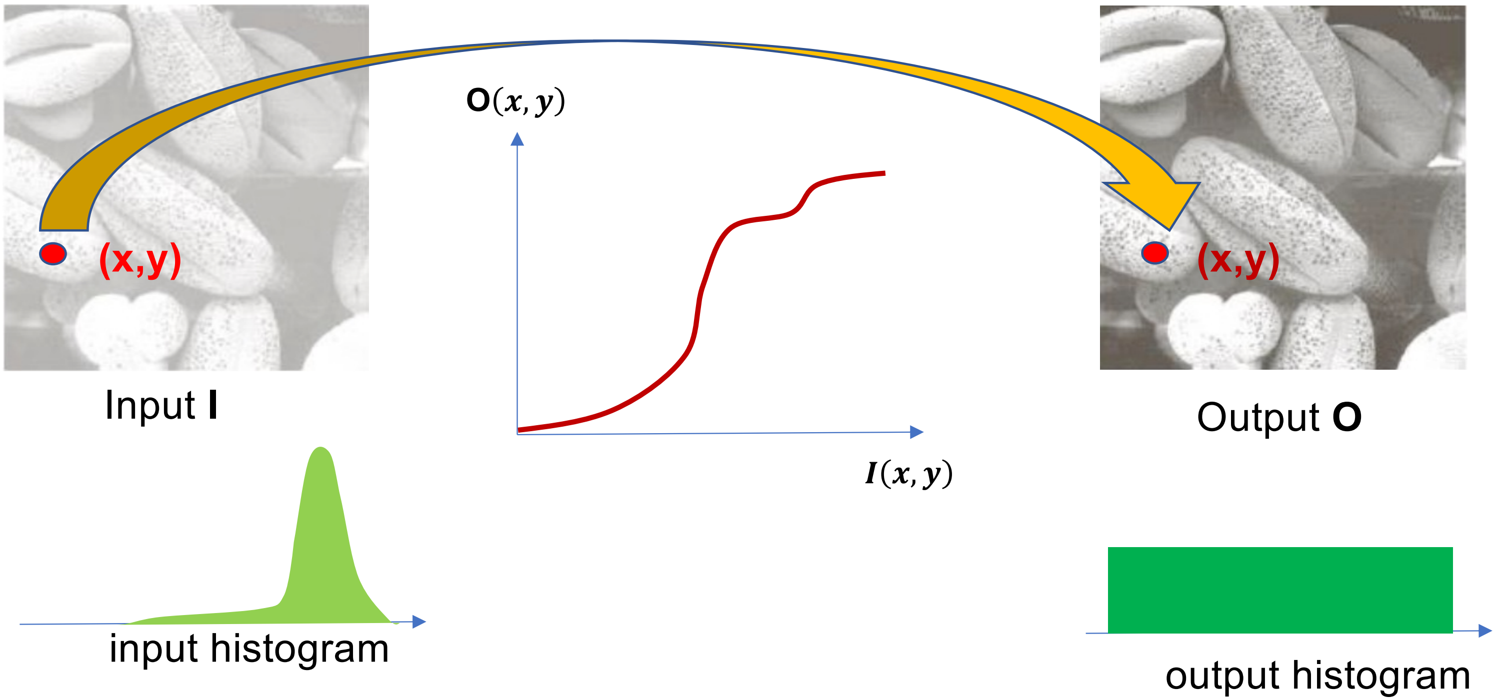
Intensity-domain Processing

Histogram

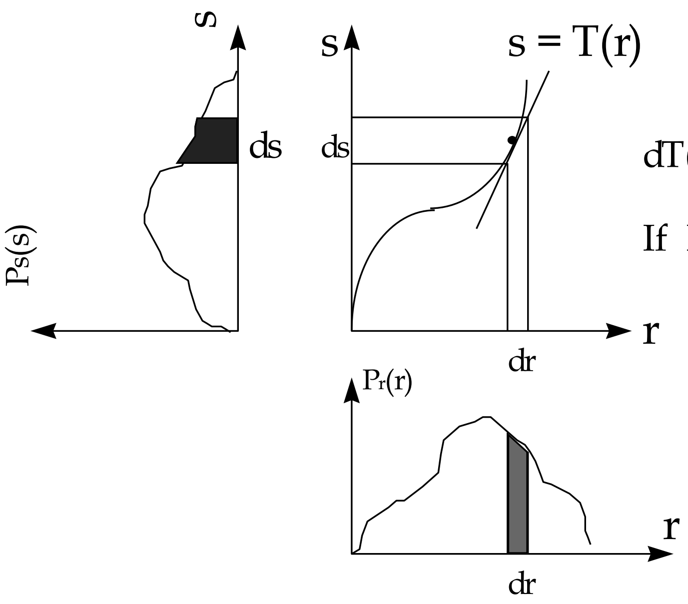


intensity
position
intensity
count
Dark Image
Bright Image
Low-contrast Image
High-contrast Image

Histogram Equalization (1/3)



Histogram Equalization (2/3)



$$P_r(r)\,dr = P_s(s)\,ds$$

$$dT(r)/dr = ds/dr = P_r(r)/P_s(s)$$

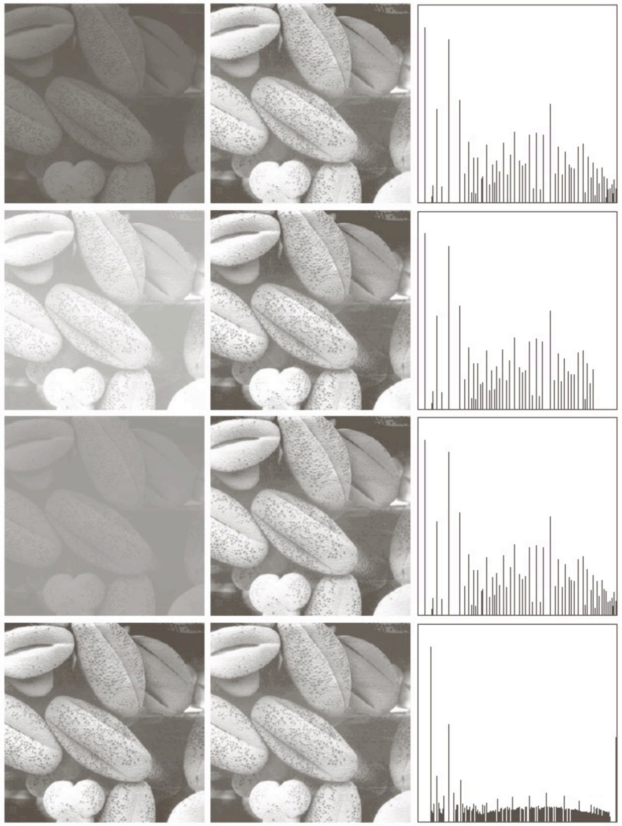
If $P_s(s)$ = constant = K

$= 1/(\mathbf{r}_{max} - \mathbf{r}_{min})$

$$\mathbf{T(r)} = \frac{\mathbf{1}}{\mathbf{K}}\int_{\mathbf{r}_{min}}^{\mathbf{r}} \mathbf{P_r(r)\ dr} + \mathbf{r}_{min}$$

for $\mathbf{r}_{min} \leq \mathbf{r} \leq \mathbf{r}_{max}$.

Histogram Equalization (3/3)



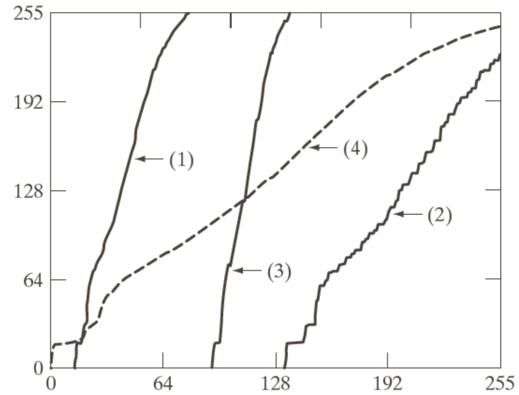
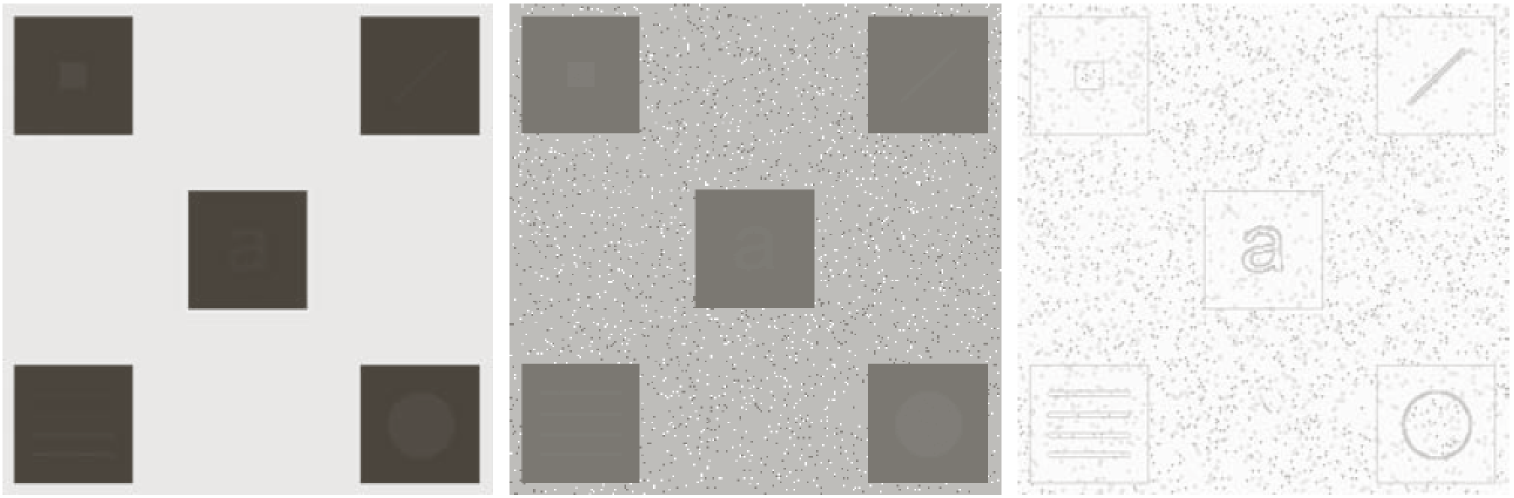


FIGURE 3.21 Transformation functions for histogram equalization. Transformations (1) through (4) were obtained from the histograms of the images (from top to bottom) in the left column of Fig. 3.20 using Eq. (3.3-8).

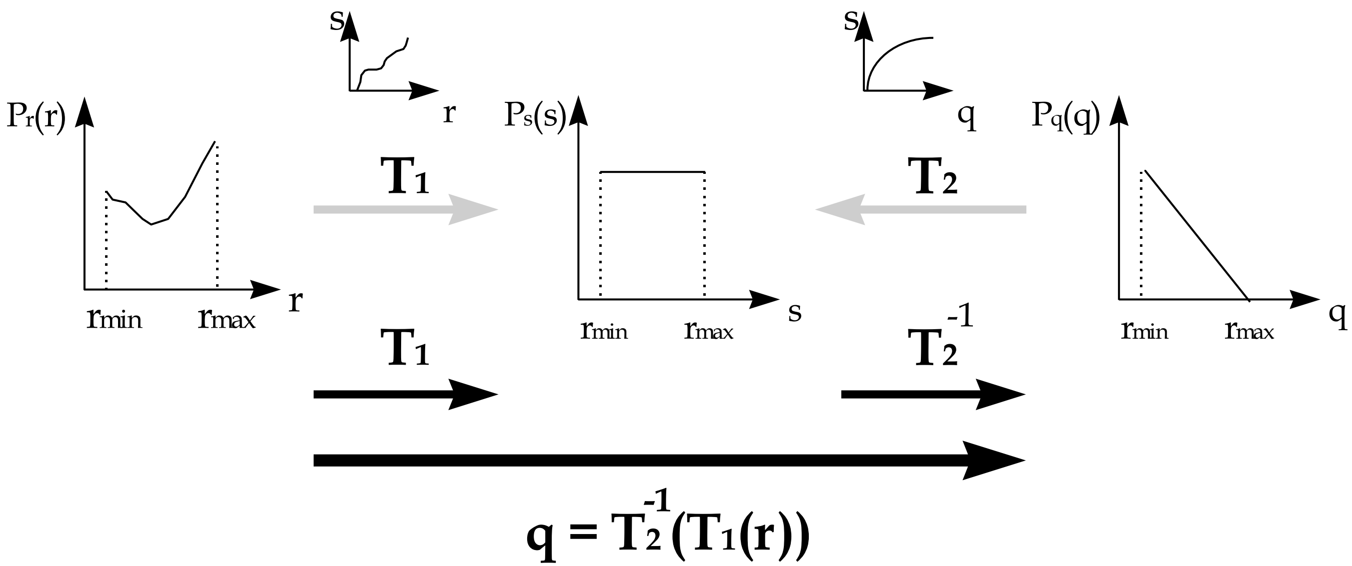
Local Histogram Equalization

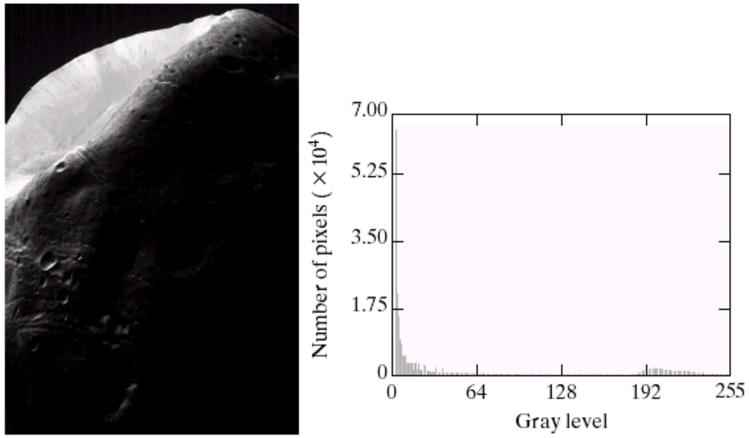


a b c

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 3×3.

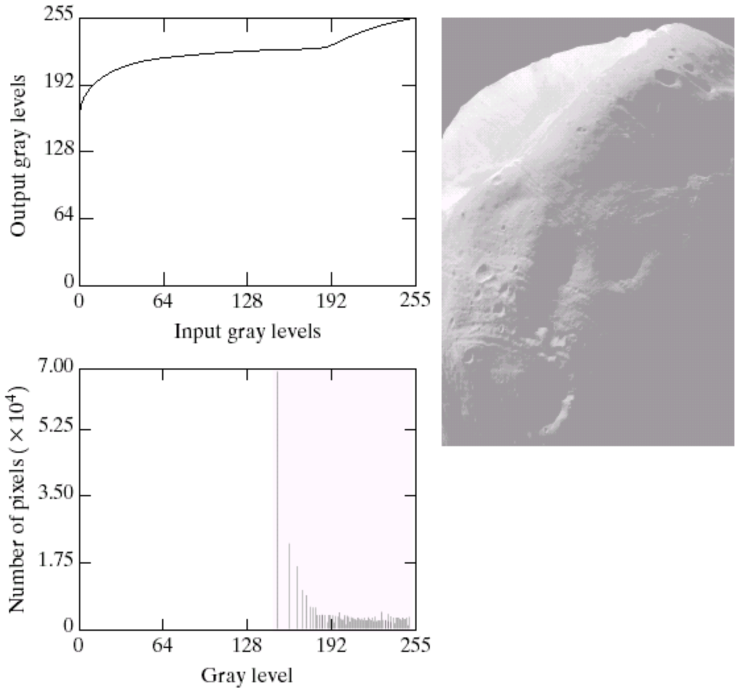
Histogram Matching





a b

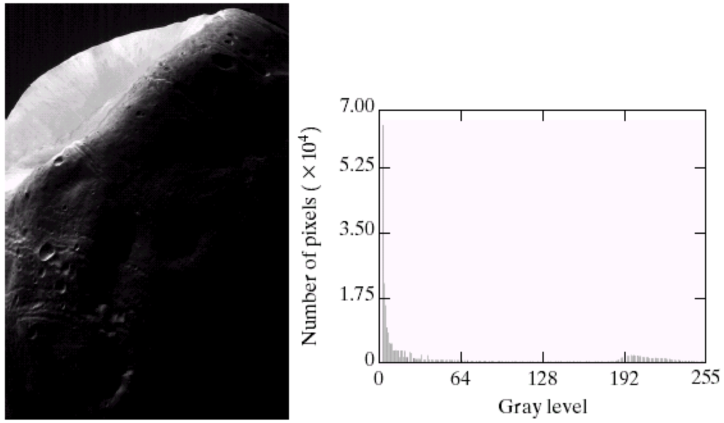
FIGURE 3.20 (a) Image of the Mars moon Photos taken by NASA's ***Mars Global Surveyor.*** (b) Histogram. (Original image courtesy of NASA.)



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

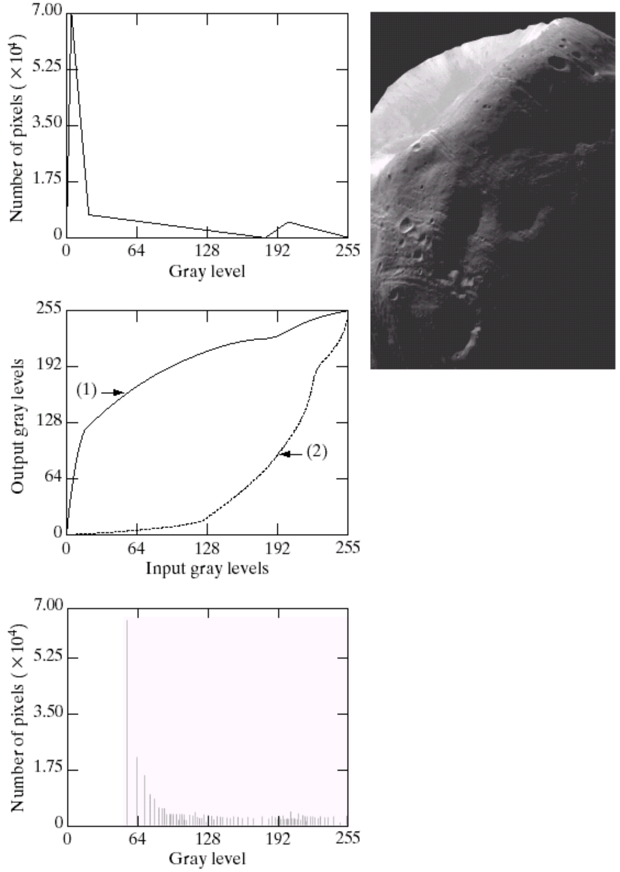


a b

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 c
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).



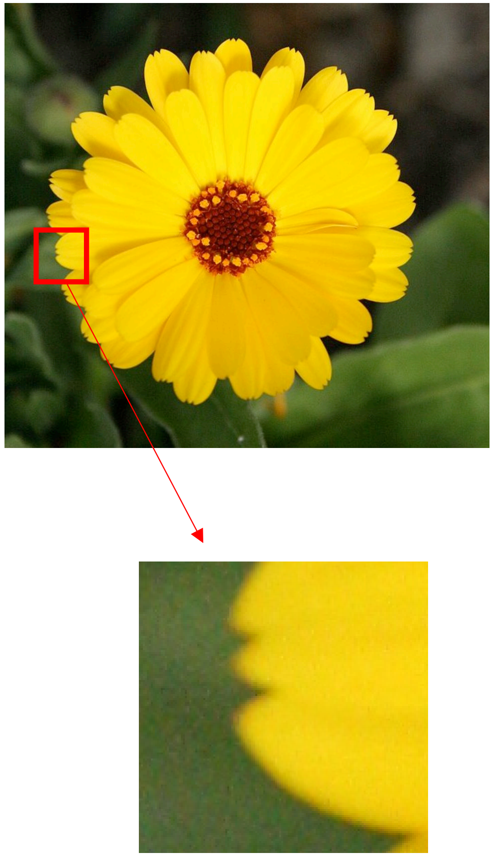
Histogram Matching

Spatial-domain Processing

Topics

- Image Smoothing
- Sharpness Enhancement
- Contrast Enhancement

Image Noise



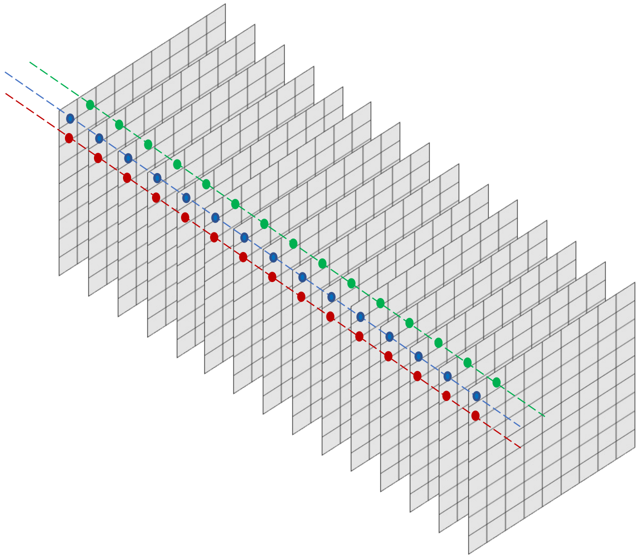
$$I(x,y) = \underbrace{S(x,y)}_{\text{signal}} + \underbrace{N(x,y)}_{\text{noise}}$$

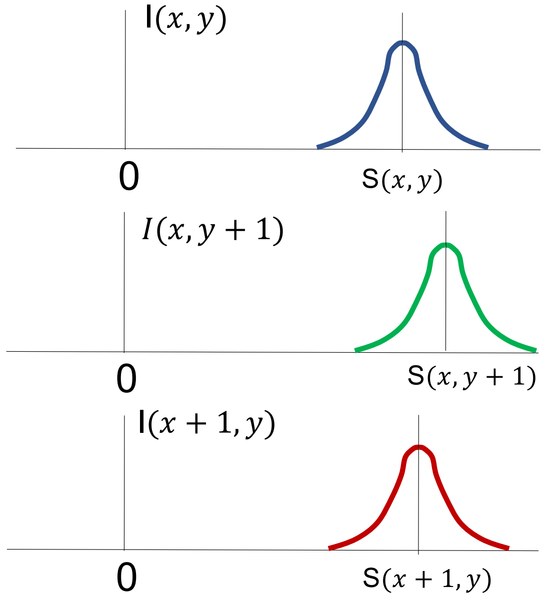
Typically, we assume
- Image noise has zero mean.
- Image noise has the same variance at different pixels.
- image noise at different pixels are uncorrelated.

Ref: https://en.wikipedia.org/wiki/Image_noise

Image Noise

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





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

signal noise

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

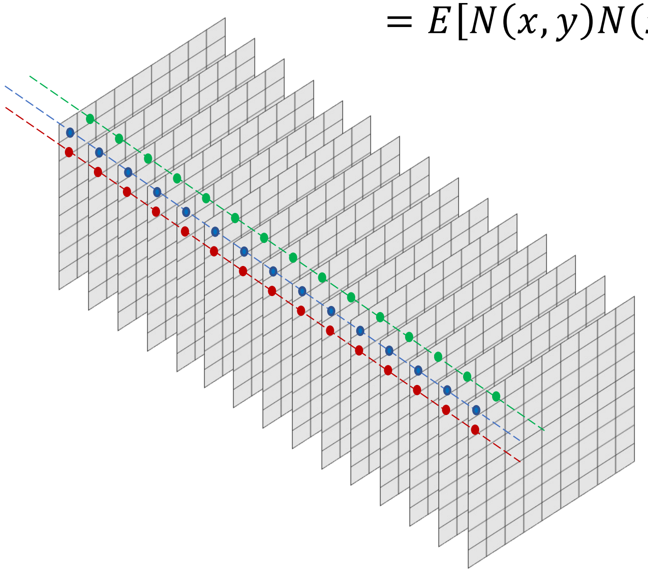
Image Noise

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

signal noise

- image noise at different pixels are uncorrelated.

$$\mathrm{E}[(I(x,y) - E[I(x,y)])(I(x',y') - E[I(x',y')])]$$
$$= E[N(x,y)N(x',y')] = E[N(x,y)]E[N(x',y')] = 0$$



Smoothing Filtering

Purpose: blurring & noise reduction

Linear Filter: Low-pass spatial filter

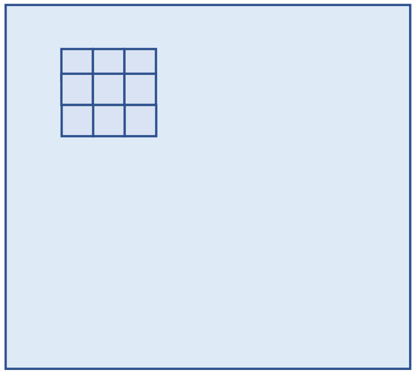
$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

Local Averaging



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

where $I(x,y) = \underbrace{S(x,y)}_{\text{signal}} + \underbrace{N(x,y)}_{\text{noise}}$

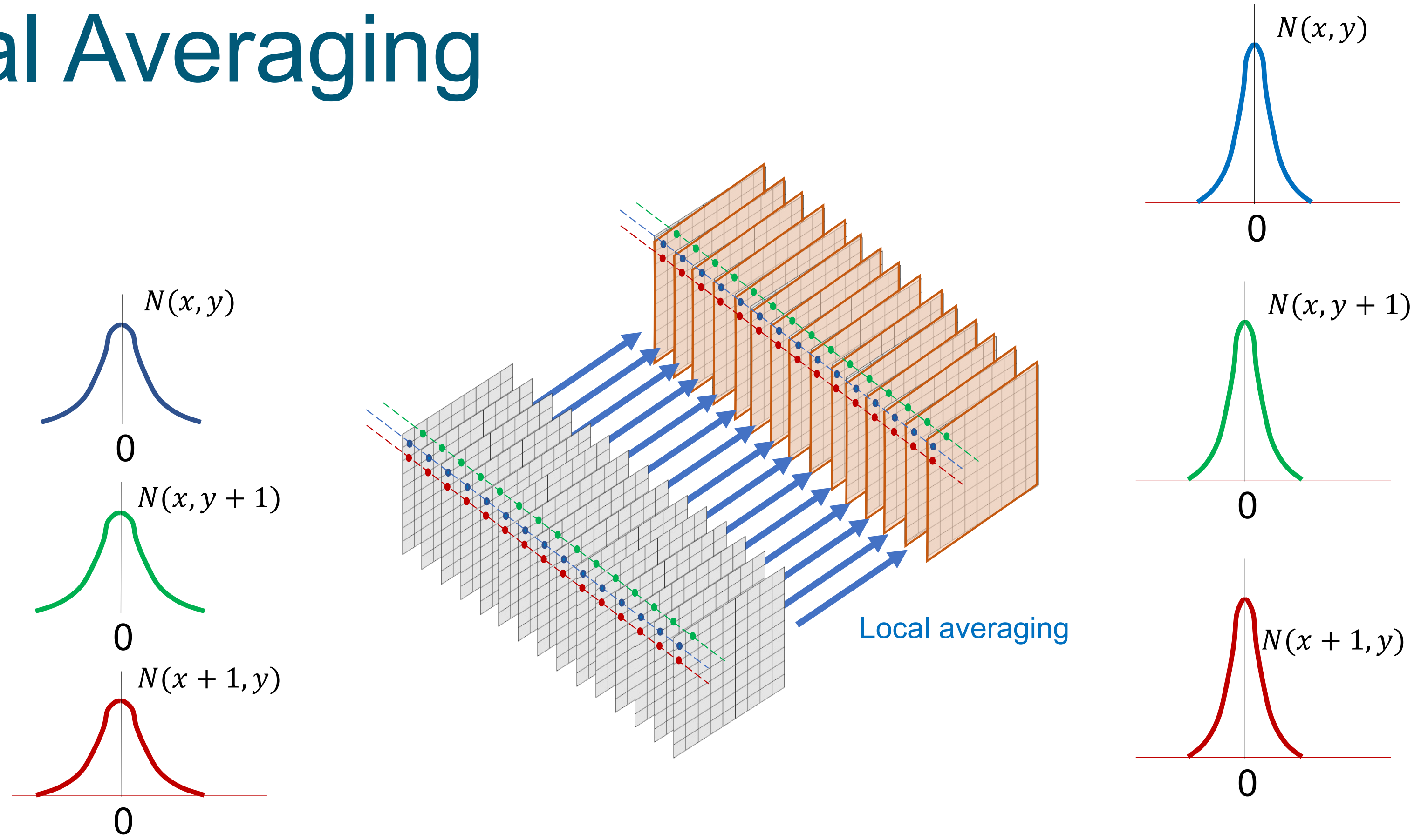
$$E\left[\hat{I}(x,y)\right] = \frac{1}{9}\sum_{i=-1}^{1}\sum_{j=-1}^{1} S(x+i, y+j) + E\left[\frac{1}{9}\sum_{i=-1}^{1}\sum_{j=-1}^{1} N(x+i, y+j)\right]$$

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

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

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

Local Averaging



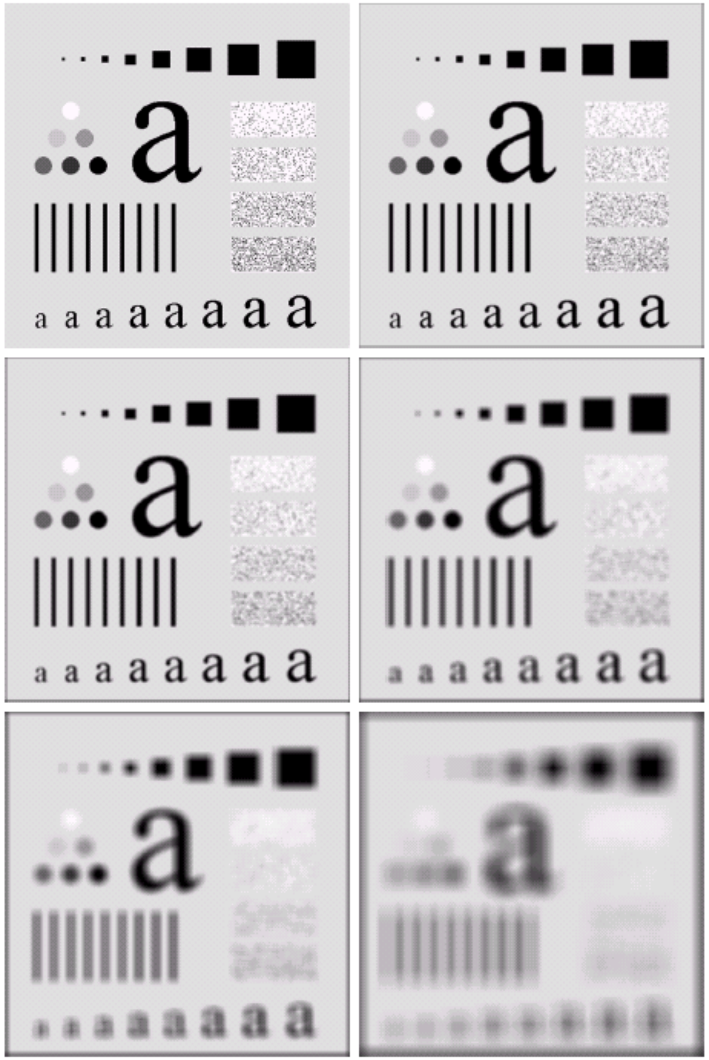
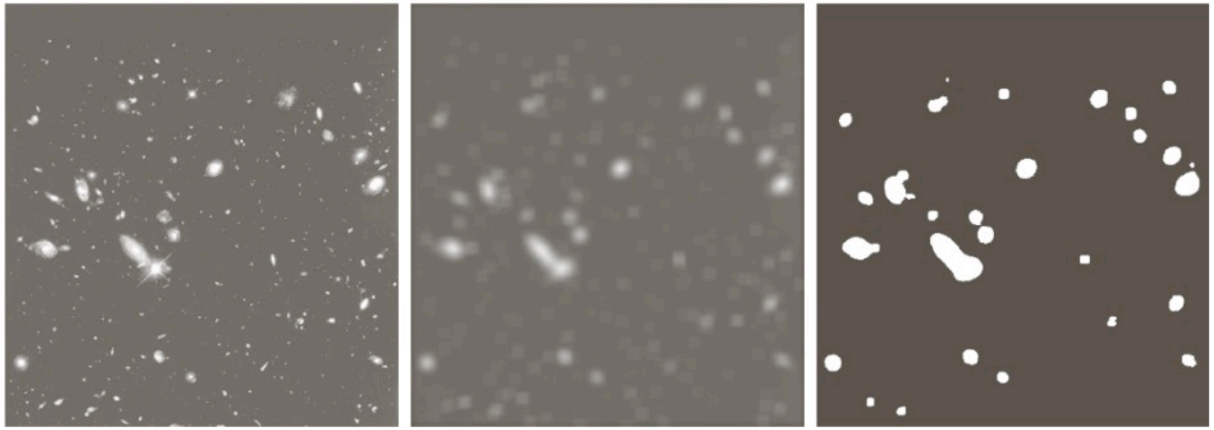


FIGURE 3.33 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. 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



a b c

FIGURE 3.34 (a) Image of size 528 × 485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15 × 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**

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

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

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

Remark: such as noise in range imaging.

Noise Models (2/6)

- **Erlang (Gamma) Noise**

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

Remark: such as noise in laser imaging.

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

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

Remark: such as noise in laser imaging.

Noise Models (3/6)

- **Uniform Noise**

$$p(z) = \begin{cases} \dfrac{1}{b-a} & \text{for } a \le z \le b \\ 0 & otherwise \end{cases}$$

$$\mu = \frac{a+b}{2}$$

$$\sigma^2 = \frac{(b-a)^2}{12}$$

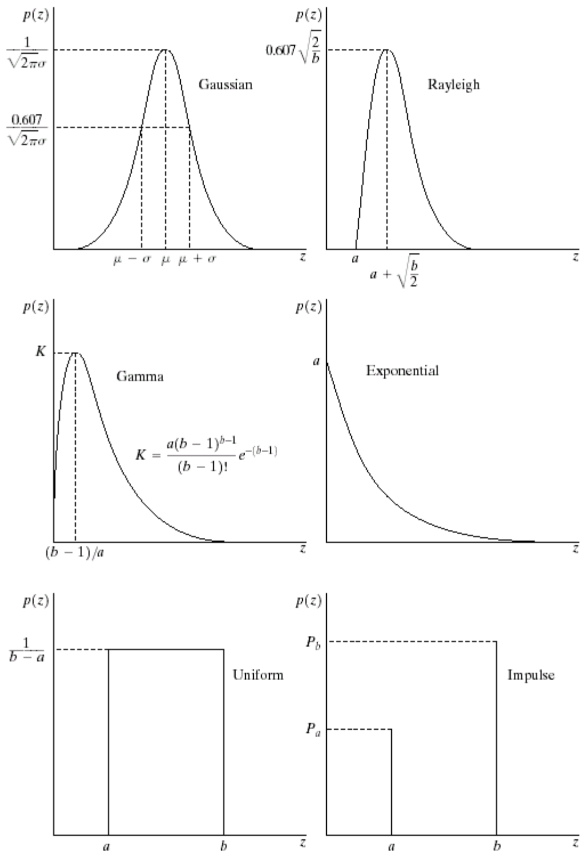
Remark: useful as the basis for numerous random number generation.

- **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)

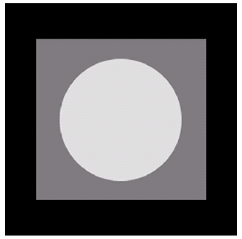


FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.

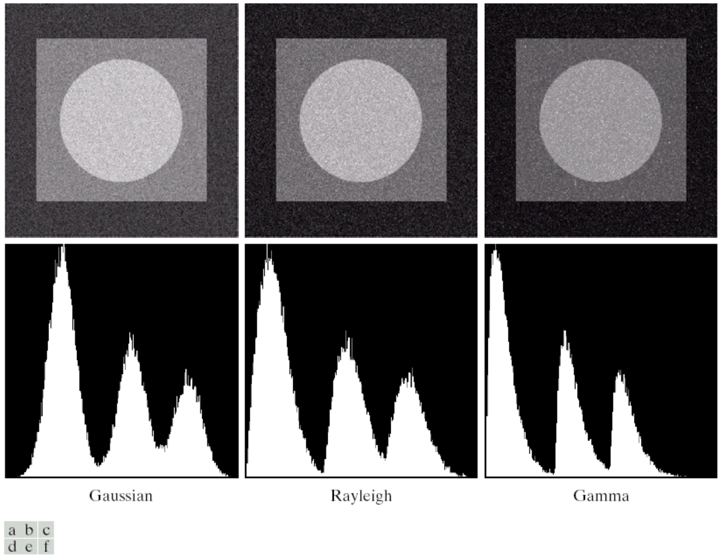


FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

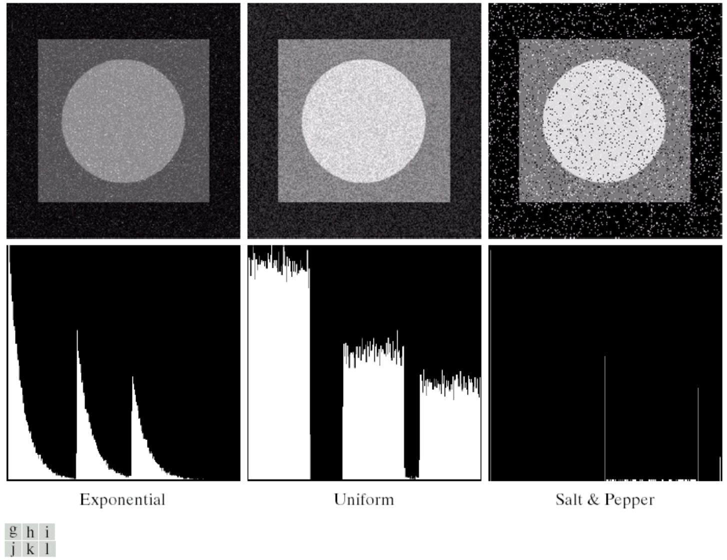


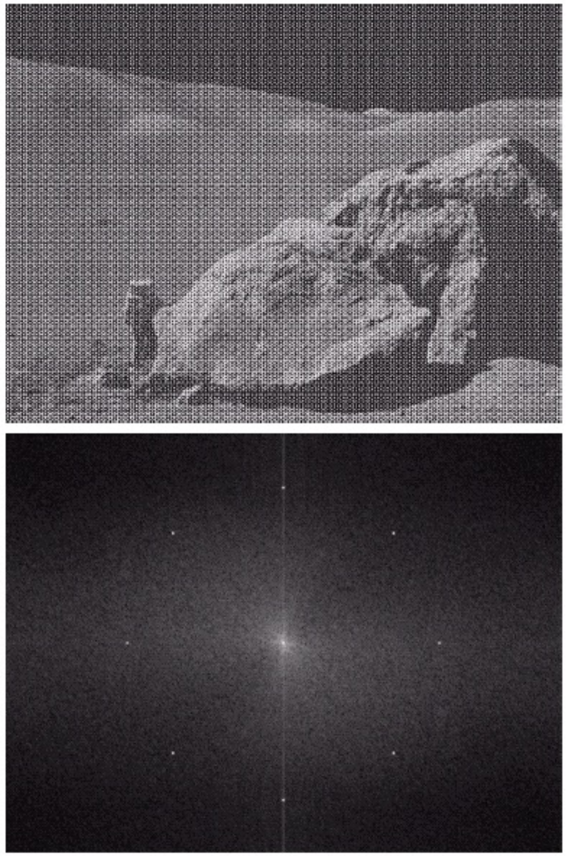
FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and impulse noise to the image in Fig. 5.3.

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

$\frac{1}{9} \times$

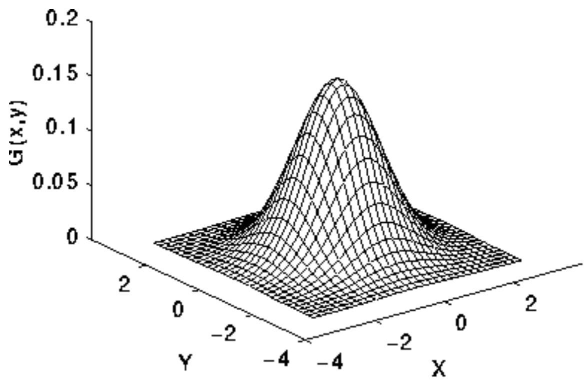
1	1	1
1	1	1
1	1	1

$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

Gaussian Smoothing

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



$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

$\sigma = 1$

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

https://en.wikipedia.org/wiki/Gaussian_blur

- Gaussian kernel is separable and symmetric

Order-Statistics Filters (1/2)

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

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

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

  Remark: Max filter works well for pepper noise.
  Min filter works well for salt noise.

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

  Remark: work well for Gaussian noise and uniform noise.

Order-Statistics Filters (2/2)

Adaptive, Local Noise Reduction Filter

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

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

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

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

f(**x**): original image

c(ξ,**x**): measure the *geometric* closeness between **x** and a nearby point ξ

Bilateral Filter (2/4)

Bilateral filter

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

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

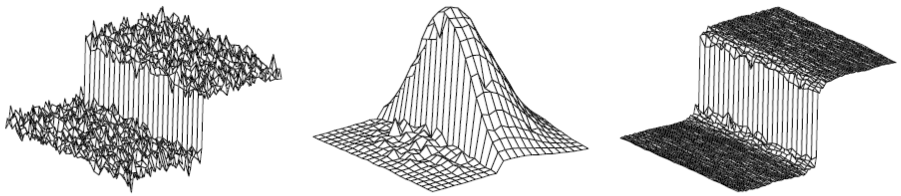
$s(\mathbf{f}(\xi),\mathbf{f}(\mathbf{x}))$: measure the *photometric* similarity between the pixel at **x** and that of a nearby point ξ.

Bilateral Filter (3/4)

Example

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

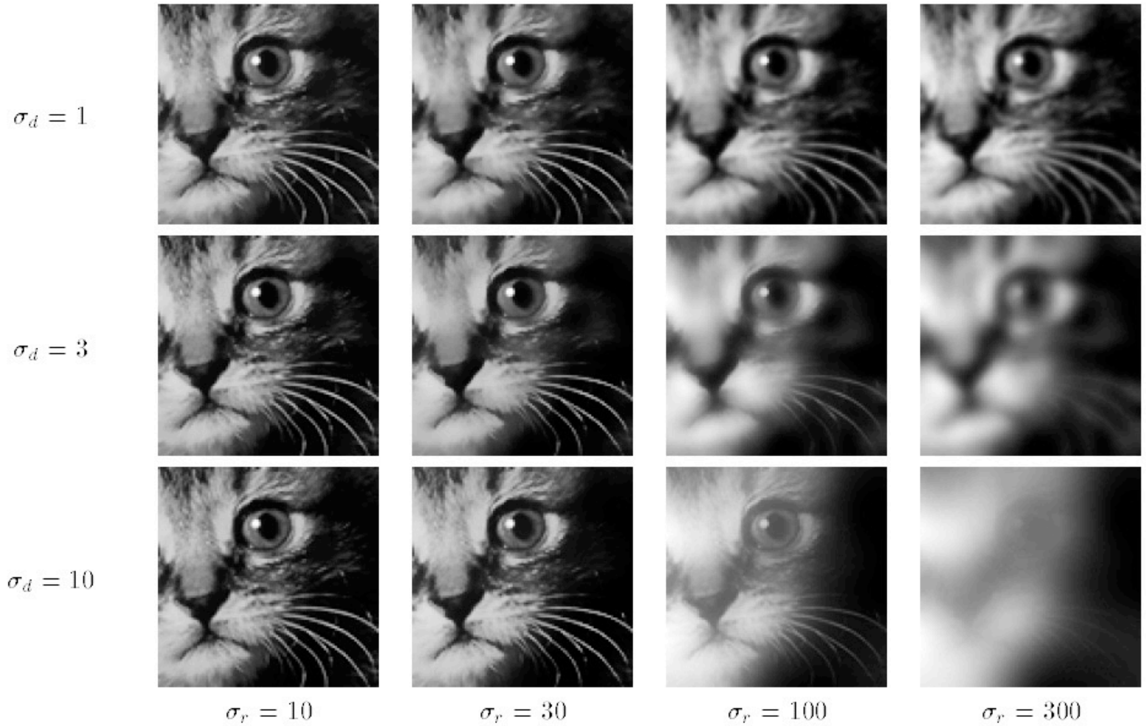
$$s(\xi,\mathbf{x}) = \exp\{-\frac{1}{2}(\frac{\delta(f(\xi), f(\mathbf{x}))}{\sigma_r})^2\} \quad \text{where} \quad \delta(\phi,\mathbf{f}) = \|\phi - \mathbf{f}\|$$



(a) (b) (c)

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 23×23 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.

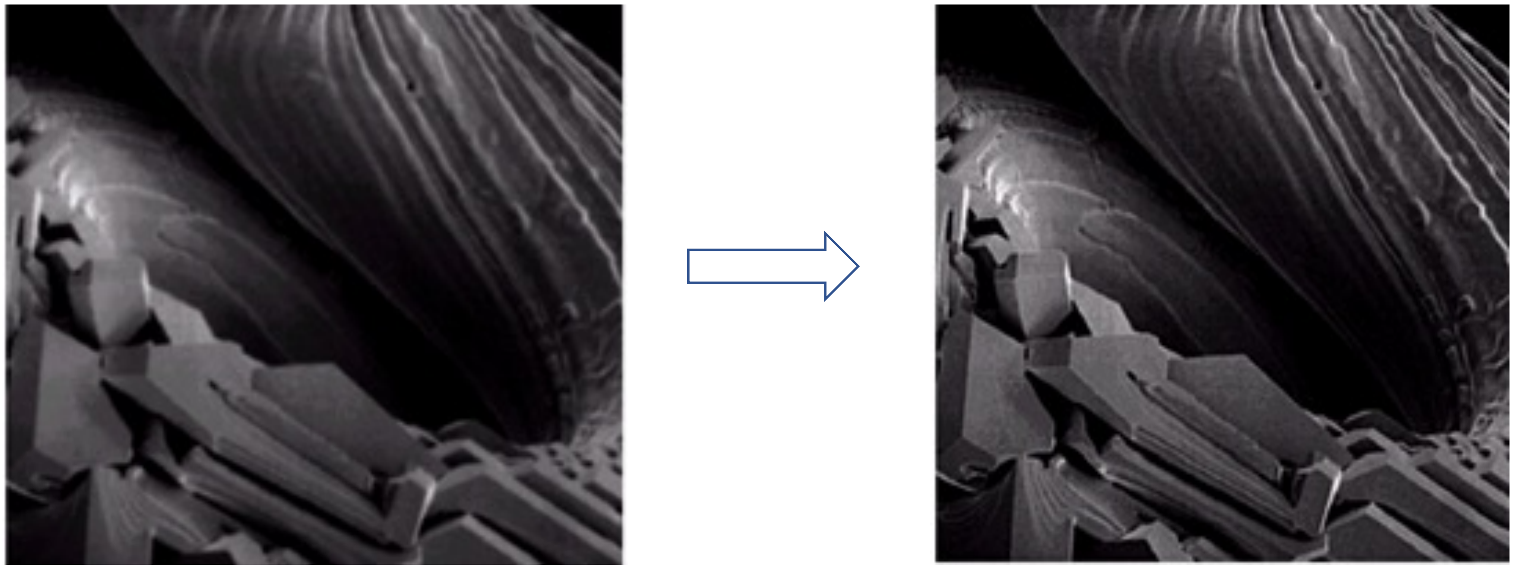
Bilateral Filter (4/4)



Ref: http://www.cs.duke.edu/~tomasi/papers/tomasi/tomasiIccv98.pdf

Sharpness Enhancement

- Purpose: highlight or enhance fine detail.

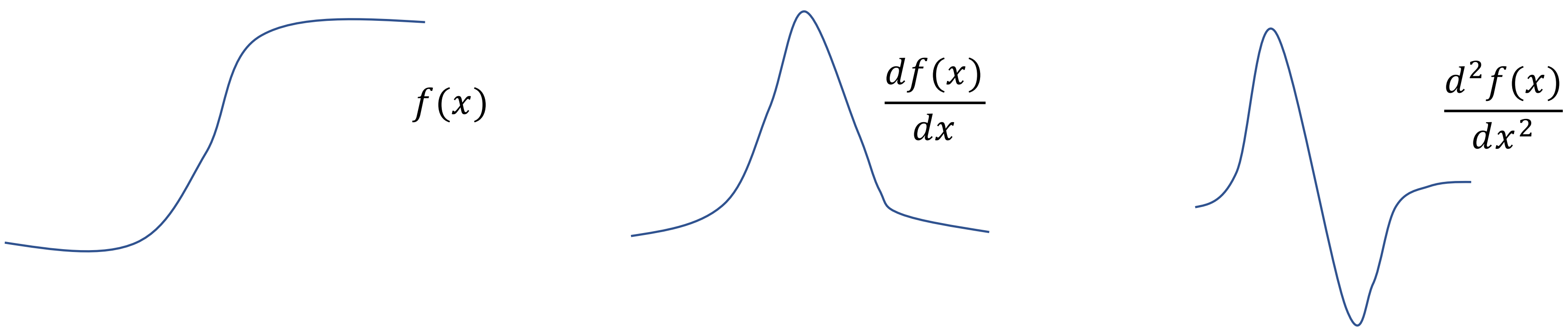


Sharpness Enhancement

- Typically used measurement:

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

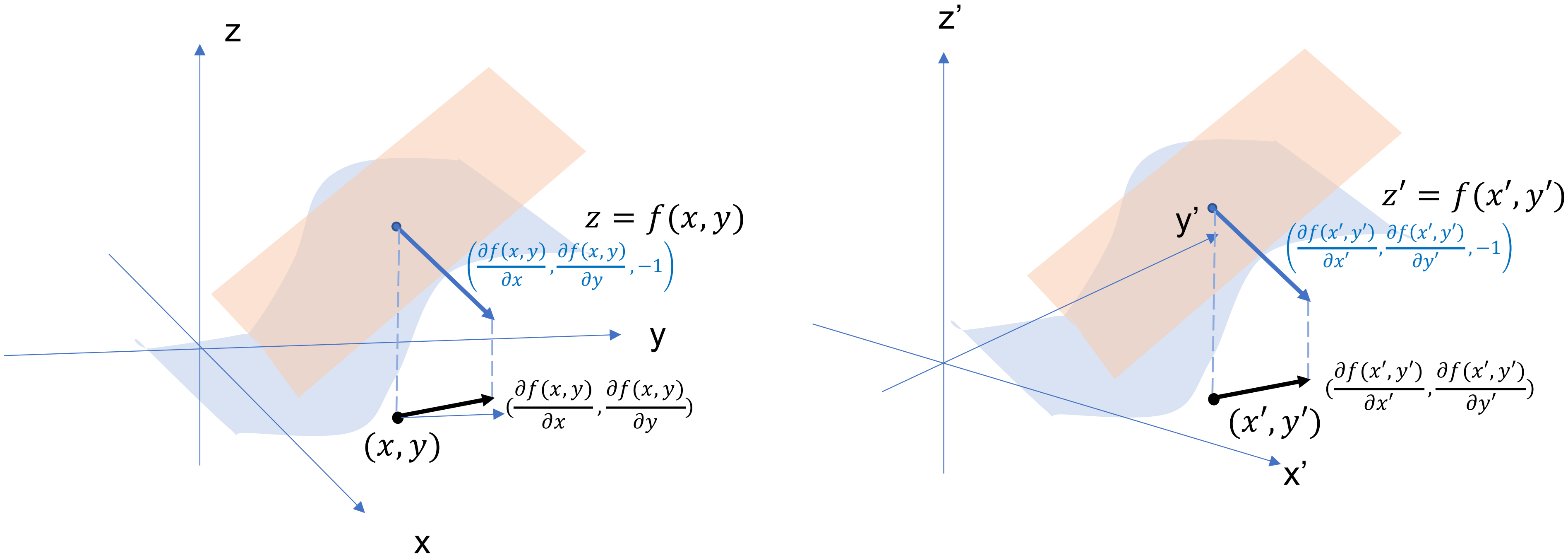
2nd-order derivative $\dfrac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$



1^{st} Derivative

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

Gradient



Roberts

1	0
0	-1

	1
0 -1	0

Prewitt

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

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

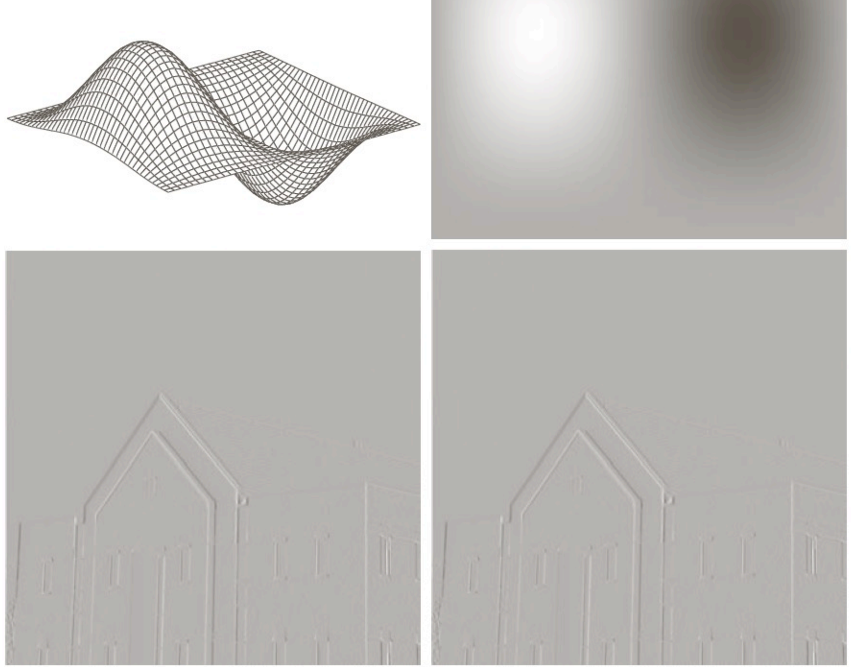
Sobel

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

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

Sobel Operator

−1	0	1
−2	0	2
−1	0	1

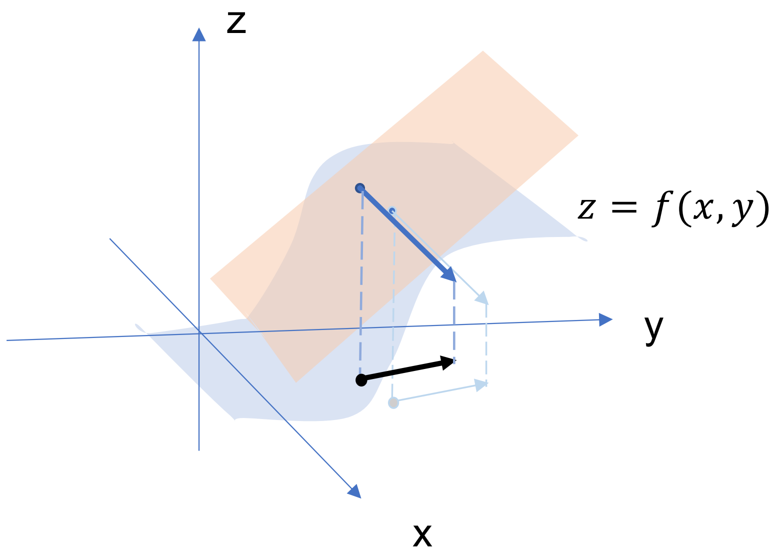


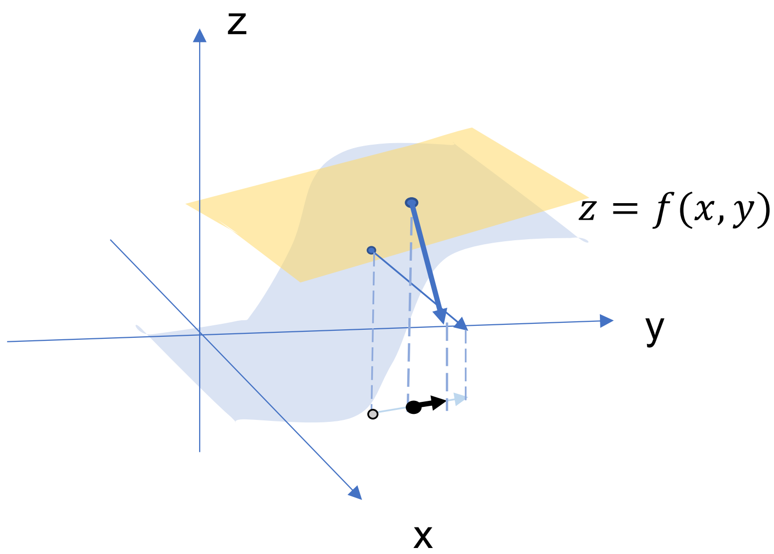
a b
c d

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.

2nd Derivatives

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





2nd-Derivatives

- **Laplacian Filter**

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

(a)

0	1	0
1	−4	1
0	1	0

(b)

1	1	1
1	−8	1
1	1	1

(c)

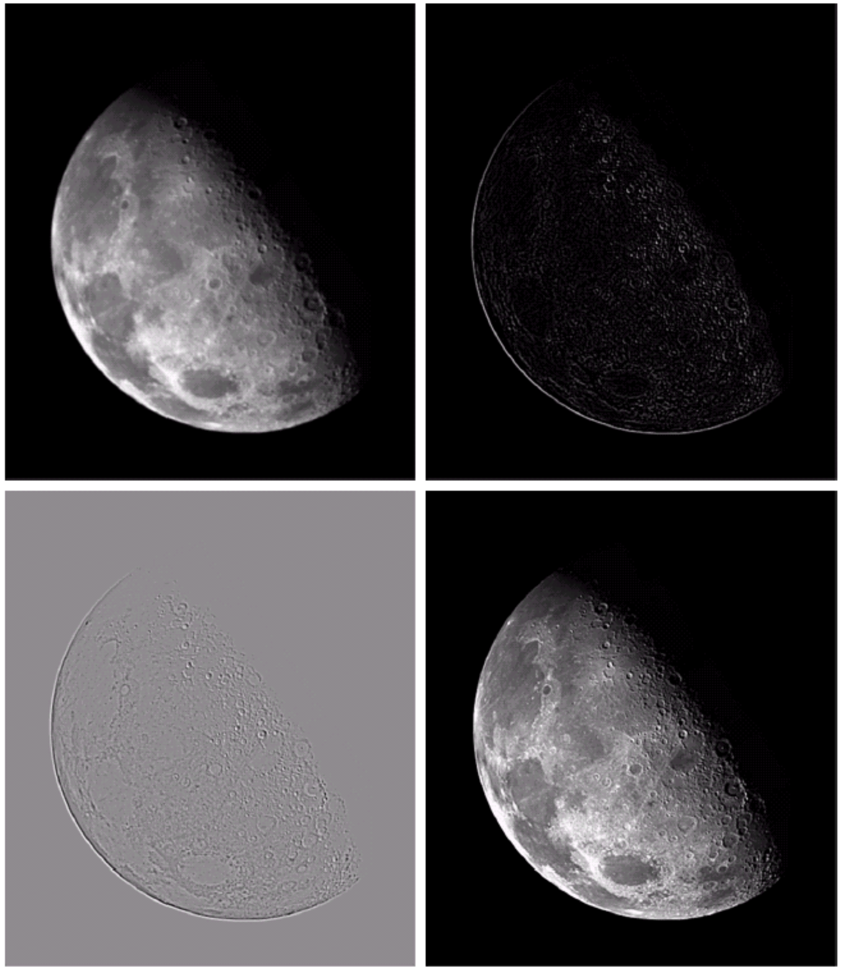
0	−1	0
−1	4	−1
0	−1	0

(d)

−1	−1	−1
−1	8	−1
−1	−1	−1

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.

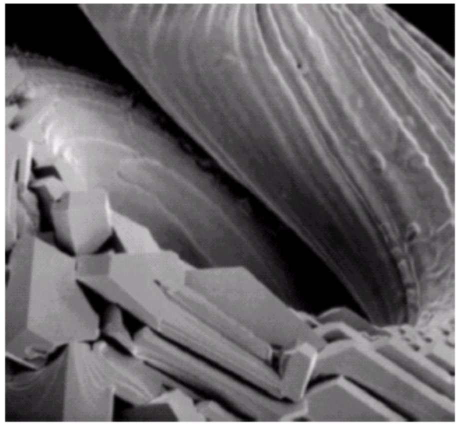


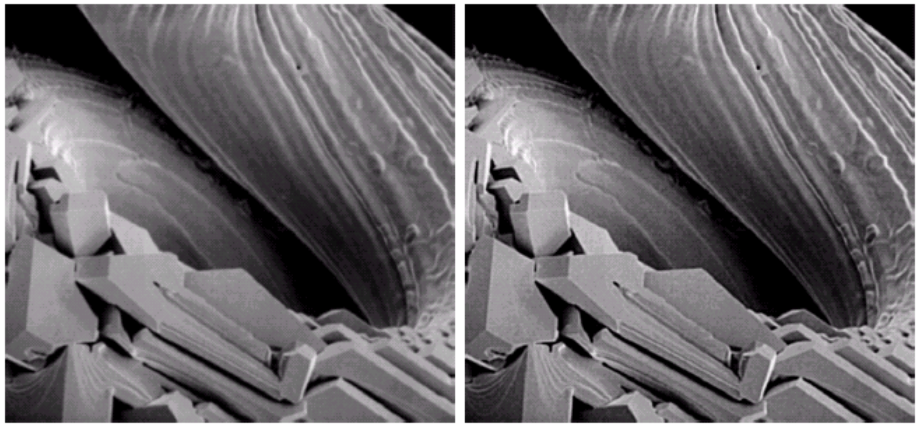
a b
c d

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.)

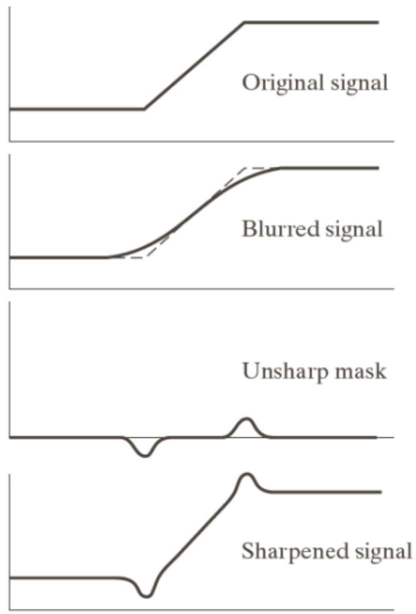
Unsharp Masking & High-Boosting Filtering

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

$\overline{f}(x,y)$: 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

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

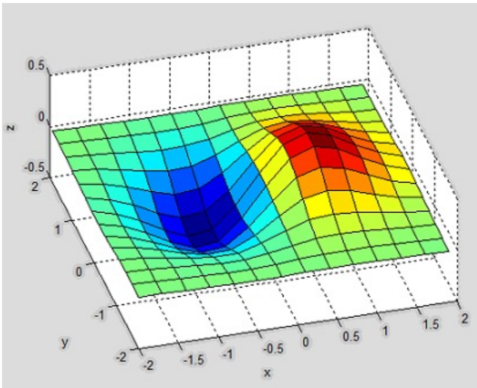
$$\frac{\partial}{\partial y}(f(x,y) * G(x,y)) = f(x,y) * (\frac{\partial G(x,y)}{\partial y})$$

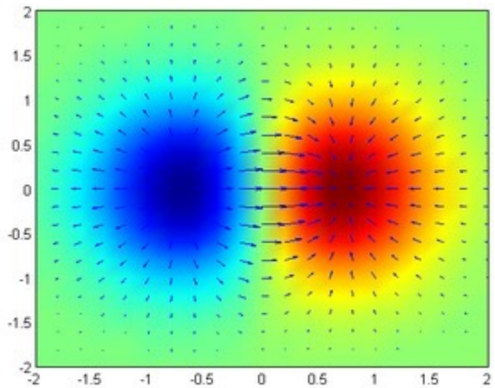
$$\nabla^2(f(x,y) * G(x,y)) = f(x,y) * \nabla^2 G(x,y)$$

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

First Derivatives (Gradient)

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



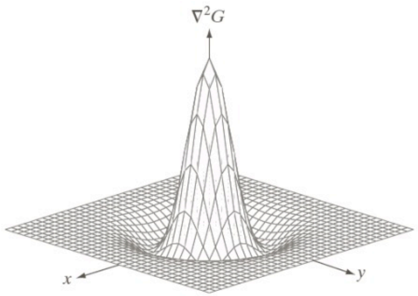


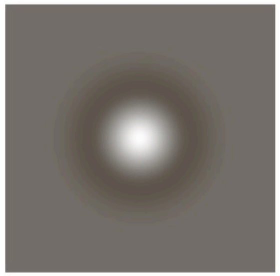
https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/12954/versions/7/previews/googleearth/html/ge_quiver.html

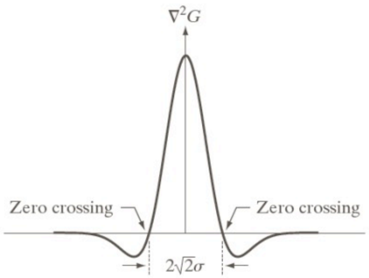
https://en.wikipedia.org/wiki/Gradient

LOG (Laplacian of Gaussian) Operator

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







0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

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

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

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 $\sigma^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2-\sigma_2^2}\ln[\frac{\sigma_1^2}{\sigma_2^2}]$