CSCE274 Robotic Applications and Design
Fall 2021
Computer Vision

Ioannis REKLEITIS, Ibrahim SALMAN

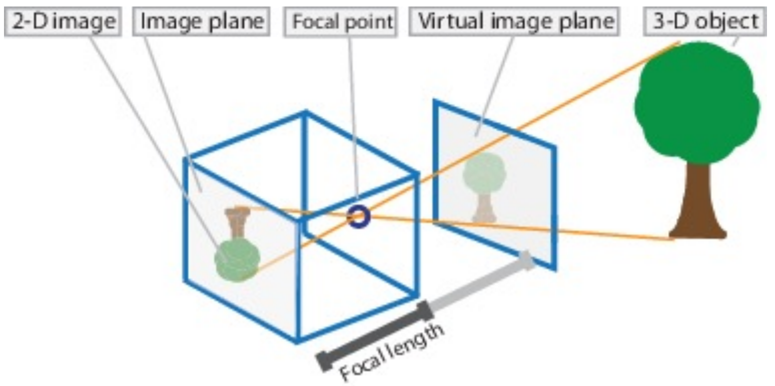
Computer Science and Engineering

University of South Carolina

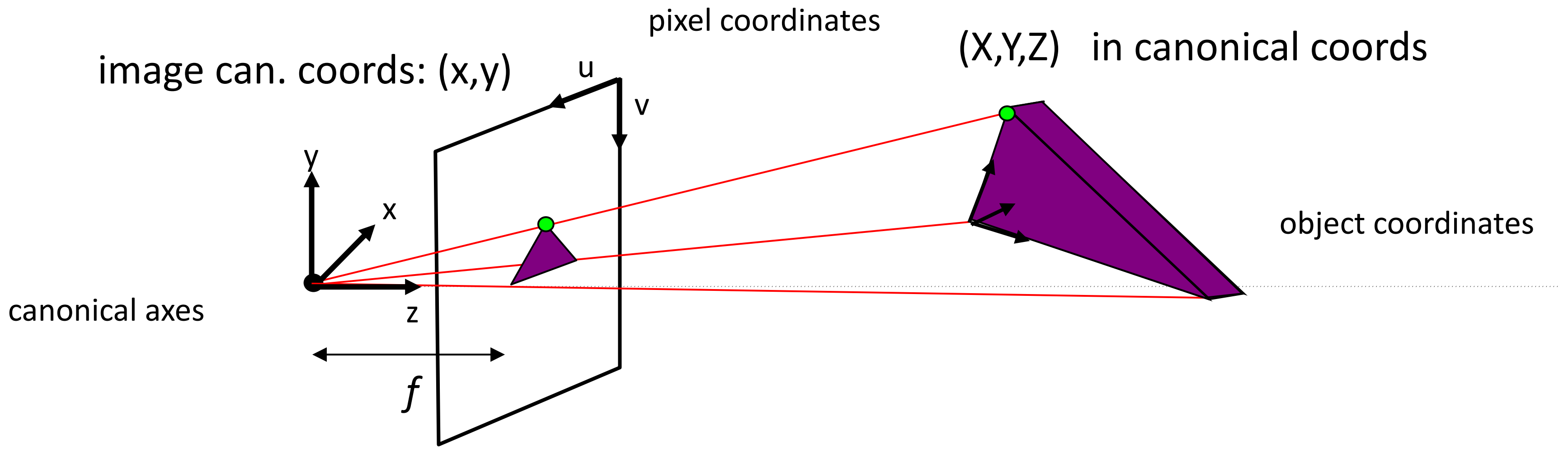
yiannisr@cse.sc.edu

Pinhole camera model
(From the Sensors Lecture)

- Pinhole camera model describes the *relationship* between the coordinates of 3D points of objects in the world and its projection onto the image plane of an ideal pinhole camera

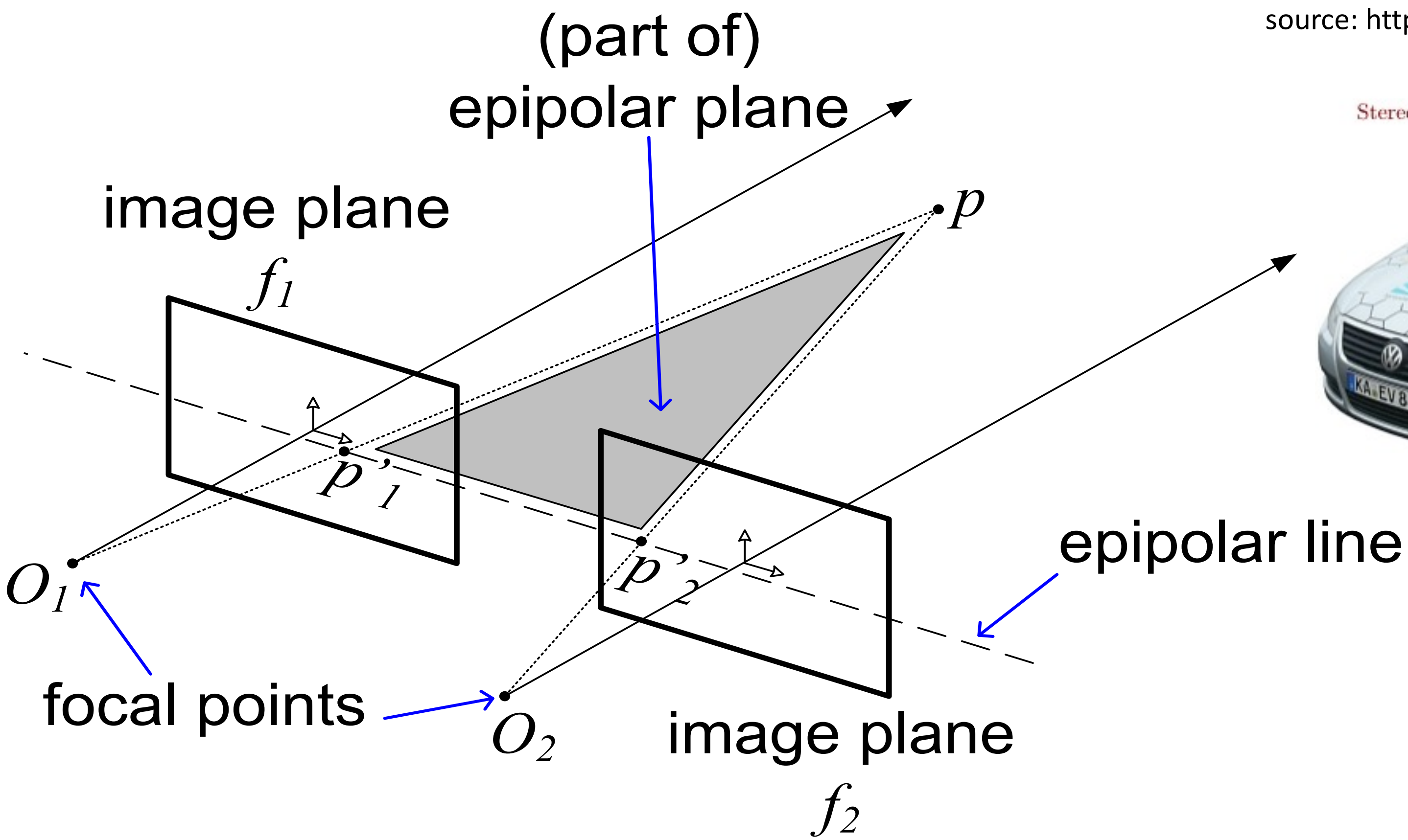


Source: mathworks.com



Stereo vision

- Stereo vision to recover depth information



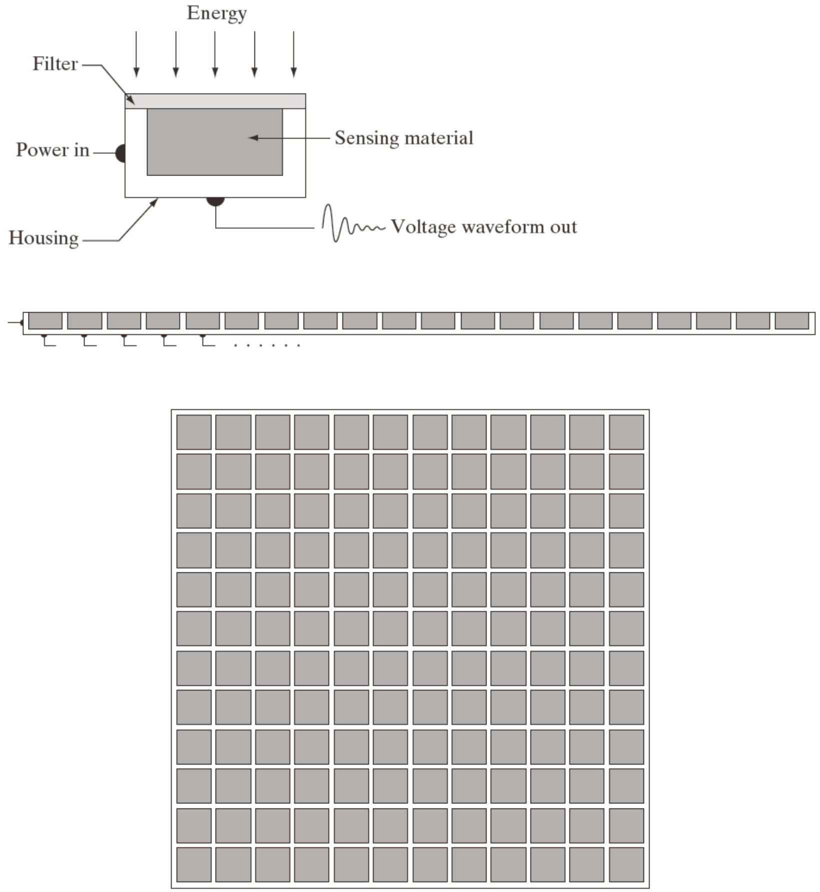
source: http://www.cvlibs.net/datasets/kitti

Image Sensing and Acquisition

Illumination energy → digital images

Incoming energy is transformed into a voltage

Digitizing the response



a
b
c

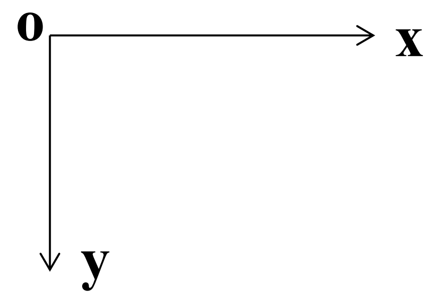
FIGURE 2.12
(a) Single imaging sensor.
(b) Line sensor.
(c) Array sensor.

Slides courtesy of Prof. Yan Tong

A (2D) Image

An image = a 2D function *f*(*x*,*y*) where

- *x* and *y* are spatial coordinates
- *f*(*x*,*y*) is the intensity or gray level



A digital image:

- *x*, *y*, and *f*(*x*,*y*) are all finite
- For example $x \in \{1,2,\ldots,M\}$, $y \in \{1,2,\ldots,N\}$

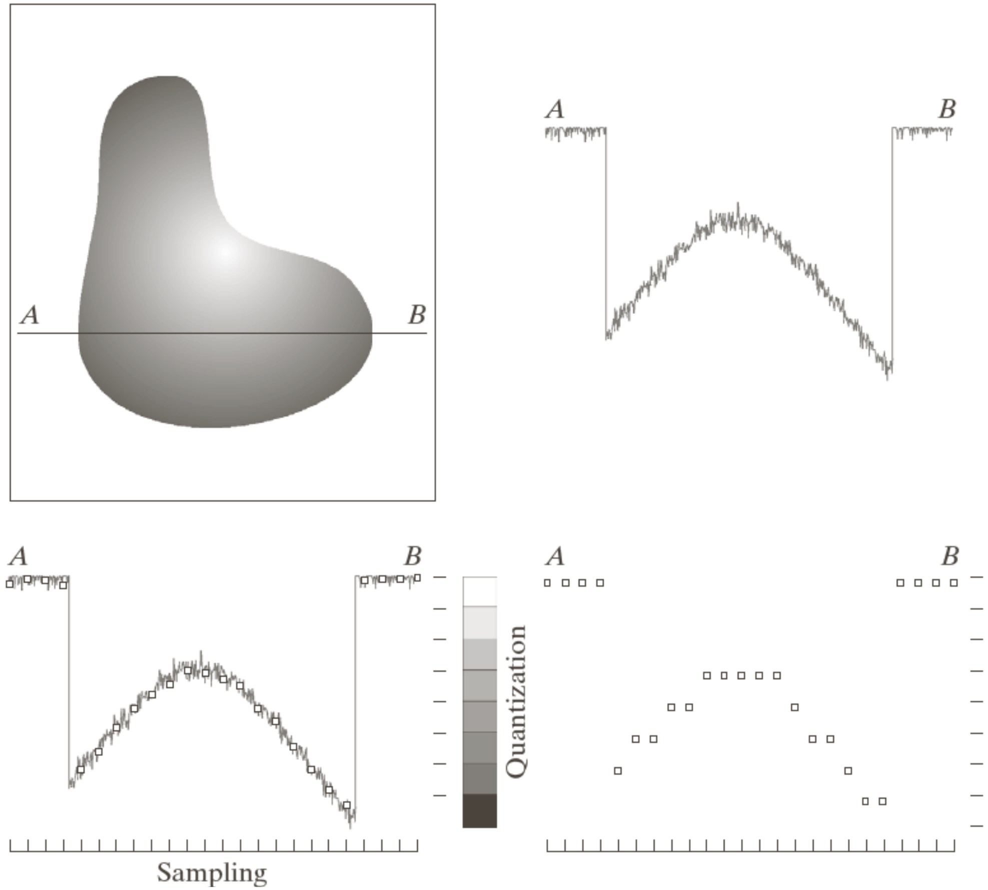
$$f(x,y) \in \{0,1,2,\ldots,255\}$$

Digital image processing → processing digital images by means of a digital computer

Each element (*x*,*y*) in a digital image is called a pixel (picture element)

Slides courtesy of Prof. Yan Tong

Image Sampling and Quantization

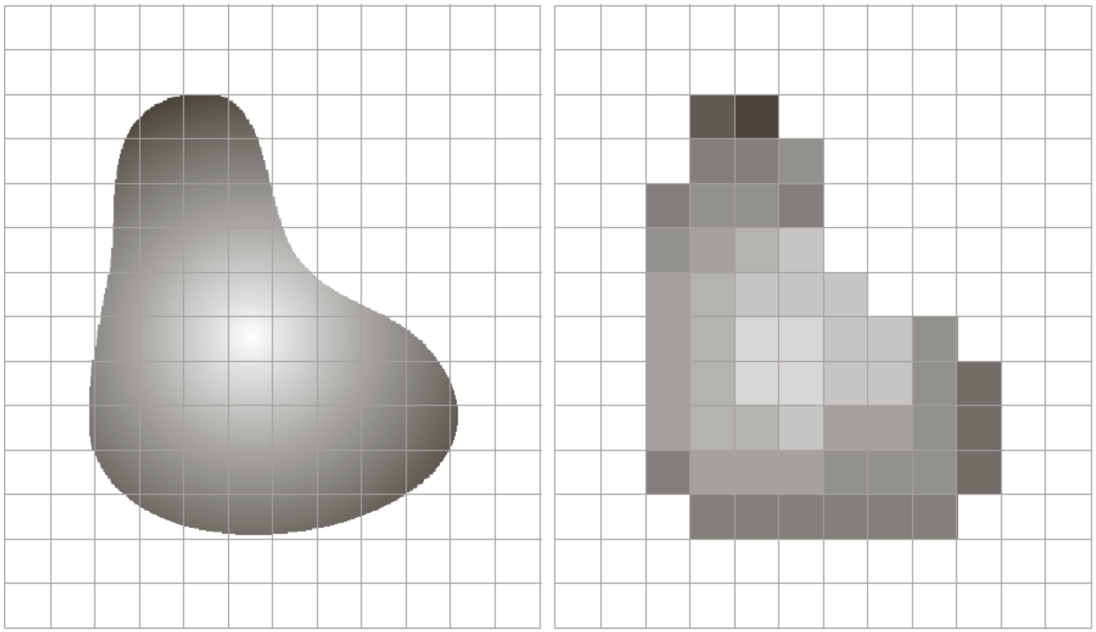


Sampling: Digitizing the coordinate values (usually determined by sensors)

Quantization: Digitizing the amplitude values

Slides courtesy of Prof. Yan Tong

Image Sampling and Quantization in a Sensor Array



CCD array

a b

FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

Slides courtesy of Prof. Yan Tong

Review: Linear Algebra

- Matrix Addition
- Matrix Multiplication **A*B**
- Matrix-Vector Multiplication **A***v
- Matrix Transpose $\mathbf{A}^T$, $(\mathbf{AB})^T = \mathbf{A}^T\mathbf{B}^T$
- Matrix Inverse $\mathbf{A}^{-1}$
- Identity Matrix $\mathbf{I}_3$=

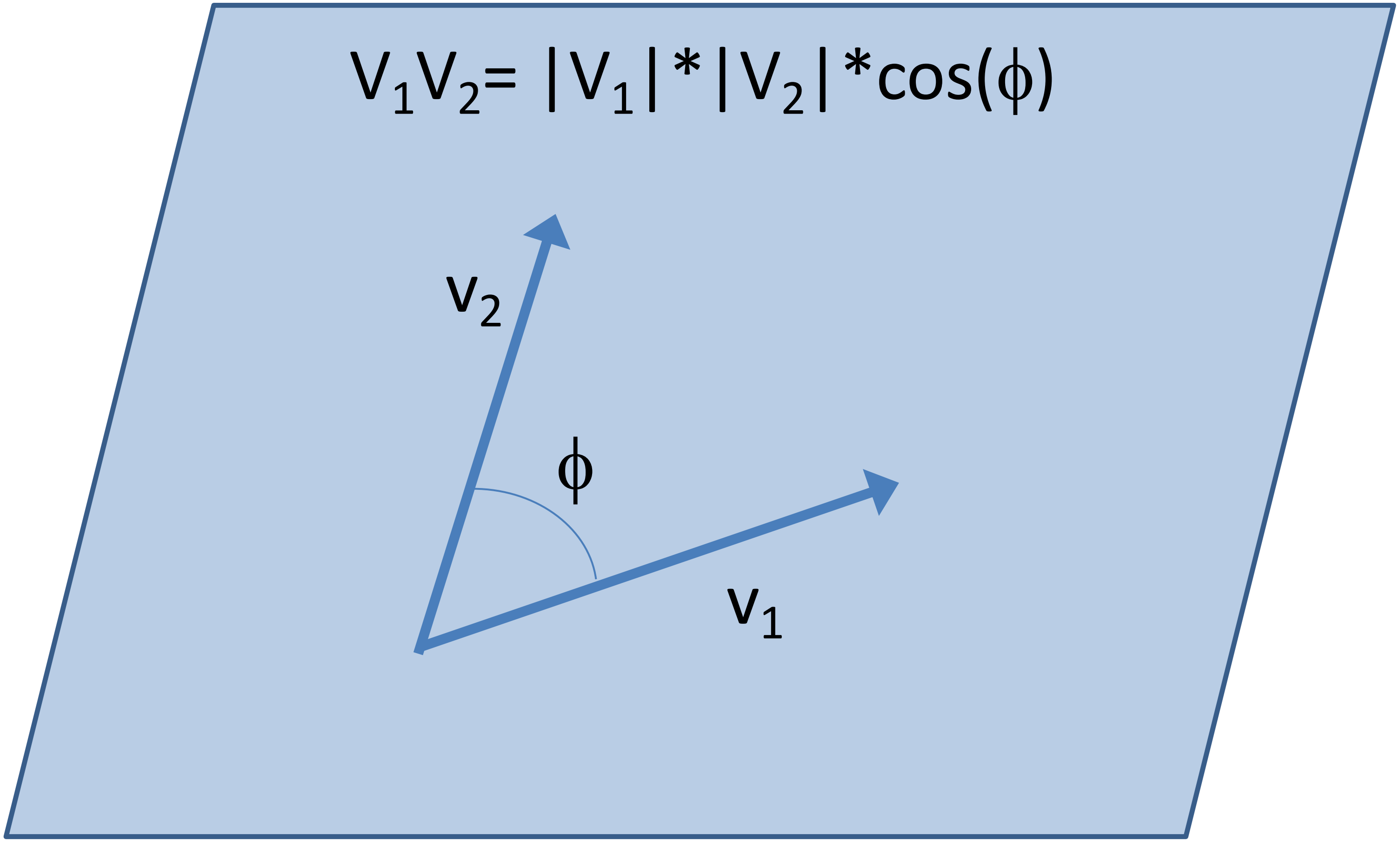
1	**0**	**0**
0	**1**	**0**
0	**0**	**1**

$$\begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{bmatrix}^{\mathrm{T}} = \begin{bmatrix}1 & 3 & 5\\ 2 & 4 & 6\end{bmatrix}$$

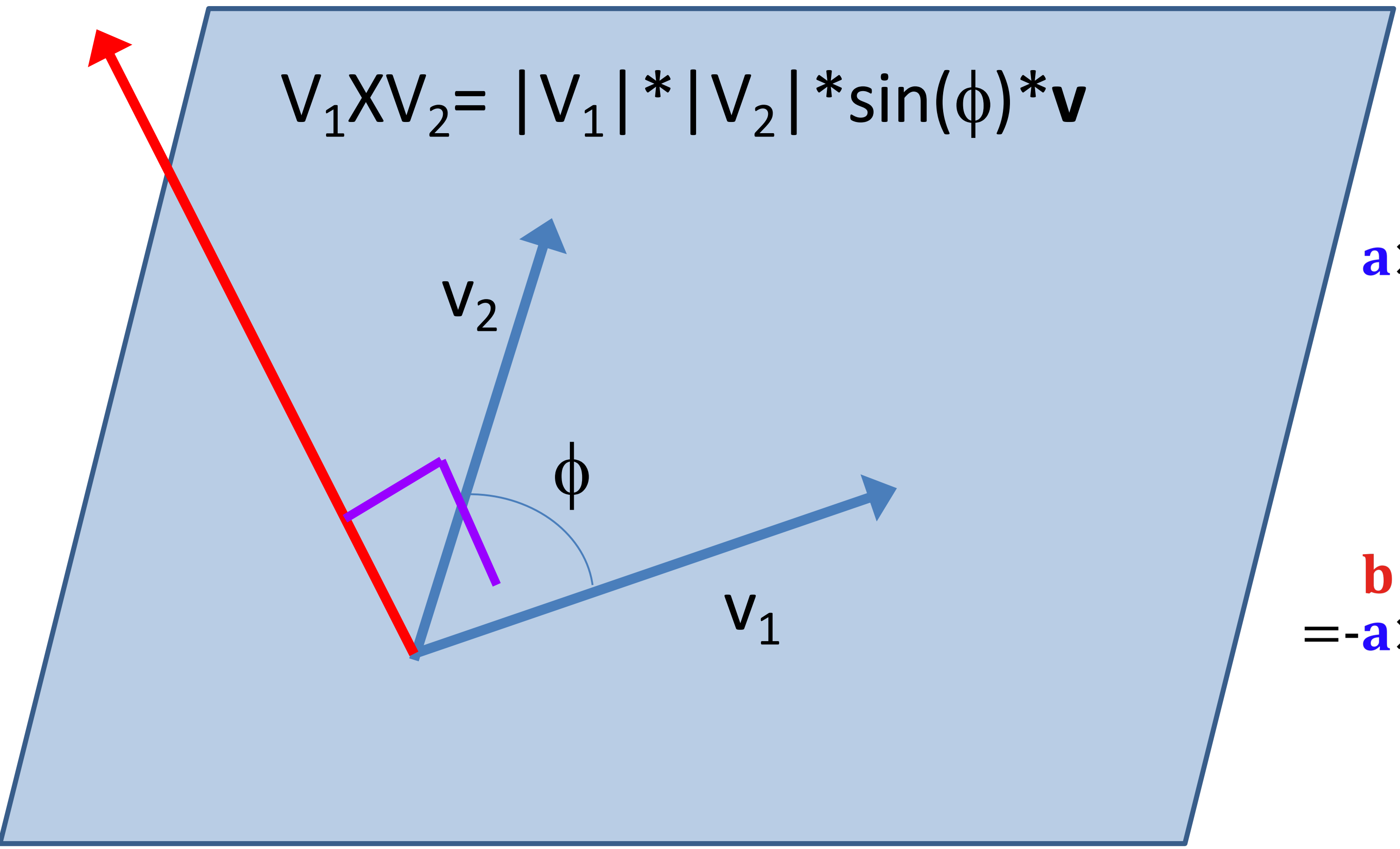
$$\mathbf{AB} = \begin{pmatrix} a & b & c \\ p & q & r \\ u & v & w \end{pmatrix} \begin{pmatrix} \alpha & \beta & \gamma \\ \lambda & \mu & \nu \\ \rho & \sigma & \tau \end{pmatrix} = \begin{pmatrix} a\alpha + b\lambda + c\rho & a\beta + b\mu + c\sigma & a\gamma + b\nu + c\tau \\ p\alpha + q\lambda + r\rho & p\beta + q\mu + r\sigma & p\gamma + q\nu + r\tau \\ u\alpha + v\lambda + w\rho & u\beta + v\mu + w\sigma & u\gamma + v\nu + w\tau \end{pmatrix},$$

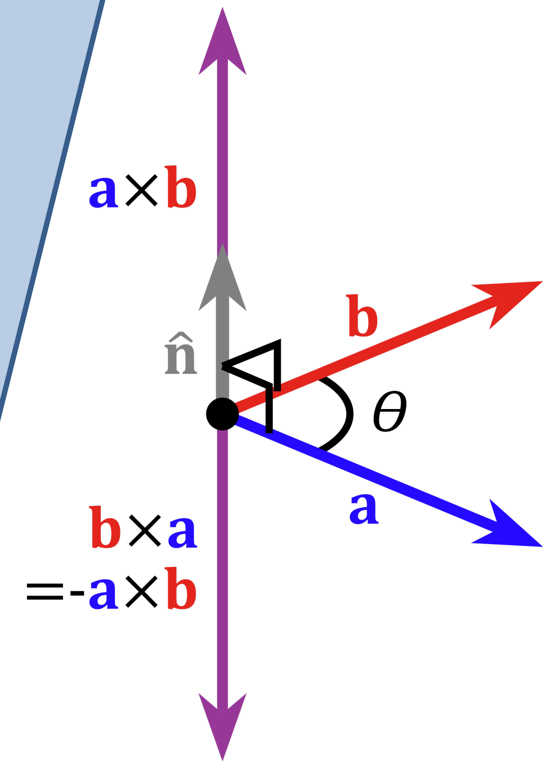
Dot Product

$$V_1 V_2 = |V_1| * |V_2| * \cos(\phi)$$



Cross Product





Single pixel operations

- Determined by
  - Transformation function T
  - Input intensity value
- Not depend on other pixels and position

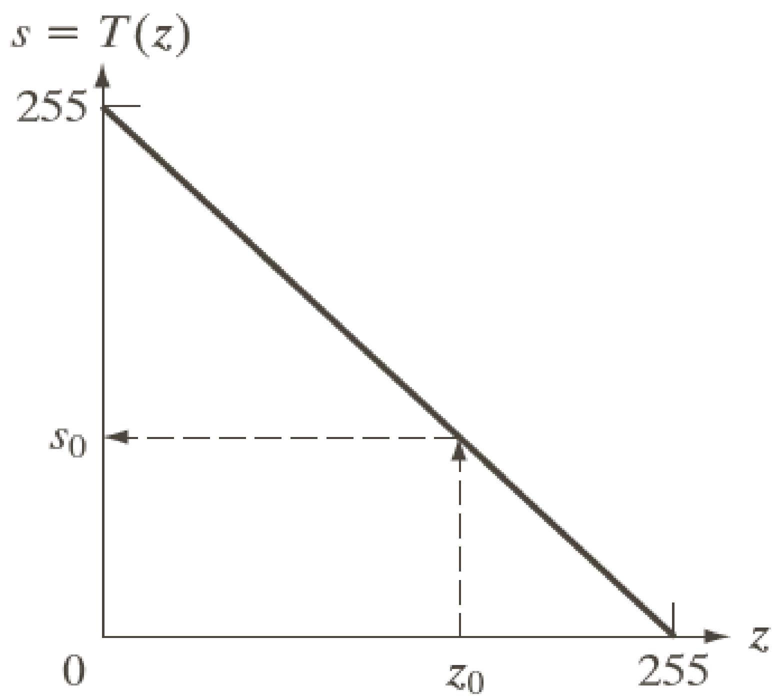
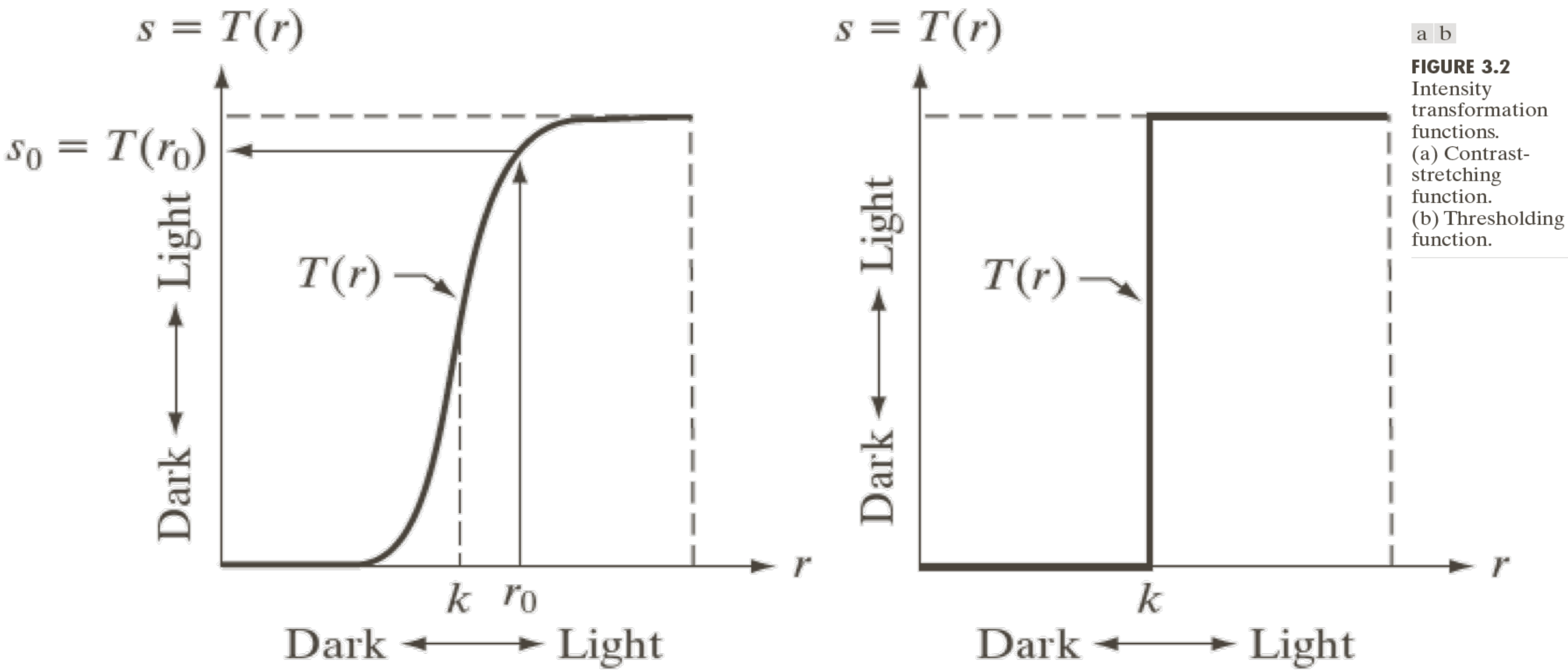


FIGURE 2.34 Intensity transformation function used to obtain the negative of an 8-bit image. The dashed arrows show transformation of an arbitrary input intensity value z_0 into its corresponding output value s_0.

Slides courtesy of Prof. Yan Tong

Intensity Transformation
Image Enhancement

Contrast stretch



Soft thresholding (logistic function)

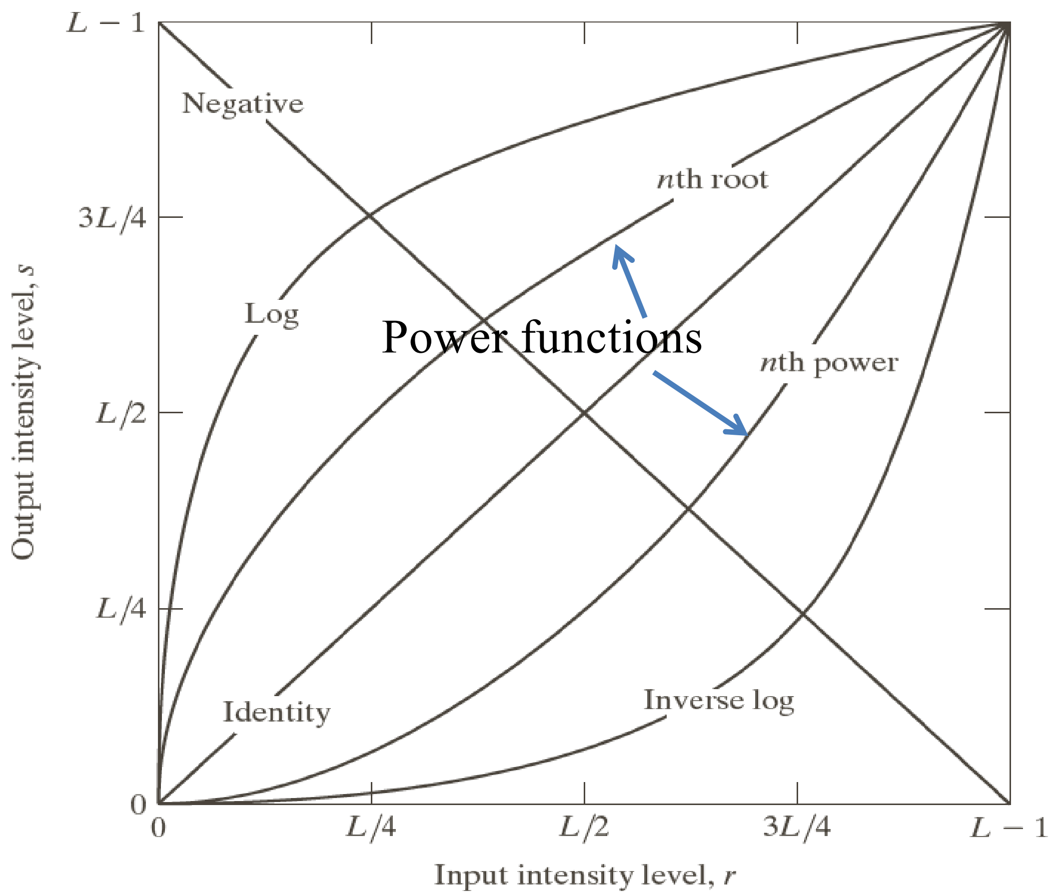
Hard thresholding (step function)

$$\sim s = \frac{1}{1 + e^r}$$

Slides courtesy of Prof. Yan Tong

Thresholded image

Basic Intensity Transformation Functions



Log function:

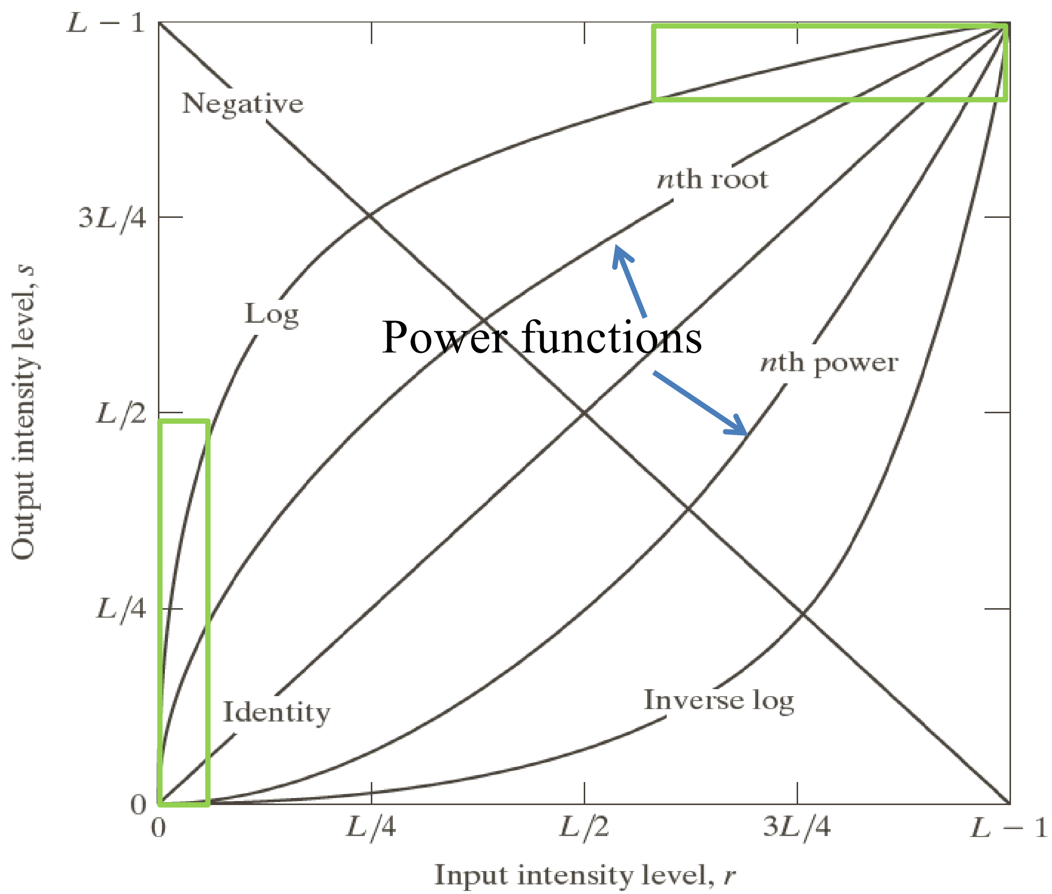
$$s = c\log(1 + r) \quad r \geq 0$$

Inverse log function:

$$s = c\log^{-1}(r)$$

FIGURE 3.3 Some basic intensity transformation functions. All curves were scaled to fit in the range shown.

Basic Intensity Transformation Functions



Log function:

$$s = c\log(1 + r) \quad r \geq 0$$

Stretch low intensity levels

Compress high intensity levels

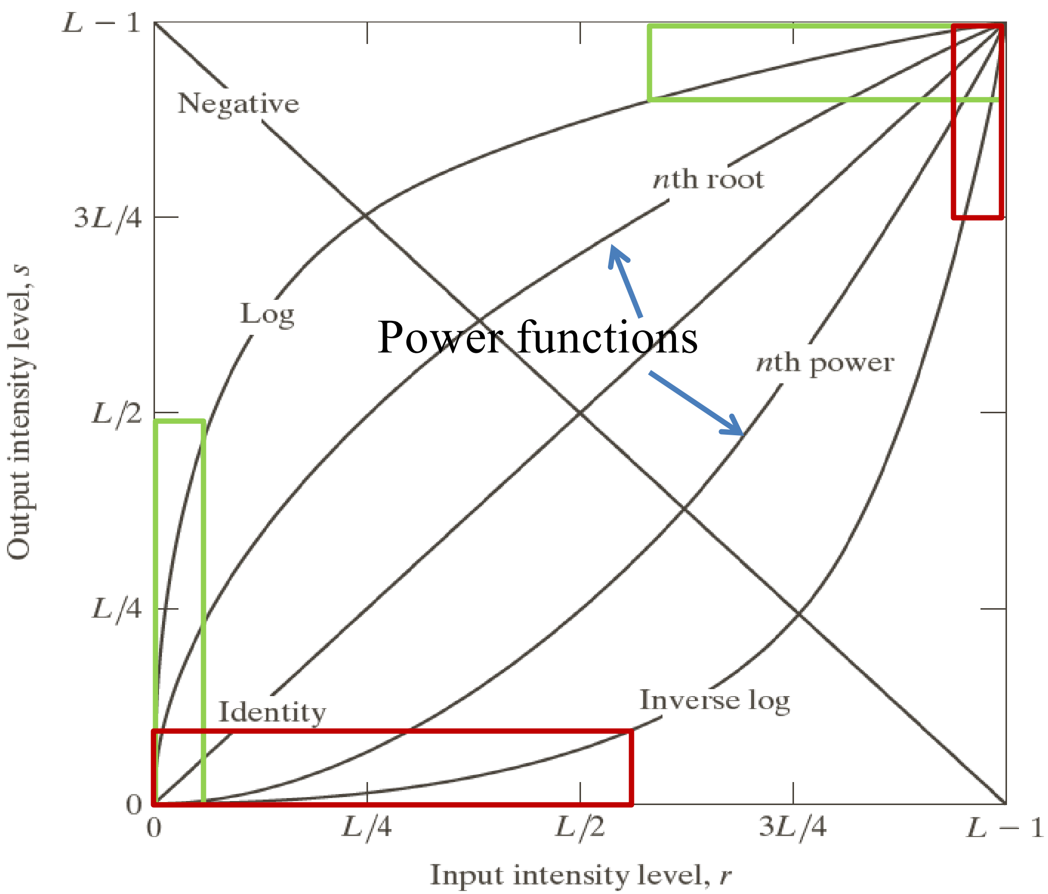
Inverse log function:

$$s = c\log^{-1}(r)$$

FIGURE 3.3 Some basic intensity transformation functions. All curves were scaled to fit in the range shown.

Slides courtesy of Prof. Yan Tong

Basic Intensity Transformation Functions



Log function:

$$s = c\log(1+r) \quad r \geq 0$$

Stretch low intensity levels

Compress high intensity levels

Inverse log function:

$$s = c\log^{-1}(r)$$

Stretch high intensity levels

Compress low intensity levels

FIGURE 3.3 Some basic intensity transformation functions. All curves were scaled to fit in the range shown.

Slides courtesy of Prof. Yan Tong

Power-Law (Gamma) Transformations

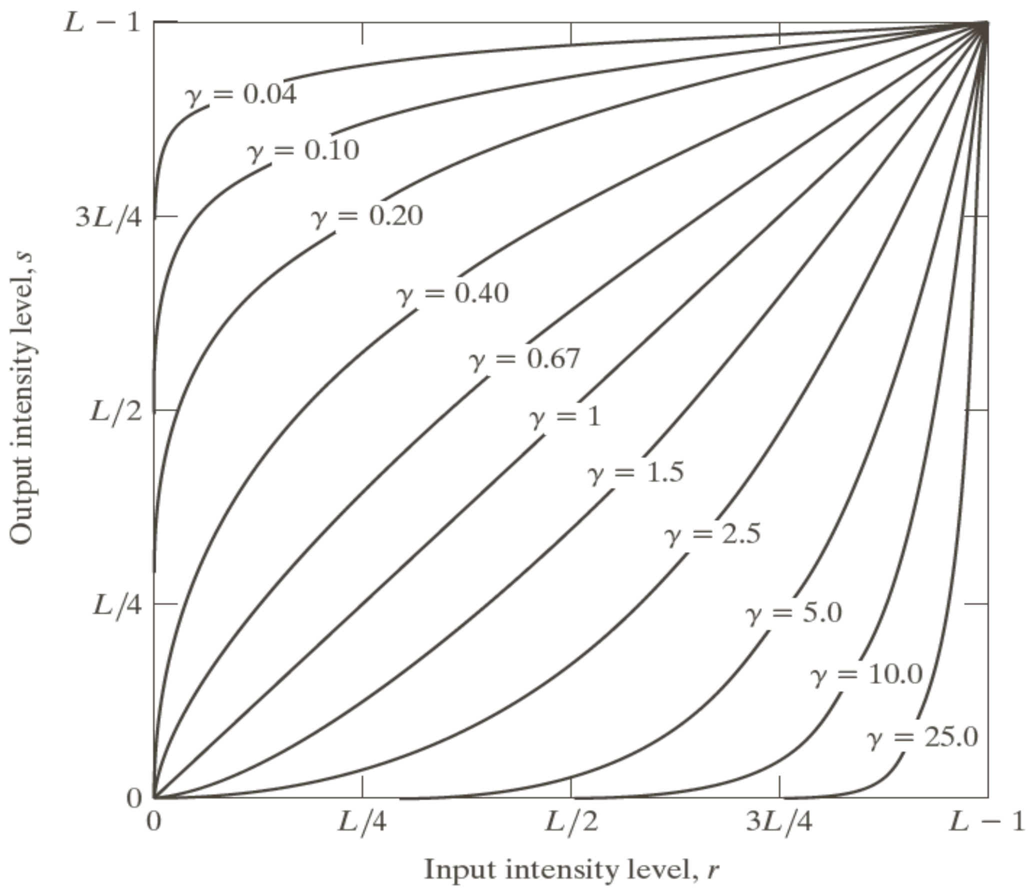


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases). All curves were scaled to fit in the range shown.

$$s = cr^{\gamma}$$

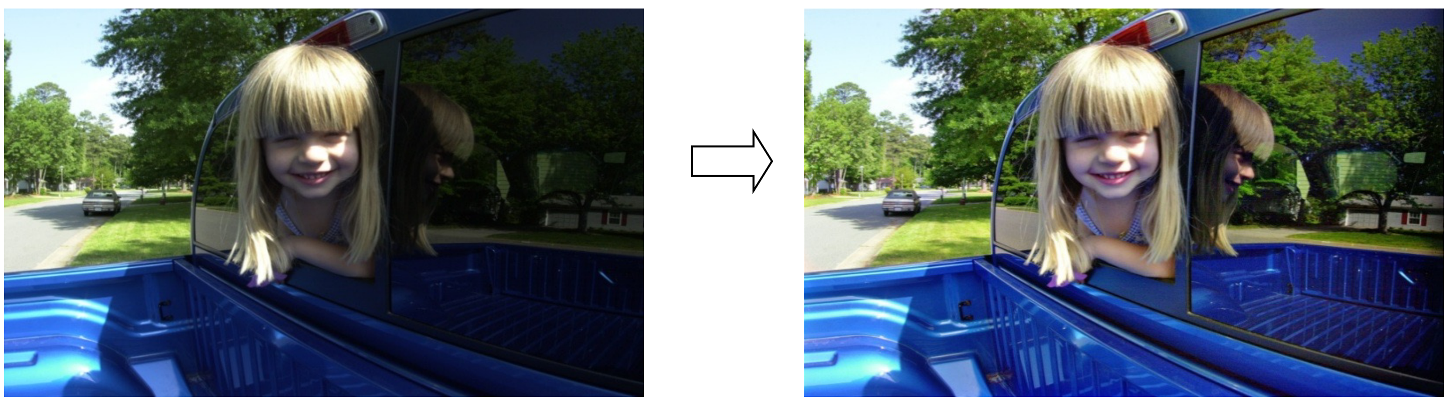
- More versatile than log transformation
- Performed by a lookup table

Slides courtesy of Prof. Yan Tong

LookUp Table Operations

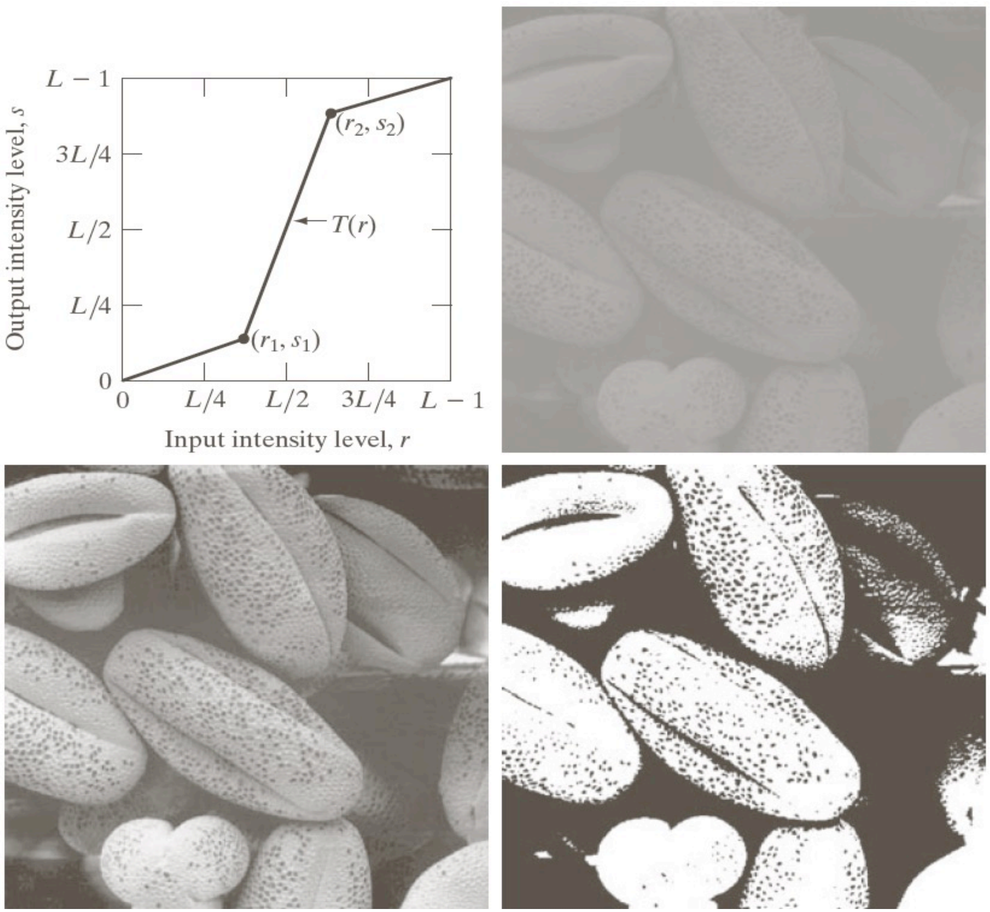
- Look Up Table: LUT[i]=c*i^gamma;
- NI[i,j]=LUT[I[i,j]];

Image Enhancement Using Gamma Correction



Slides courtesy of Prof. Yan Tong

Piecewise-Linear Transformation Functions: Contrast Stretching

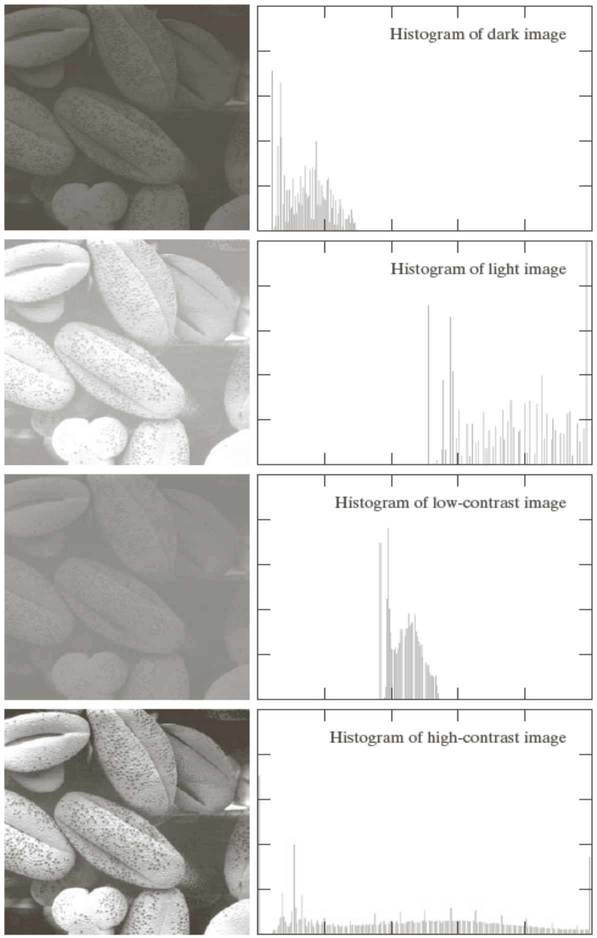


a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Slides courtesy of Prof. Yan Tong

Histogram Processing



Histogram

$$h(r_k) = n_k$$

Normalized histogram

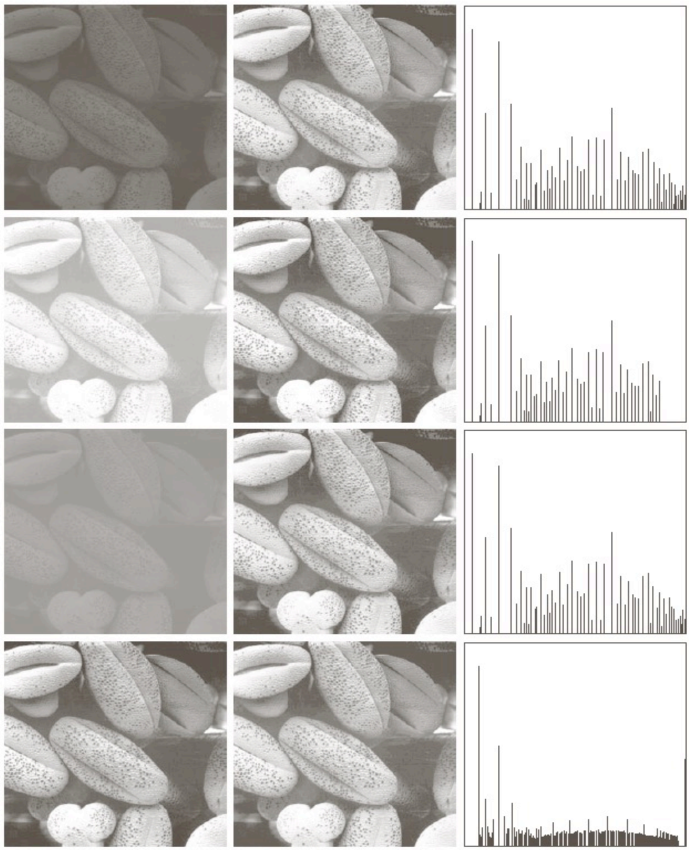
$$p(r_k) = n_k / MN$$

$$\sum_{k=0}^{255} p(r_k) = 1$$

FIGURE 3.16 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms.

Slides courtesy of Prof. Yan Tong

Examples



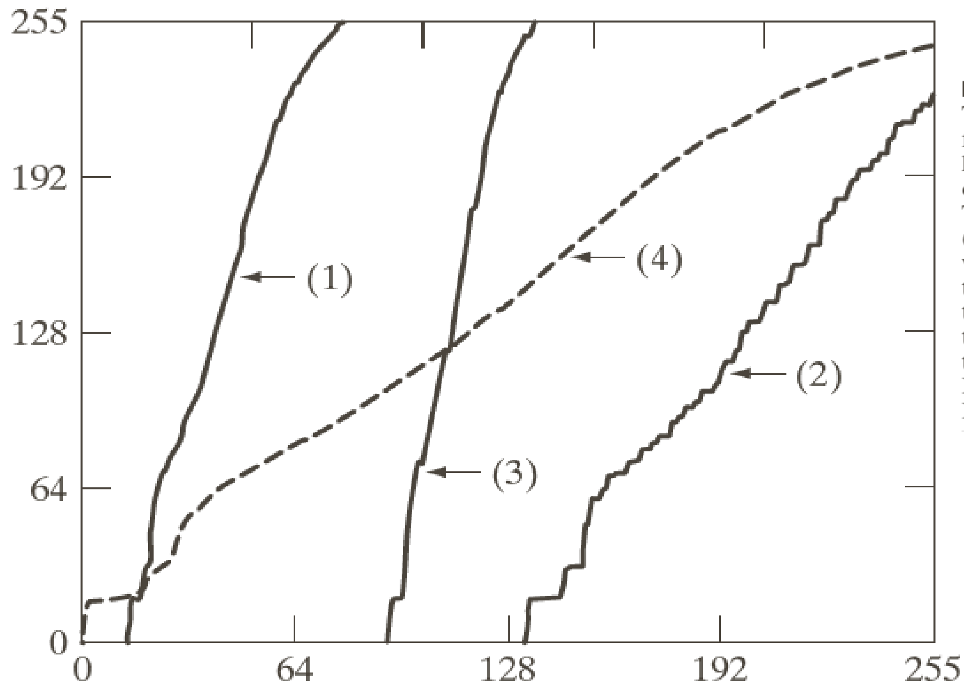
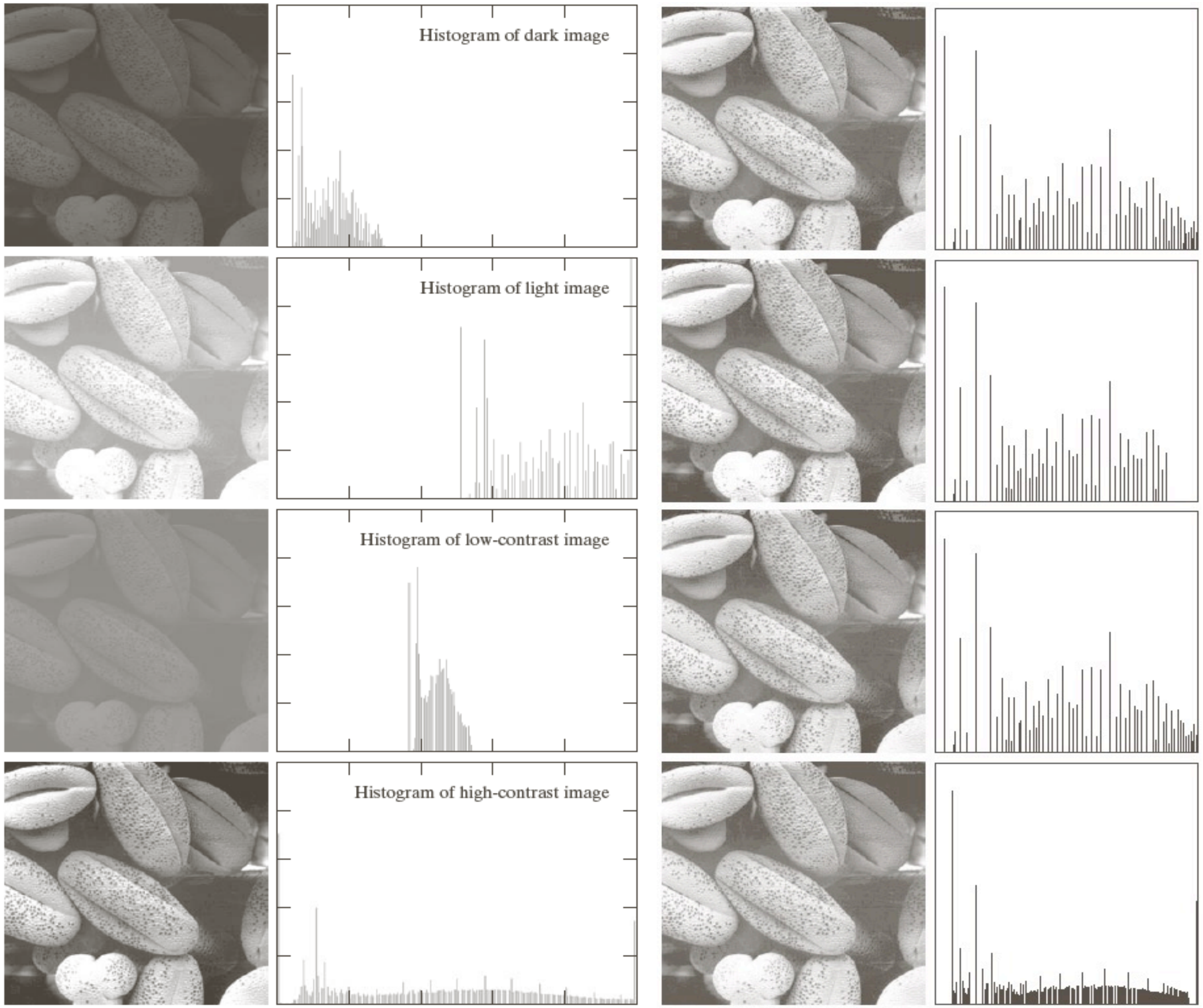


FIGURE 3.21 Transformation functions for histogram equalization. Transformations (1) through (4) were obtained from the histograms of the images (from top to bottom) in the left column of Fig. 3.20 using Eq. (3.3-8).

FIGURE 3.20 Left column: images from Fig. 3.16. Center column: corresponding histogram-equalized images. Right column: histograms of the images in the center column.

Slides courtesy of Prof. Yan Tong

Examples



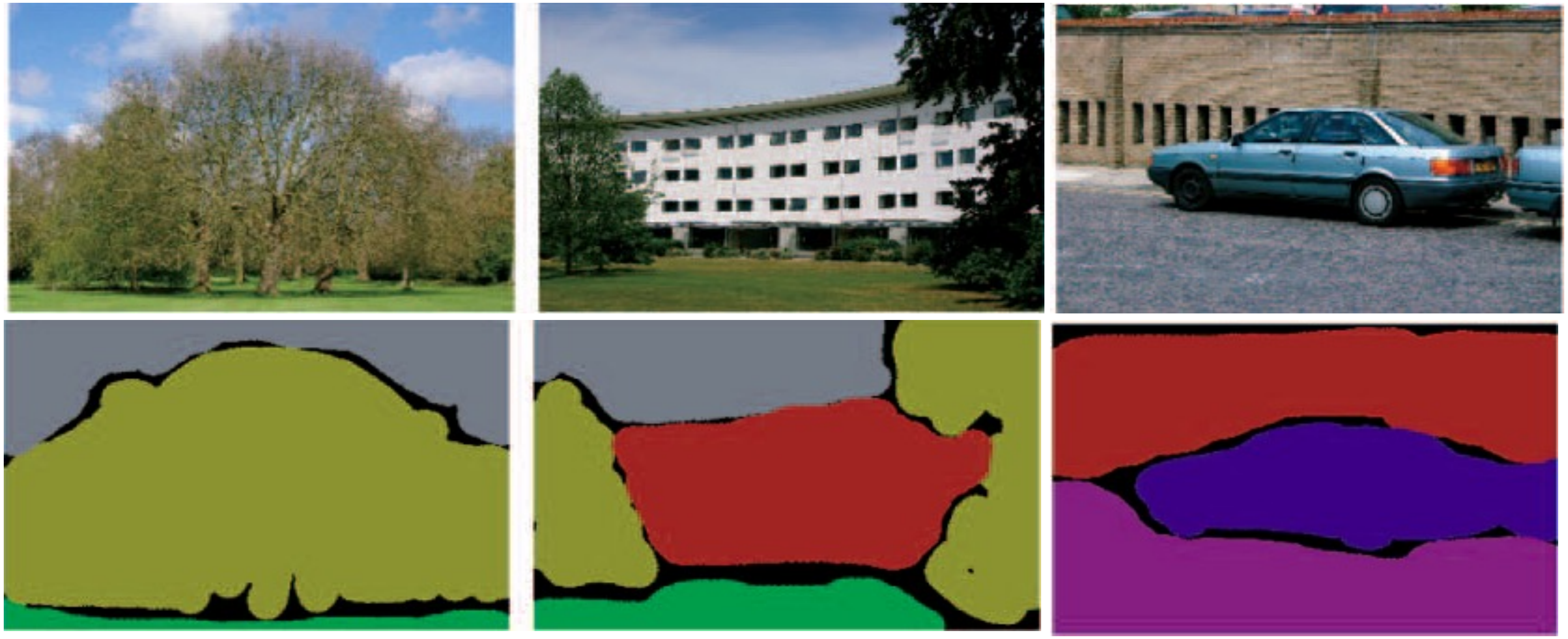
Slides courtesy of Prof. Yan Tong

Additional Information

- Mean value of intensity
- Median Value of intensity
- Max and Min Value of intensity

Image Segmentation

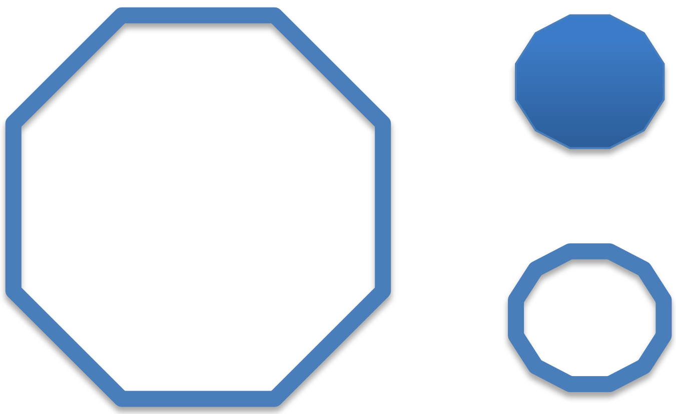
A process that partitions R into subregions $R_1, R_2, \ldots, R_n$



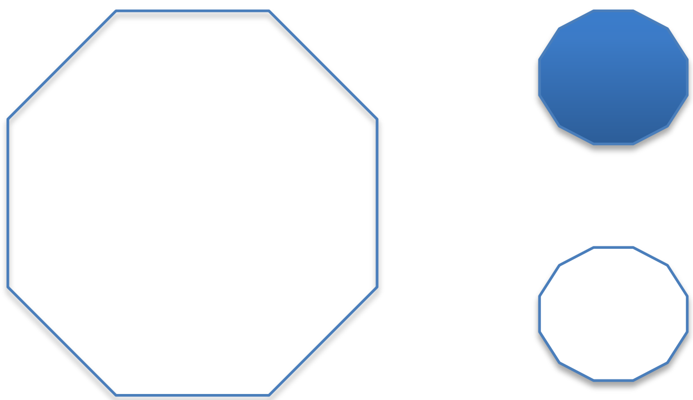
Microsoft multiclass segmentation data set

Slides courtesy of Prof. Yan Tong

Dilation and Erosion



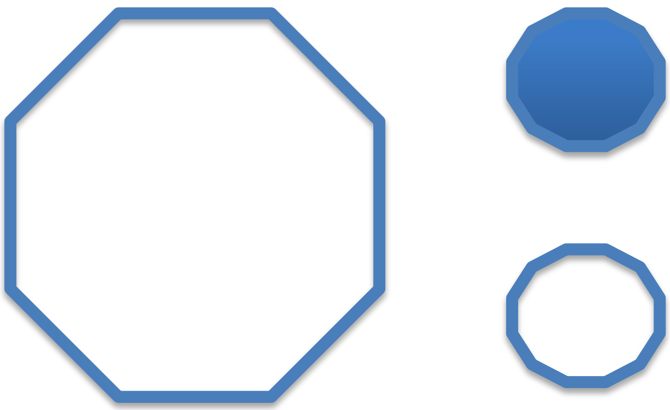
Erosion



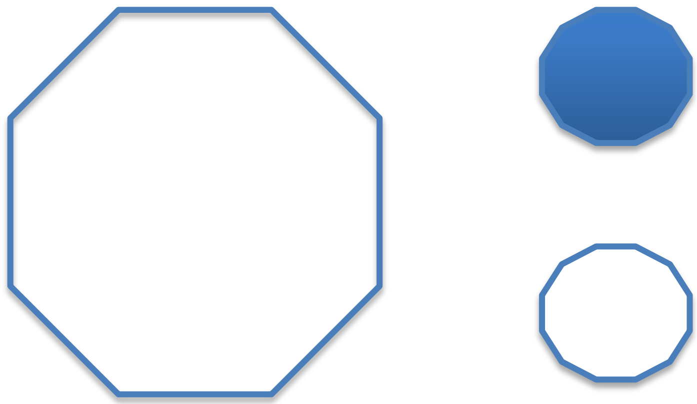
Dilation

Usually on binary images, after thresholding and/or segmentation

Dilation and Erosion

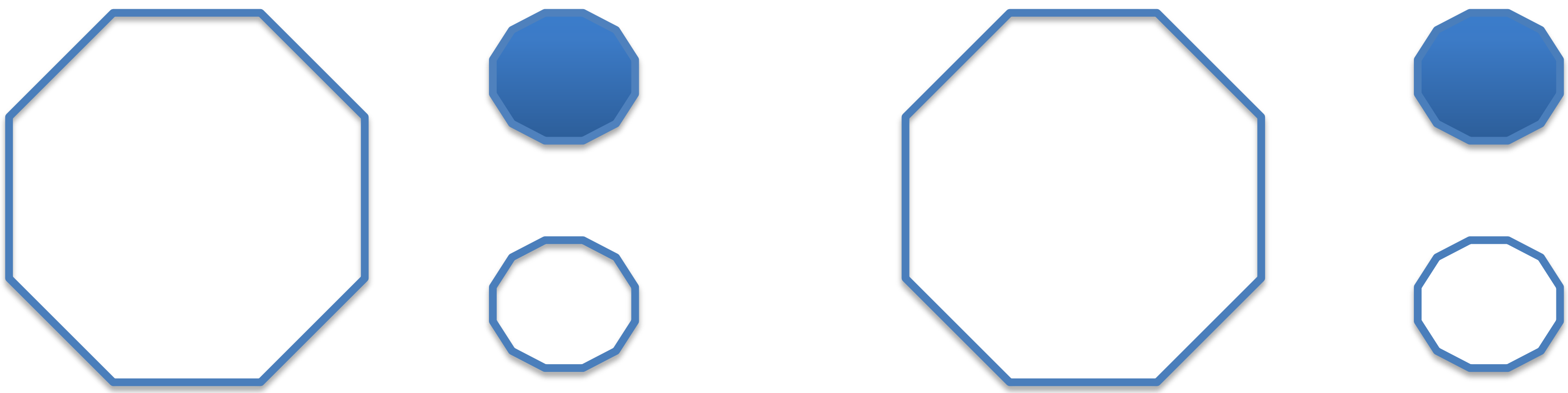


Erosion



Dilation

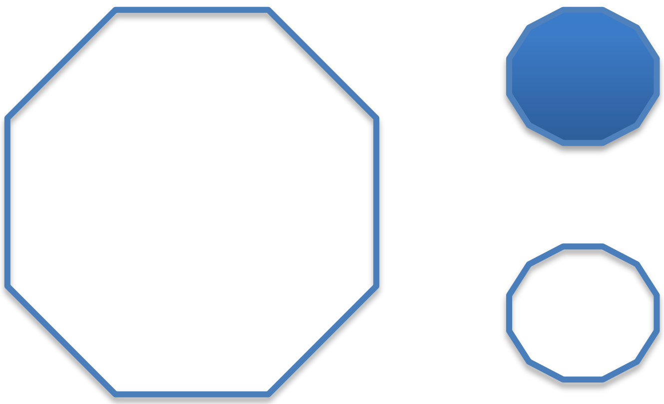
Dilation and Erosion



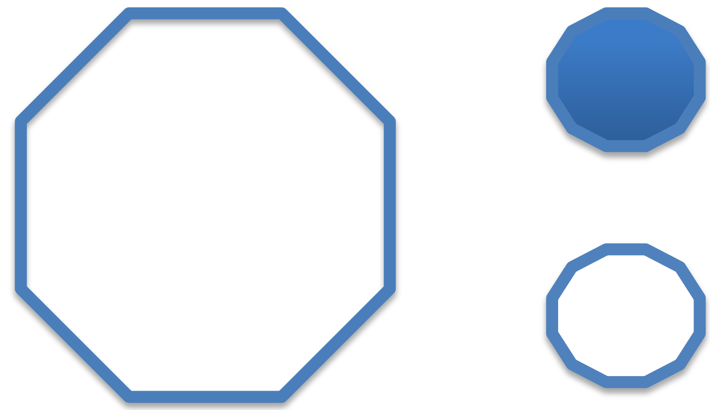
Erosion

Dilation

Dilation and Erosion

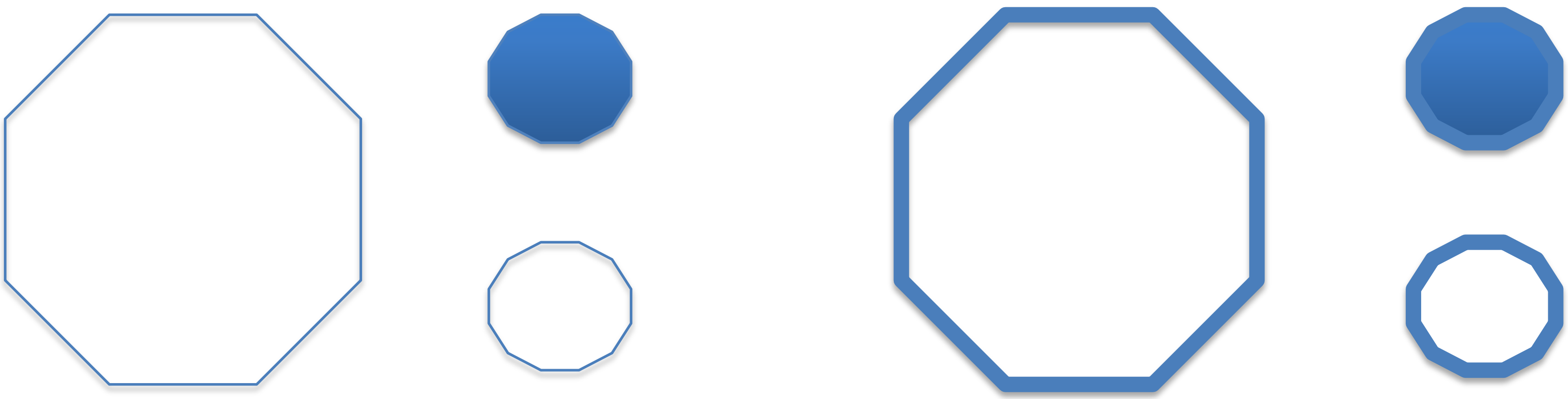


Erosion



Dilation

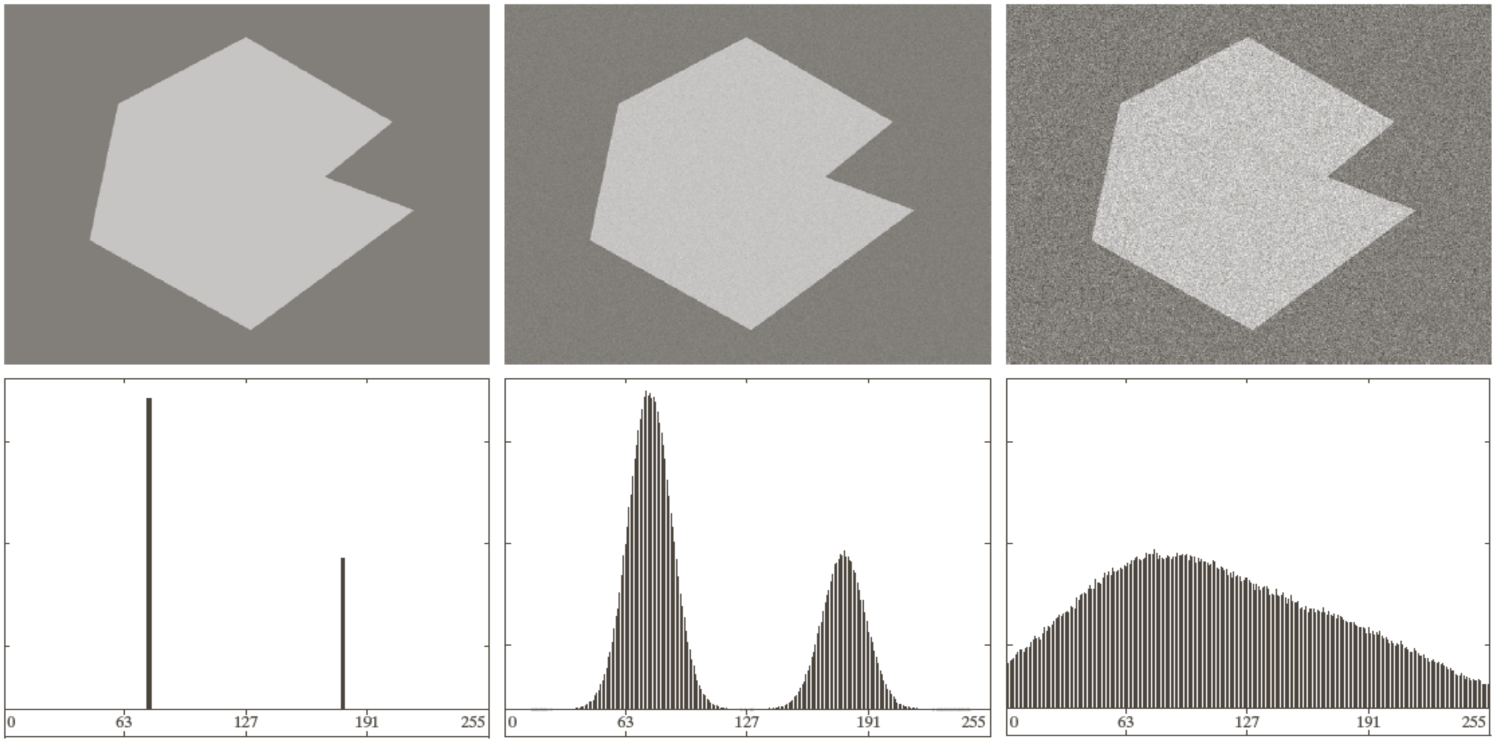
Dilation and Erosion



Erosion

Dilation

The Role of Noise in Image Thresholding

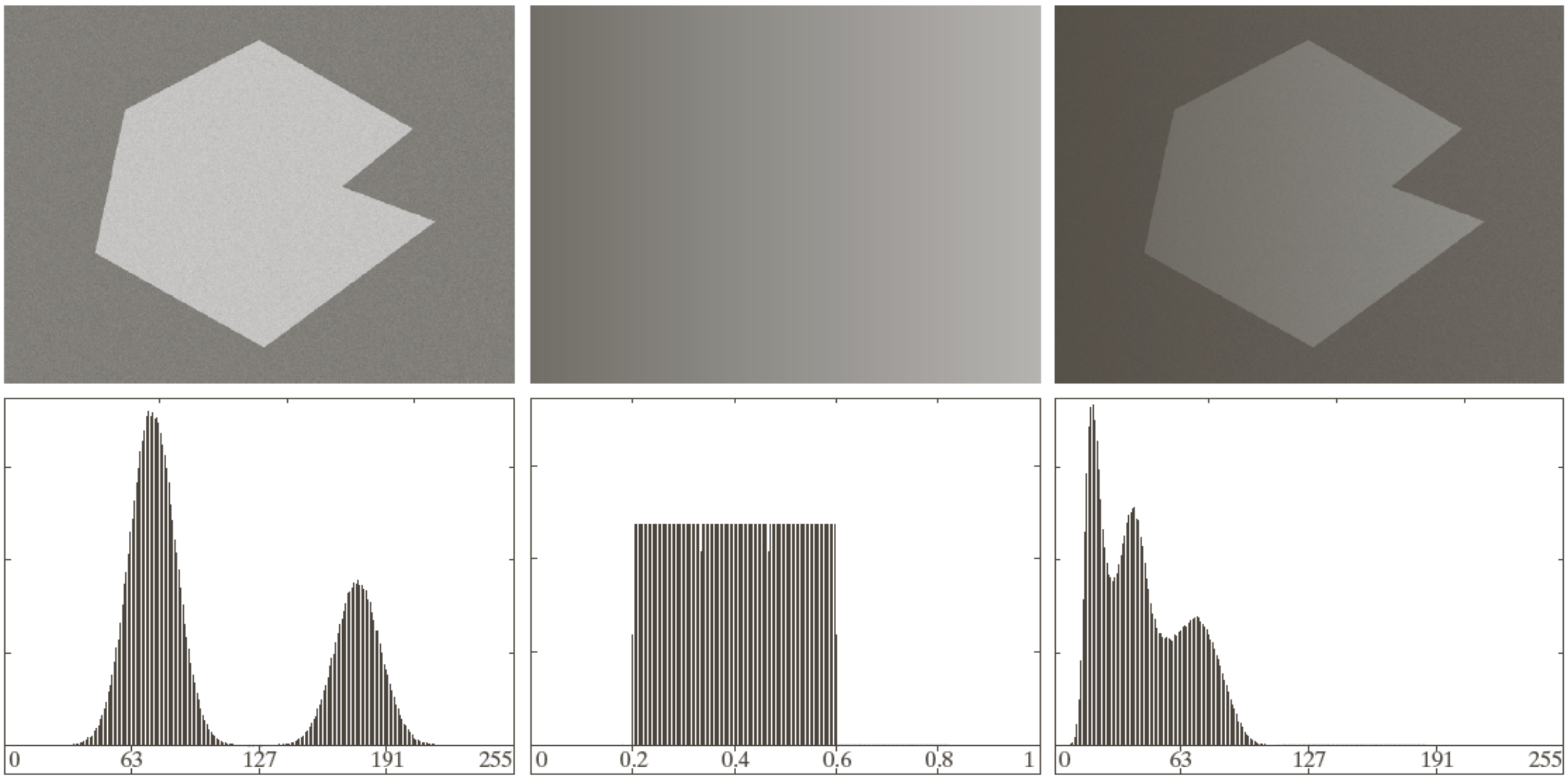


a	b	c
d	e	f

FIGURE 10.36 (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.

Slides courtesy of Prof. Yan Tong

The Role of Illumination in Thresholding



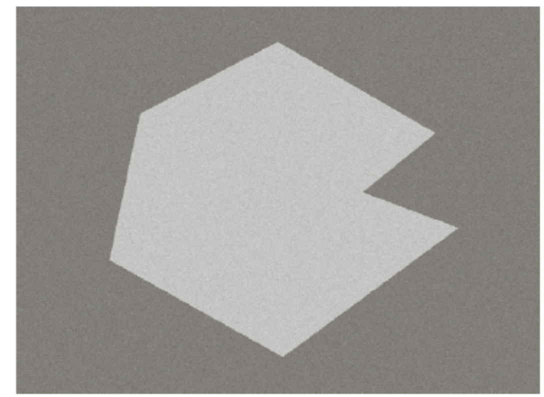
a b c
d e f

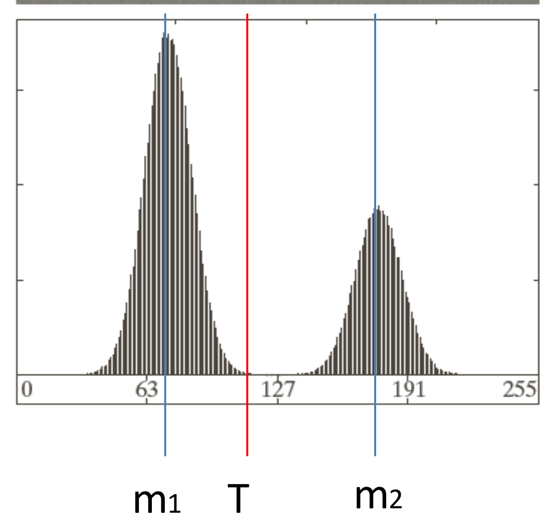
FIGURE 10.37 (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d)–(f) Corresponding histograms.

Slides courtesy of Prof. Yan Tong

How to Pick the Threshold

1. Select an initial estimate for the global threshold, *T*.
2. Segment the image using *T* by producing two groups of pixels
3. Compute the mean of these two groups of pixels, say m_1 and m_2.
4. Update the threshold $T=(m_1 + m_2)/2$
5. Repeat Steps 2 through 4 until convergence

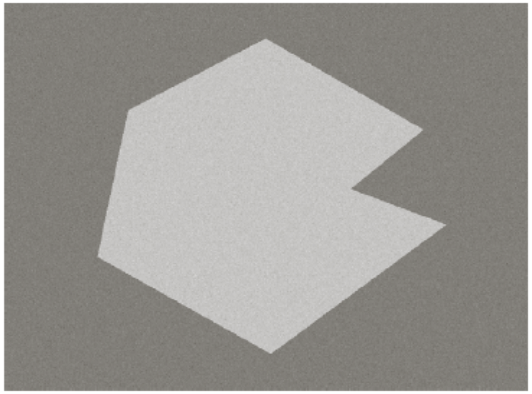


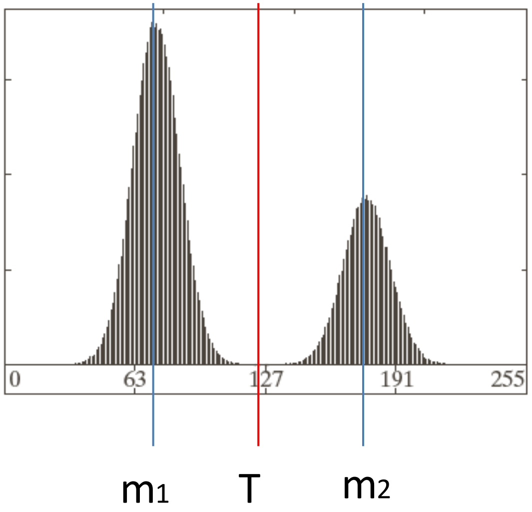


Slides courtesy of Prof. Yan Tong

How to Pick the Threshold

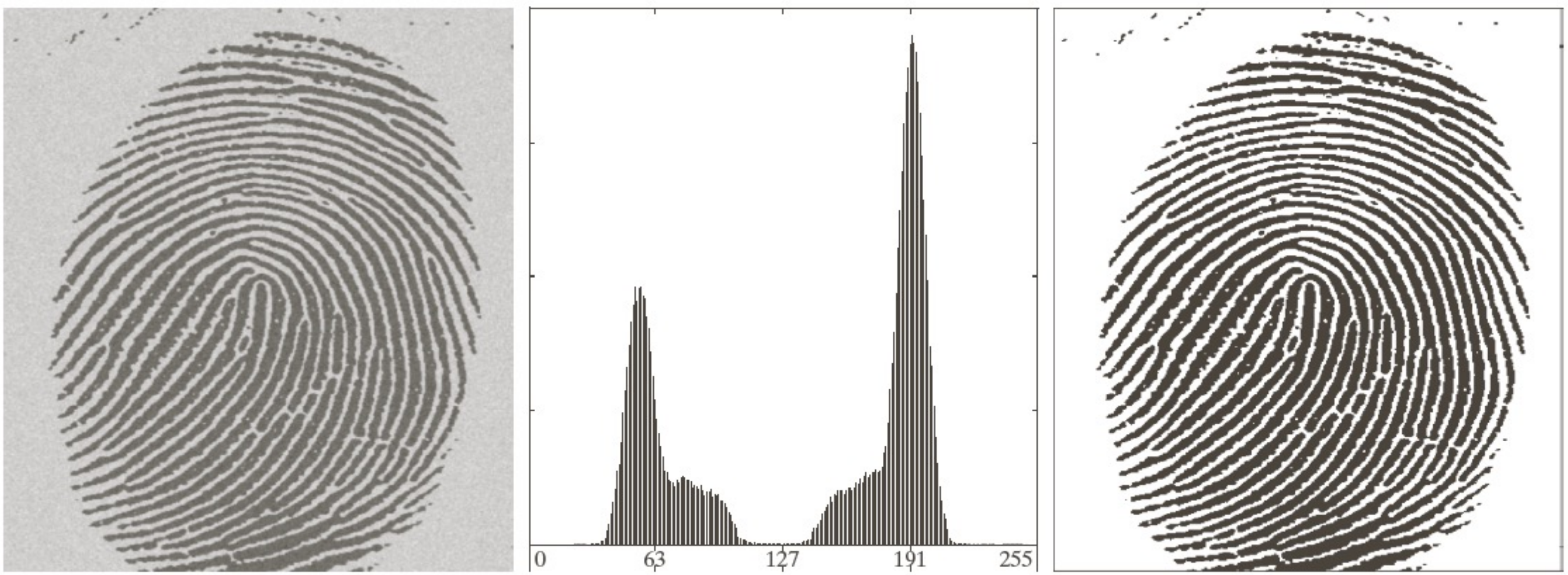
1. Select an initial estimate for the global threshold, *T*.
2. Segment the image using *T* by producing two groups of pixels
3. Compute the mean of these two groups of pixels, say m_1 and m_2.
4. Update the threshold $T=(m_1 + m_2)/2$
5. Repeat Steps 2 through 4 until convergence





Slides courtesy of Prof. Yan Tong

An Example



a b c

FIGURE 10.38 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added for clarity). (Original courtesy of the National Institute of Standards and Technology.)

Slides courtesy of Prof. Yan Tong

An Example (cont.)

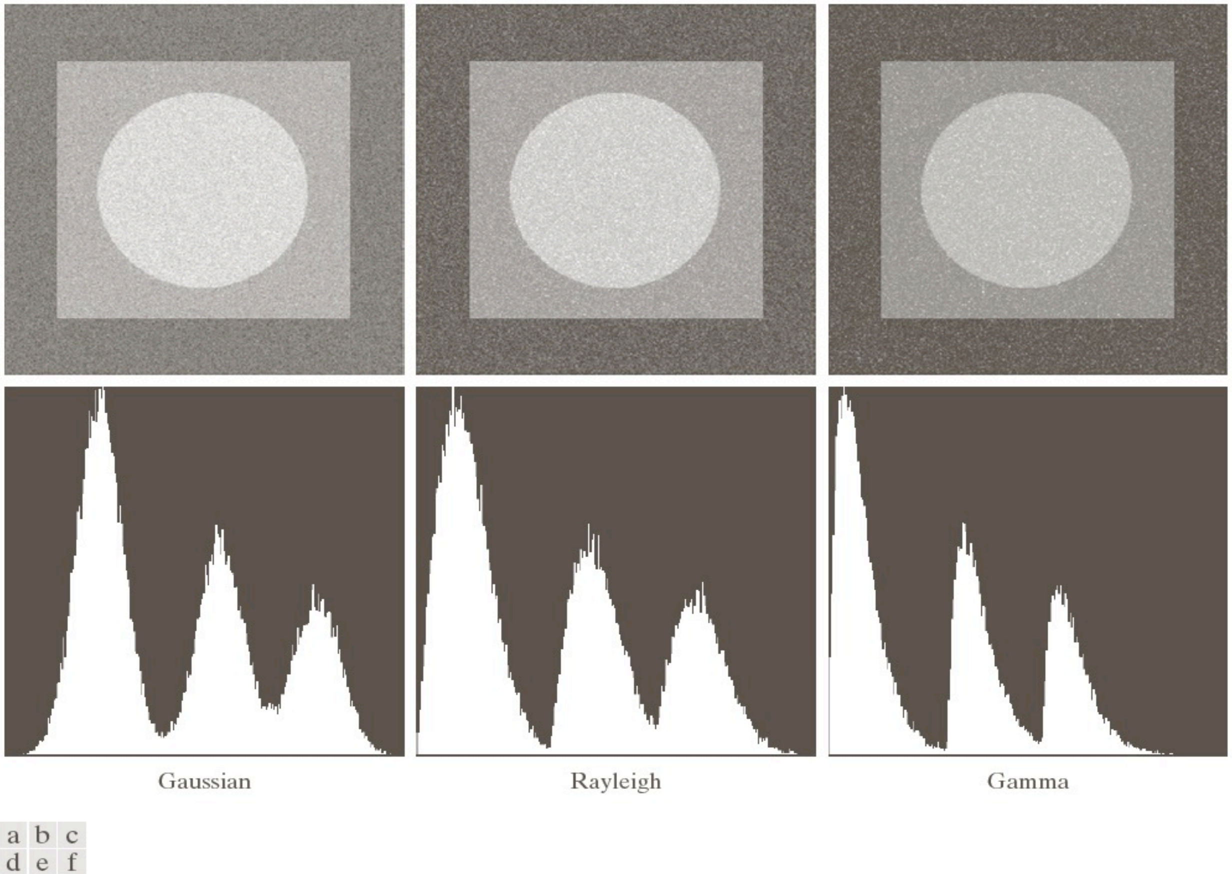


FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

Slides courtesy of Prof. Yan Tong

An Example (cont.)

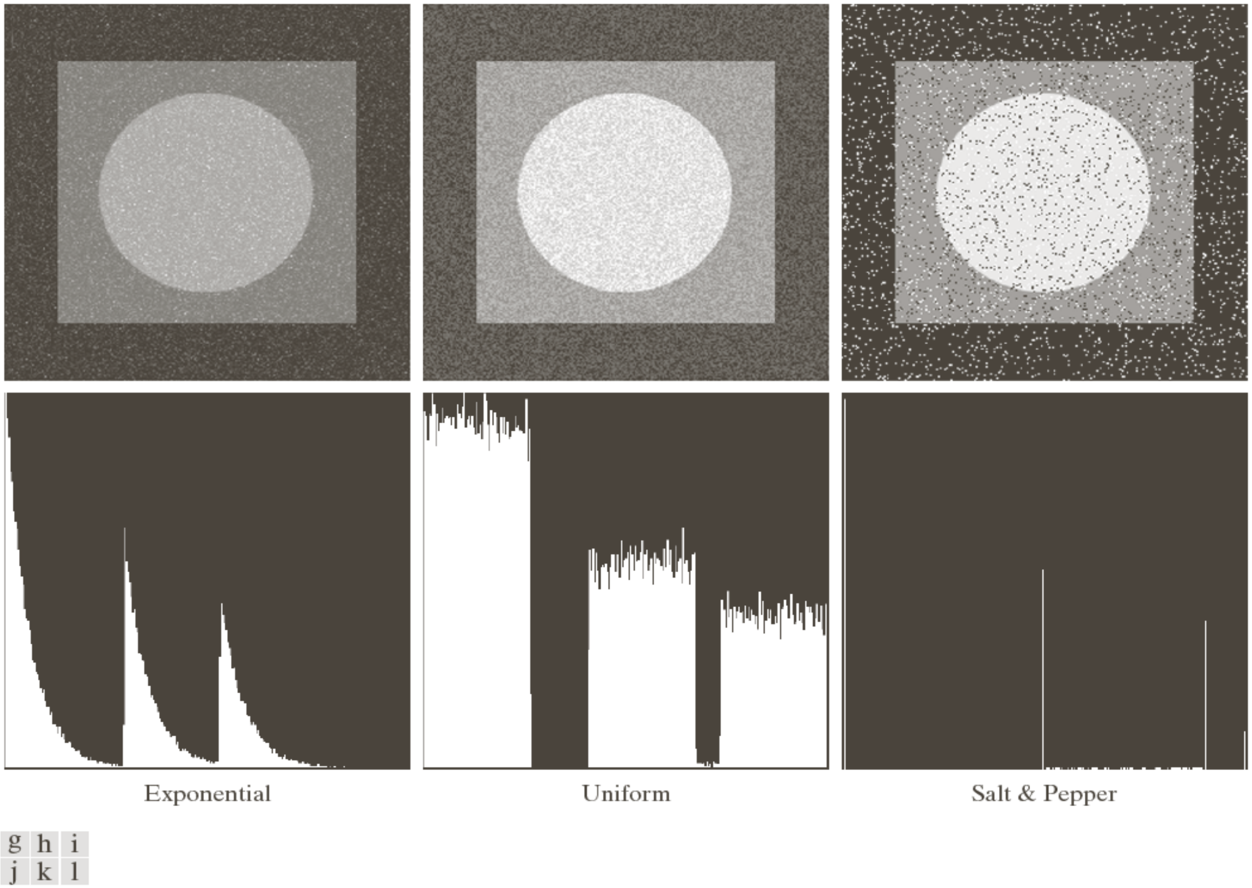
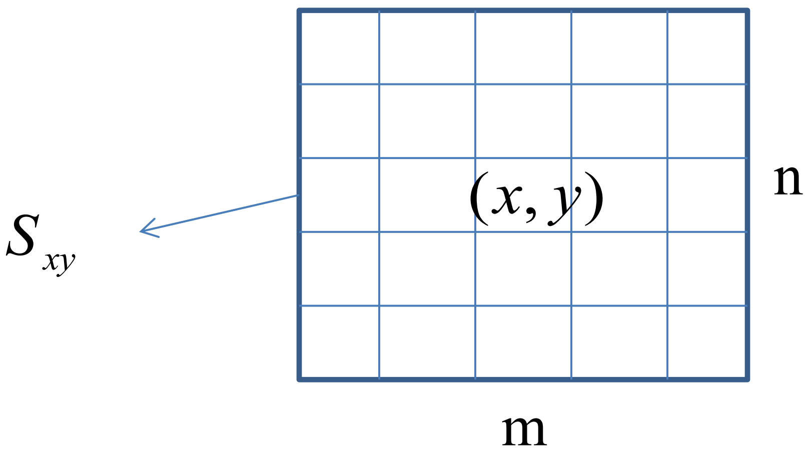


FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and salt and pepper noise to the image in Fig. 5.3.

Slides courtesy of Prof. Yan Tong

Mean Filters for Continuous Noise Models



Arithmetic Mean Filter: a linear filter

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

Slides courtesy of Prof. Yan Tong

Non-linear Mean Filters

Geometric Mean Filter

Harmonic Mean Filter

Contraharmonic Mean Filter

Slides courtesy of Prof. Yan Tong

Non-linear Mean Filters

Geometric Mean Filter

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

Harmonic Mean Filter

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

Work well for
- Continuous noise
- Salt noise

Fail for the pepper noise

Contraharmonic Mean Filter

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

Q is the order of the filter
- positive Q removes pepper noise
- negative Q removes salt noise
- Special cases: Q=0, Q=-1

Slides courtesy of Prof. Yan Tong

Estimation by Modeling – Motion Blur

Constant velocity along x and y direction:

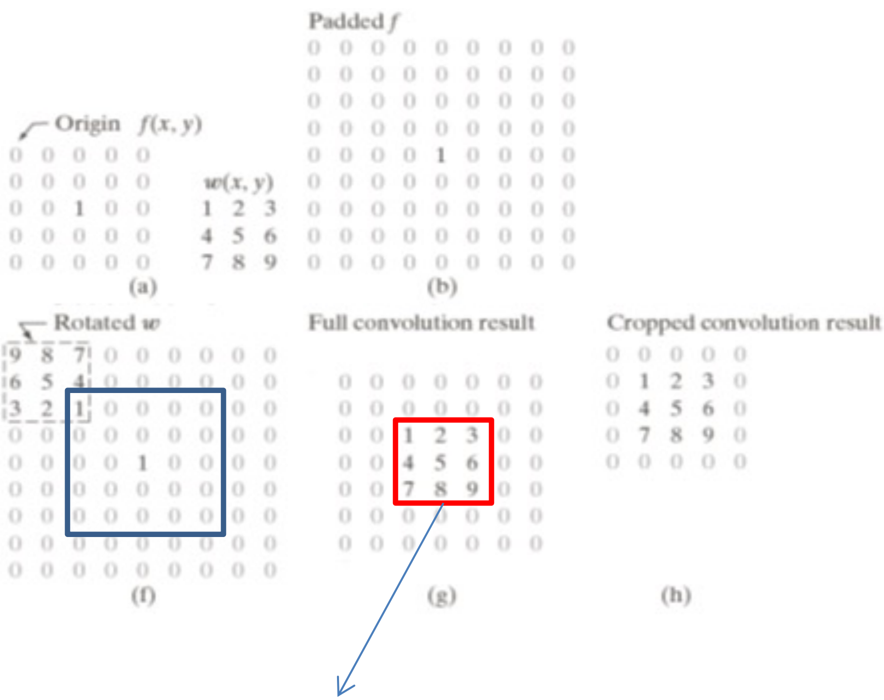
$$x_0(t) = at/T \qquad y_0(t) = bt/T$$

a b

FIGURE 5.26 (a) Original image. (b) Result of blurring using the function in Eq. (5.6-11) with $a = b = 0.1$ and $T = 1$.

Slides courtesy of Prof. Yan Tong

Extend to 2D Image: 2D Image Convolution



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

- Flip in both horizontal and vertical directions (rotate 180 degree) -> same if the filter is symmetric
- Convolution filter/mask/kernel
- Full convolution result has the size of $(M+2a, N+2b)$
- Cropped result has the size of (M, N) – the size of the original image

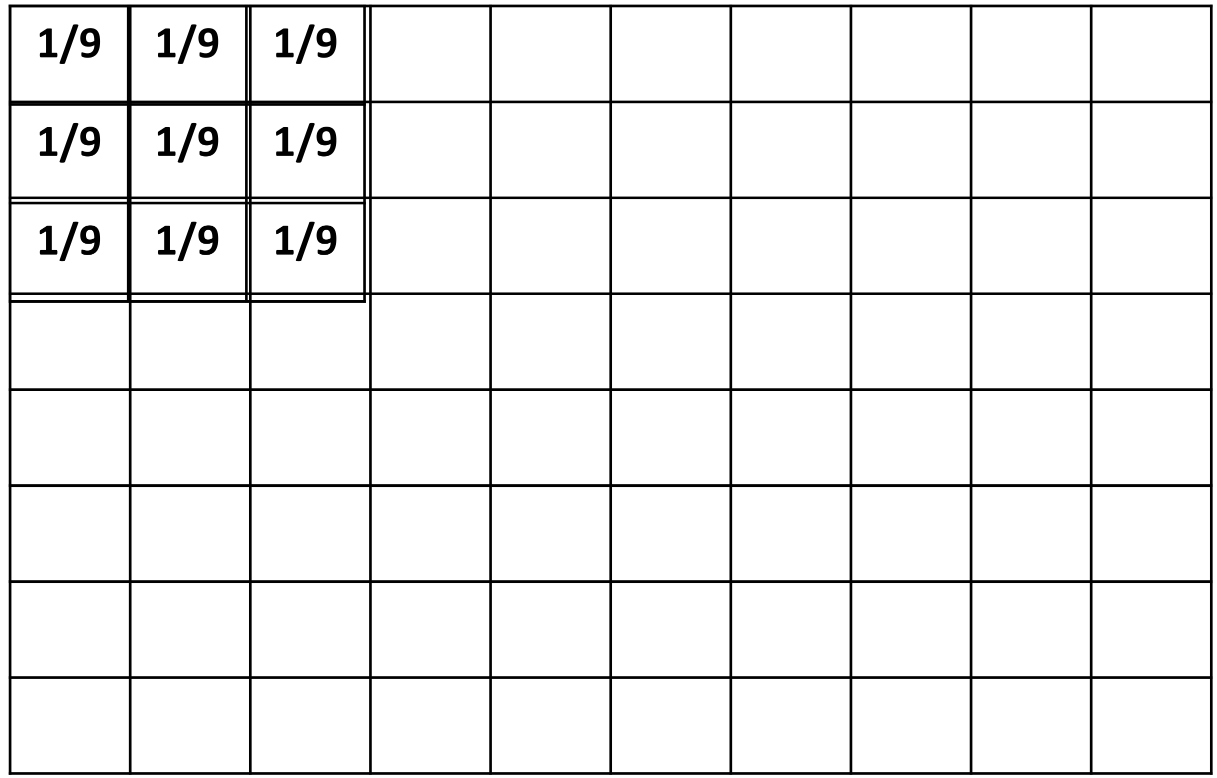
The 2D impulse response of image convolution is the same as the filter

Slides courtesy of Prof. Yan Tong

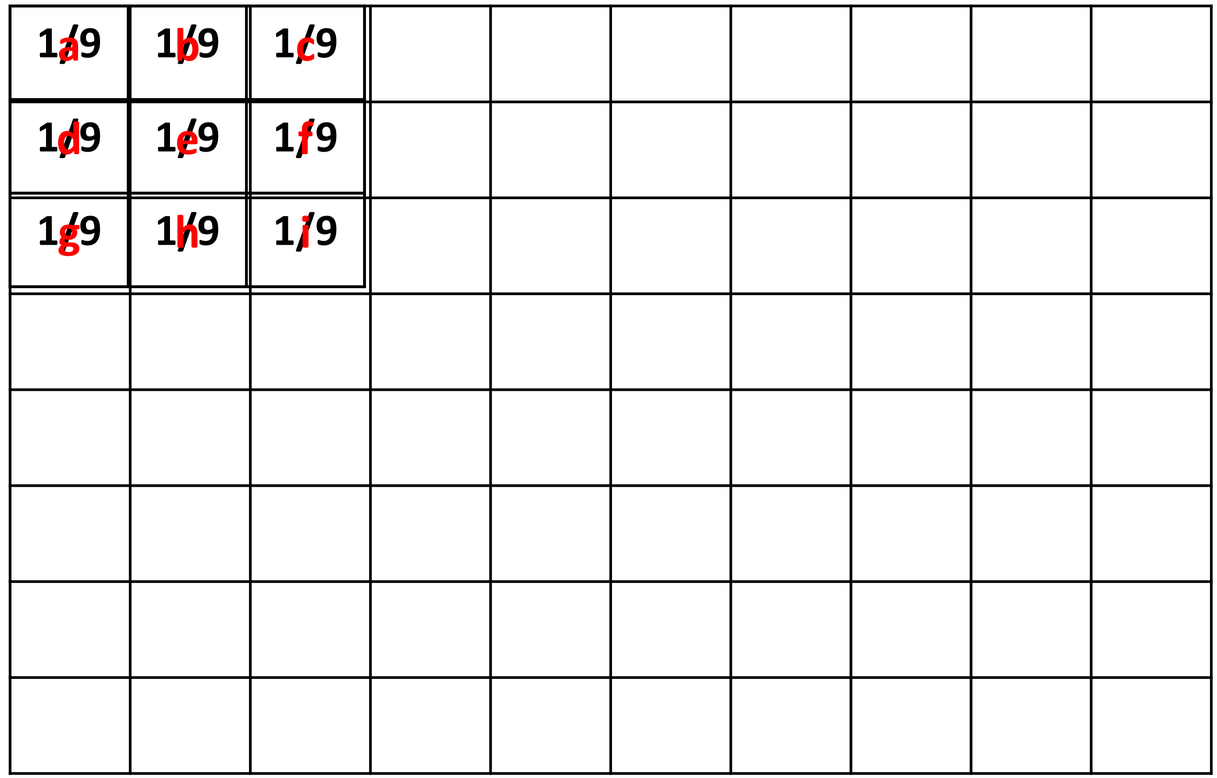
Convolution

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

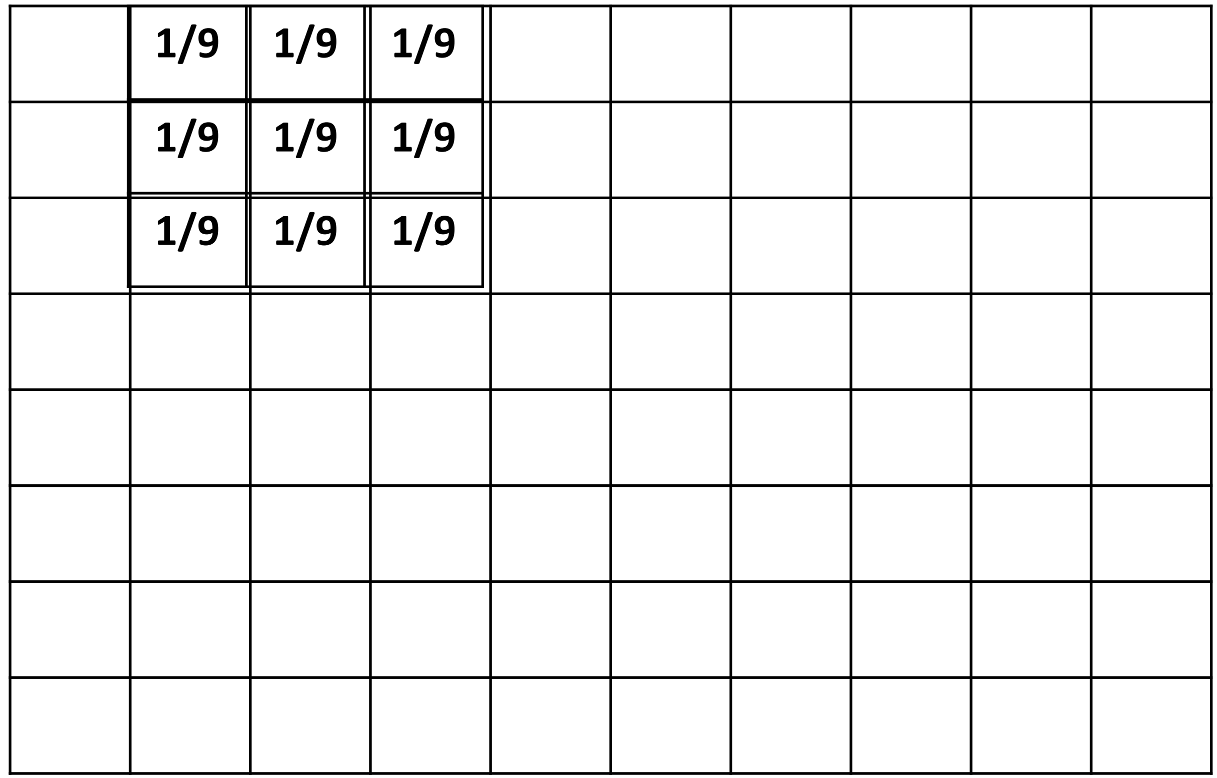
Convolution



Convolution



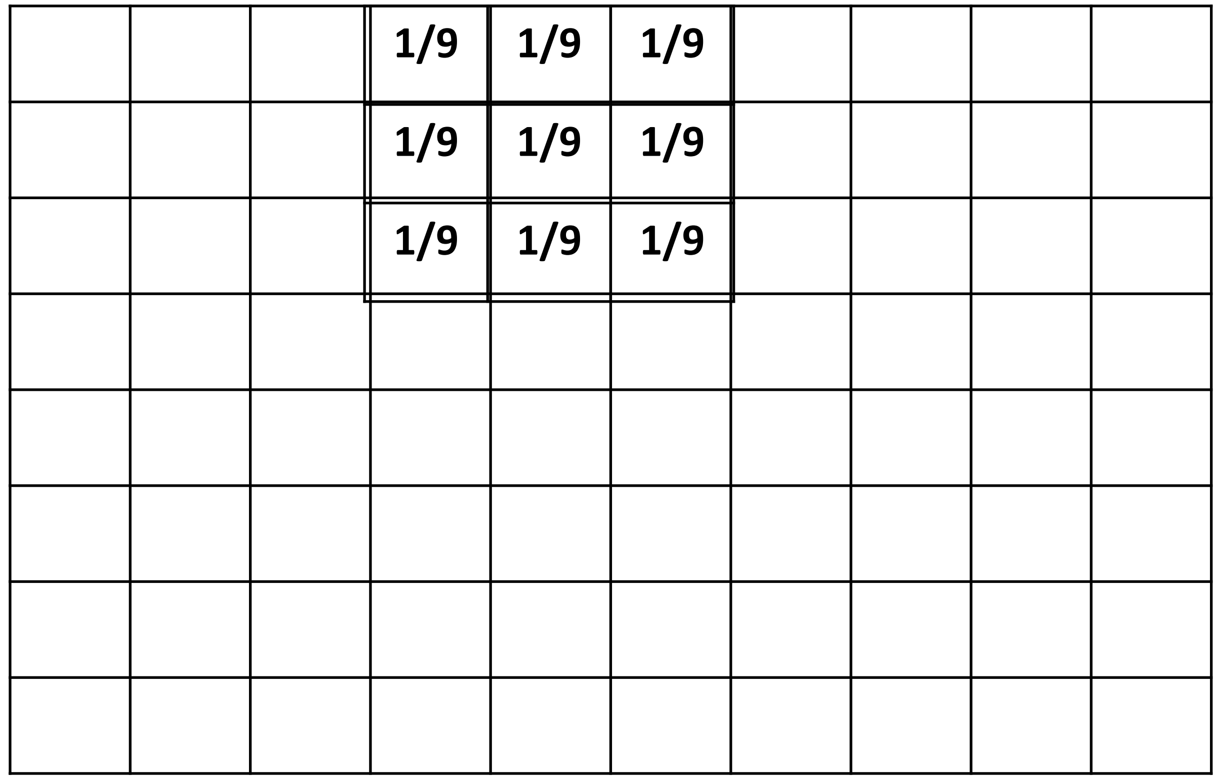
Convolution



Convolution

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

Convolution



Convolution

1/9	**1/9**	**1/9**							
1/9	**1/9**	**1/9**							
1/9	**1/9**	**1/9**							

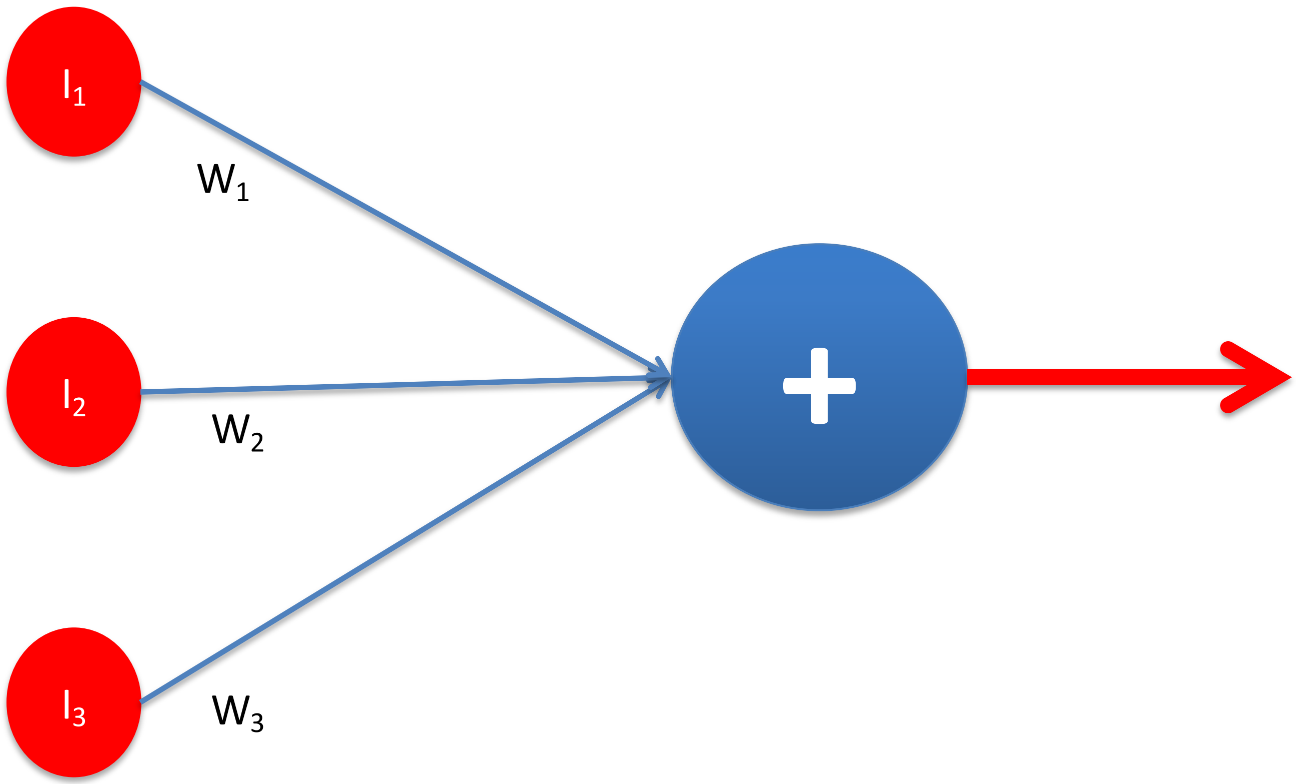
…

Serialization

a	b	c
d	e	f
g	h	i

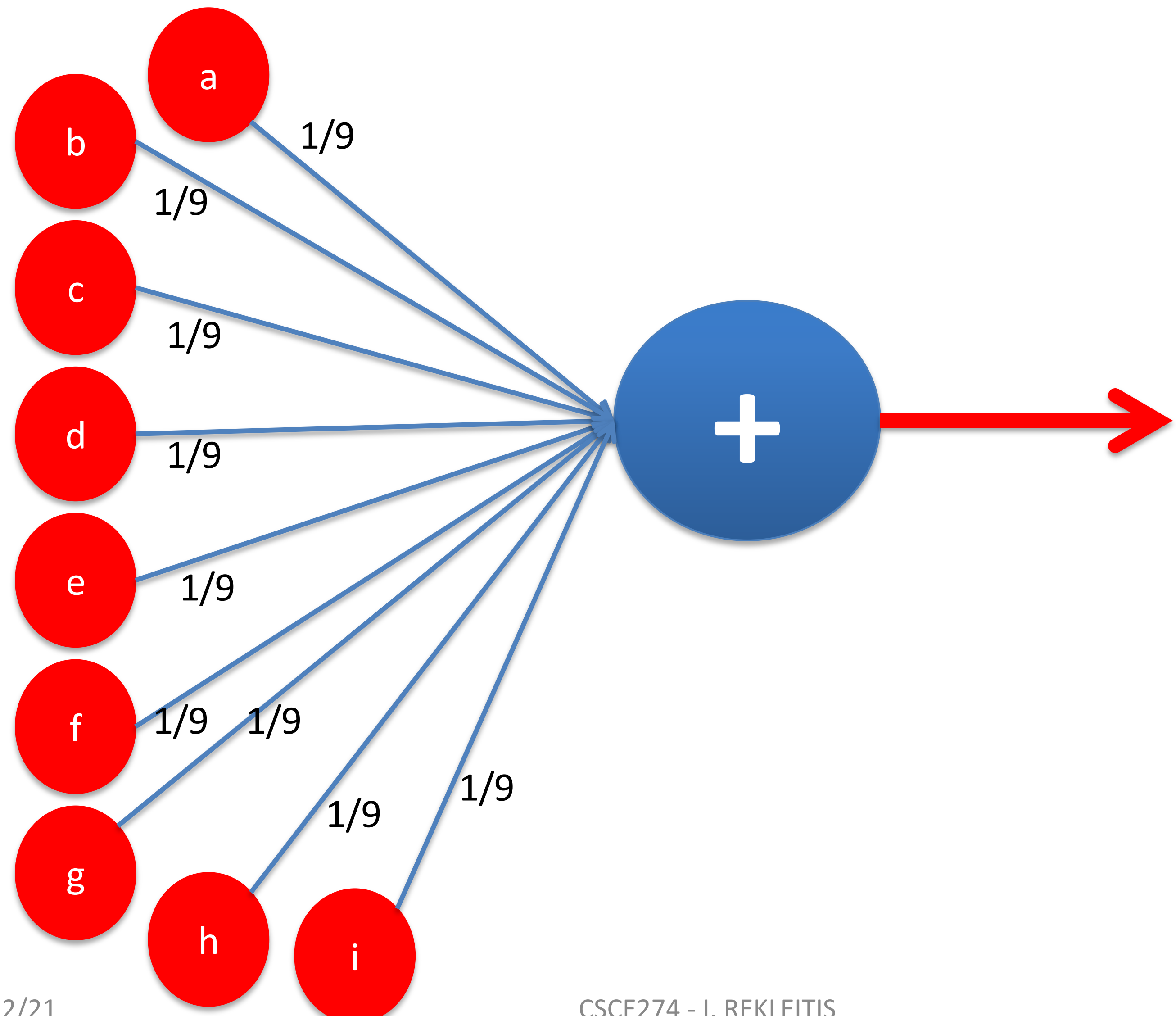
a	b	c	d	e	f	g	h	i

Neuron



$$O = \sum_{i=1\ldots3} W_i * I_i$$

Neuron



Linear Filters

- General process:
  - Form new image whose pixels are a weighted sum of original pixel values, using the same set of weights at each point.
- Properties
  - Output is a linear function of the input
  - Output is a shift-invariant function of the input (i.e. shift the input image two pixels to the left, the output is shifted two pixels to the left)
- Example: smoothing by averaging
  - form the average of pixels in a neighborhood
- Example: smoothing with a Gaussian
  - form a weighted average of pixels in a neighborhood
- Example: finding an edge

Slides courtesy of Prof. Yan Tong

Smoothing Spatial Filter – Low Pass Filters

Weighted average

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

1	1	1
1	1	1
1	1	1

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

1	2	1
2	4	2
1	2	1

a b

FIGURE 3.32 Two 3×3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to 1 divided by the sum of the values of its coefficients, as is required to compute an average.

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

- Noise reduction
- reduction of “irrelevant details”
- edge blurred

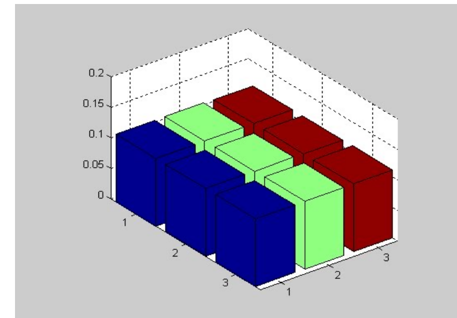
Normalization factor

Slides courtesy of Prof. Yan Tong

Smoothing Spatial Filter

Image averaging $$R = \frac{1}{9}\sum_{i=1}^{9} z_i$$

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



$* \frac{1}{9}$

1	1	1
1	1	1
1	1	1

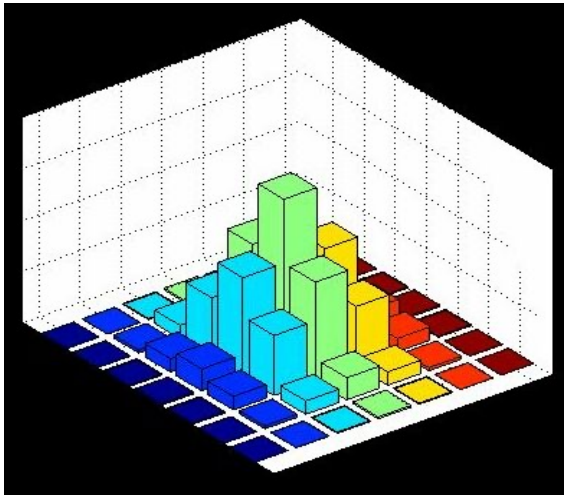
$=$

Slides courtesy of Prof. Yan Tong

Smoothing Spatial Filter

2D Gaussian filter $$h(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$*$



$=$

Slides courtesy of Prof. Yan Tong

Comparison using Different Smoothing Filters – Different Kernels

Average

Gaussian

Slides courtesy of Prof. Yan Tong

Basic Edge Detection

First-order derivative:

Gradient

$$\nabla f(x,y) = grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix}$$

Edge strength

$$M(x,y) = \sqrt{g_x^2 + g_y^2}$$

$$\approx |g_x| + |g_y|$$

Edge direction

$$\alpha(x,y) = \tan^{-1}\left[\frac{g_y}{g_x}\right]$$

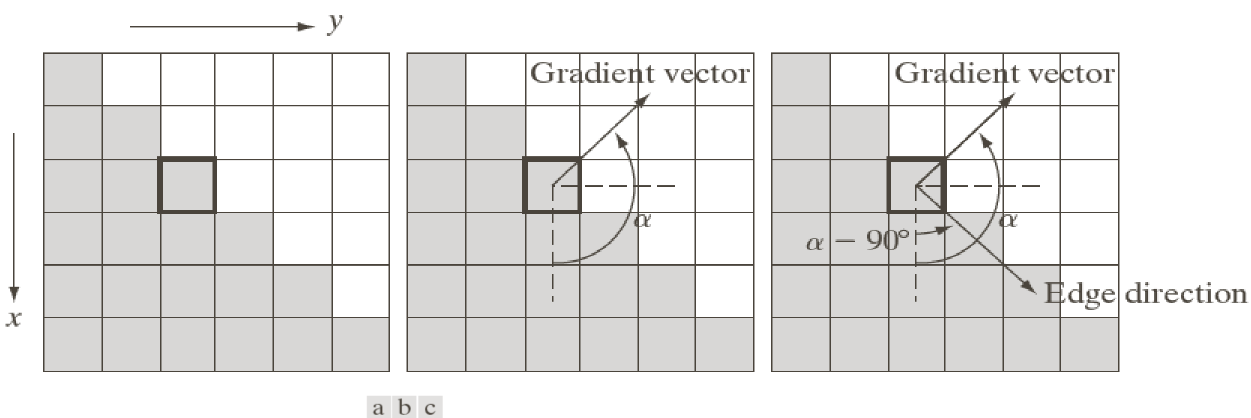


FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

Slides courtesy of Prof. Yan Tong

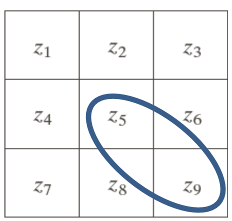
Masks for Calculating the Gradient (2x2)

Gradient in vertical/horizontal direction

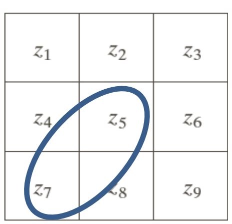
-1
1

-1	1

Gradient in diagonal direction



-1	0
0	1



0	-1
1	0

Slides courtesy of Prof. Yan Tong

Masks for Calculating the Gradient (3x3)

Gradient in vertical/horizontal

−1	−1	−1	−1	0	1
0	0	0	−1	0	1
1	1	1	−1	0	1

Prewitt

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

Sobel

Gradient in diagonal

0	1	1	−1	−1	0
−1	0	1	−1	0	1
−1	−1	0	0	1	1

Prewitt

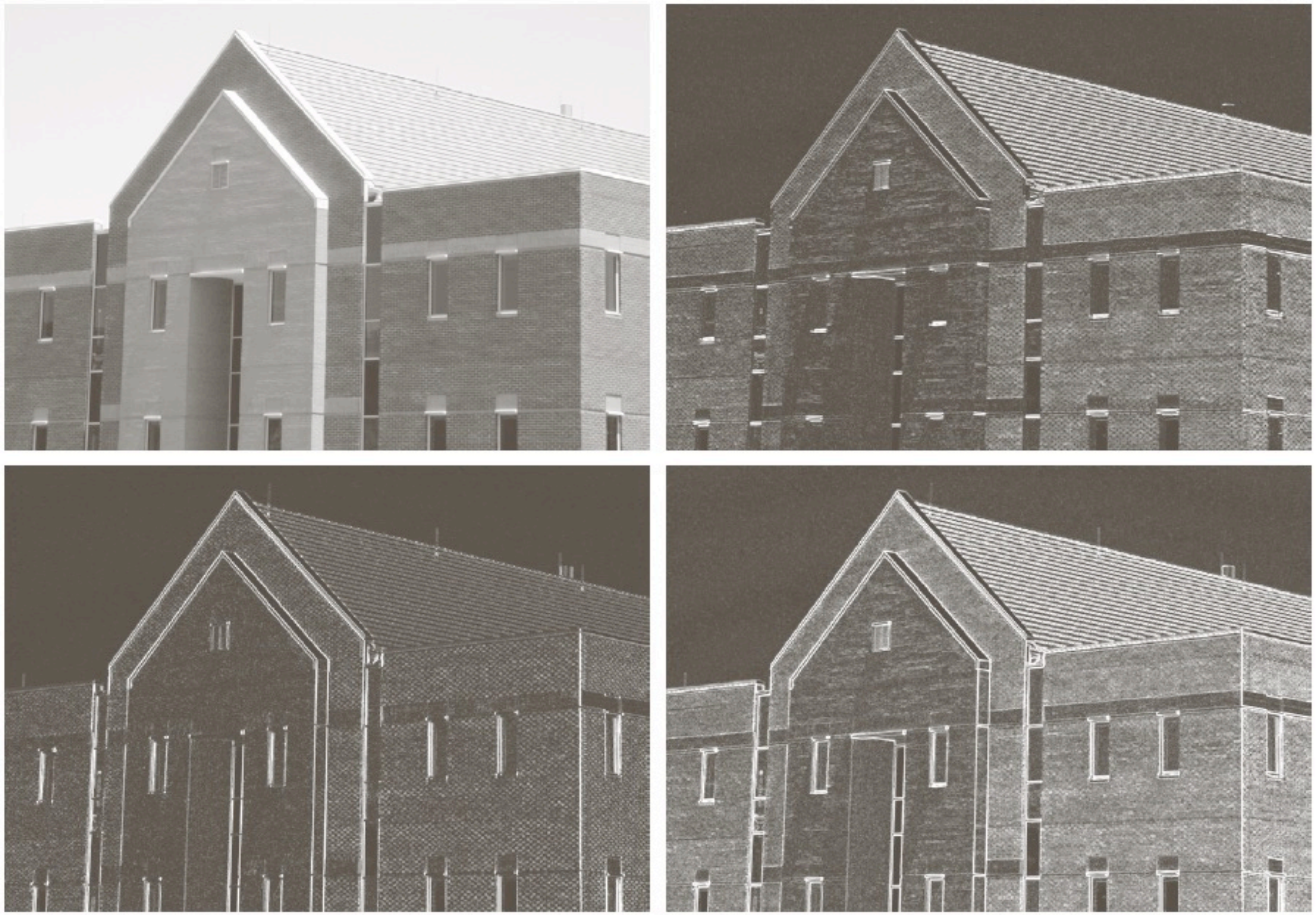
0	1	2	−2	−1	0
−1	0	1	−1	0	1
−2	−1	0	0	1	2

Sobel

Sobel operator performs edge detection and smoothing simultaneously.

Slides courtesy of Prof. Yan Tong

An Example



a b
c d

FIGURE 10.16 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) $|g_x|$, the component of the gradient in the x-direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image. (c) $|g_y|$, obtained using the mask in Fig. 10.14(g). (d) The gradient image, $|g_x| + |g_y|$.

Slides courtesy of Prof. Yan Tong

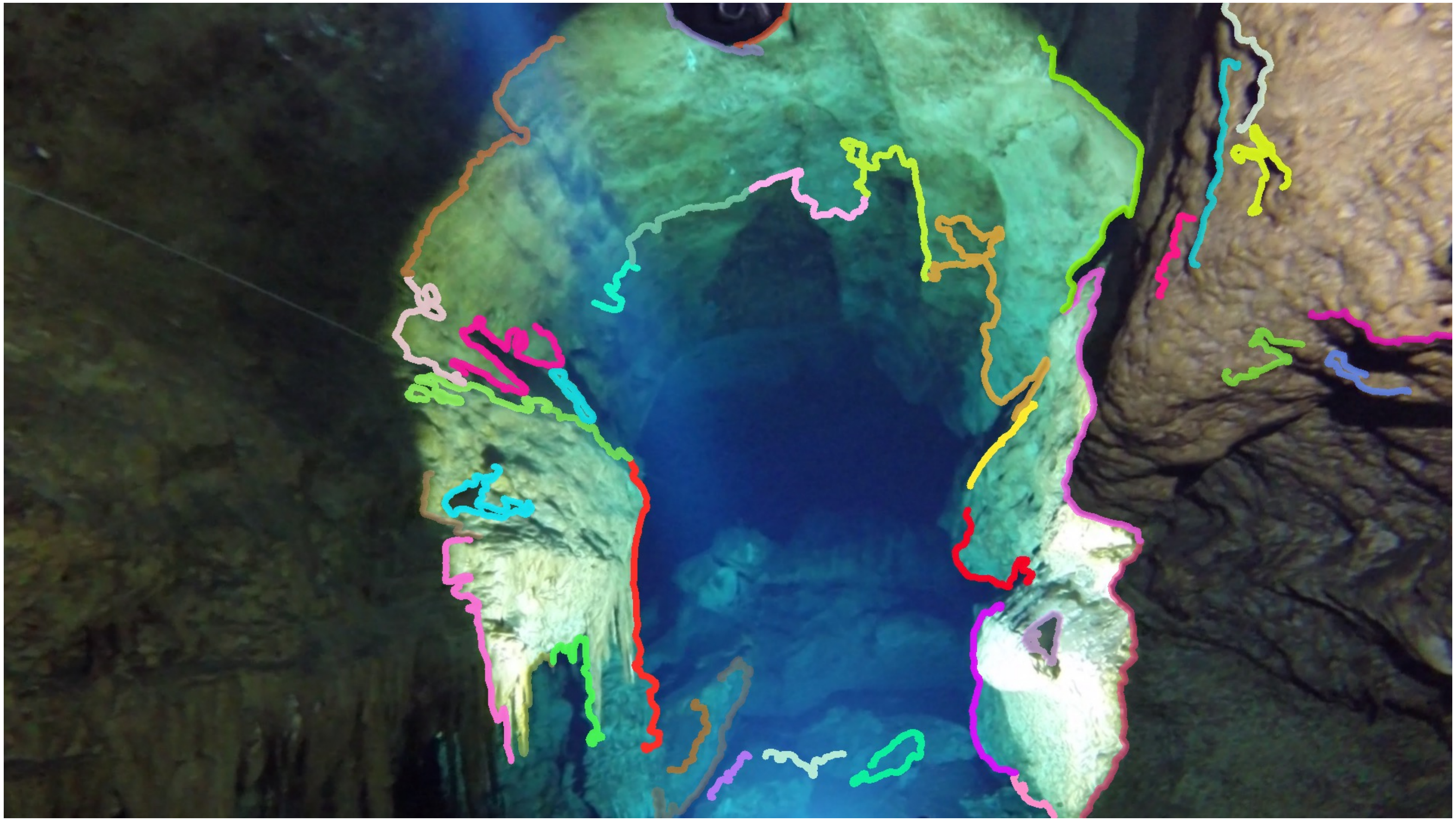
An Example – Cont.

FIGURE 10.17 Gradient angle image computed using Eq. (10.2-11). Areas of constant intensity in this image indicate that the direction of the gradient vector is the same at all the pixel locations in those regions.

Angle information is employed in Canny edge detector and other feature representation, such as Histogram of Orientation (HOG).

Slides courtesy of Prof. Yan Tong

Edge detection



Brief Review on Simple Edge Detectors

- First-order derivative
  - Roberts (2x2)
  - Prewitt (3x3)
  - Sobel (3x3, smooth + difference)
  - Thicker edge
  - One operator for one edge direction
- Second-order derivative
  - Laplacian (3x3)
  - Double edge
  - Zero-crossing
- Common issues:
  - Sensitive to image noise
  - Cannot deal with the scale change of the image

Slides courtesy of Prof. Yan Tong

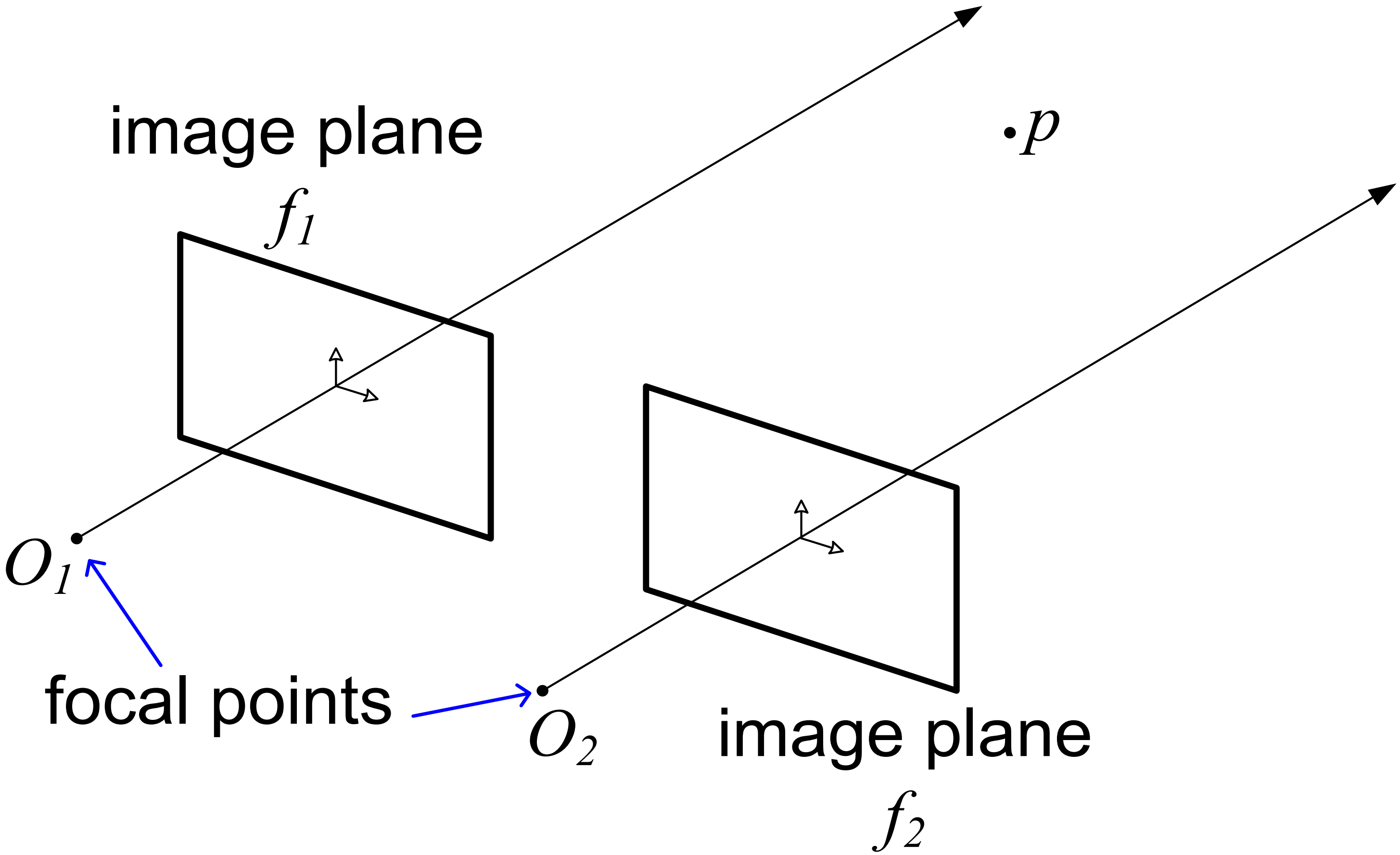
Advanced Edge Detection Techniques

- Deal with image noise
- Exploit the properties of image
- → Work much better for real images

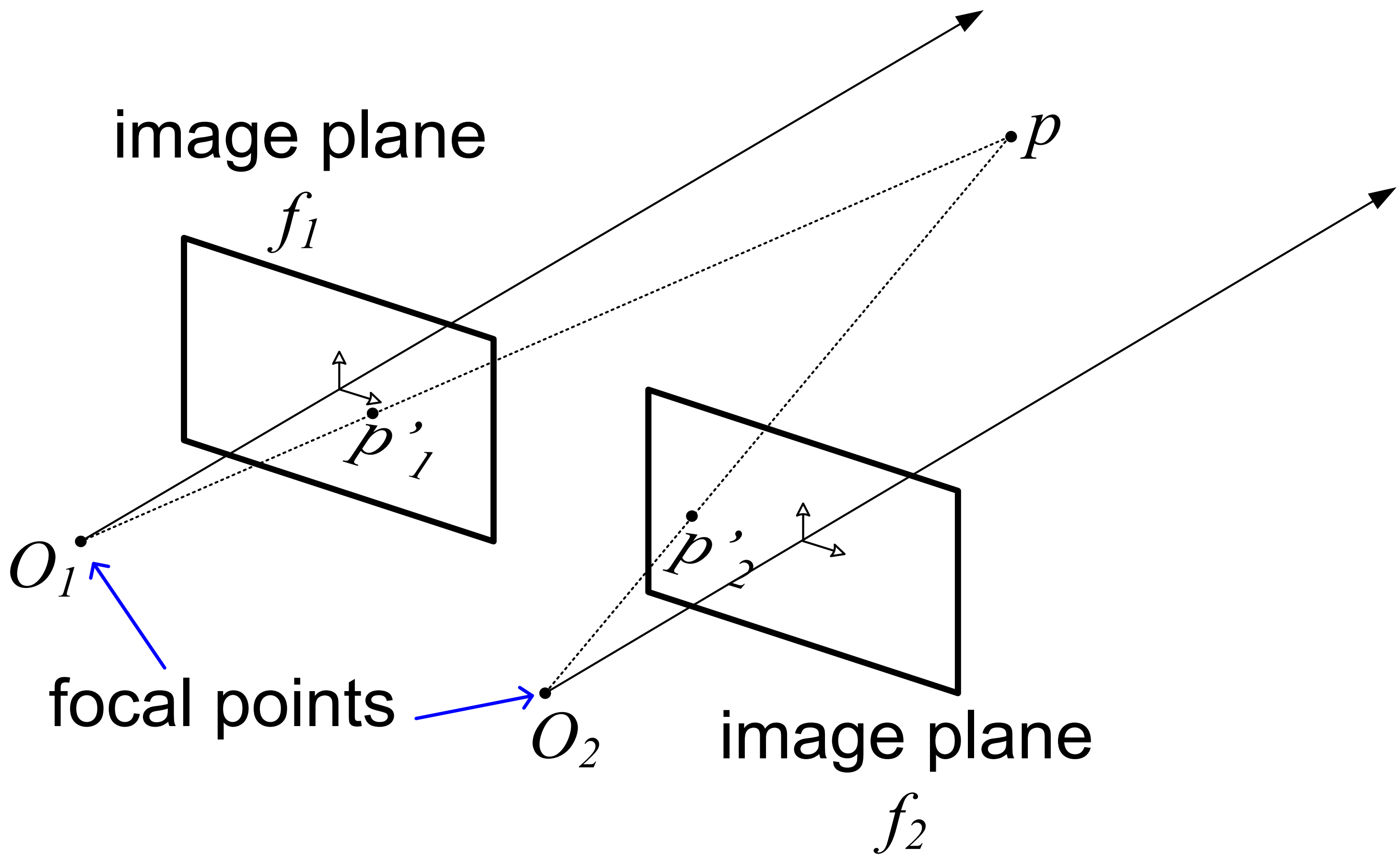
- Advanced edge detectors:
  - Laplacian of Gaussian (LoG)
  - Difference of Gaussian (DoG)
  - Canny

Slides courtesy of Prof. Yan Tong

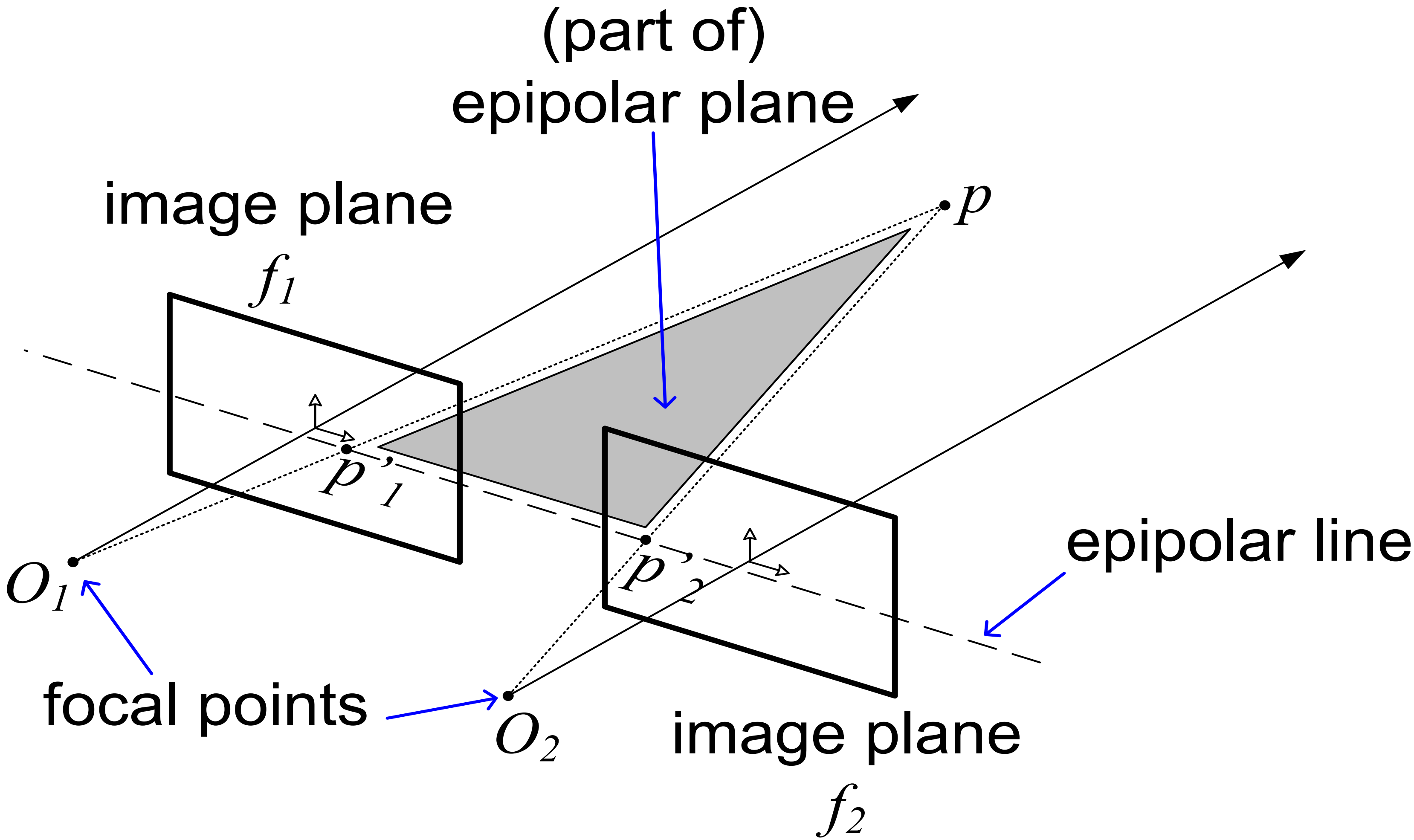
Stereo Vision: Pinhole Camera



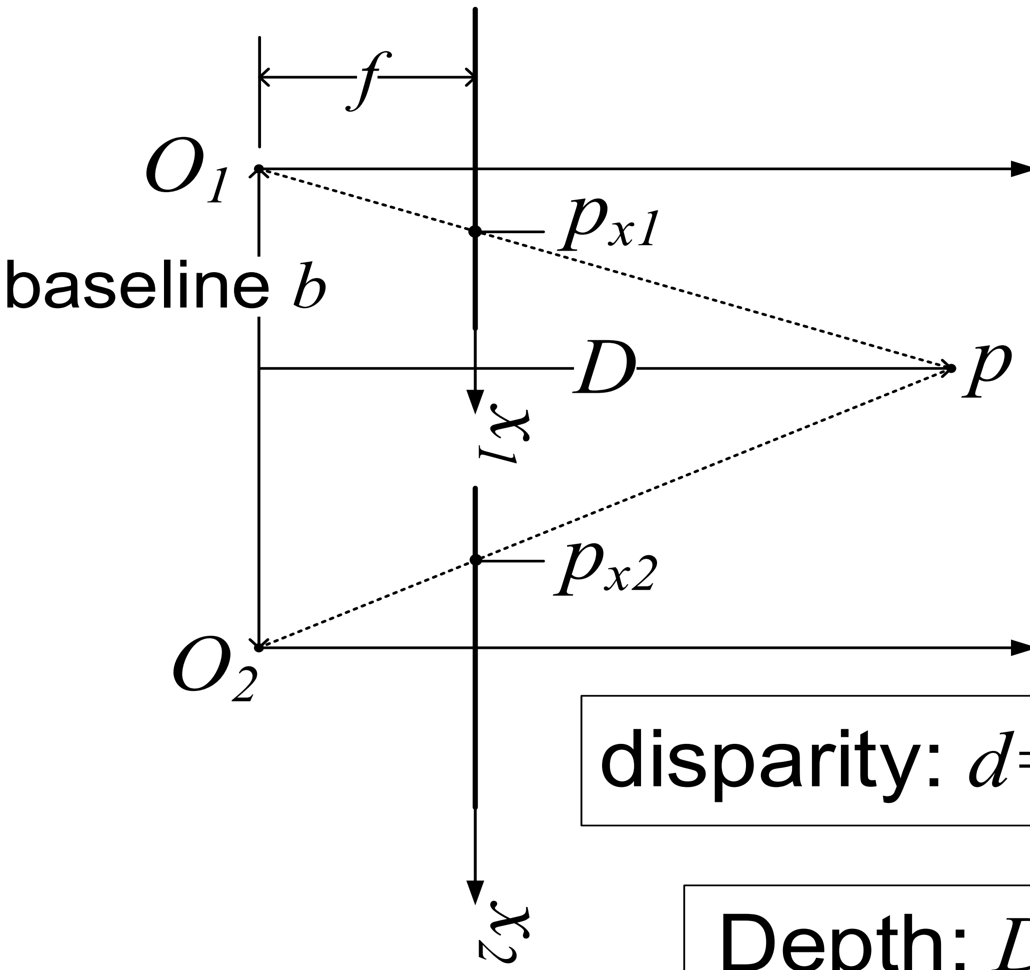
Stereo Vision: Pinhole Camera



Stereo Vision: Pinhole Camera



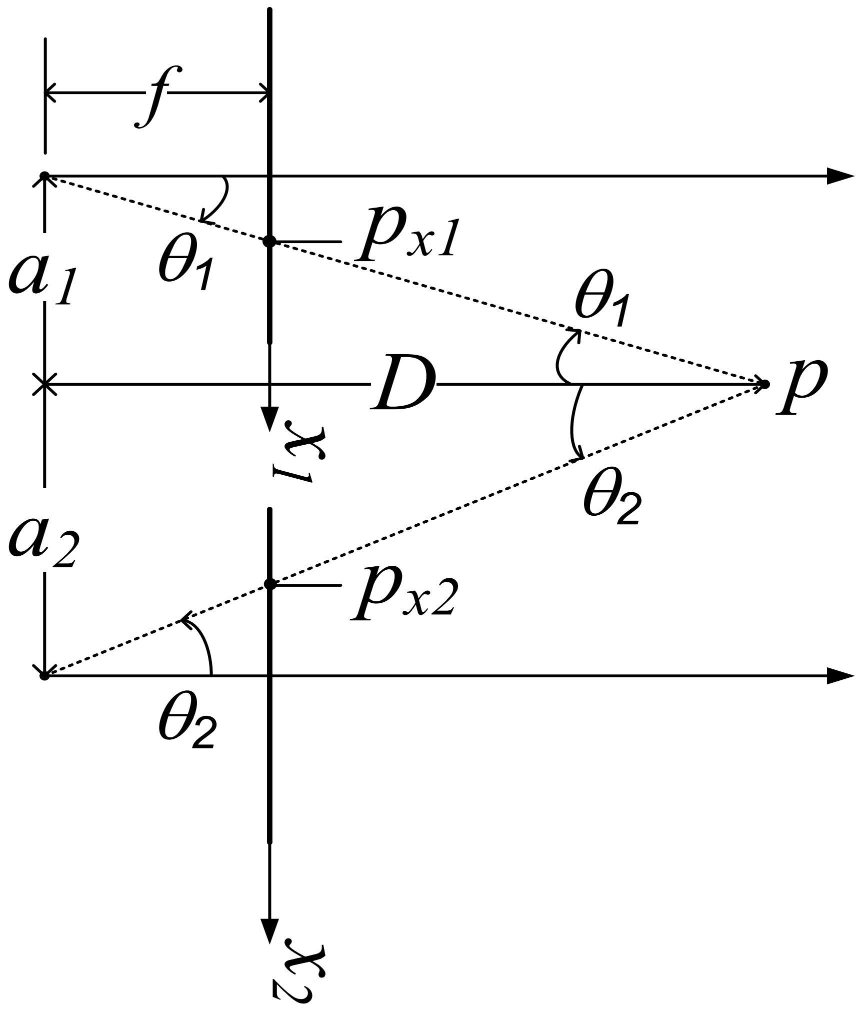
Stereo Vision: Pinhole



disparity: $d = p_{x1} - p_{x2}$

Depth: $D = fb/d$

Stereo Vision: Pinhole



$$\frac{p_{x1}}{f} = \frac{a_1}{D}$$

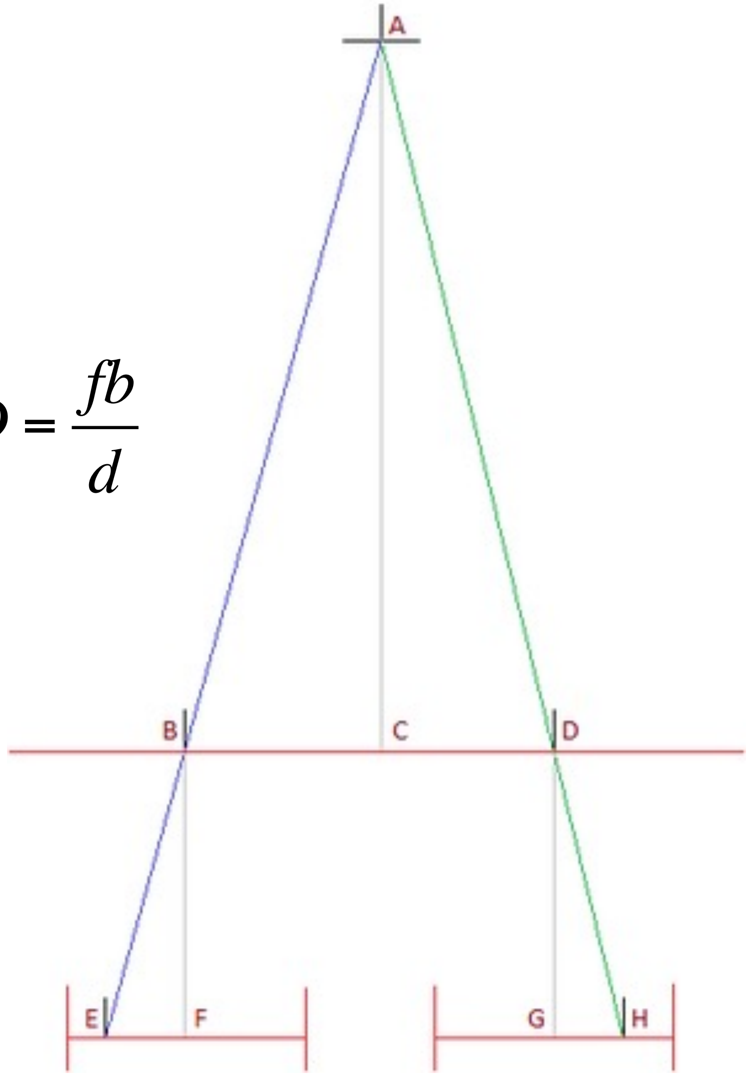
$$\frac{p_{x2}}{f} = \frac{a_2}{D}$$

$$a_1 + a_2 = b$$

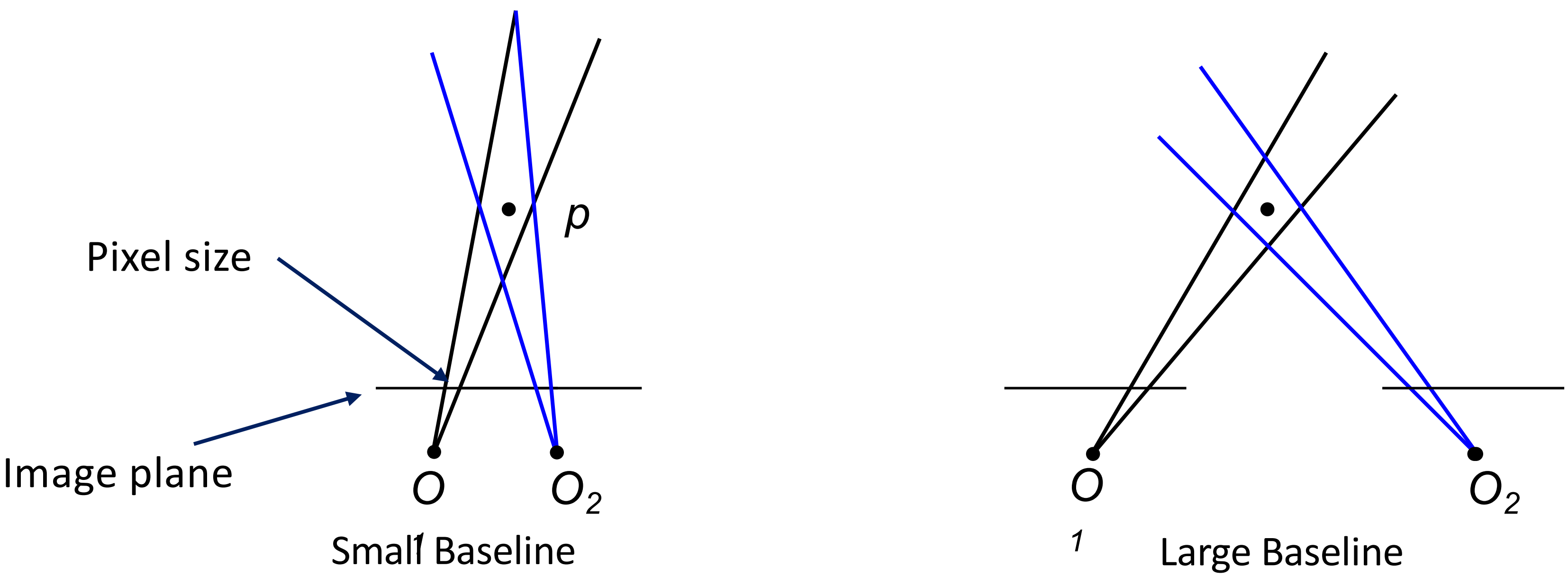
Stereo continuation

The disparity is EF+GH
so:

$$\frac{|EF|+|GH|}{f} = \frac{\alpha_1+\alpha_2}{D} \Leftrightarrow \frac{d}{f} = \frac{b}{D} \Leftrightarrow D = \frac{fb}{d}$$



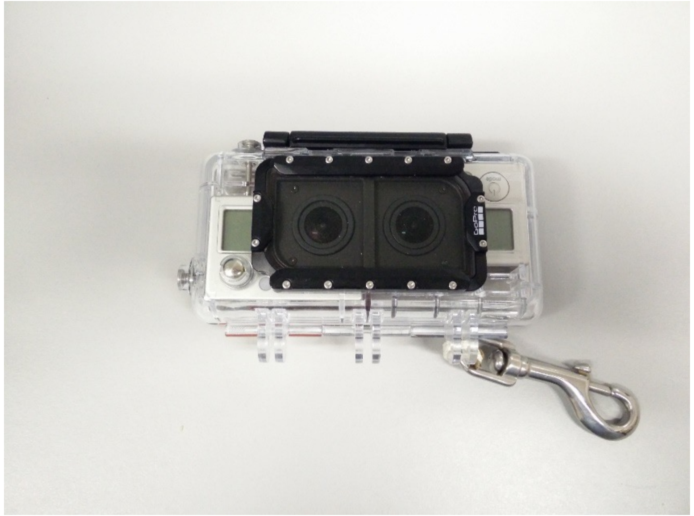
Baseline



- What's the optimal baseline?
  - Too small: large depth error
  - Too large: difficult search problem

Baseline

GoPro 3D HERO System



b=3.2 cm

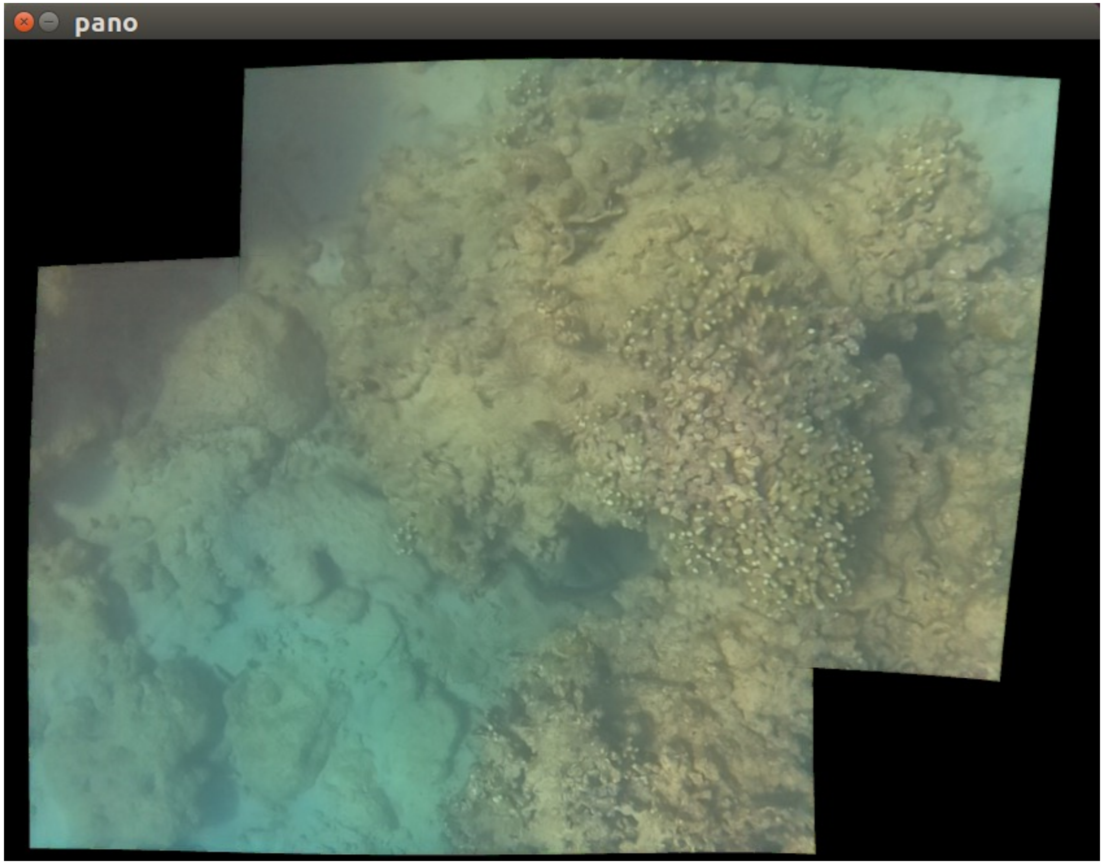
source: http://www.cvlibs.net/datasets/kitti

b=54 cm

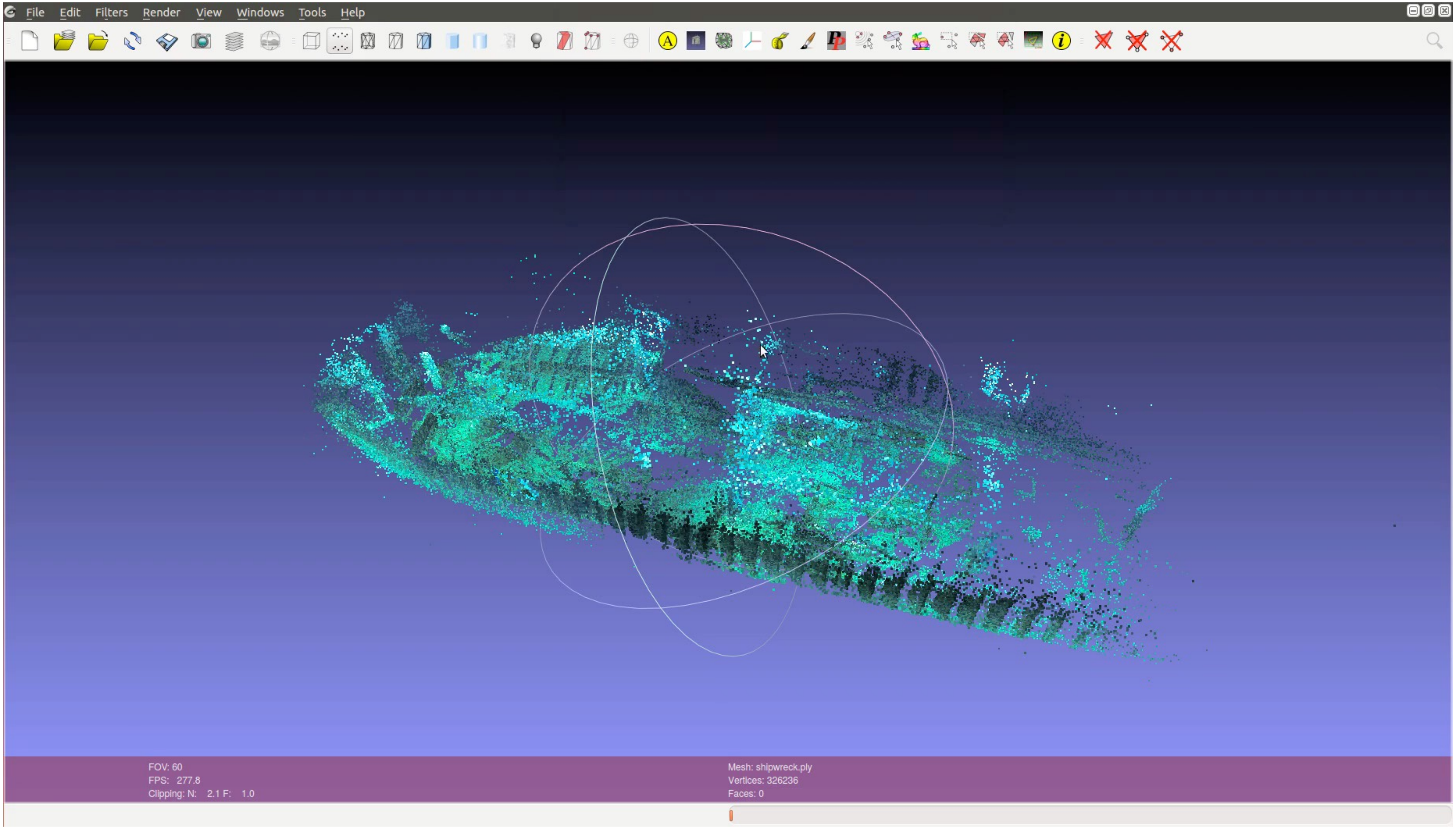
Matching Left and Right

Visual Odometry/Structure from Motion

Mosaic



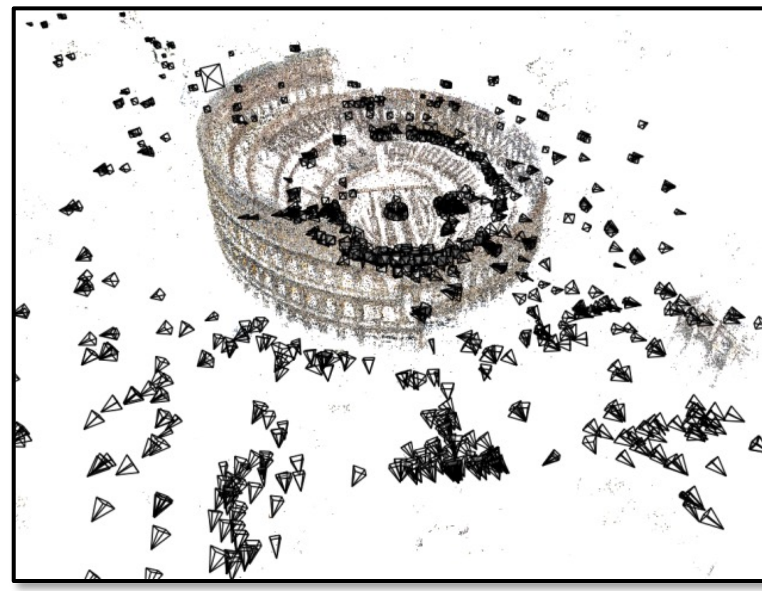
3D Sparse reconstruction



3D Sparse reconstruction

Source: https://grail.cs.washington.edu/rome/

Internet Photos ("Colosseum")



Reconstructed 3D cameras and points

OpenCV