# Bioimage Informatics

## Lecture 11, Spring 2012

Bioimage Data Analysis (III):

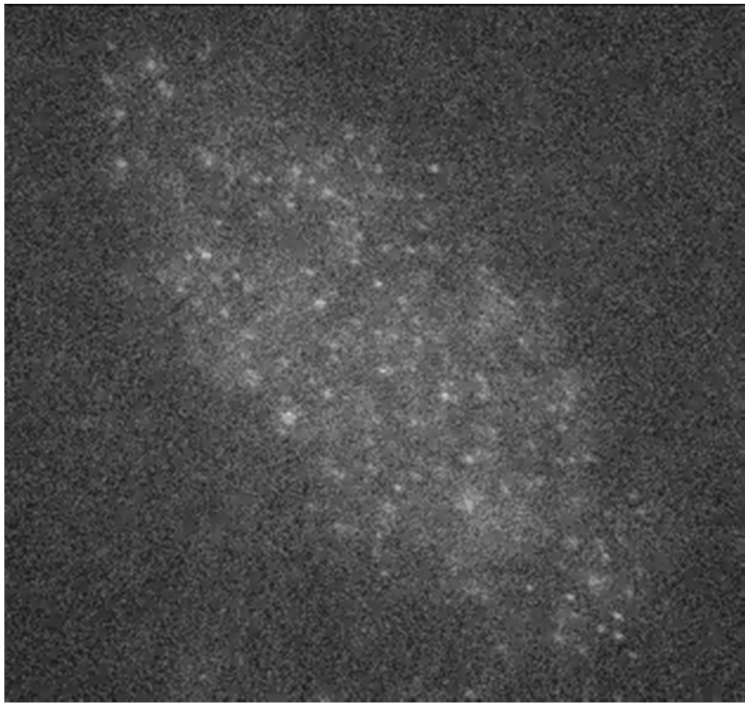
## Edge Detection; Line/Curve Detection

# Outline

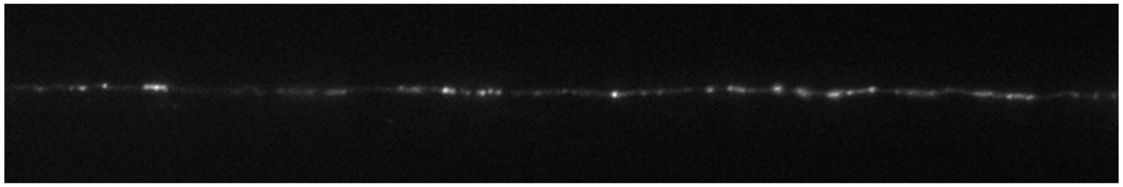
- Review: low-level feature detection
- Overview of edge detection
- Line/curve detection using the Hough transform

- Review: low-level feature detection
- Overview of edge detection
- Line/curve detection using the Hough transform

# Feature Detection: Points/Particles

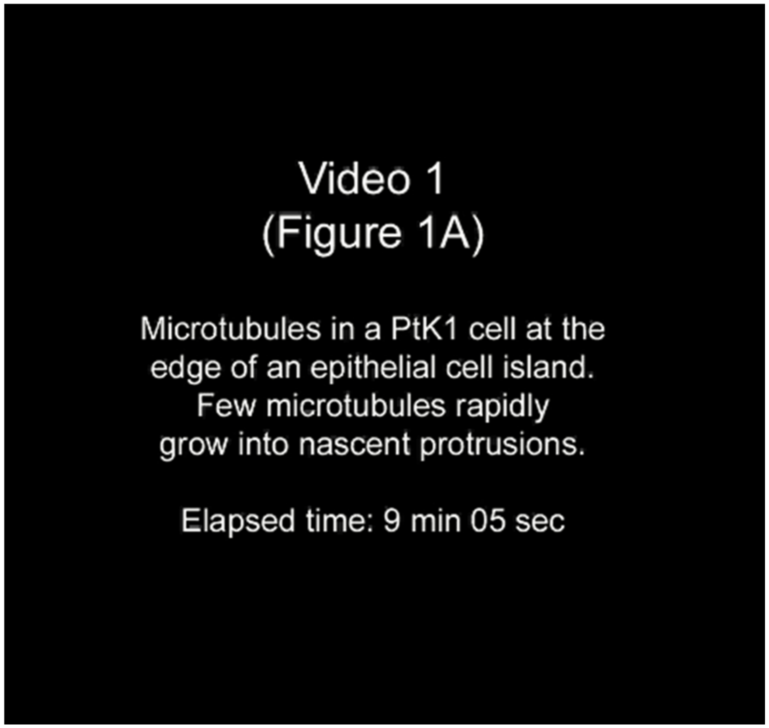


Fluorescent speckles in a Xenopus extract spindle



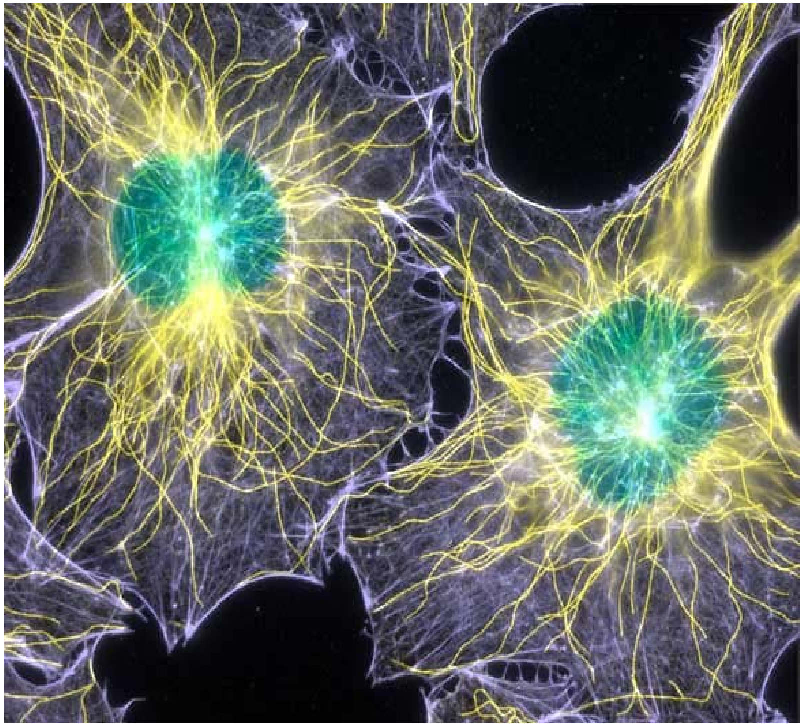
Vesicles transported in a Drosophila motor neuron

# Feature Detection: Lines/Curves



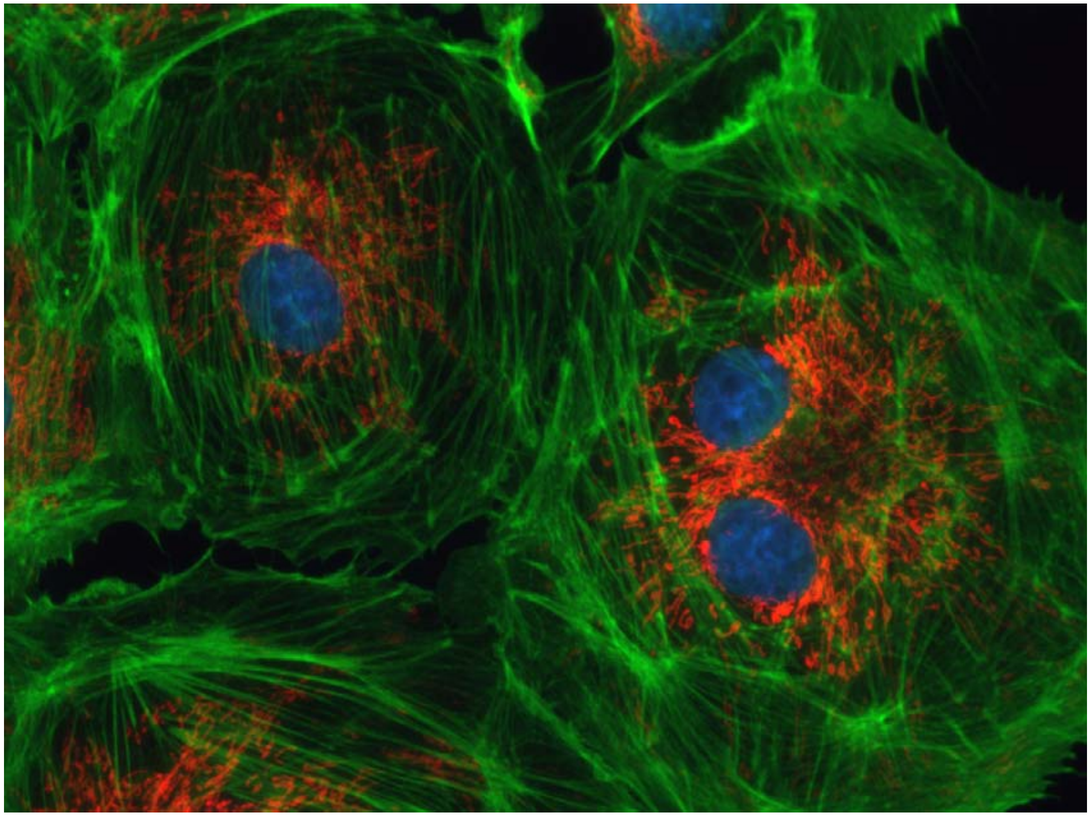

T. Wittmann et al, *J. Cell Biol.*, 161:845, 2003.

http://www.cell.com/cell_picture_show



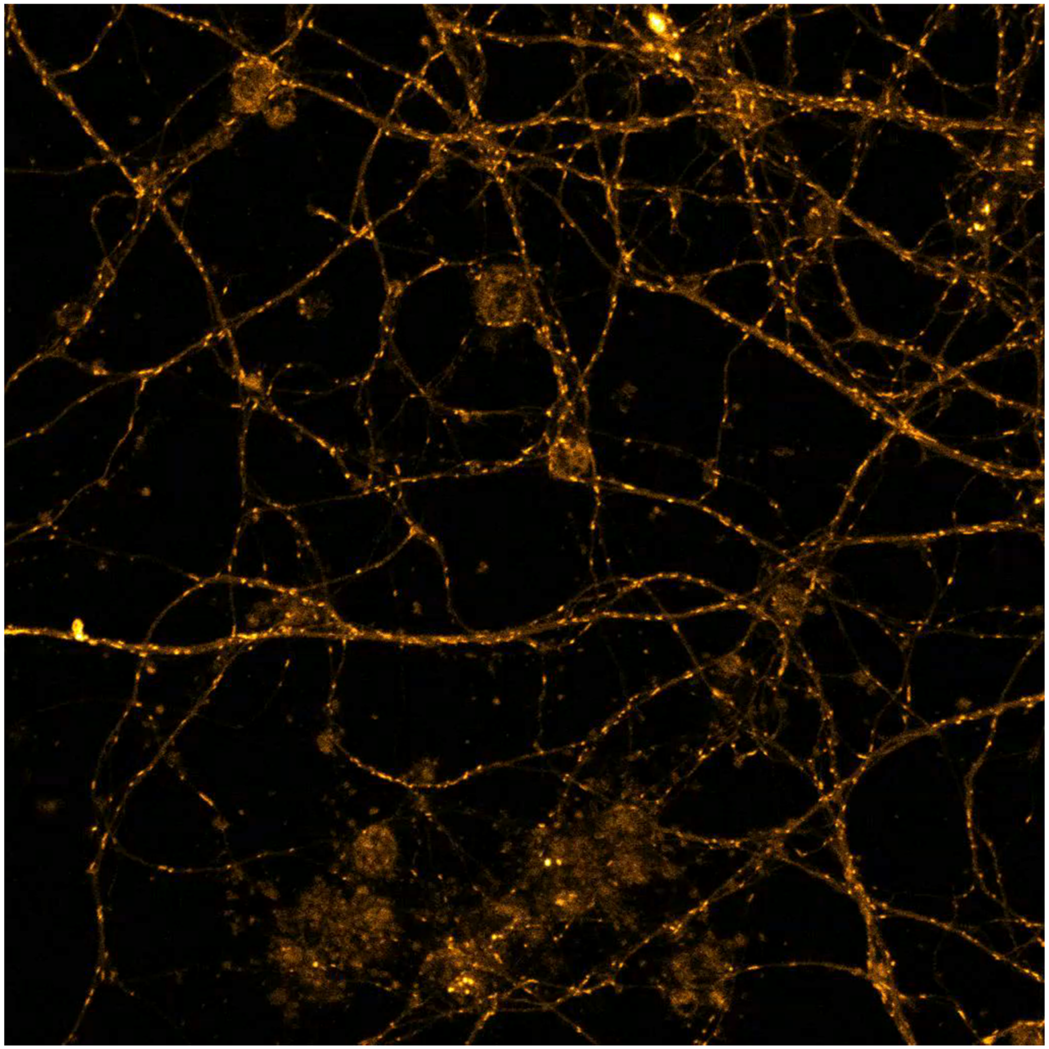
**Nikon Small World, 2003**
Torsten Wittmann, UCSF
Filamentous actin and microtubules (structural proteins) in mouse fibroblasts (cells) (1000x)

# An Image from Lab Visit

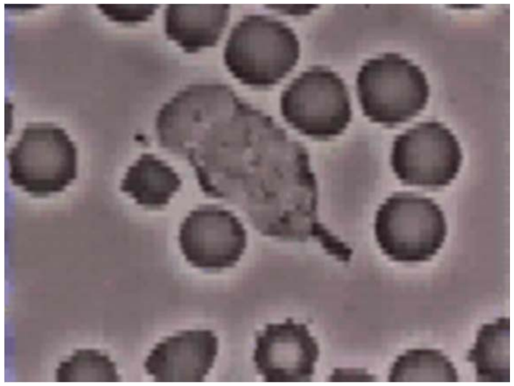


Bovine pulmonary artery endothelial (BPAE) cells stained with a combination of fluorescent dyes. Mitochondria were labeled with red-fluorescent MitoTracker Red CMXRos, F-actin was stained using green-fluorescent Alexa Fluor 488 phalloidin, and blue-fluorescent DAPI was used to label the nuclei.

# Feature Detection: Regions



Mitochondria in mouse hippocampal neuron, James Lim, LBNL



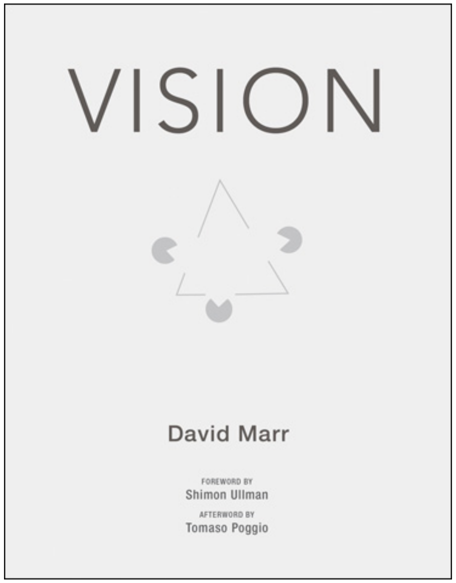
A neutrophil chasing a bacterium.
*Devreotes Lab, Johns Hopkins U.*

# Marr's Theory of Visual Information Processing

- David Marr (Jan. 19,1945 – Nov. 17, 1980)

- Three levels of cognition
  - Computational level
  - Algorithmic/representation level
  - Implementational level

- Three stages of vision
  - Primal sketch
  - 2.5D sketch
  - 3D model



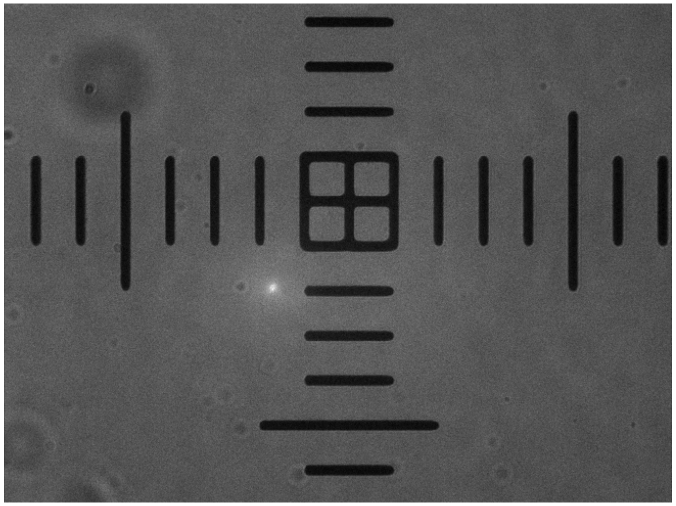

*Vision: A computational investigation into the human representation and processing of visual information*

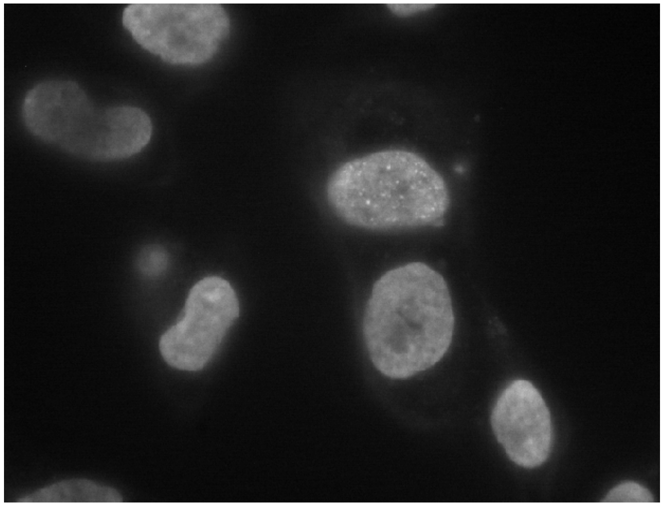
- Review: low-level feature detection
- Overview of edge detection
- Line/curve detection using the Hough transform

# Edge Detection

- What is an edge?

An edge point, or an edge, is a pixel at or around which the image intensities undergo a sharp change.





# Motivation: Edge in Images

- The edge can be treated as a 1D signal when examined in the normal direction.

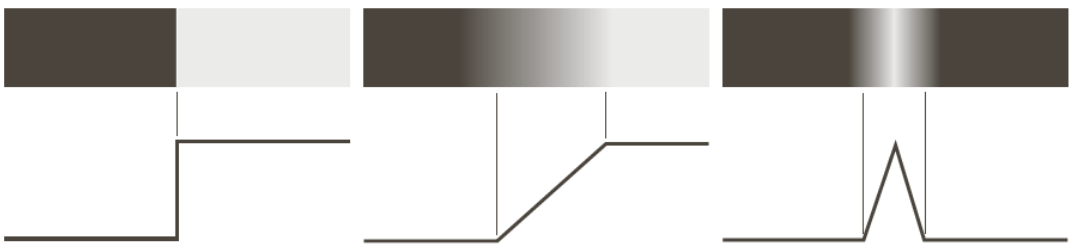
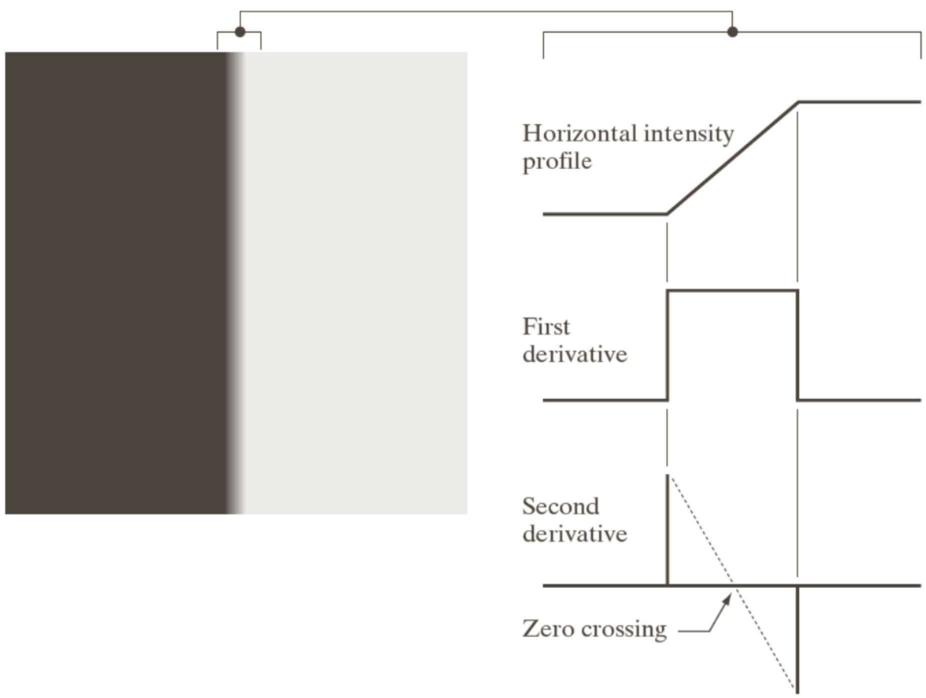



**FIGURE 10.8** From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.

Gonzalez & Woods, DIP 3/e

# Edge Can be Identified by Calculating Gradient




a b

**FIGURE 10.10** (a) Two regions of constant intensity separated by an ideal vertical ramp edge. (b) Detail near the edge, showing a horizontal intensity profile, together with its first and second derivatives.

Gonzalez & Woods, DIP 3/e

# How to Calculate Image Gradient

- Calculation of image gradient follows standard numerical differentiation scheme:

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(x) + O(h^3)$$
$$f(x-h) = f(x) - hf'(x) + \frac{1}{2}h^2 f''(x) + O(h^3)$$
$$f'(x_0) = \frac{f(x_0+h) - f(x_0-h)}{2h} + O(h^2)$$

- In an image, the first and second derivatives:

$$\frac{\partial I(i,j)}{\partial x} = I_x(i,j) = \frac{I(i+1,j) - I(i-1,j)}{2 \cdot h} + O(h^2)$$
$$\frac{\partial^2 I(i,j)}{\partial x^2} = I_{xx}(i,j) = \frac{I(i+1,j) - 2I(i,j) + I(i-1,j)}{h^2} + O(h)$$

# Gradient Calculation is Sensitive to Noise

- Without image smoothing, calculation of derivatives becomes highly sensitive to noise.

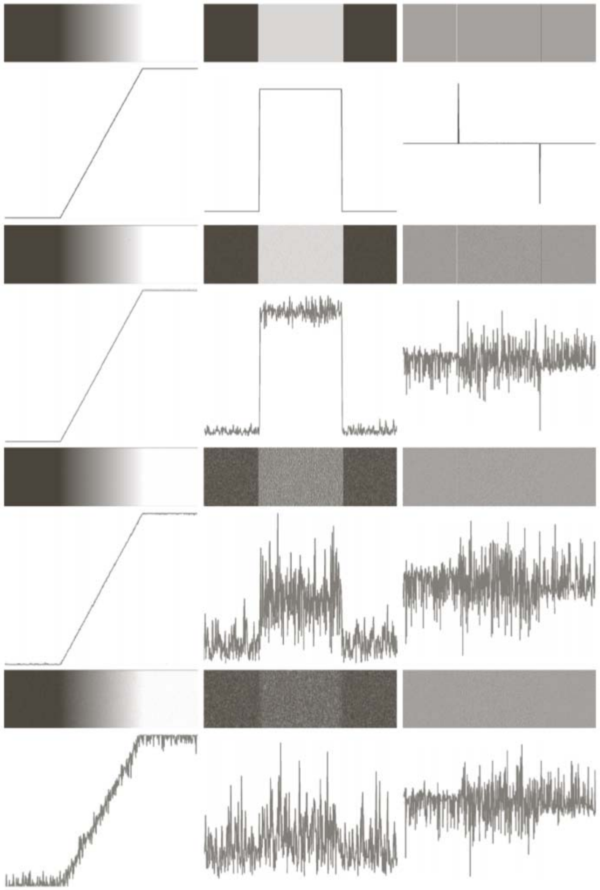


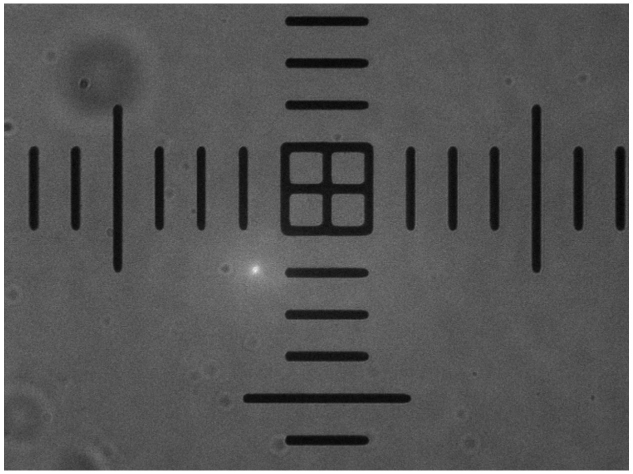
**FIGURE 10.11** First column: Images and intensity profiles of a ramp edge corrupted by random Gaussian noise of zero mean and standard deviations of 0.0, 0.1, 1.0, and 10.0 intensity levels, respectively. Second column: First-derivative images and intensity profiles. Third column: Second-derivative images and intensity profiles.

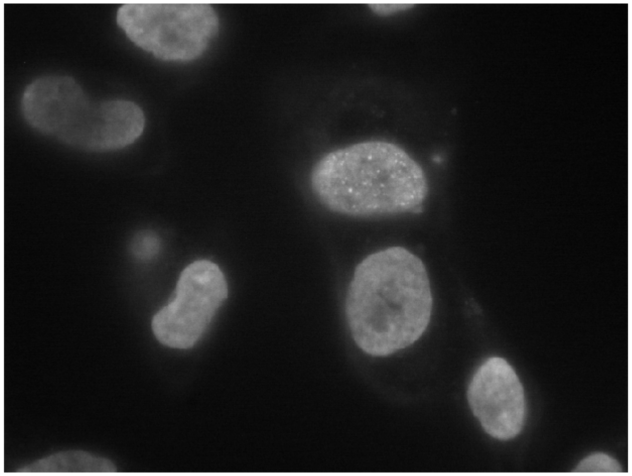
Gonzalez & Woods, DIP 3/e

# Edge Detection Procedure

- Step I: noise suppression
- Step II: edge enhancement
- Step III: edge localization

Canny, J., *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.





# Why Use Gaussian Kernel for Smoothing

- Gaussian kernel is not the only smoothing kernel.

- It has several important advantages:

  - Convolution of a Gaussian with another Gaussian is Gaussian.

  - Efficiency. Gaussian kernel is separable.

  - Repeated smoothing with a low-pass filter will eventually converge to Gaussian smoothing.

# Combination of Noise Suppression and Gradient Estimation (I)

- Notation:

  *J*: raw image;

  *I*: filtered image after convolution with Gaussian kernel G.

- A basic property of convolution

$$\frac{\partial (G * J)}{\partial x} = \frac{\partial I}{\partial x} = I_x = \frac{\partial G}{\partial x} * J \qquad \frac{\partial (G * J)}{\partial y} = \frac{\partial I}{\partial y} = I_y = \frac{\partial G}{\partial y} * J$$

$$E_s(x_0, y_0) = \sqrt{I_x^2(x_0, y_0) + I_y^2(x_0, y_0)}$$ ←Edge strength

$$E_o(x_0, y_0) = \arctan \frac{I_y(x_0, y_0)}{I_x(x_0, y_0)}$$ ←Edge orientation

# Combination of Noise Suppression and Gradient Estimation (II)

- Gaussian kernel in 1D

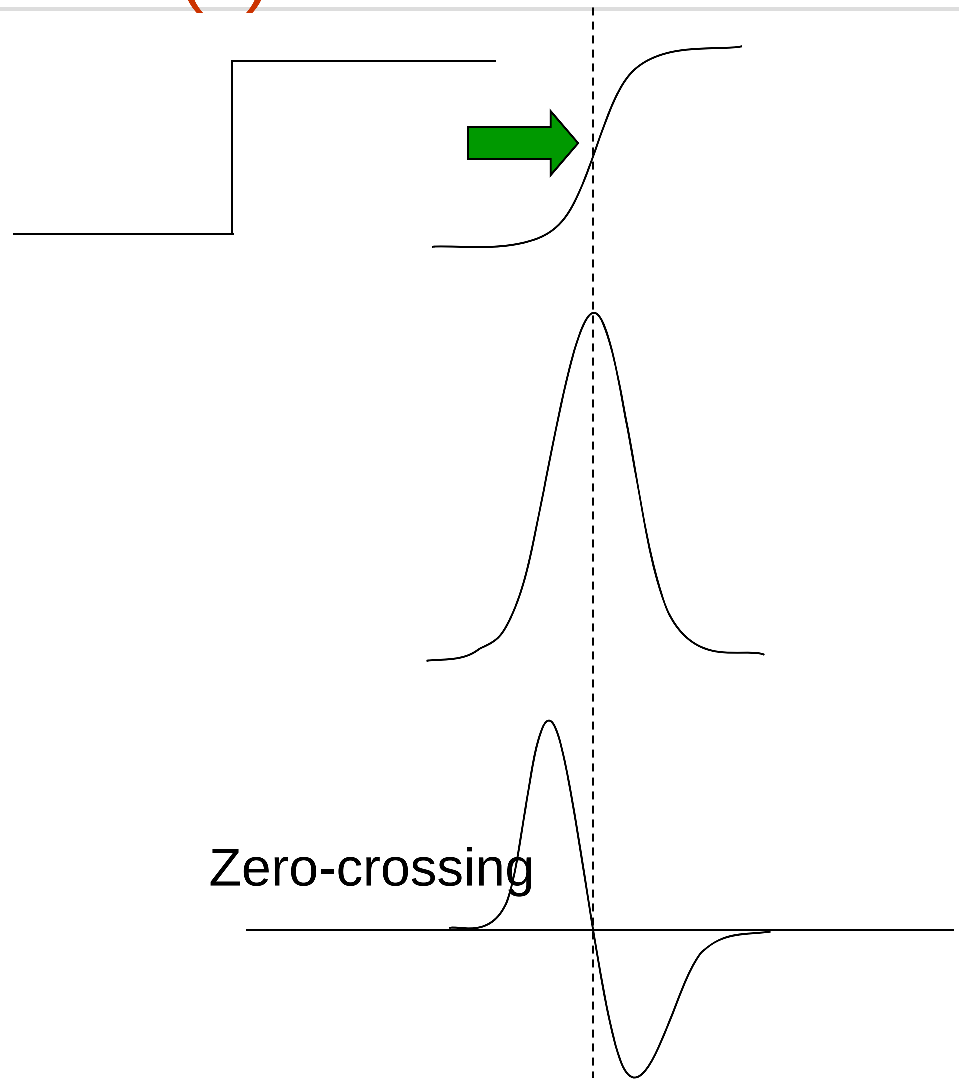
$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

- First order derivative

$$G'(x) = \frac{-x}{\sqrt{2\pi}\sigma^3} e^{-\frac{x^2}{2\sigma^2}}$$

- Second order derivative

$$G''(x) = \frac{-x}{\sqrt{2\pi}\sigma^3} e^{-\frac{x^2}{2\sigma^2}} \left[1 - \frac{x^2}{\sigma^2}\right]$$



# Combination of Noise Suppression and Gradient Estimation (III)

- Implementation

$$G(x,y;\sigma_x,\sigma_y)=\frac{1}{2\pi\sigma_x\sigma_y}\exp\left\{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)\right\}=\frac{1}{\sqrt{2\pi}\sigma_x}\exp\left\{-\frac{1}{2}\frac{x^2}{\sigma_x^2}\right\}\cdot\frac{1}{\sqrt{2\pi}\sigma_y}\exp\left\{-\frac{1}{2}\frac{y^2}{\sigma_y^2}\right\}$$

$$=G(x,\sigma_x)\cdot G(y,\sigma_y)$$

$$\frac{\partial G(x,y;\sigma_x,\sigma_y,\theta)}{\partial x}=\frac{dG(x,\sigma_x)}{dx}G(y,\sigma_y)$$

$$\frac{\partial G(x,y;\sigma_x,\sigma_y,\theta)}{\partial y}=G(x,\sigma_x)\frac{dG(y,\sigma_y)}{dy}$$

- Advantages:

  - Reduced computational cost
  - Calculation of gradient can run in parallel in both directions

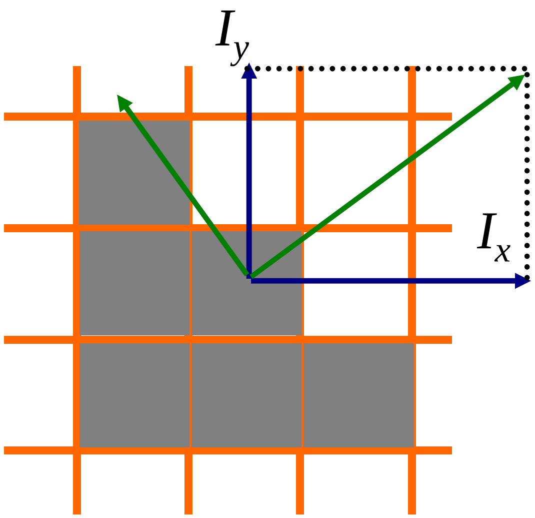
# Edge Enhancement

- Step I: For each pixel $I(x_0,y_0)$, calculate the gradient

$$\left.\frac{\partial I}{\partial x}\right|_{x=x_0} \qquad \left.\frac{\partial I}{\partial y}\right|_{y=y_0}$$

- Step II: Estimate edge strength

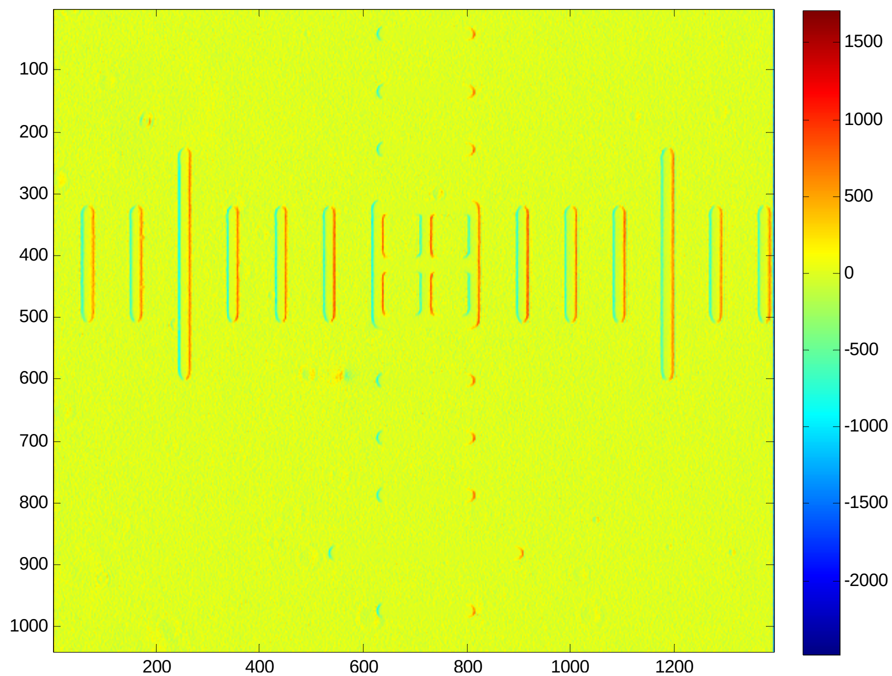
$$E_s\left(x_0,y_0\right)=\sqrt{I_x^2\left(x_0,y_0\right)+I_y^2\left(x_0,y_0\right)}$$

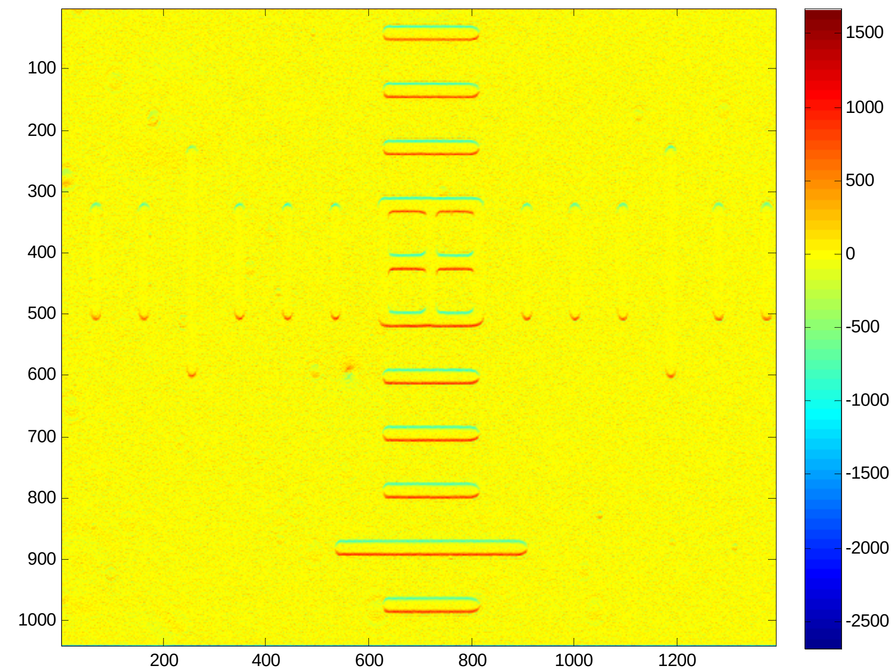


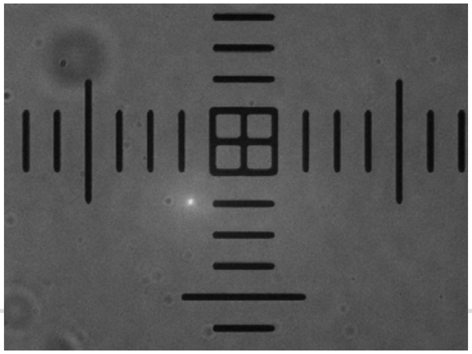

- Step III: Estimate edge direction

$$E_o\left(x_0,y_0\right)=\arctan\frac{I_y\left(x_0,y_0\right)}{I_x\left(x_0,y_0\right)}$$

# Calculation of Image Gradient



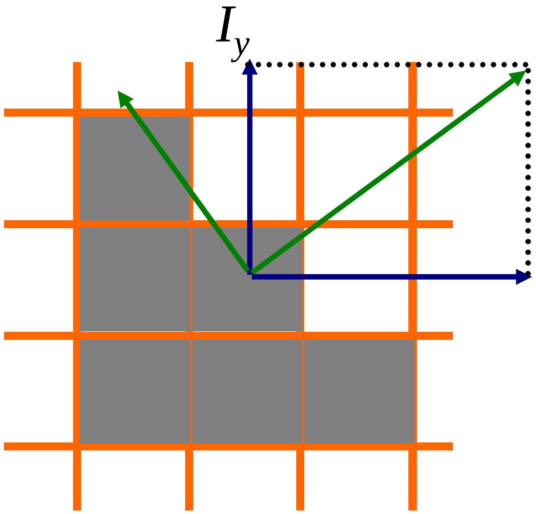



# Non-Maximum Suppression

- For each pixel $I(x_0,y_0)$,compare the edge strength along the direction perpendicular to the edge

- An edge point must have its edge strength no less than its two neighbors.



# Hysteresis Thresholding

- The main purpose is to link detected edge points while minimizing breakage.

- Basic idea

  - Using two thresholds $T_L$ and $T_H$

  - Starting from a point where edge gradient magnitude higher than $T_H$

  - Link to neighboring edge points with edge gradient magnitude higher than $T_L$

# Influence of Scale Selection on Edge Detection

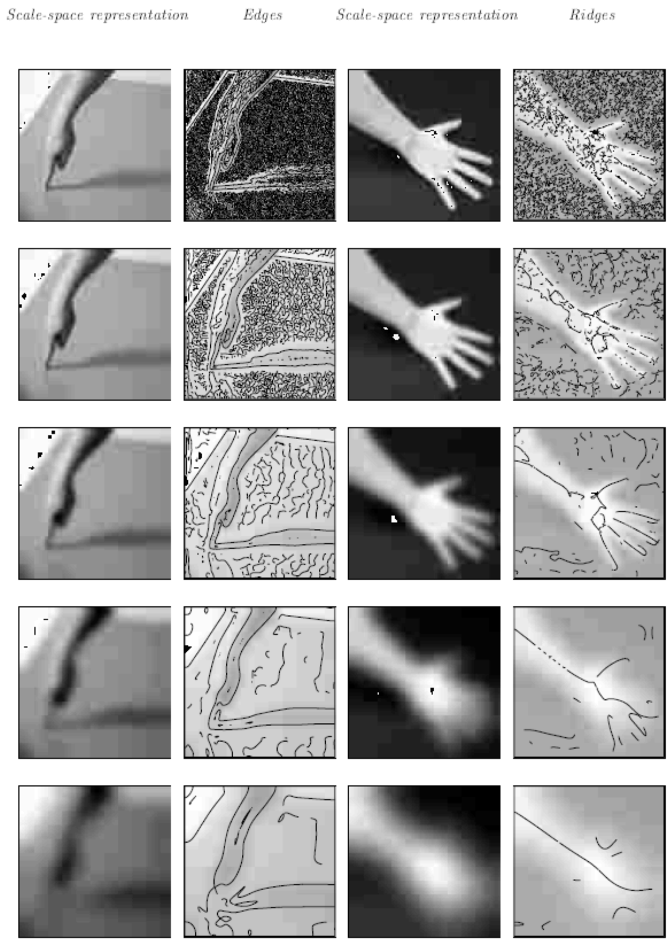



**Figure 1.4:** *Edges and ridges computed at different scales in scale-space (scale levels $t = 1.0$, $4.0$, $16.0$, $64.0$ and $256.0$ from top to bottom) using a differential geometric edge detector and ridge detector, respectively. (Image size: 256*256 pixels.)*

*Lindeberg (1999) [`` Principles for automatic scale selection'',] in: B. J"ahne (et al., eds.), Handbook on Computer Vision and Applications, volume 2, pp 239--274, Academic Press.*

# Edge Detection Demo

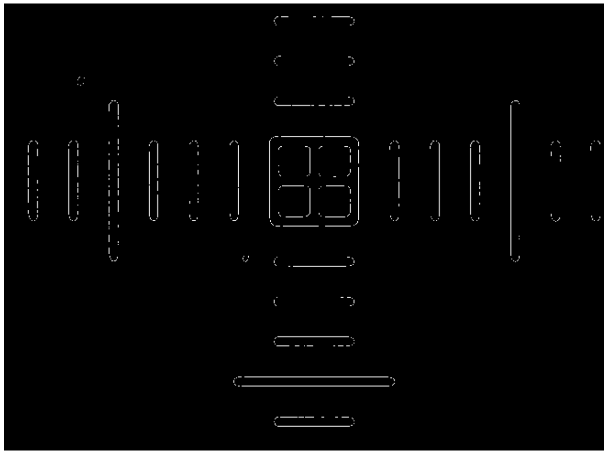
- Review: low-level feature detection
- Overview of edge detection
- Line/curve detection using the Hough transform

# Line/Curve Detection by

- A variety of techniques are available.

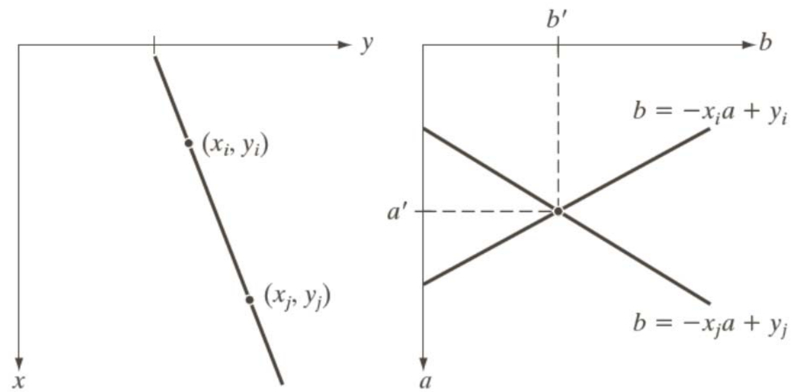
  - Spatial domain
    - e.g.by edge point detection and grouping

  - Transform domain
    - e.g. by Hough transform



# Basic Concept of Hough Transform

- A simple example: representing the lines passing through ($x_0$, $y_0$) in the parameter space.




$$y_1 = a \cdot x_1 + b \qquad y_2 = a \cdot x_2 + b$$

$$b = (-x_1)a + y_1 \qquad b = (-x_2)a + y_2$$

$$y_1 = \tilde{a} \cdot x_1 + \tilde{b}$$

$$y_2 = \tilde{a} \cdot x_2 + \tilde{b}$$

The two lines in the transform domain must intersect at $(a, b)$

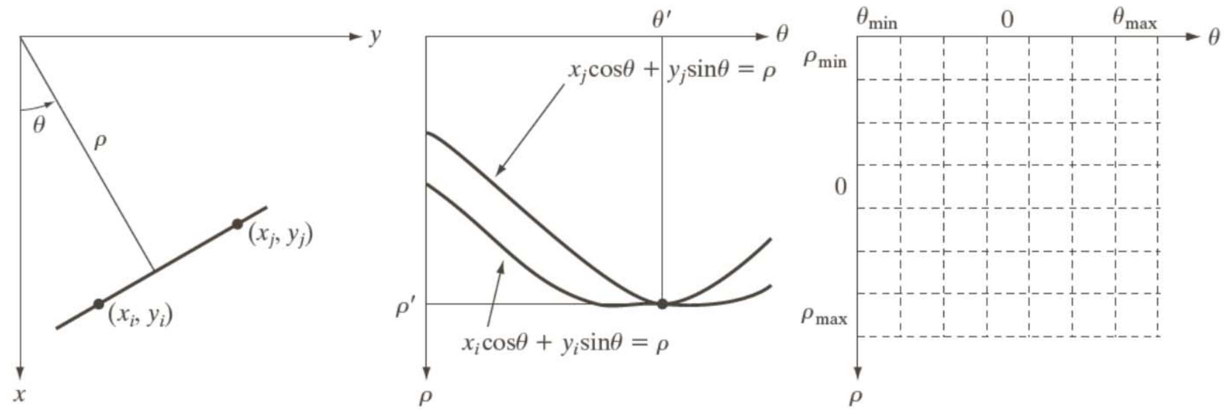
# HT Algorithm Implementation Details

- Parameterization using y=mx+n fails for the case of vertical lines.

- A different way of parameterization:

$$\rho = x \cdot cos\,\theta + y \cdot sin\,\theta$$

- Exhaustive search the space of [ρ, θ] can be time-consuming.



# Generalization of the HT Algorithm for Curve Detection

- The HT algorithm is a voting algorithm. The key idea is to convert a (difficult) pattern recognition problem into a (simple) peak detection problem.

- Hough transform can be generalized to detect circles, ellipses, or any curve that can be parameterized.

- Examples

Circles with known radius but unknown center

$$(x-x_c)^2+(y-y_c)^2=R^2 \Longrightarrow (x_c-x_i)^2+(y_c-y_i)^2=R^2$$

Ellipses with known major and minor semi-axes but unknown center

$$\frac{(x-x_c)^2}{a^2}+\frac{(y-y_c)^2}{b^2}=1 \Longrightarrow \frac{(x_c-x_i)^2}{a^2}+\frac{(y_c-y_i)^2}{b^2}=1$$

# Evaluation of Parametric Transform Based Curve Detection

- The curve to be detected can be of arbitrary form as long as it can be parameterized.

- Strengths:
  - Handles occlusion and partial line/curves well.
  - Relatively robust to noise
  - Capable of detecting multiple instances

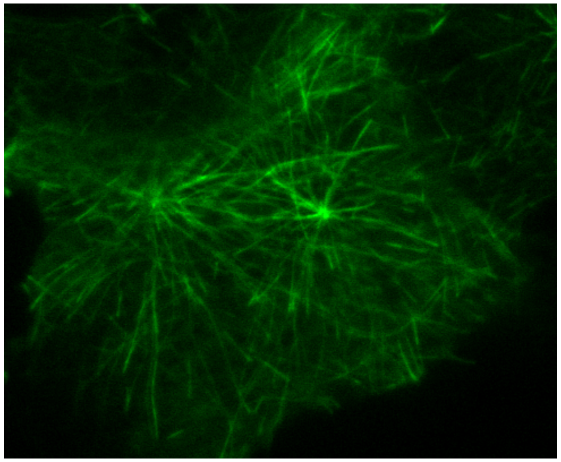
- Limitations

  → For curves with multiple parameters, the voting/search can be costly.

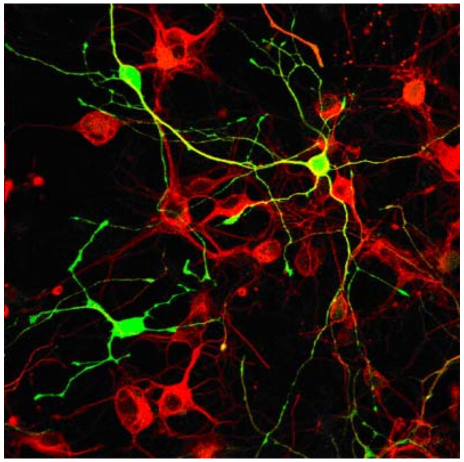
  → Other shapes can also generate spurious peaks.

# Comments on Line/Curve Detection

- Curves features are very common in bioimages.



microtubules



neurons

- General curve feature detection will be addressed in the next lecture.

**Questions?**