Image Derivatives and Edge Detection

Albert Chung
Department of Computer Science
University of Exeter

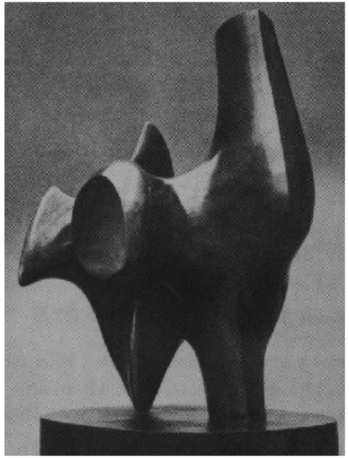
Spring Term 2023

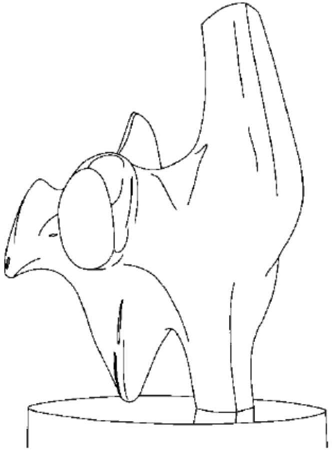
Main Reference

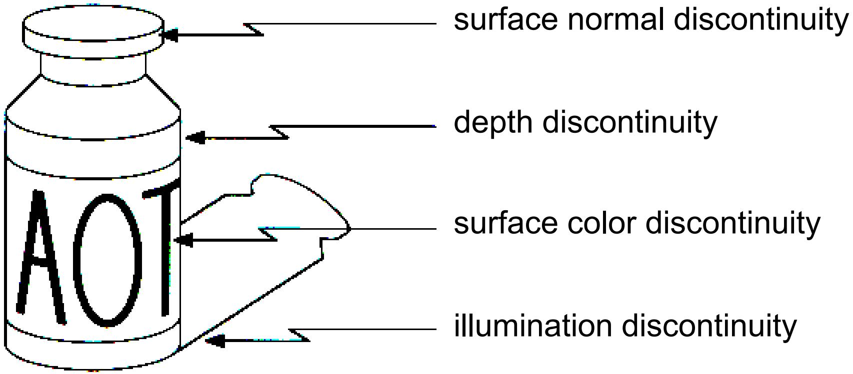
- R. Szeliski, “Computer Vision: Algorithms and Applications” (2nd Edition):
  - Image gradient: Chapter 3.2
  - Edges and contours: Chapter 7.2

Edge detection

- Convert a 2D image into a set of edge segments (curves)
  - Extracts salient features of the scene
  - More compact than pixels
- Edges are caused by a variety of factors







Characterizing edges

- An edge is a place of rapid change in the image intensity function

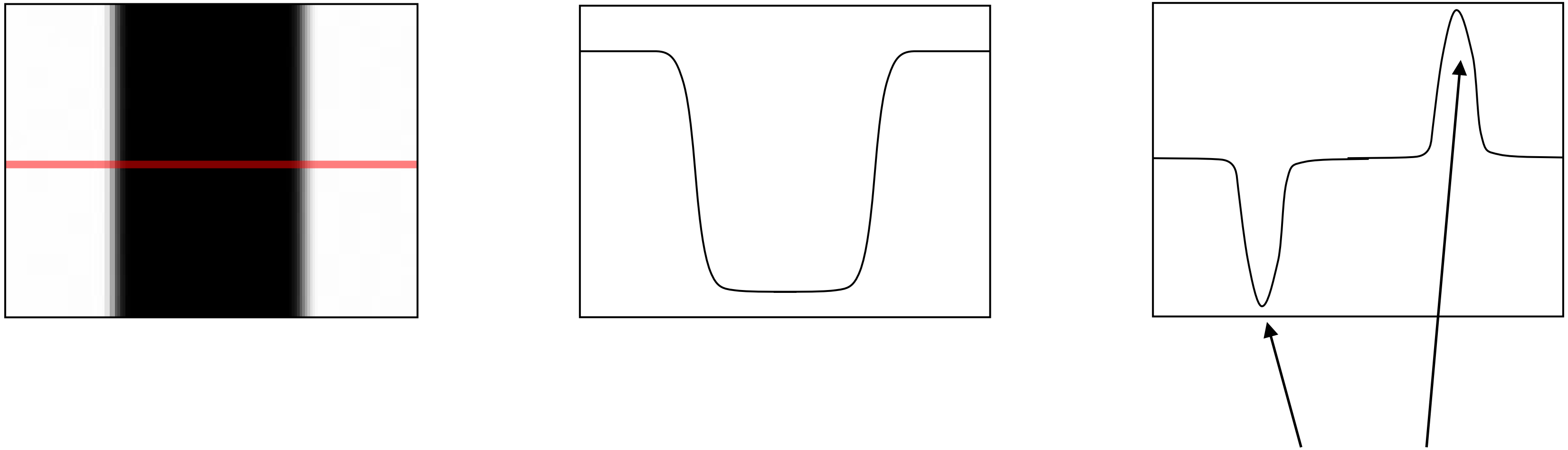


Image derivatives

- How can we differentiate a digital image F[x,y]?
  - Option 1: reconstruct a continuous image, f, then compute the partial derivative
  - Option 2: take discrete derivative (finite difference)

$$\frac{\partial f}{\partial x}[x,y] \approx F[x+1,y] - F[x,y]$$

- How would you implement this as a linear filter?

$\frac{\partial f}{\partial x}$:

1	-1	

H_x

$\frac{\partial f}{\partial y}$:

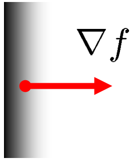
	-1	
	1	

H_y

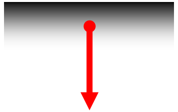
Source: S. Seitz

Image gradient

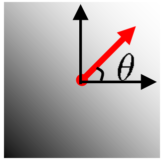
- The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$
- The gradient points in the direction of most rapid increase in intensity:



$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$



$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$



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

- The edge strength is given by the gradient magnitude:

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

- The gradient direction is given by:

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} \Big/ \frac{\partial f}{\partial x}\right)$$

- how does this relate to the direction of the edge?

Sobel operator

- Common approximation of image derivative operator

$$\frac{1}{8}\begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \qquad \frac{1}{8}\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$$

$$s_x \qquad\qquad s_y$$

- Standard Sobel operator omits the 1/8 term
  - No difference for edge detection
  - Only needed for right gradient magnitude

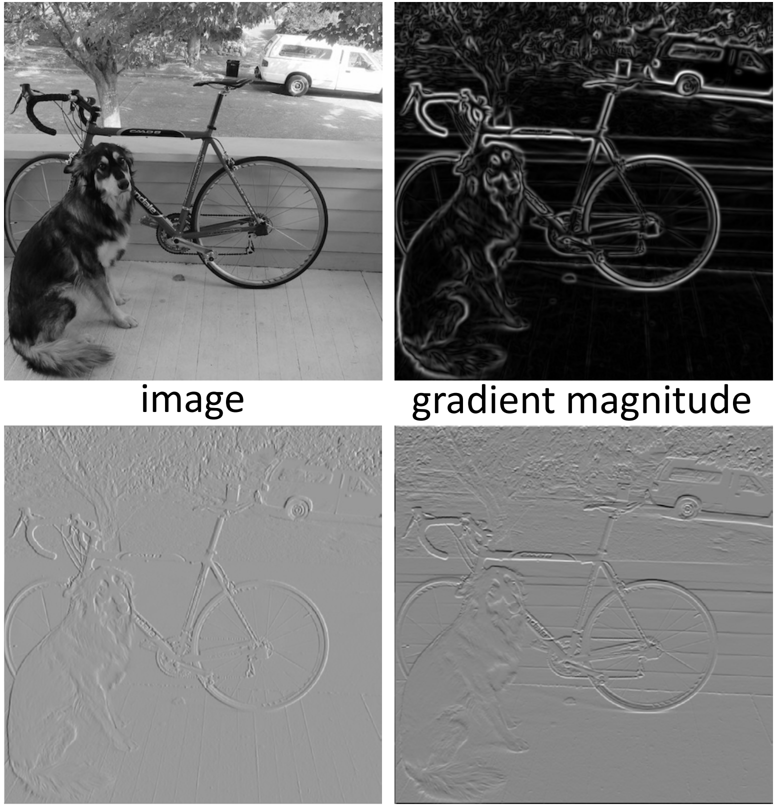
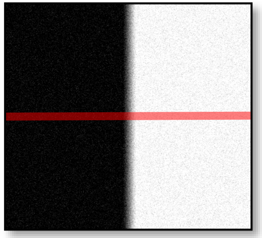


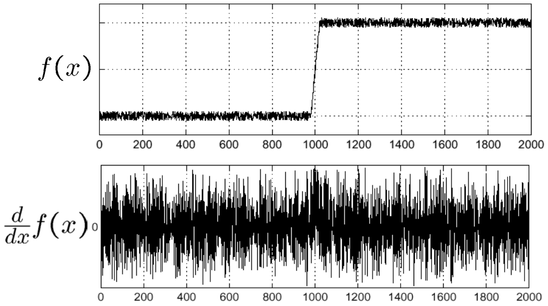
image | gradient magnitude

$\frac{\partial f}{\partial x}$ | $\frac{\partial f}{\partial y}$

Sobel operator: approximation of Gaussian derivative



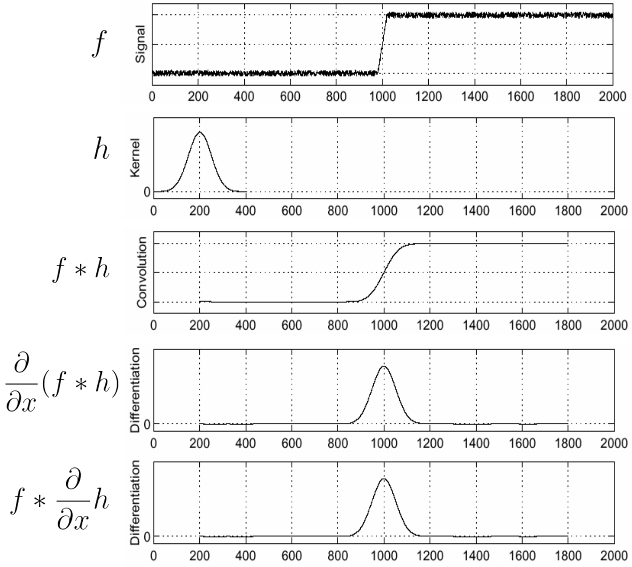
Noisy input image

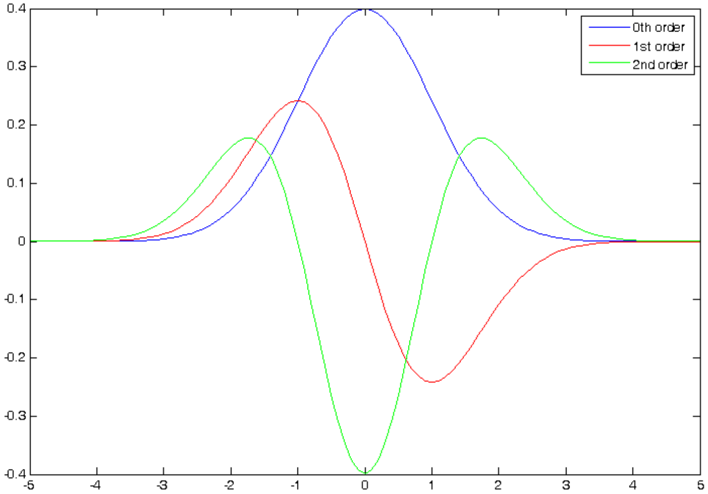


$$G_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

$$G'_\sigma(x) = \frac{d}{dx} G_\sigma(x) = -\frac{1}{\sigma}\left(\frac{x}{\sigma}\right) G_\sigma(x)$$

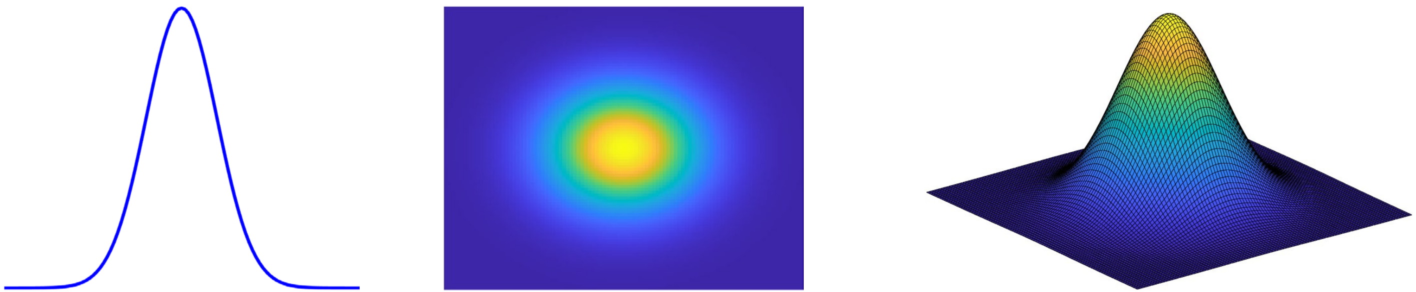
$$G''_\sigma(x) = \frac{d^2}{dx^2} G_\sigma(x) = -\frac{1}{\sigma^2}\left(1 - \frac{x^2}{\sigma^2}\right) G_\sigma(x)$$



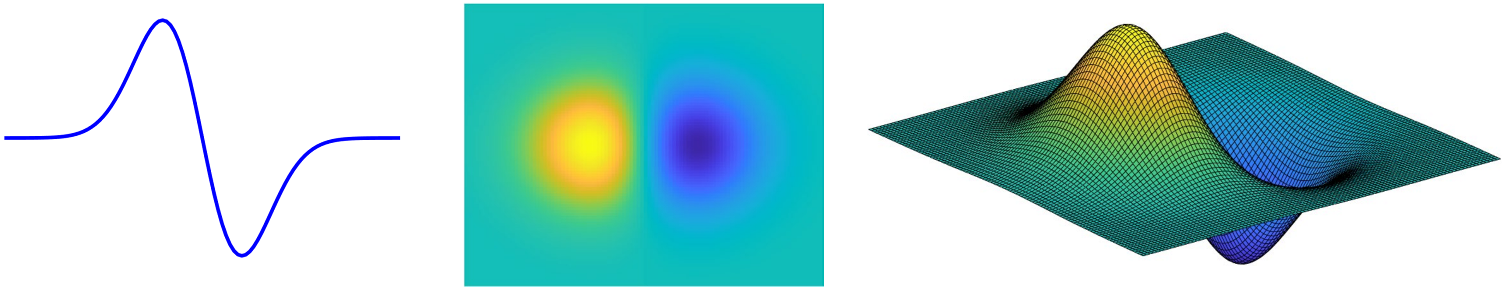


Gaussian and Gaussian derivatives

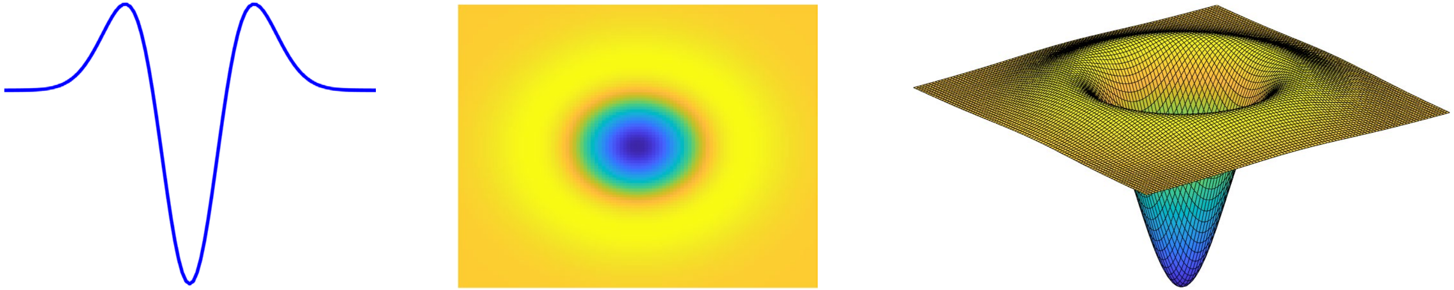
Gaussian



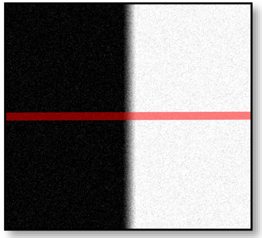
1^{st} Order Derivative



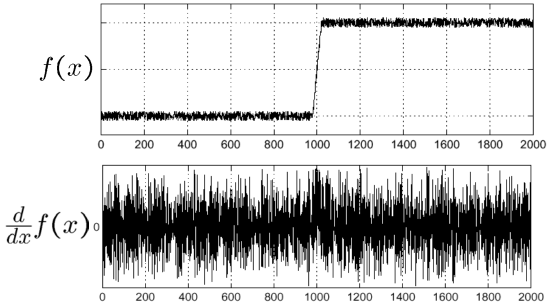
2^{nd} Order Derivative

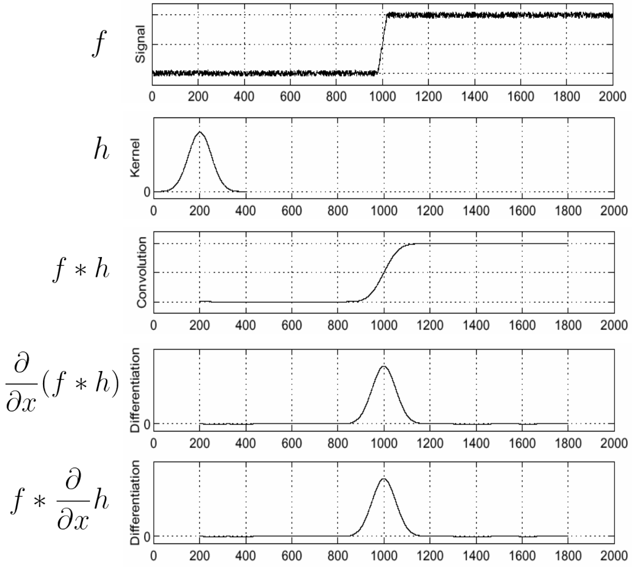


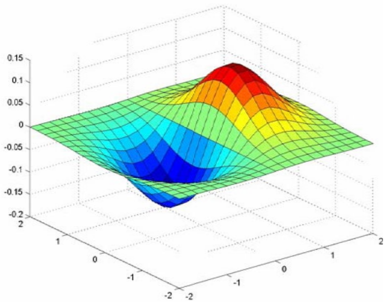
Sobel operator: approximation of Gaussian derivative

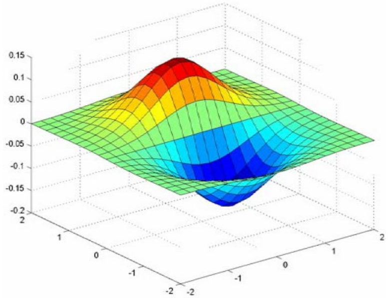


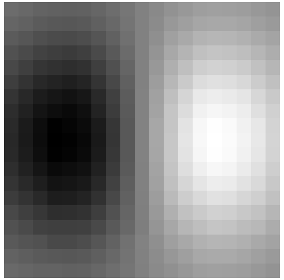
Noisy input image

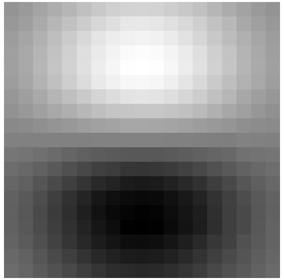












$\frac{1}{8}$

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

s_x

$\frac{1}{8}$

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

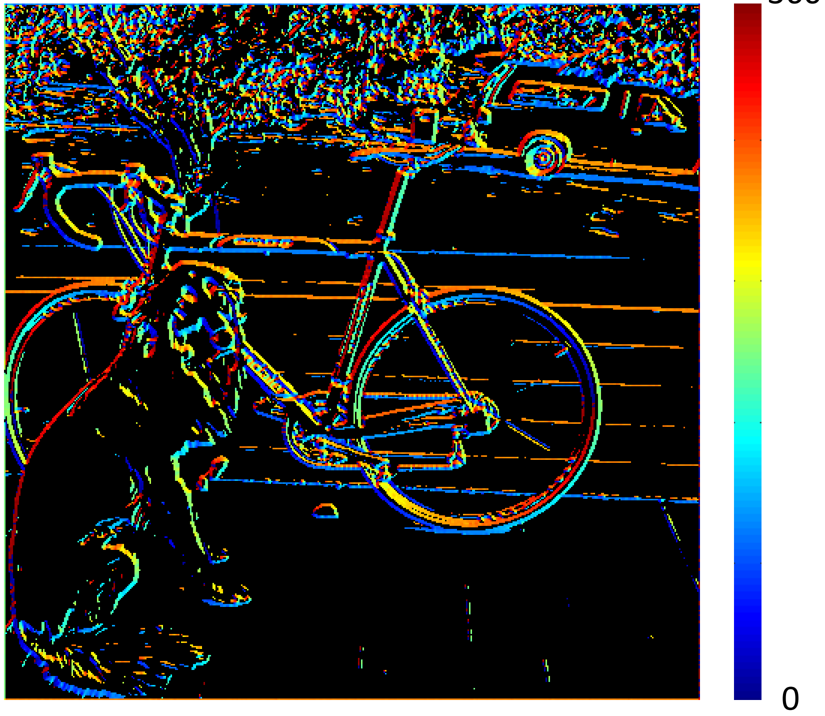
s_y

Finding edges

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

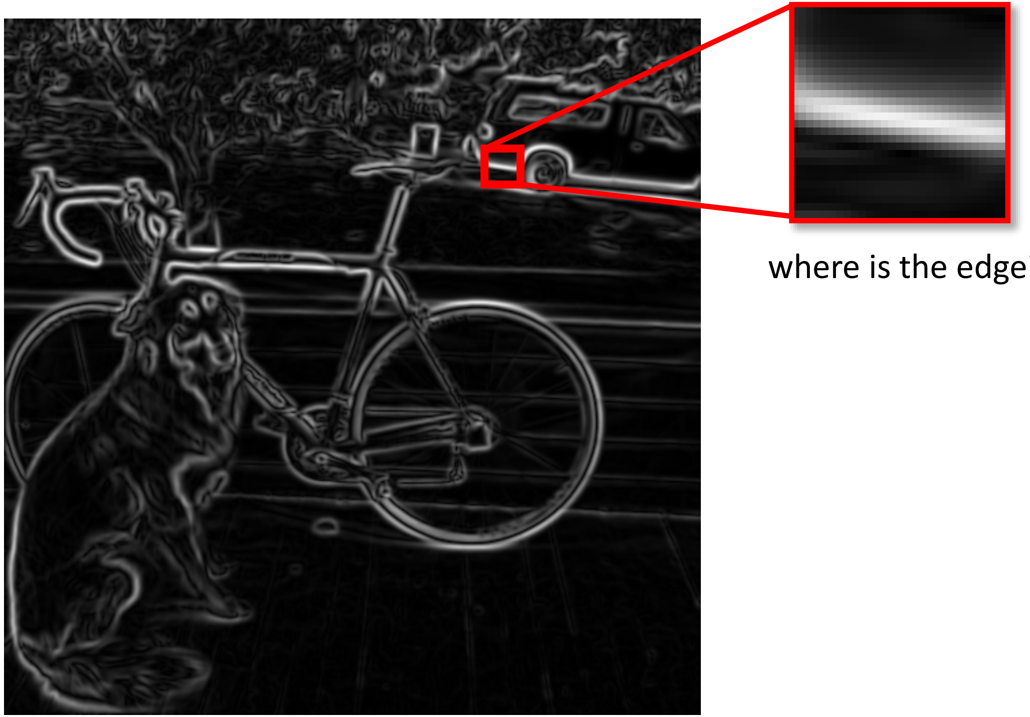
$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

Gradient magnitude



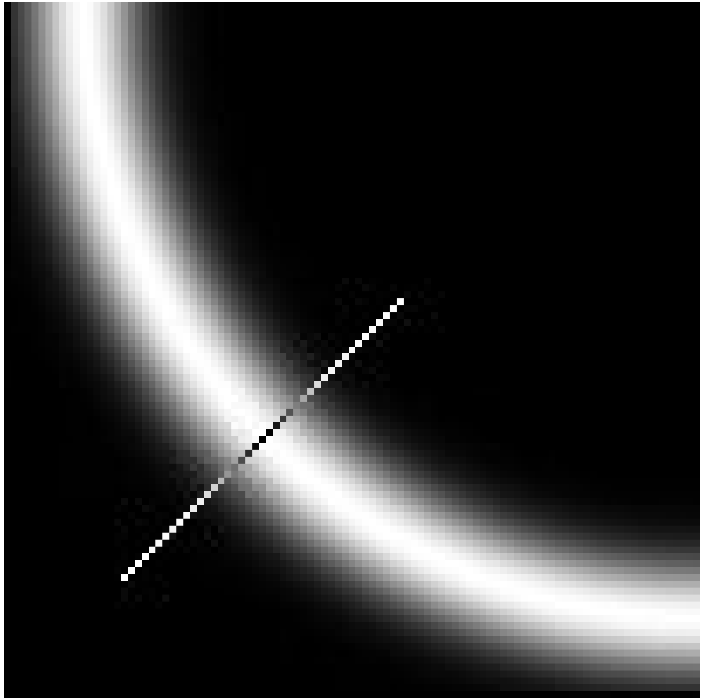
Gradient orientation

Finding edges: Canny edge detection



where is the edge?

Gradient magnitude



Non-maximal suppression

Nonmaximum suppression

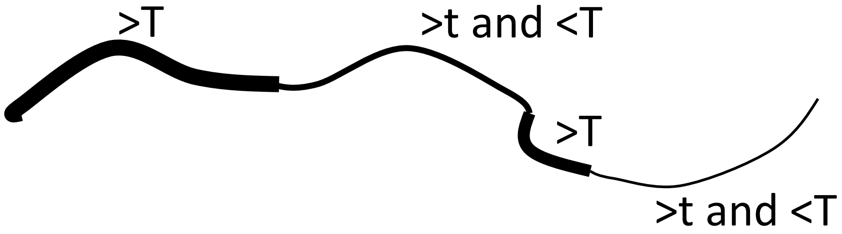
Nonmaxima suppression

After Non-maximal Suppression

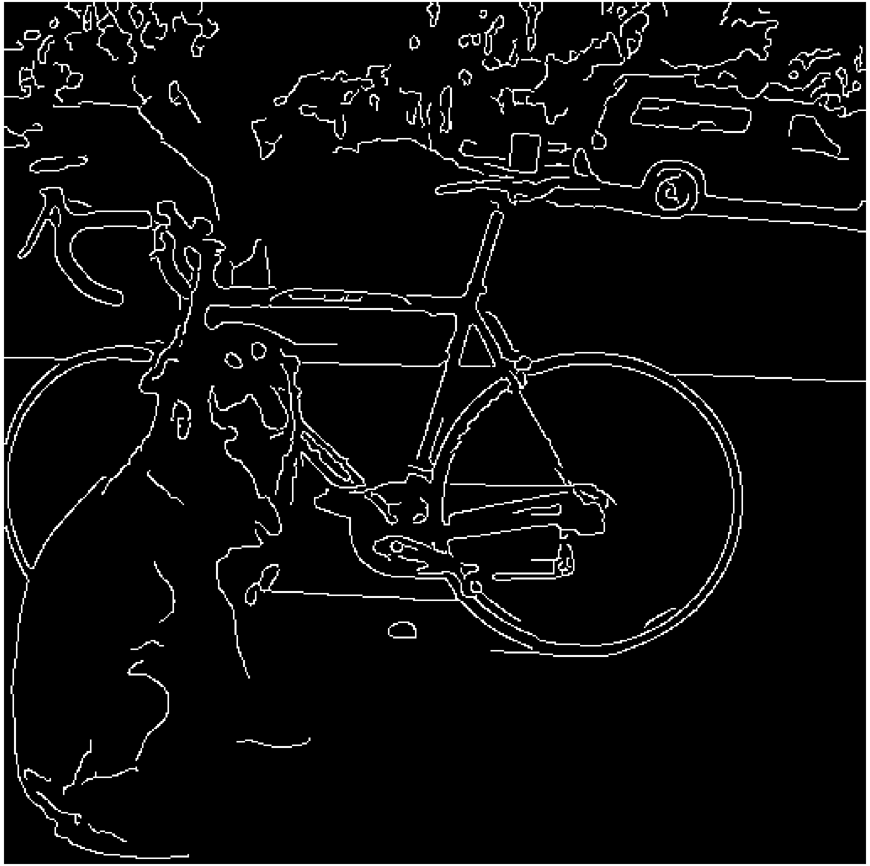
- Still some noise
- Only want strong edges
- 2 thresholds, 3 cases
  - R > T: strong edge
  - R < T but R > t: weak edge
  - R < t: no edge

Why two thresholds: connecting edges

- Strong edges are edges!
- Weak edges are edges if and only if they connect to strong
- Look in some neighborhood of pixel (usually 8 closest)



Canny edge detector



- Depends on several parameters:
  - High threshold: T
  - Low threshold: t
  - Width of Gaussian blur: σ
- Still a widely used edge detector in computer vision

J. Canny, A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

Canny edge detector

- The choice of σ depends on desired behavior
  - Large σ detects “large-scale” edges
  - Small σ detects fine edges

image | $\sigma = 1$ | $\sigma = 2$ | $\sigma = 3$ | $\sigma = 4$ | $\sigma = 5$

Similar filters

$$S_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \qquad S_x = \begin{bmatrix} +1 & 0 & -1 \\ +1 & 0 & -1 \\ +1 & 0 & -1 \end{bmatrix} \qquad \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$S_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad S_y = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \qquad \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$$

Sobel filter | Prewitt filter | Roberts filter

Edge detection with zero-crossings of LoG filter

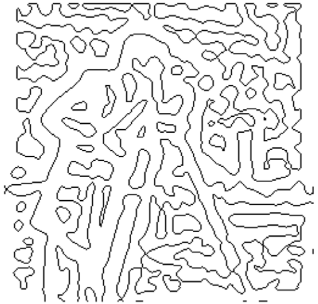
- sub-pixel **edge elements** (edgels) from zero-crossings:
  - look for **adjacent pixels** x_i and x_j where the LoG S(x) changes value
  - compute zero-crossing of the line connecting $S(x_i)$ and $S(x_j)$:

$$\boldsymbol{x}_{\mathrm{z}} = \frac{\boldsymbol{x}_i S(\boldsymbol{x}_j) - \boldsymbol{x}_j S(\boldsymbol{x}_i)}{S(\boldsymbol{x}_j) - S(\boldsymbol{x}_i)}$$


input $I(\boldsymbol{x})$


LoG $[\nabla^2 G_\sigma](\boldsymbol{x}) * I(\boldsymbol{x})$


Sign of LoG


Zero-crossings of LoG

Summary

- Edge is a place of rapid change in the image intensity function
- Image derivatives: 1^{st} and 2^{nd} order Gaussian derivatives
- Image operators or filters as approximation of Gaussian derivatives
- Steps of Canny edge detectors
- Other similar filters: Prewitt and Roberts filters
- Edge detection by finding position of zero-crossing on LoG responses

Questions?