SAIR

Spatial AI & Robotics Lab

CSE 473/573-A

L7: EDGE DETECTION

Chen Wang

Spatial AI & Robotics Lab

Department of Computer Science and Engineering

University at Buffalo The State University of New York

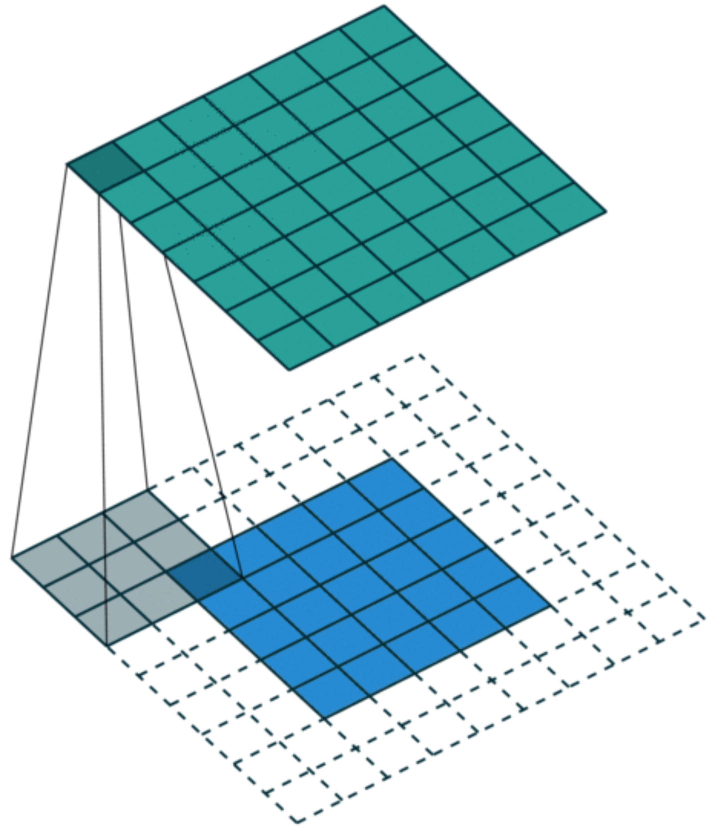
Content

- Template Matching
  - Cross-correlation
- Edge Detection
  - Image differentiation and gradient
  - Derivative theorem of convolution
  - Derivative of Gaussian filter, Laplacian of Gaussian
  - 2D edge detection filters
  - Canny edge detector, Hysteresis thresholding

TEMPLATE MATCHING

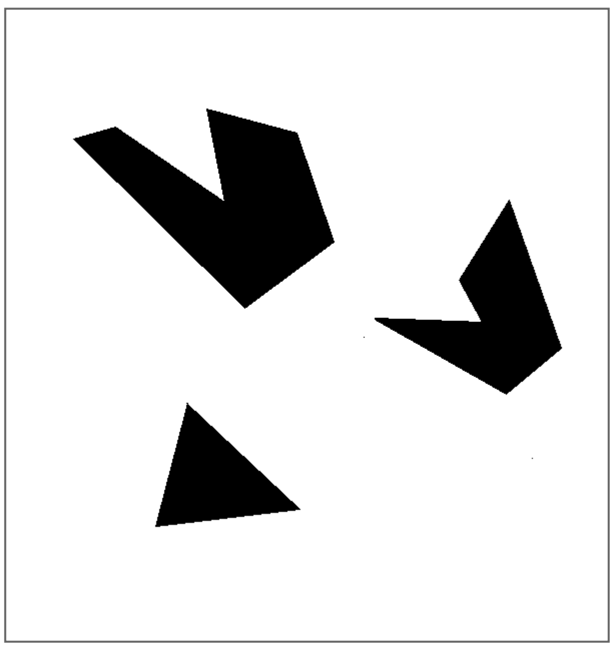
Similarity/Distance of Signals

- L1-norm / Manhattan distance
  - $|\boldsymbol{x}-\boldsymbol{w}|_1$
- L2-norm / Euclidean distance
  - $||\boldsymbol{x}-\boldsymbol{w}||_2$
- Inner Product
  - $\boldsymbol{x}\cdot\boldsymbol{w}$
- Cosine Similarity
  - $\frac{\boldsymbol{x}\cdot\boldsymbol{w}}{||\boldsymbol{x}||_2||\boldsymbol{w}||_2}$
- Filtering gives us a kind of similarity measurement, i.e., inner product.

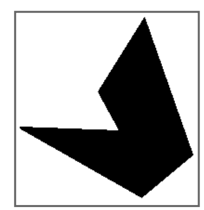


Template matching

- Each element of the output is a similarity measure of a specific pattern, i.e., a filter or a template.
- Each similarity measure is also called a ``response''.
- This process is called template matching.



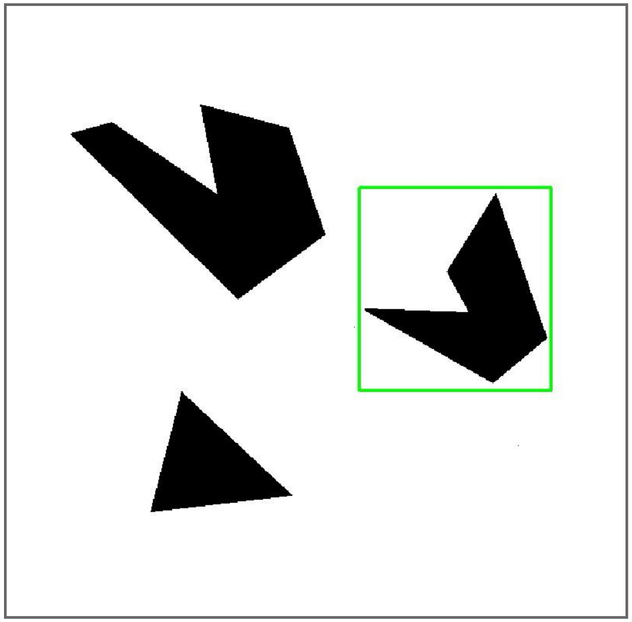
A toy example



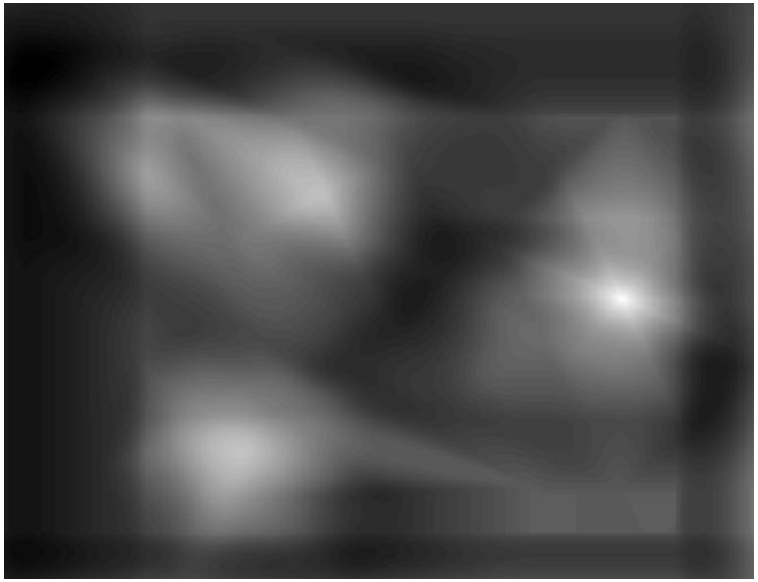
Template (mask)

Is there only one match?
What if the pattern is not exact?

Correlation of Template and Imgae



Detected template



Correlation map

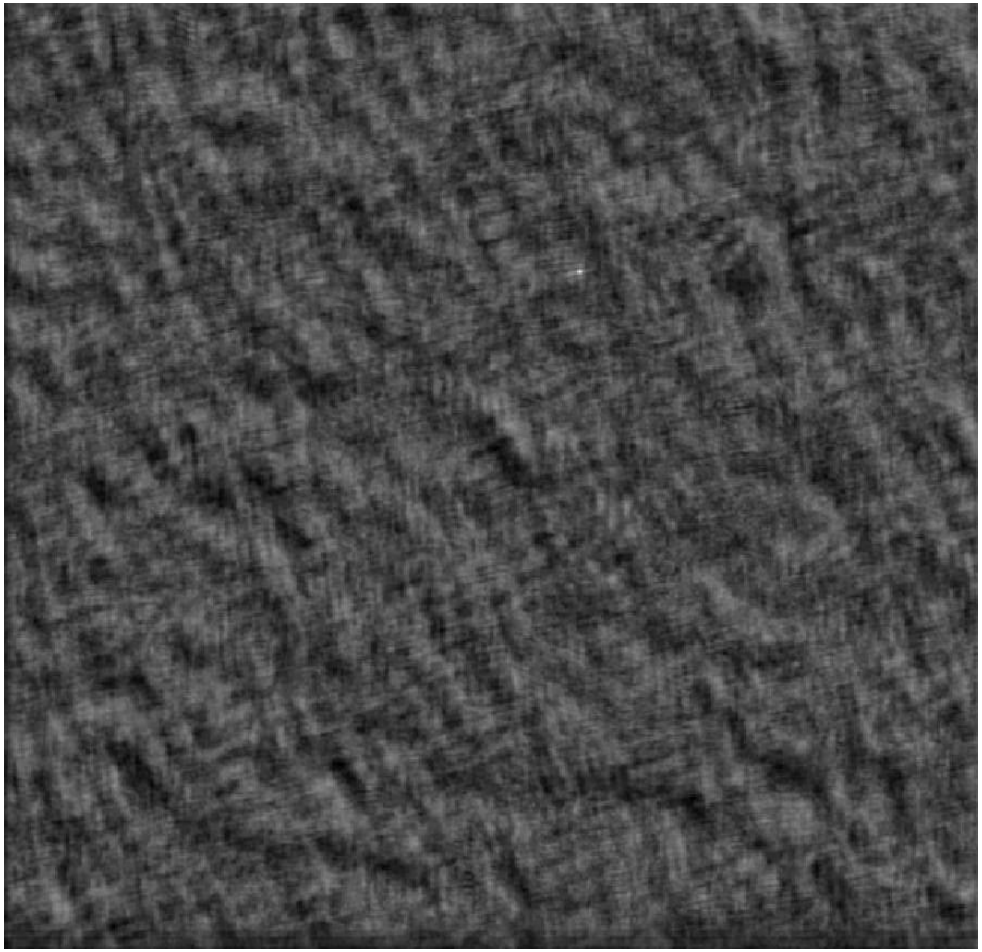
Is there only one match?
What if the pattern is not exact?

Where’s Waldo?

Scene

Template

Where's Waldo?

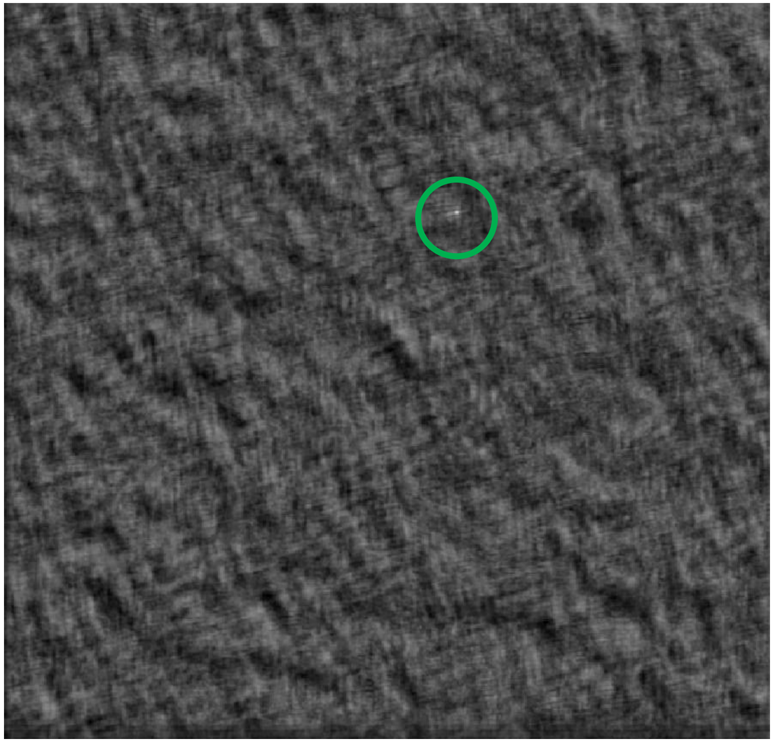


Scene

Template

Where's Waldo?

Detected template



Correlation map

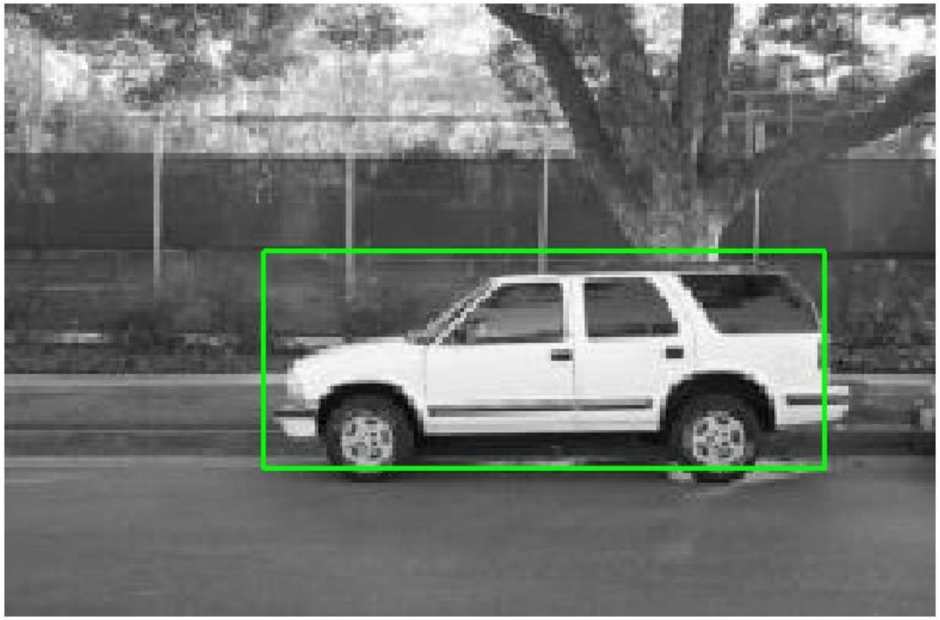
9

- Use normalized cross-correlation score to find a given pattern (template) in the image (Szeliski Eq. 8.11 in textbook).
- Normalization needed to control for relative brightness.

SAIR
Spatial AI & Robotics Lab

Template matching

- Match can be meaningful, if scale, orientation, and general appearance is right.



Detected template

Template

Template matching

- What if the template is not identical to some sub image in the scene?

Scene

Template

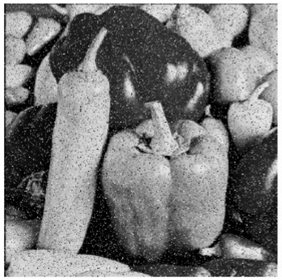
Template matching

- We need more flexible, powerful and forgiving representations.
  - Bolme, D. S., Beveridge, J. R., Draper, B. A., & Lui, Y. M. Visual object tracking using adaptive correlation filters. CVPR, 2010.
    - Computational complexity: $\mathbb{O}(N \log N)$
    - Equivariant to translation, robust to small appearance variance.
  - Wang, C., Zhang, L., Xie, L., & Yuan, J. (2018, April). Kernel cross-correlator. *AAAI, 2018*.
    - Nonlinear cross-correlation with the kernel trick.
    - Equivariant to any transforms:
      - Translation, Scale, Rotation, Affine, etc.
    - Same computational complexity with linear filter: $\mathbb{O}(N \log N)$

EDGE DETECTION

Filters for features

- Previously, filtering is a way to
  - Remove or reduce **noise.**
  - **Template matching**
- Filters also allows us to abstract higher-level “**features**”.
  - Map raw pixels to intermediate representations used for **subsequent processing**.
  - Reduce amount of data, discard redundancy, preserve useful information.



Edge detection

- **Goal**: map image from 2D array of pixels to a set of **curves** or **line segments**, or **contours**.
- **Why?**

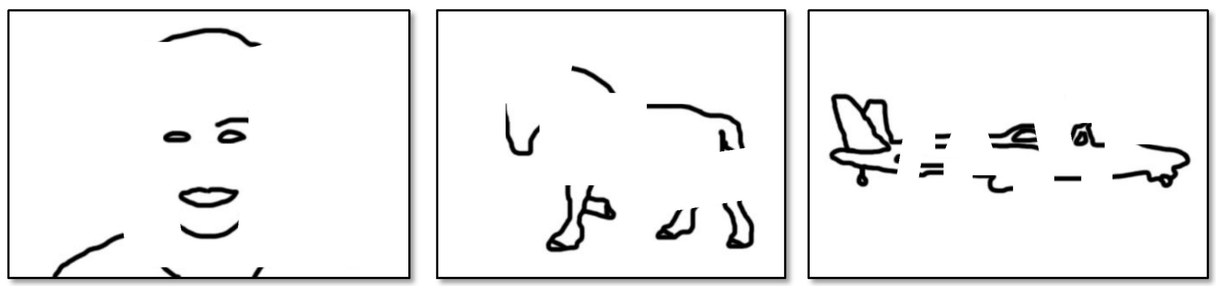


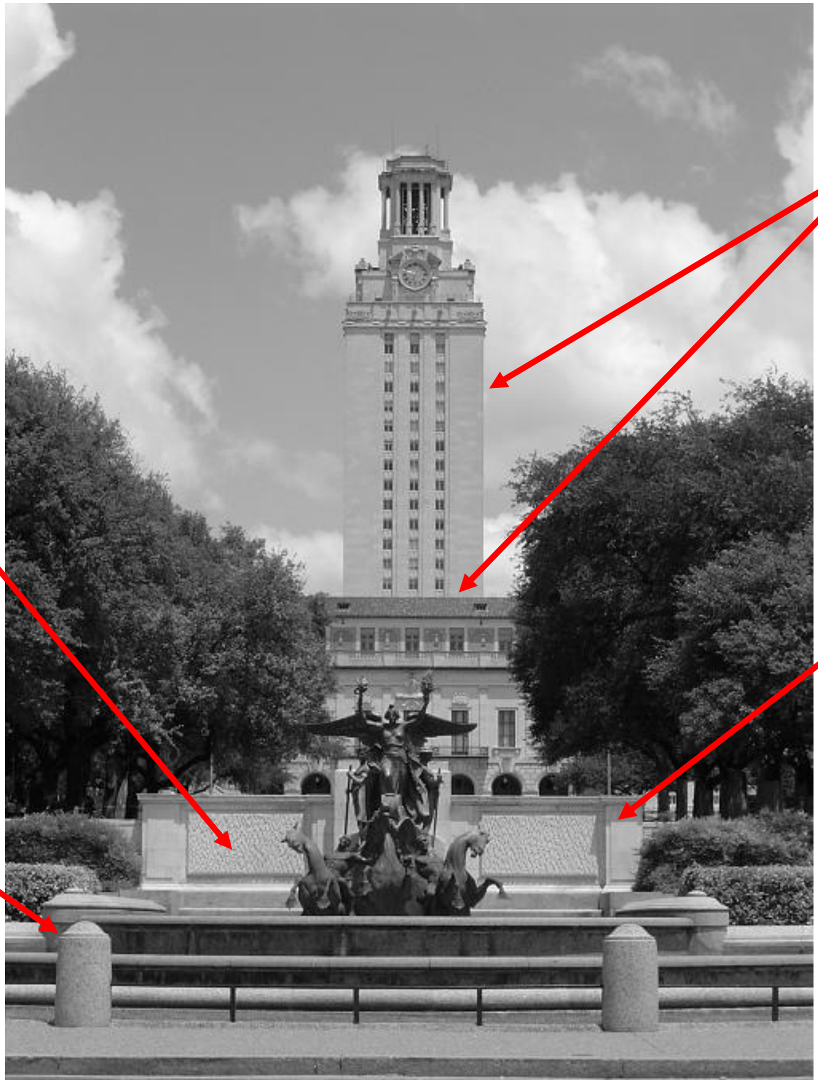
Figure from J. Shotton et al., PAMI 2007

- **Even simple sketch of the objects are quite meaningful.**
- **Main idea**: look for strong gradients, post-process.

What can cause an edge?

Reflectance change: appearance information, texture

Change in surface orientation: shape

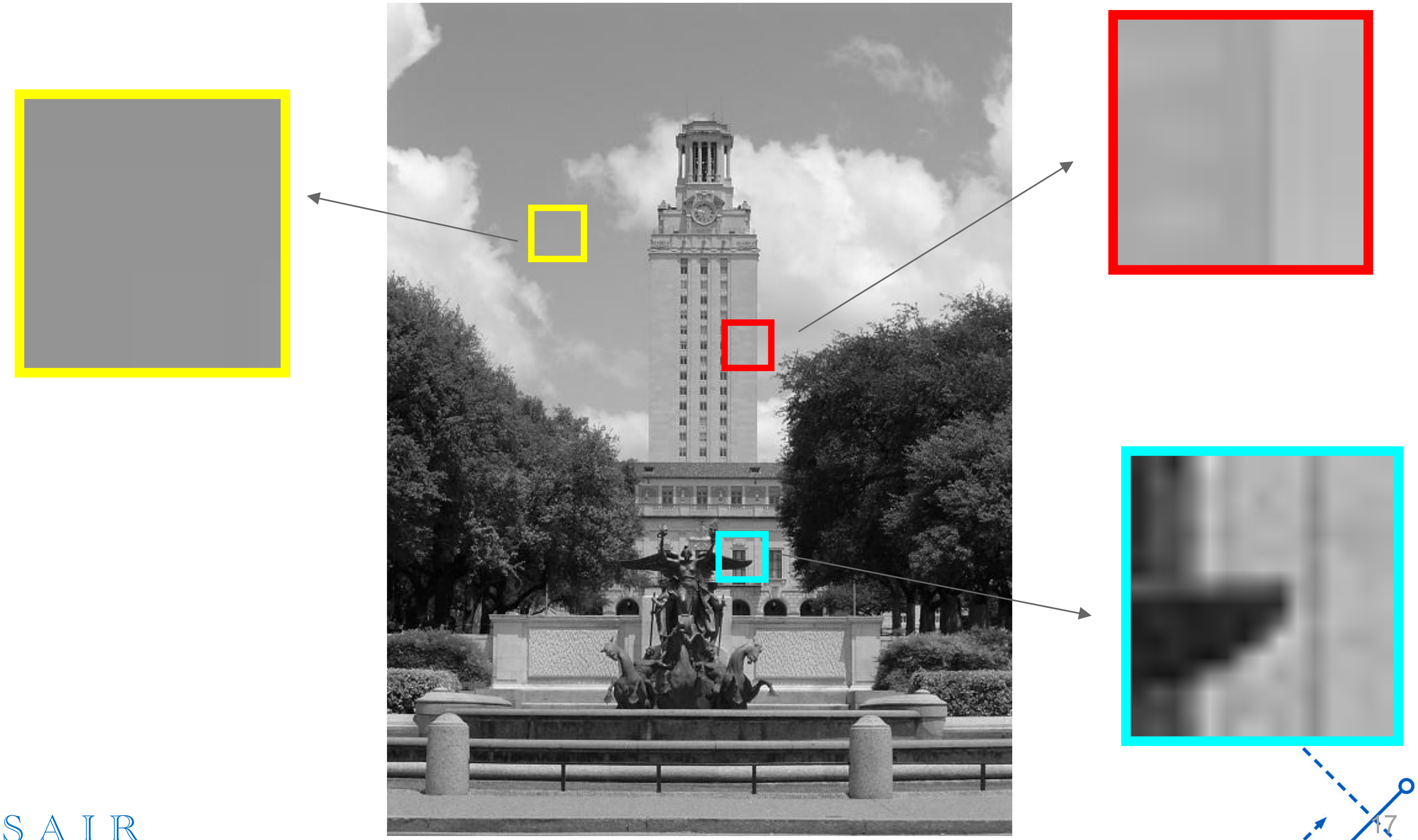


Depth discontinuity: object boundary

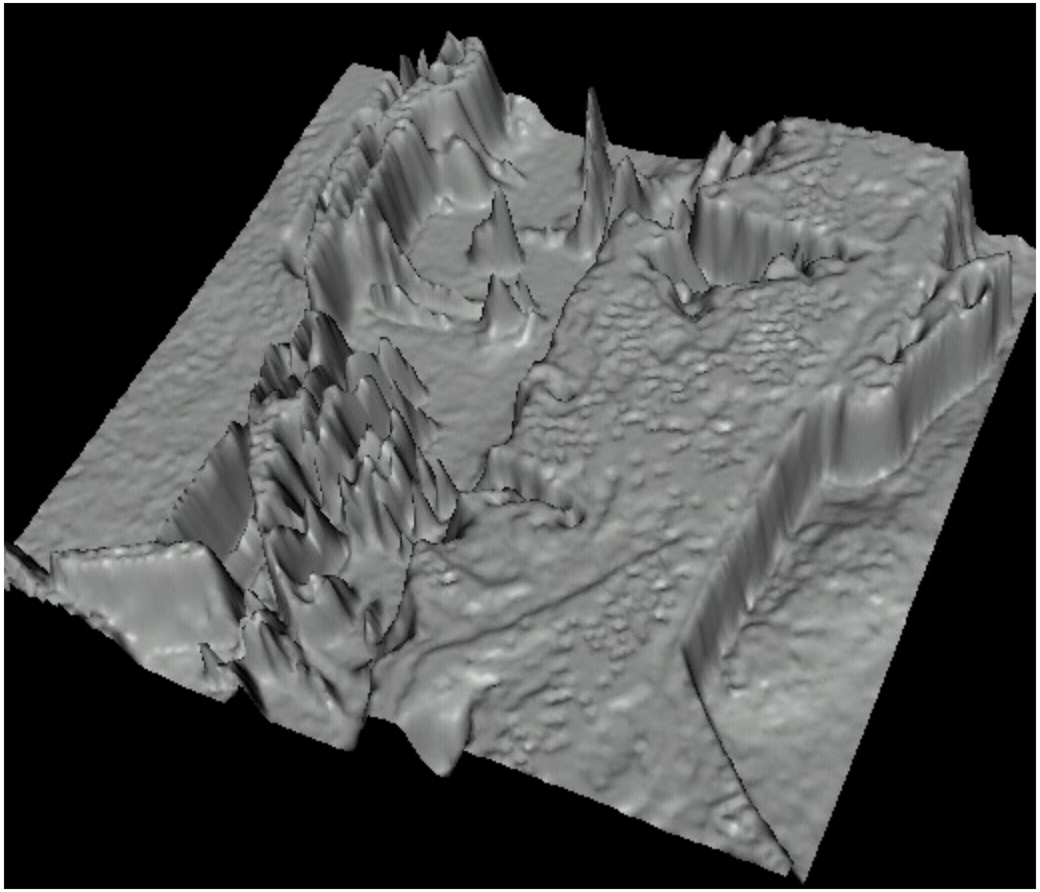
Cast shadows

16

Contrast and invariance



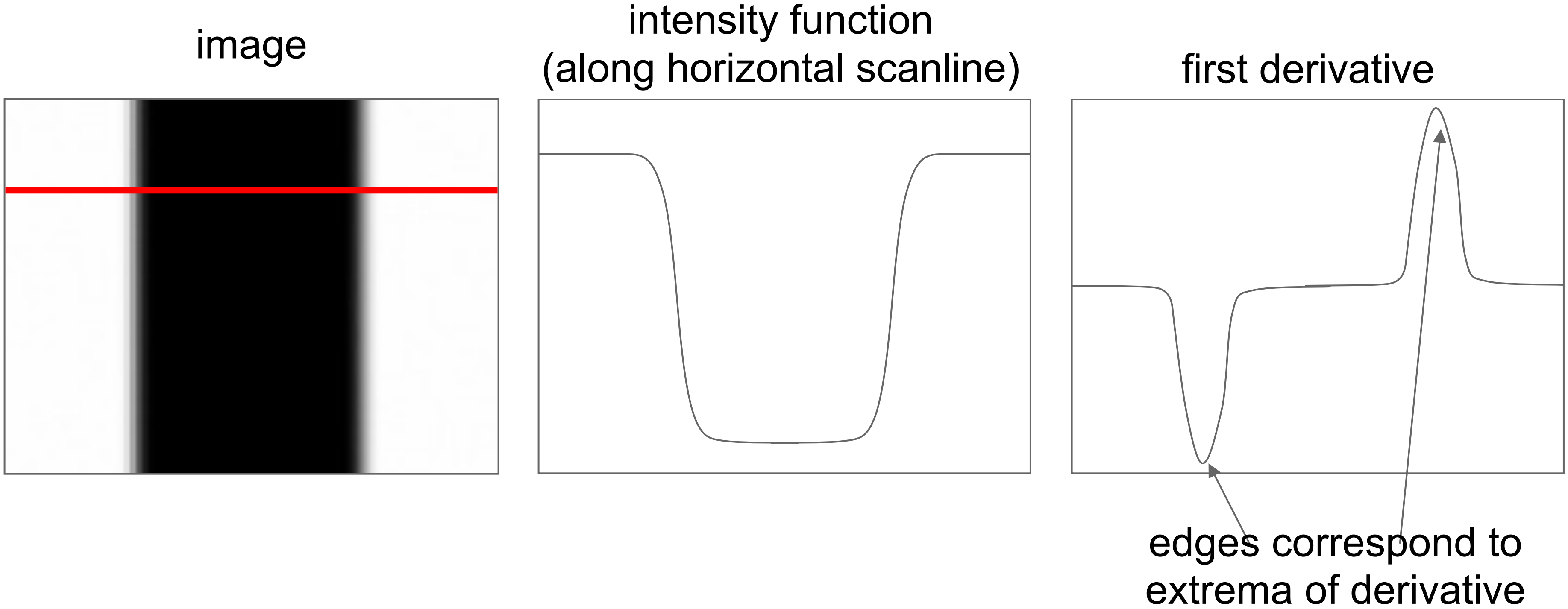
Edges look like steep cliffs



Source: S. Seitz

Derivatives and edges

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

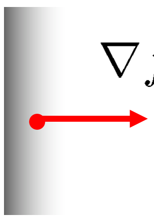


Source: L. Lazebnik

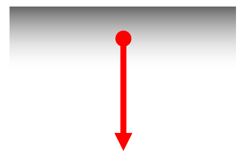
Image gradient

The gradient points in the direction of most rapid change in intensity

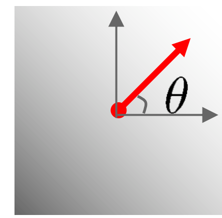
$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$



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



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



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

The gradient direction (orientation of edge normal) is given by:

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

The *edge strength* is given by the gradient magnitude

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

Differentiation and convolution

For 2D function, $f(x,y)$, the partial derivative is:

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon, y) - f(x,y)}{\varepsilon}$$

For discrete data, we can approximate using finite differences:

$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1, y) - f(x,y)}{1}$$

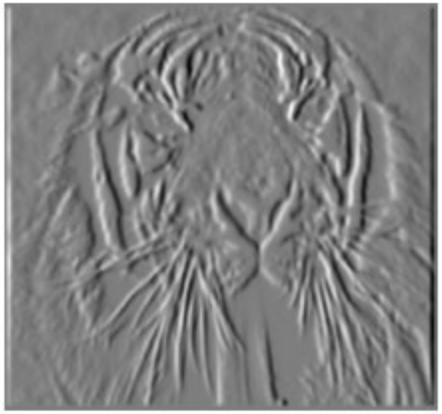
To implement above as correlation, what would be the filter?

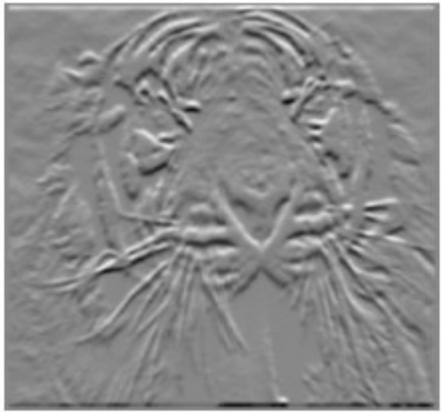
Partial derivatives of an image

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

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

-1	1





-1
1

? or

1
-1

Which shows changes with respect to x?

(showing flipped filters)

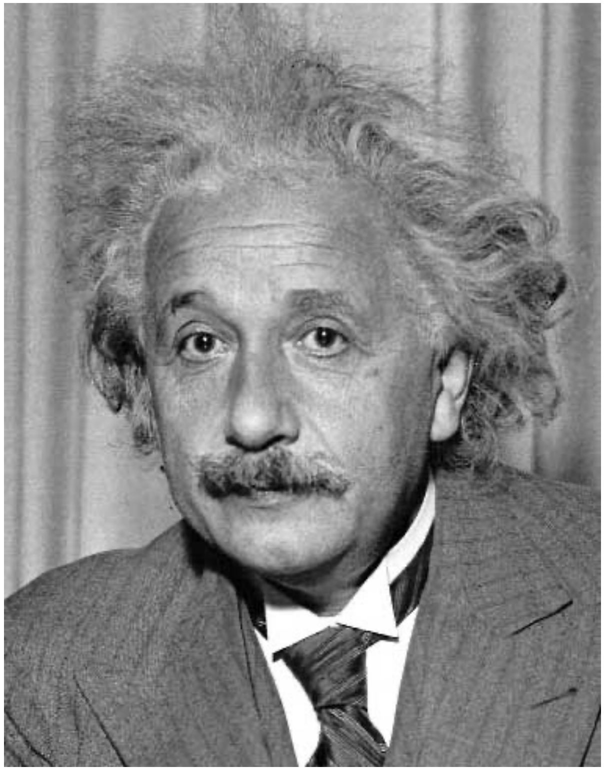
Prewitt operator

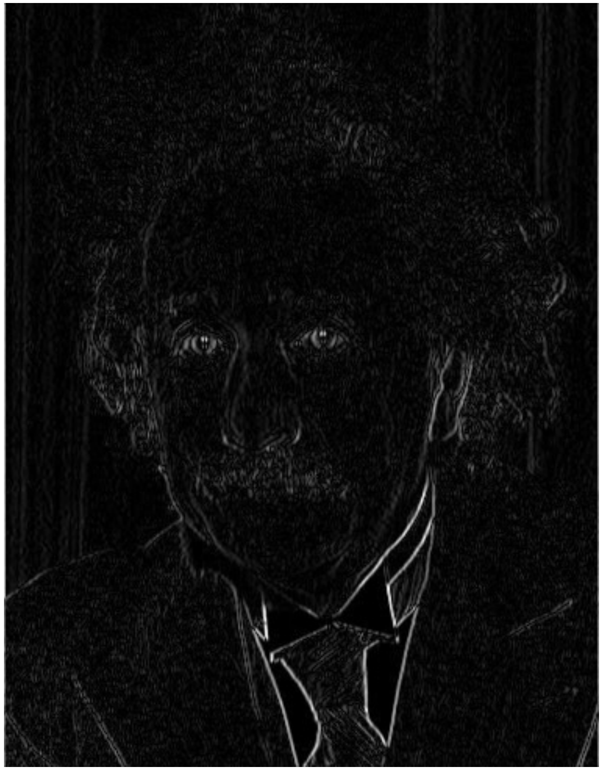
$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

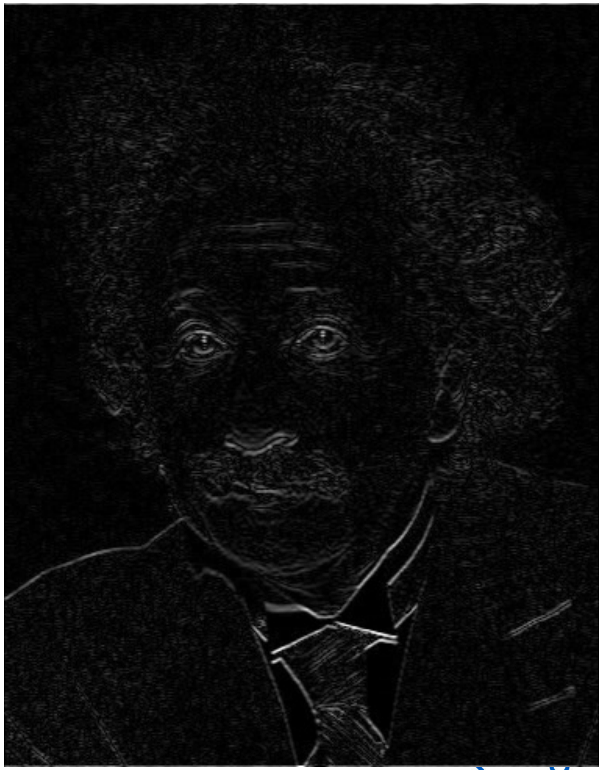
Fig. 1. The horizontal and vertical Prewitt edge detection masks.

Sobel Operator

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$







SAIR
Spatial AI & Robotics Lab

Roberts Operator

+1	0
0	-1

Gx

0	+1
-1	0

Gy

$$|G| = \sqrt{Gx^2 + Gy^2}$$

$$|G| = |Gx| + |Gy|$$

$$|G| = |P_1 - P_4| + |P_2 - P_3|$$

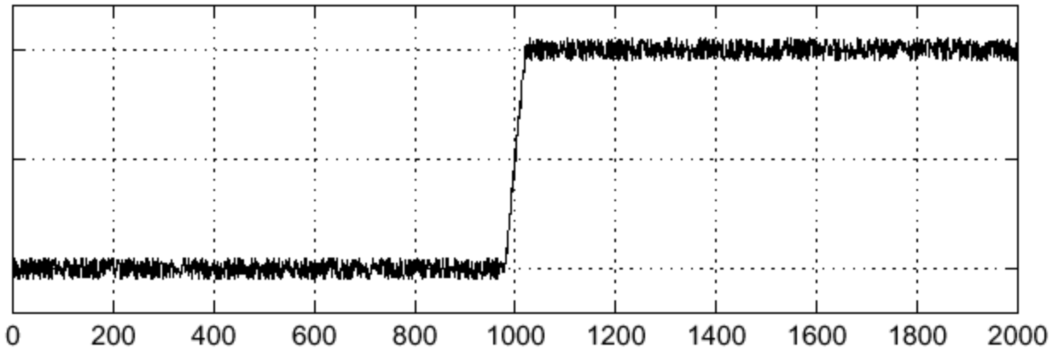
P_1	P_2
P_3	P_4

Effects of noise

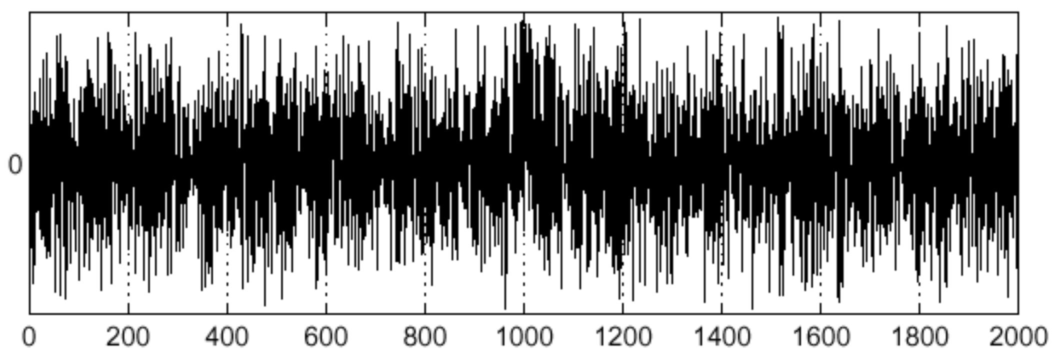
Consider a single row or column of the image

- Plotting intensity as a function of position gives a signal

$f(x)$



$\frac{d}{dx}f(x)$



Where is the edge?

S A I R
Spatial AI & Robotics Lab

26

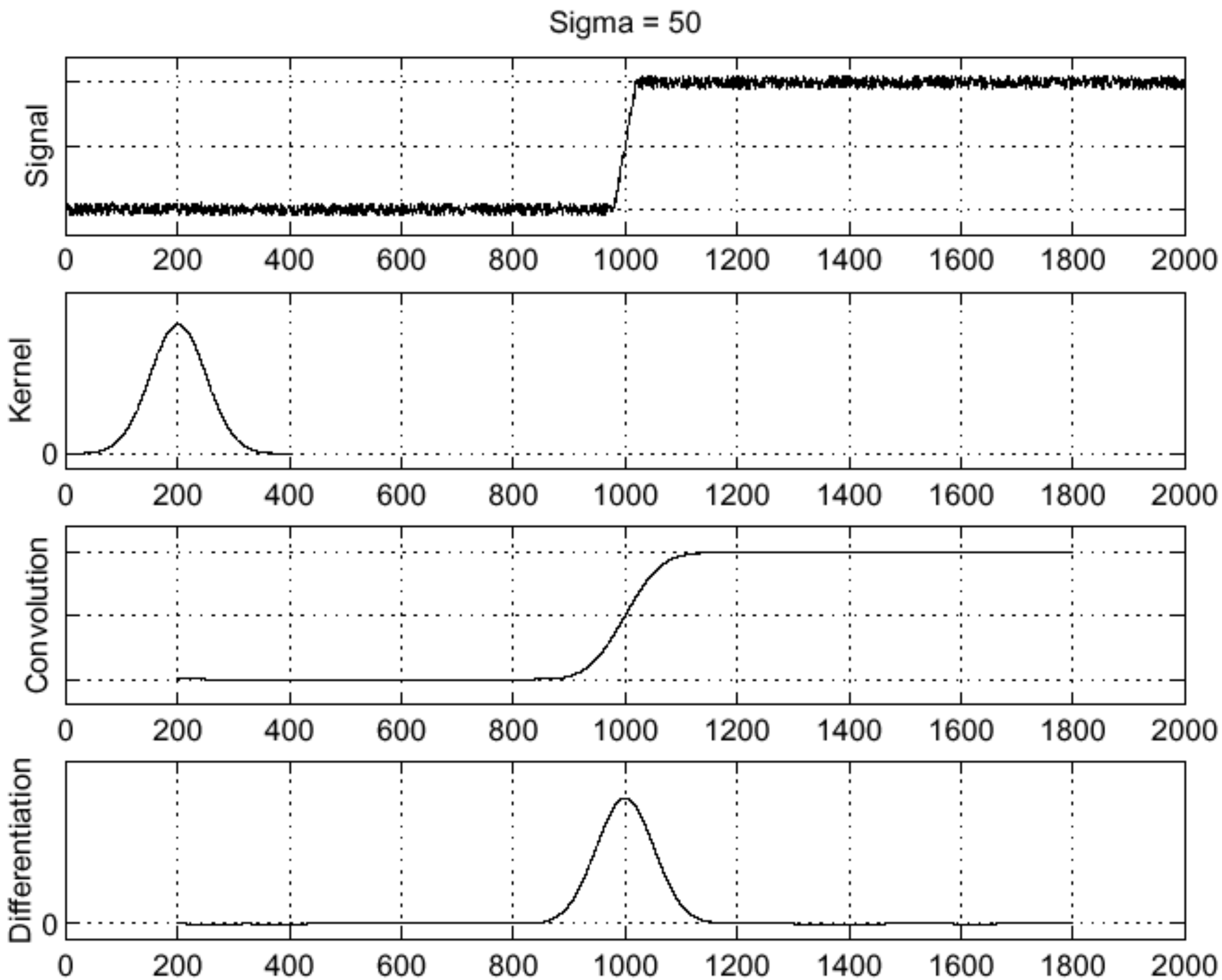
Solution: smooth first

f

h

$h \star f$

$\frac{\partial}{\partial x}(h \star f)$



Where is the edge?

Look for peaks in $\frac{\partial}{\partial x}(h \star f)$

Derivative theorem of convolution

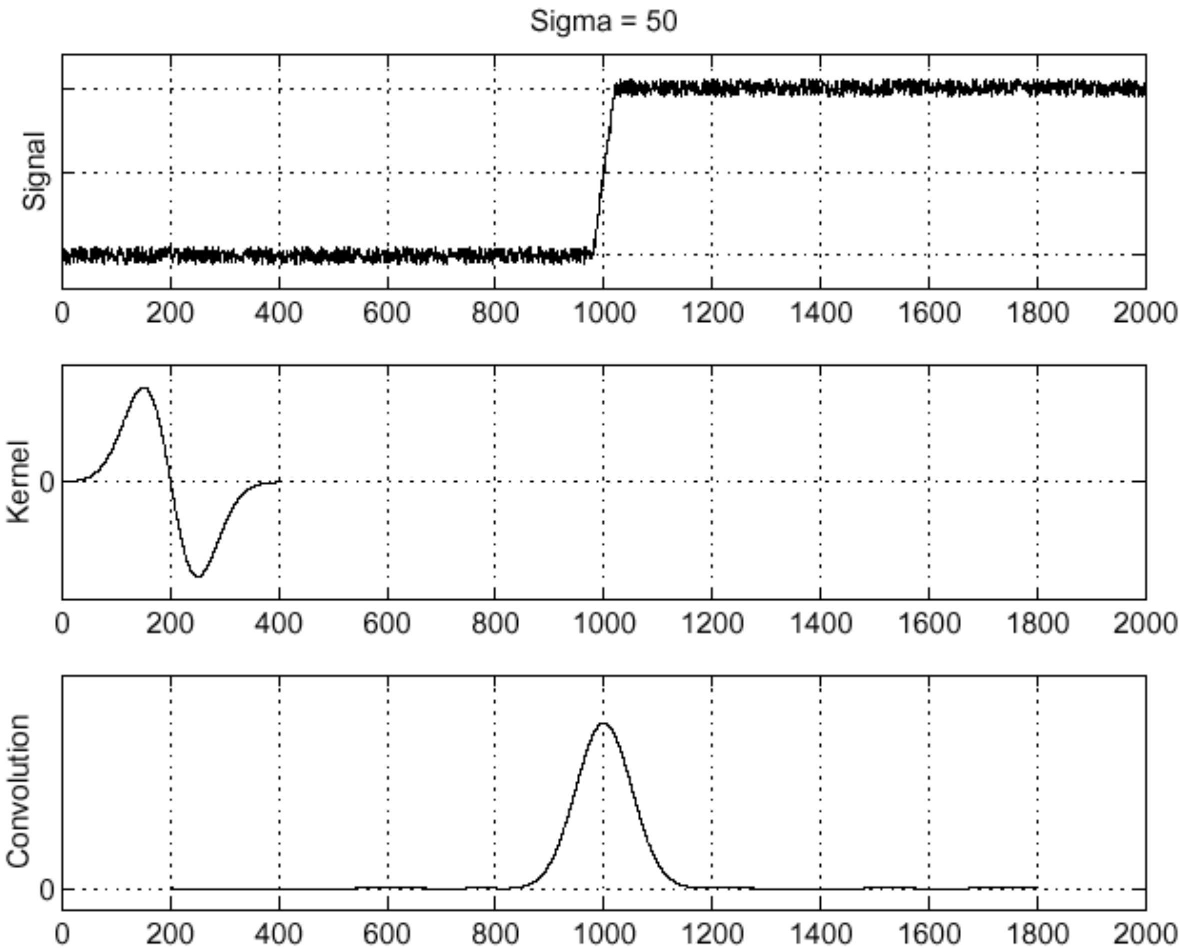
Differentiation property of convolution.

$$\frac{\partial}{\partial x}(h \star f) = \left(\frac{\partial}{\partial x}h\right) \star f$$

f

$\frac{\partial}{\partial x}h$

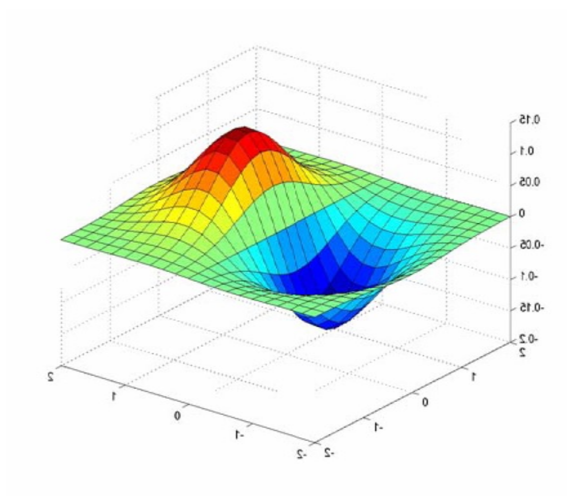
$\left(\frac{\partial}{\partial x}h\right) \star f$



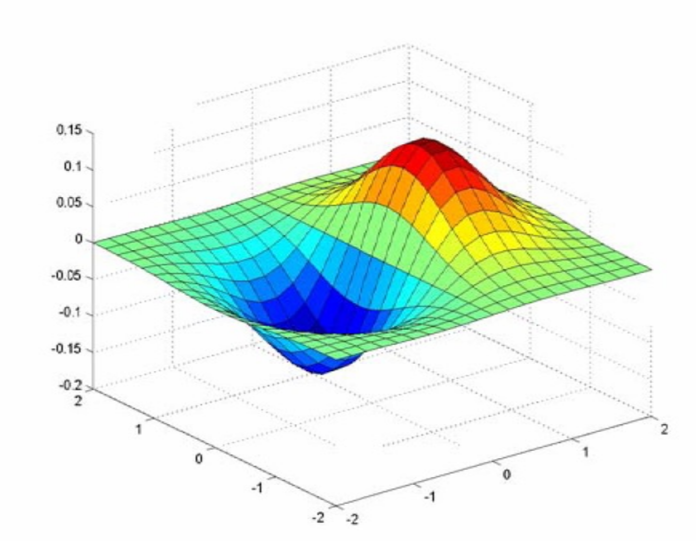
Derivative of Gaussian filter

$$(I \otimes g) \otimes h = I \otimes (g \otimes h)$$

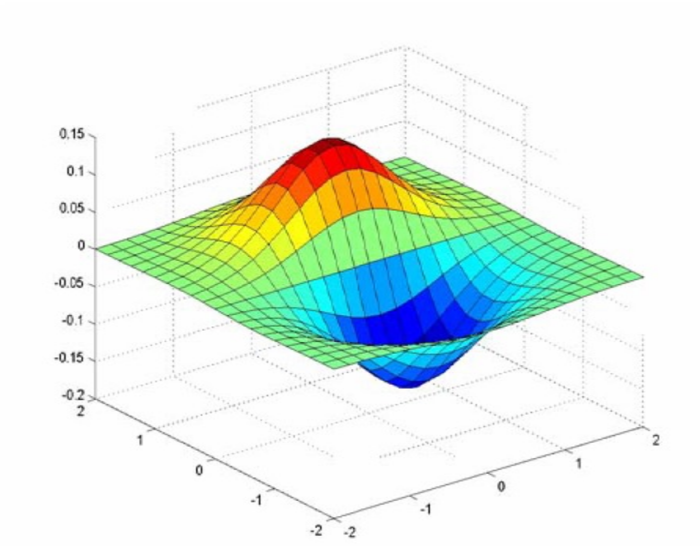
$$\begin{bmatrix} 0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \\ 0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\ 0.0219 & 0.0983 & 0.1621 & 0.0983 & 0.0219 \\ 0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\ 0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \end{bmatrix} \otimes \begin{bmatrix} 1 & -1 \end{bmatrix}$$



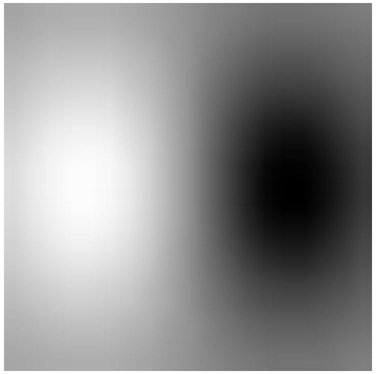
Derivative of Gaussian filters

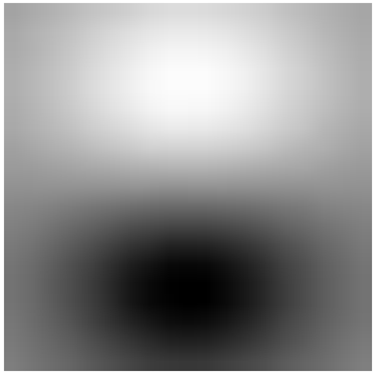


x-direction



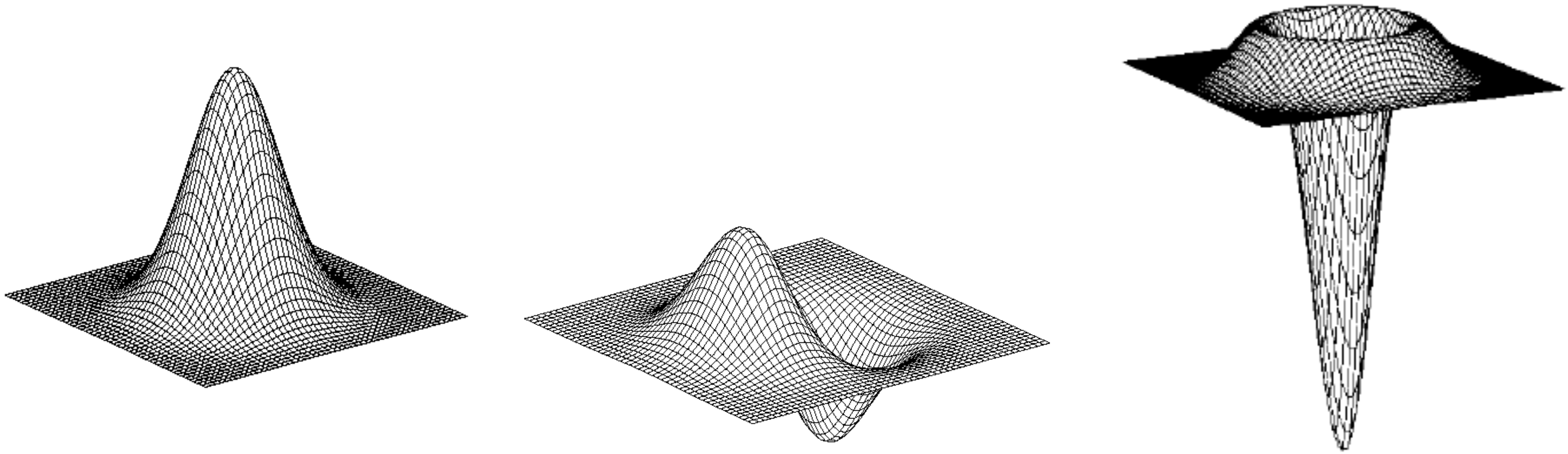
y-direction





Source: L. Lazebnik

2D edge detection filters



Gaussian

$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

derivative of Gaussian

$$\frac{\partial}{\partial x} h_\sigma(u,v)$$

Laplacian of Gaussian

$$\nabla^2 h_\sigma(u,v)$$

- ∇^2 is the Laplacian operator:

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

SAIR Spatial AI & Robotics Lab

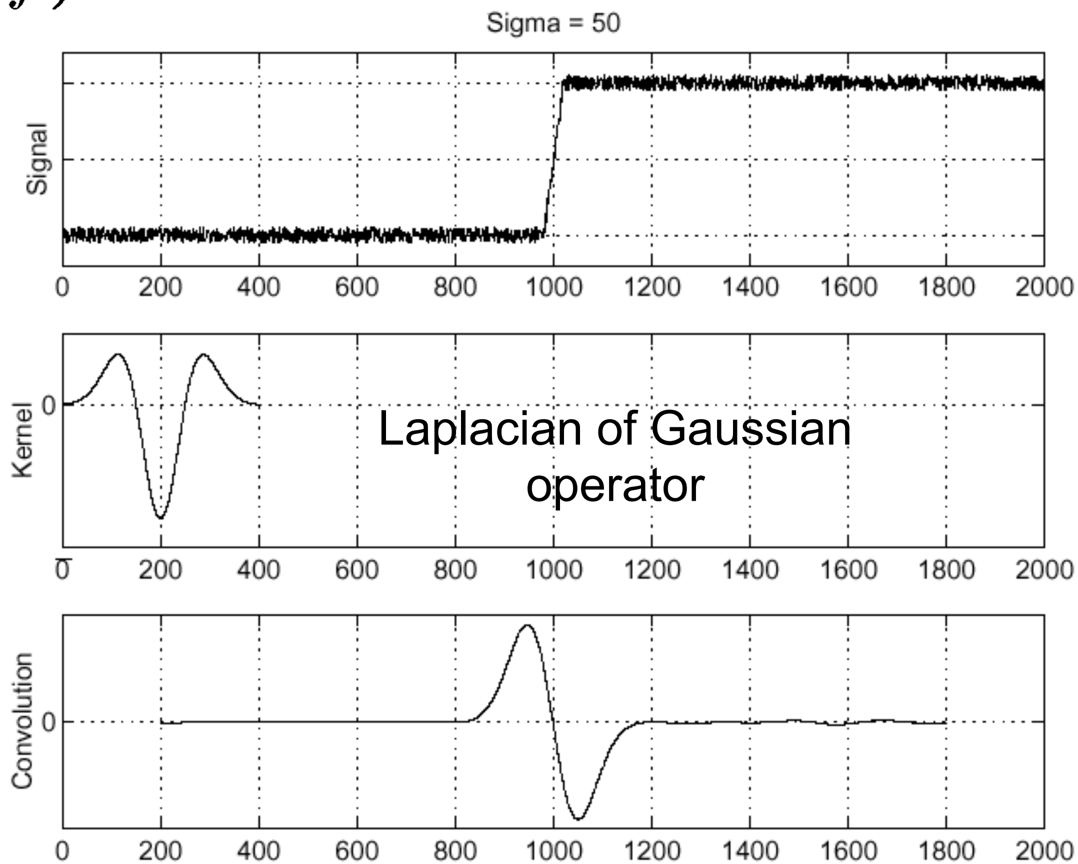
Laplacian of Gaussian

Consider $\frac{\partial^2}{\partial x^2}(h \star f)$

f

$\frac{\partial^2}{\partial x^2}h$

$(\frac{\partial^2}{\partial x^2}h) \star f$



Where is the edge?

Zero-crossings of bottom graph

Mask/Filter/Kernel Properties

- Smoothing
  - Values positive
  - **Sum to 1:** constant regions → same as input (no change)
  - Amount of smoothing proportional to mask size
  - Remove "high-frequency" components; "low-pass" filter
- Template Matching
  - Dot product as correlation (inner product).
  - Highest response for regions "look the most like the filter"
- Derivatives
  - Opposite signs get high response in high contrast regions
  - **Sum to 0**: constant regions → no response
  - High contrast → high absolute values

Gradients -> edges

Primary edge detection steps:

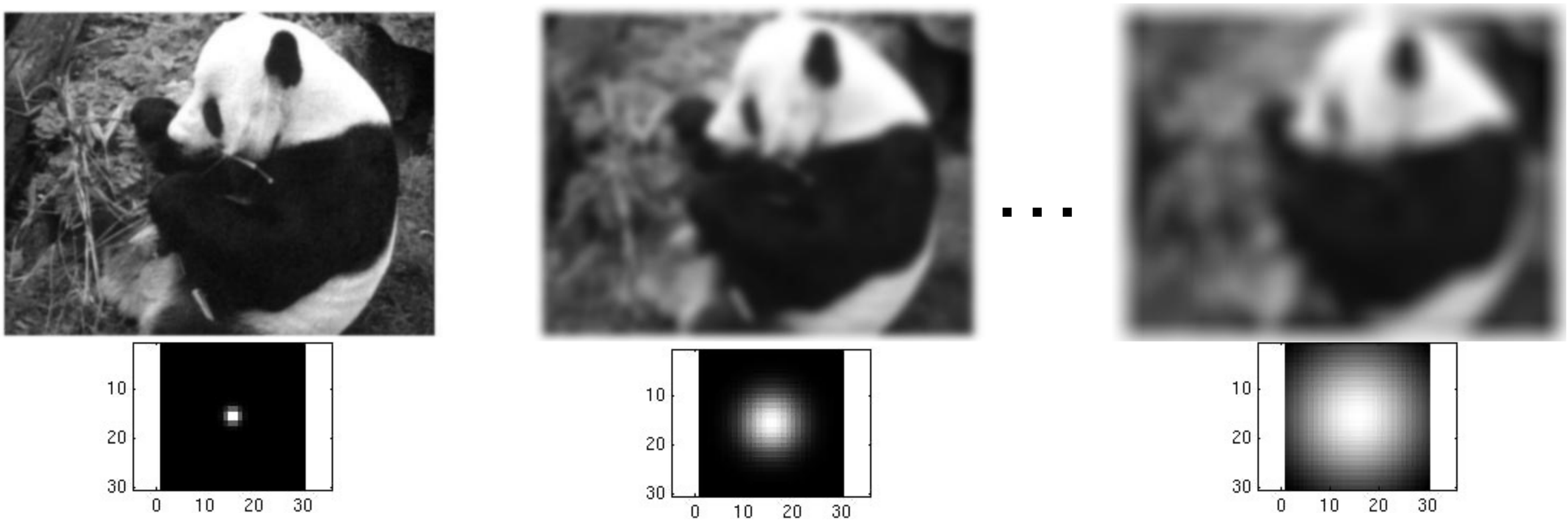
1. Smoothing: suppress noise
2. Edge enhancement: Filter for contrast
3. Edge localization

   Determine which local maxima from filter output are actually edges vs. noise

   - Threshold, Thin

Smoothing with a Gaussian

Recall: parameter σ is the “scale” / “width” / “spread” of the Gaussian kernel, and controls the amount of smoothing.



Effect of σ on derivatives

σ = 1 pixel

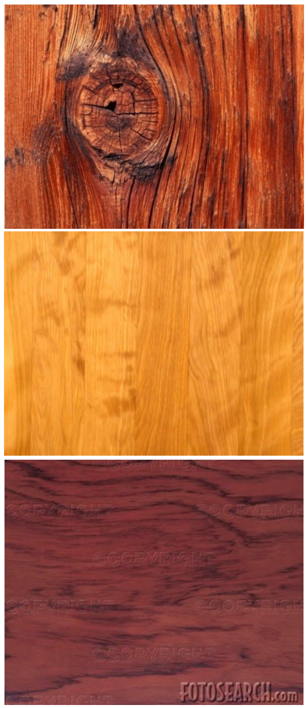
σ = 3 pixels

The apparent structures differ depending on Gaussian's scale/width parameter.

Larger values: larger scale edges detected
Smaller values: finer features detected

So, what scale to choose?

It depends what we’re looking for.



Too small of a scale…can’t see the forest for the trees.
Too big of a scale…can’t tell the maple grain from the cherry.

Thresholding

- Choose a threshold value t.
- Set any pixels less than t to zero (off)
- Set any pixels greater than or equal to t to one (on)

Gradient magnitude image

Thresholding gradient

lower threshold

higher threshold

Canny edge detector

- Filter image with derivative of Gaussian
- Find magnitude and orientation of gradient
- **Local non-maximum suppression**:
  - Thin multi-pixel wide “ridges” down to single pixel width
- Linking and thresholding (**hysteresis**):
  - Define two thresholds: low and high
  - Use the high threshold to start edge curves and the low threshold to continue them.

Source: D. Lowe, L. Fei-Fei

The Canny edge detector

original image (Lena)

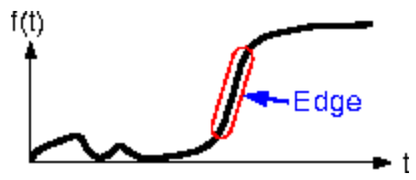
The Canny edge detector

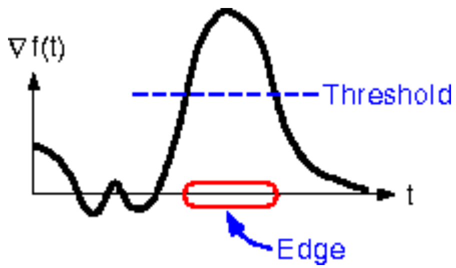
norm of the gradient

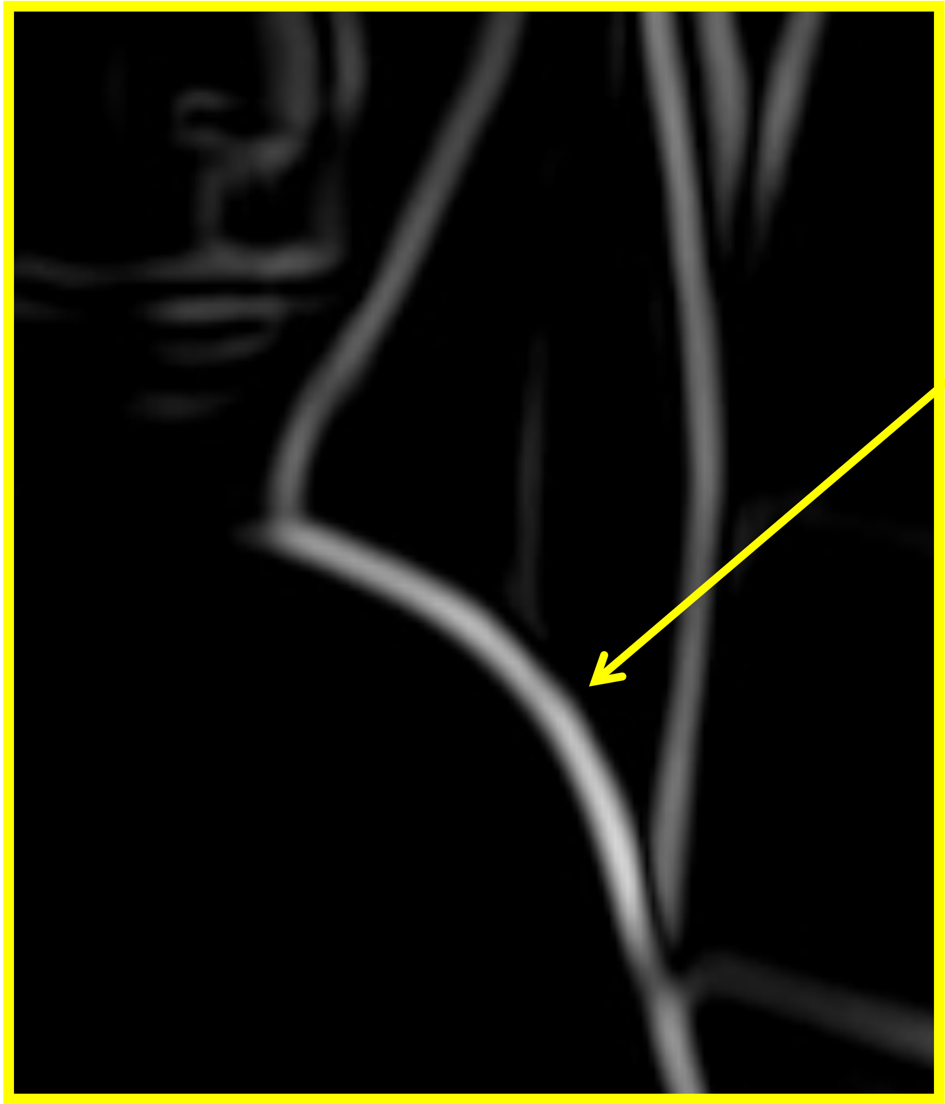
The Canny edge detector

thresholding

The Canny edge detector







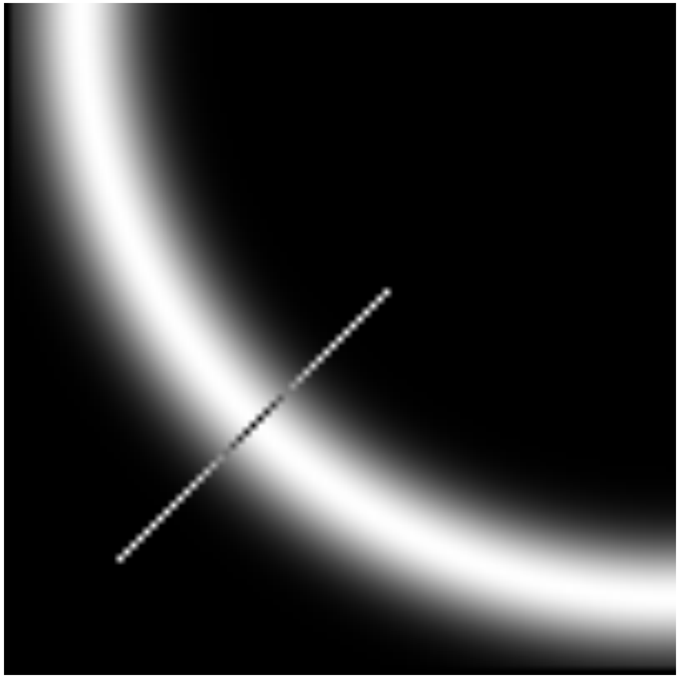
thresholding

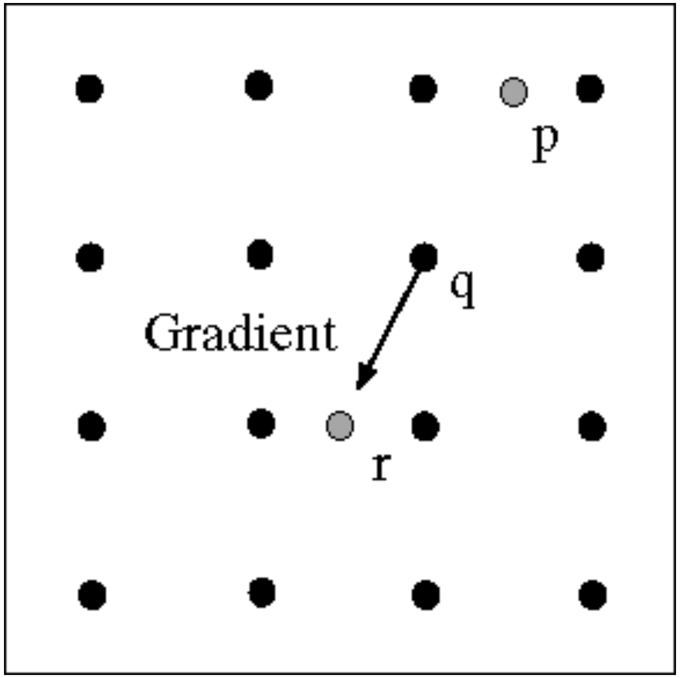
How to turn these thick regions of the gradient into curves?

Non-maximum suppression

Check if pixel is local maximum along gradient direction, select single max across width of the edge

- requires checking interpolated pixels p and r





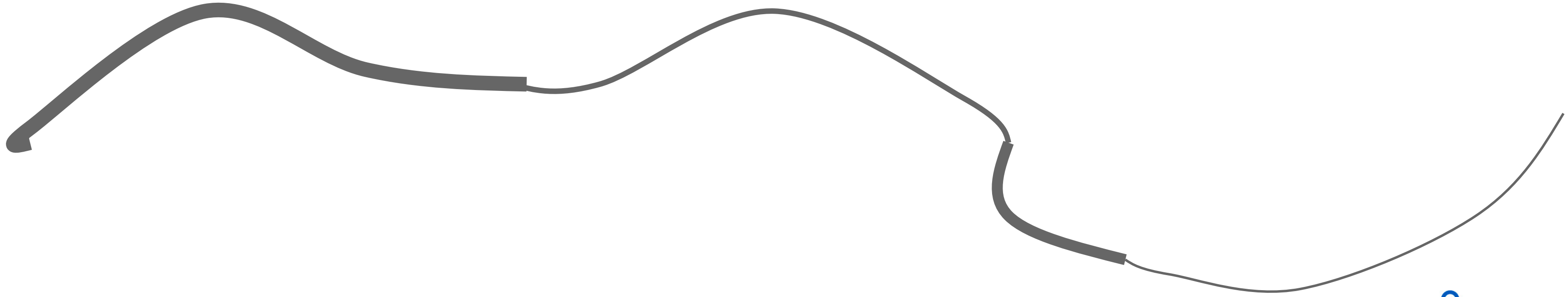
The Canny edge detector

thinning
(non-maximum suppression)

Problem: pixels along this edge didn't survive the thresholding

Hysteresis thresholding

- Check that maximum value of gradient value is sufficiently large
  - drop-outs? use **hysteresis**
    - use a high threshold to start edge curves and a low threshold to continue them.



Source: S. Seitz

Hysteresis thresholding

original image

high threshold
(strong edges)

low threshold
(weak edges)

hysteresis threshold

Source: L. Fei-Fei

Object boundaries vs. edges

Background

Texture

Shadows

Object boundaries may not be edges.
Edges may not be object boundaries.

Edge detection is just the beginning…

image gradient magnitude human segmentation

http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

Source: L. Lazebnik

Important Concepts

- Template Matching
  - Cross-correlation
- Edge Detection
  - Image differentiation and gradient
  - Derivative theorem of convolution
  - 2D edge detection filters, Sobel operator
  - Canny edge detector, Hysteresis thresholding

SAIR
Spatial AI & Robotics Lab