Robert Collins
CSE486, Penn State

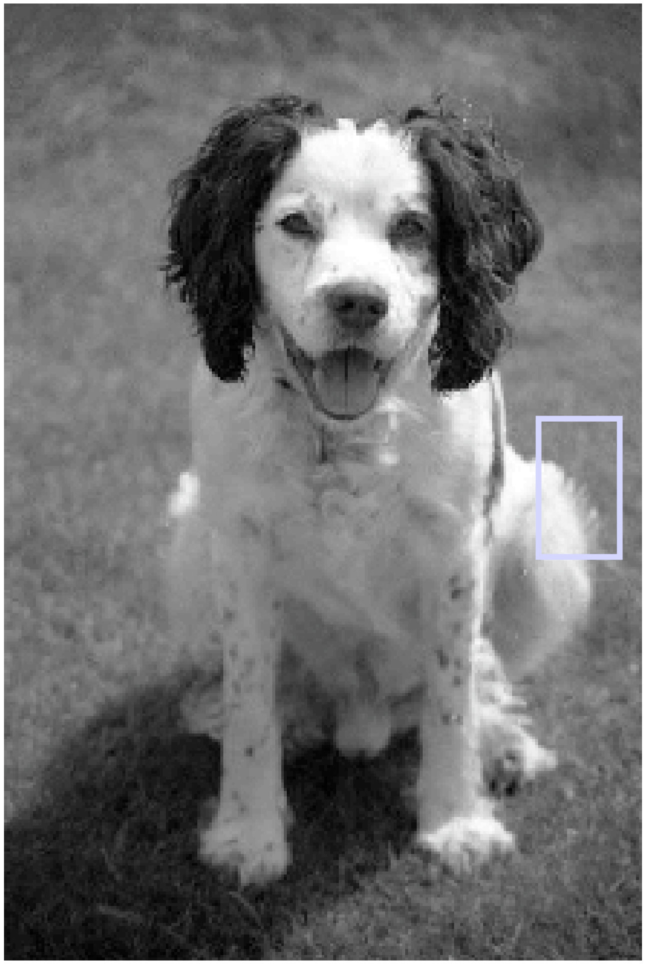
Lecture 5:
Gradients and Edge Detection

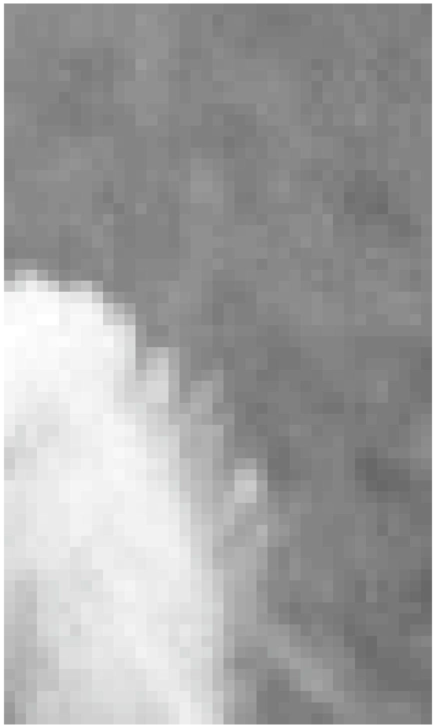
Reading: T&V Section 4.1 and 4.2

What Are Edges?

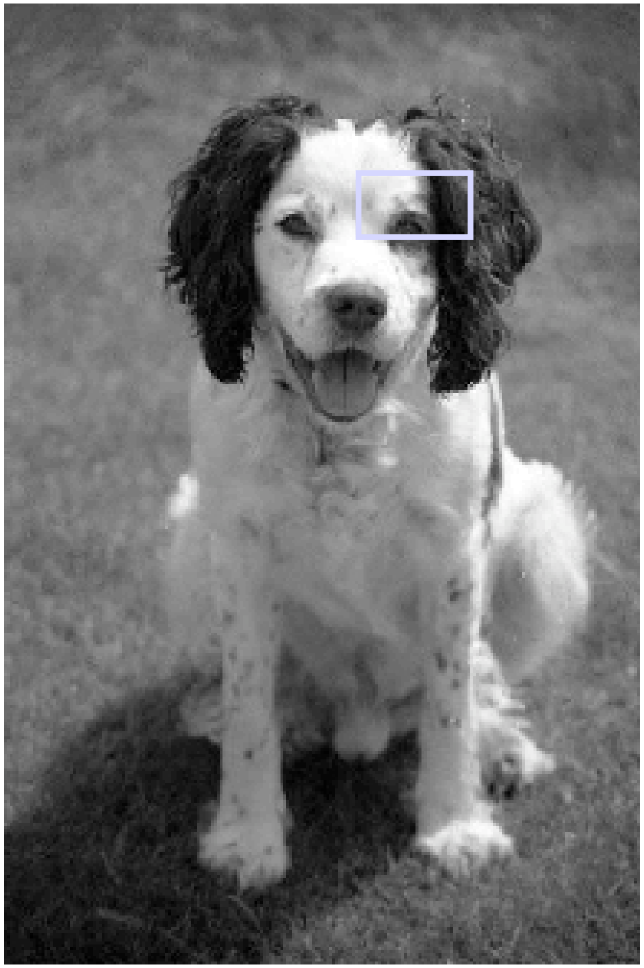
Simple answer: discontinuities in intensity.

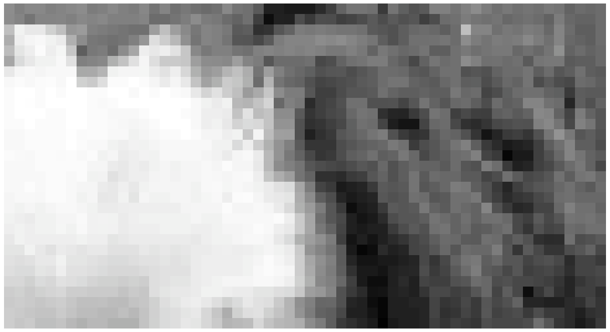
Boundaries of objects



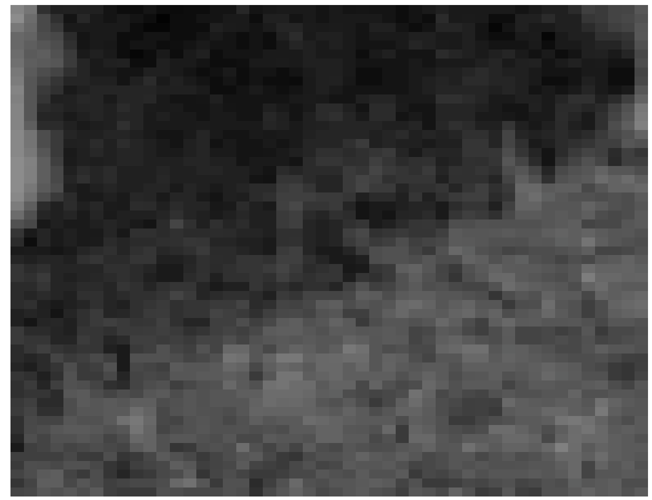


Boundaries of Material Properties





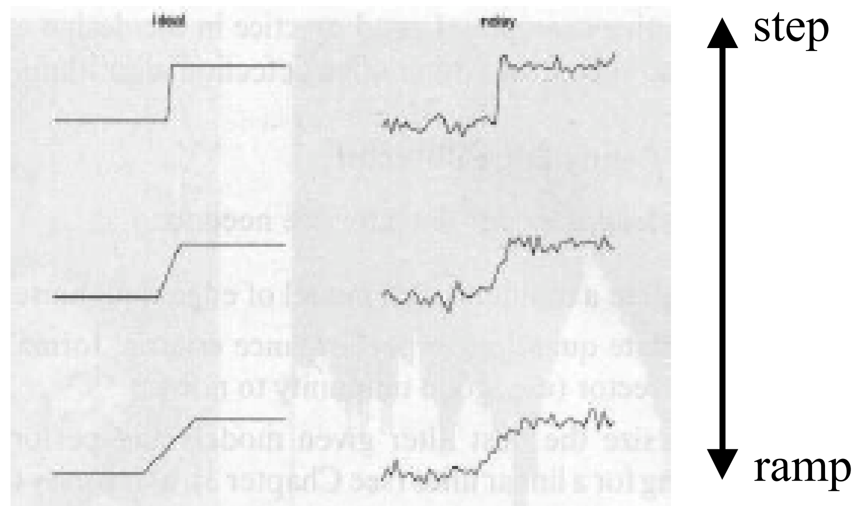
Boundaries of Lighting



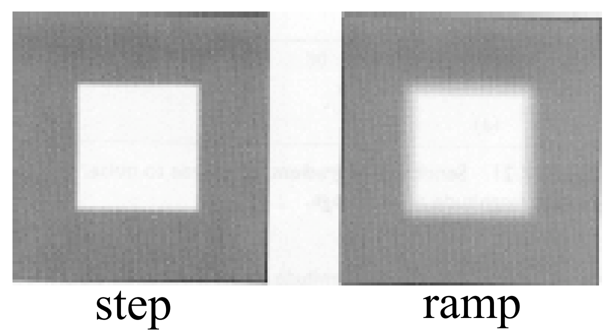
Types of Edges (1D Profiles)

- Edges can be modeled according to their intensity profiles:

- <u>Step edge</u>:
  - the image intensity abruptly changes from one value to one side of the discontinuity to a different value on the opposite side.

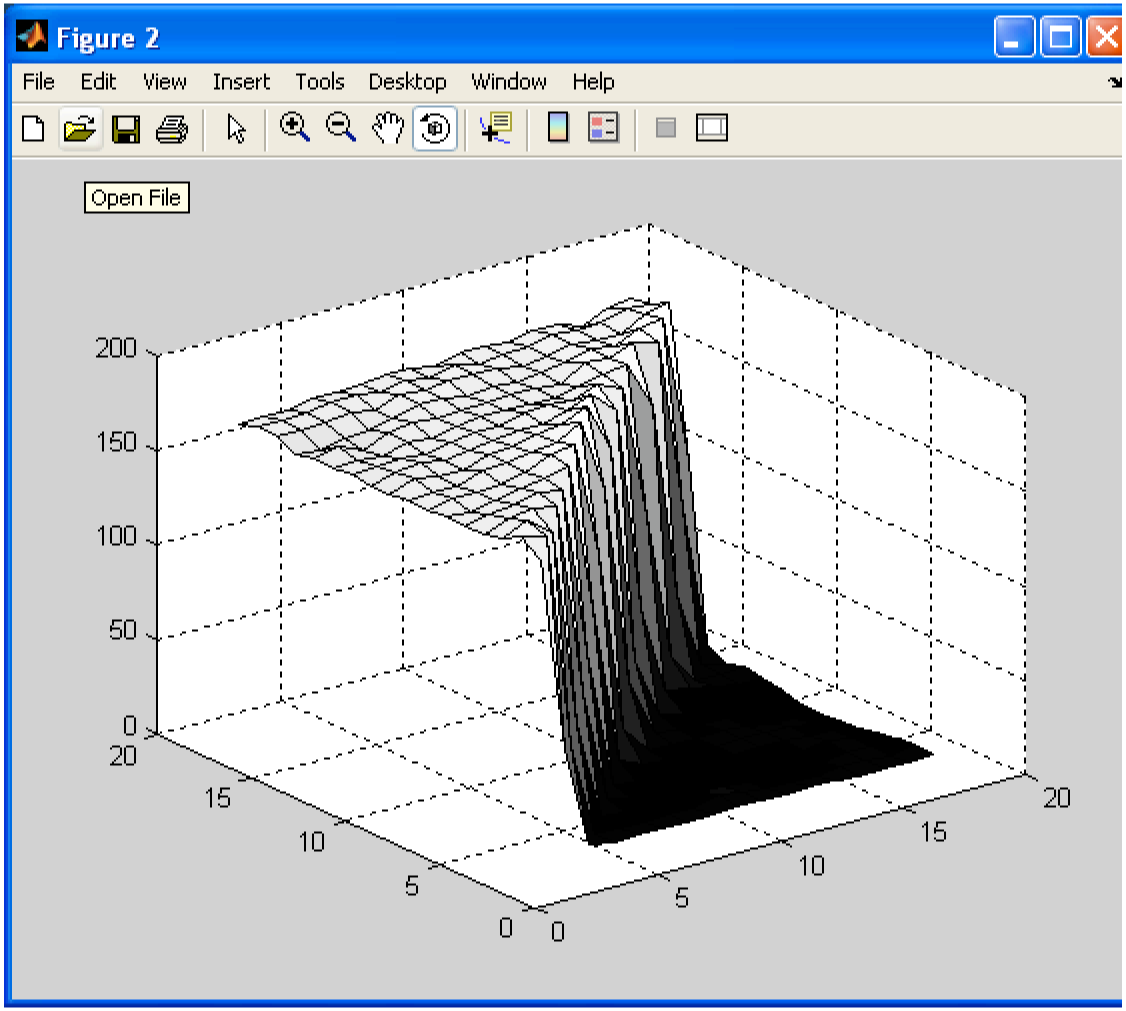


- <u>Ramp edge</u>:
  - a step edge where the intensity change is not instantaneous but occurs over a finite distance.



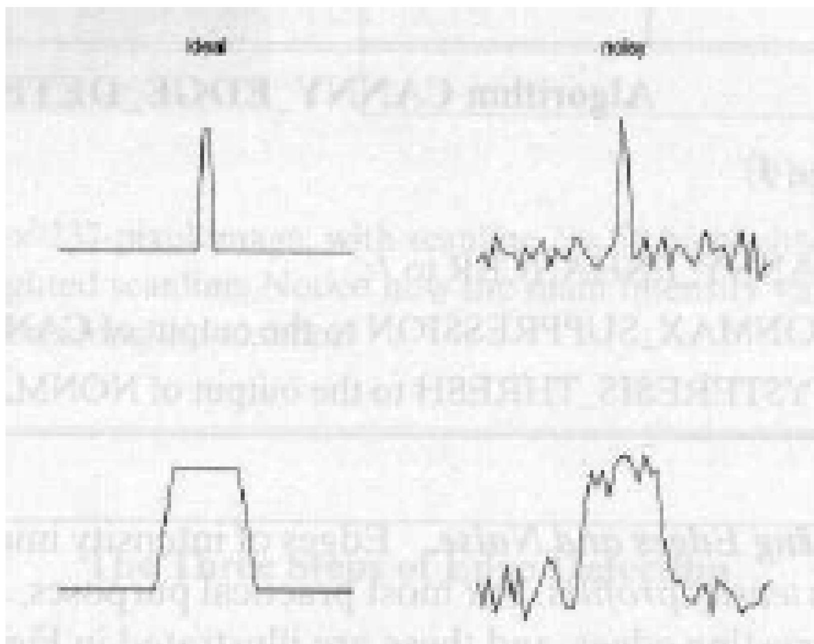
Robert Collins
CSE486, Penn State

Examples



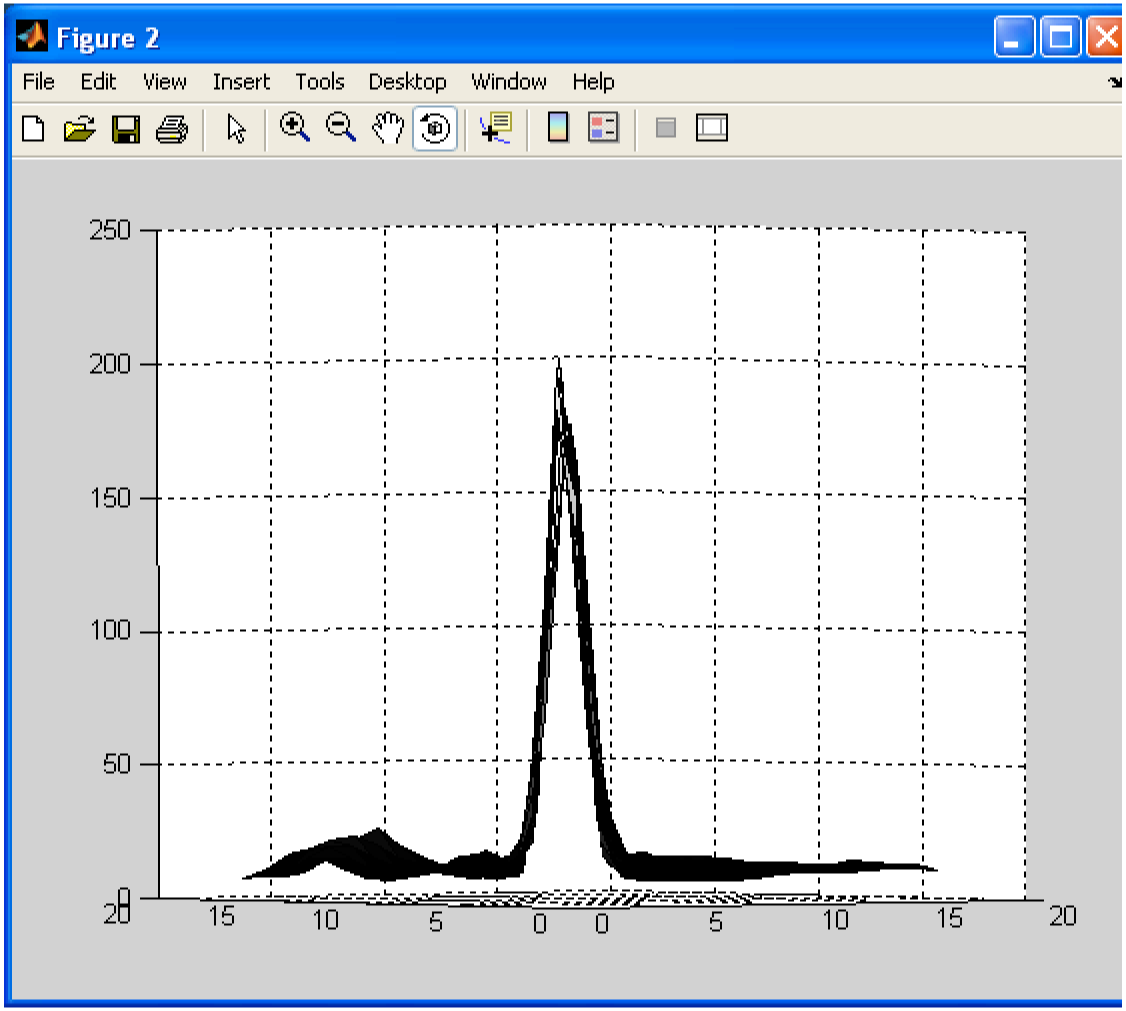
Types of Edges (1D Profiles)

- Ridge edge:
  - the image intensity abruptly changes value but then returns to the starting value within some short distance
  - generated usually by lines



Robert Collins
CSE486, Penn State

Examples



Examples

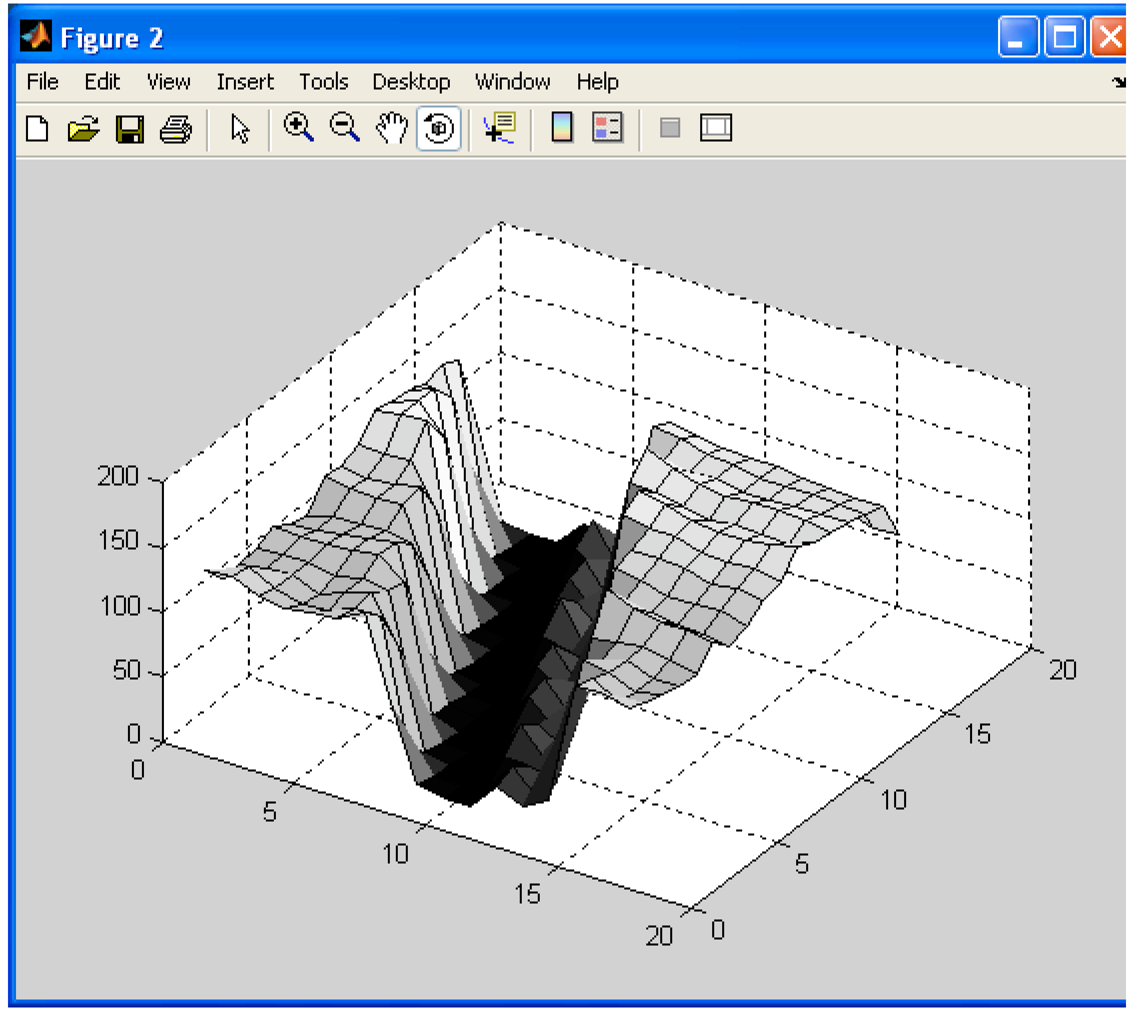
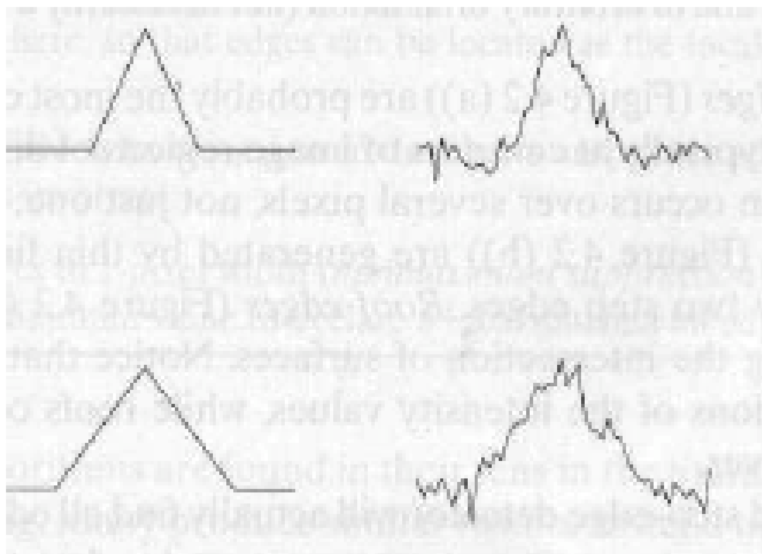


Figure 2
File Edit View Insert Tools Desktop Window Help
200
150
100
50
0
0
5
10
15
20
0
5
10
15
20

Types of Edges (1D Profiles)

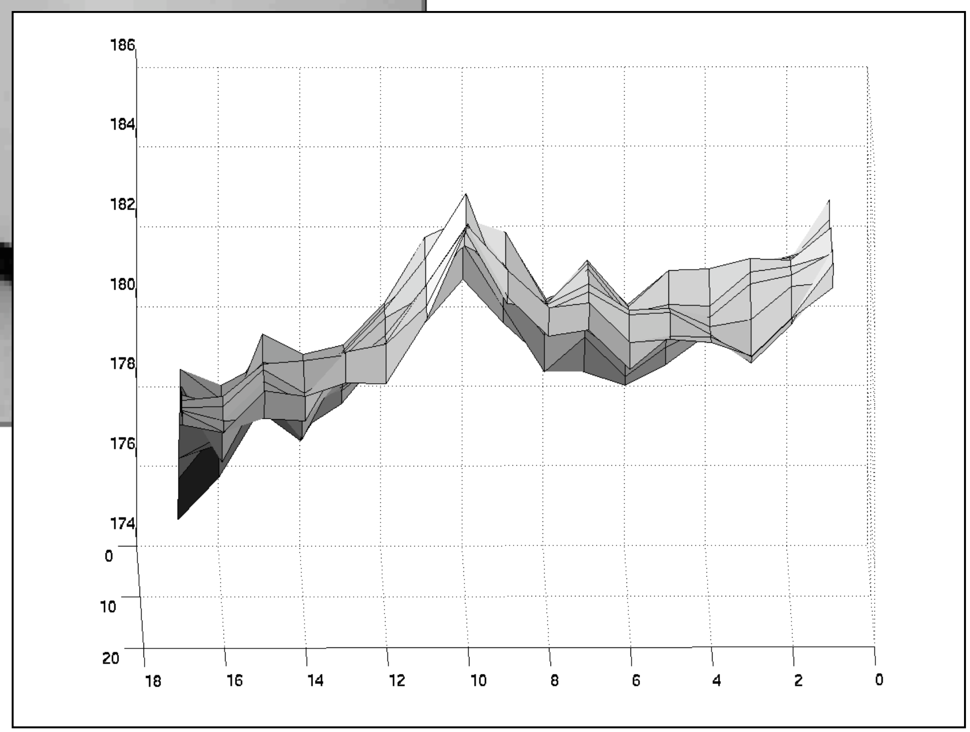
- Roof edge:
  - a ridge edge where the intensity change is not instantaneous but occurs over a finite distance
  - generated usually by the intersection of surfaces



(c)

Robert Collins
CSE486, Penn State

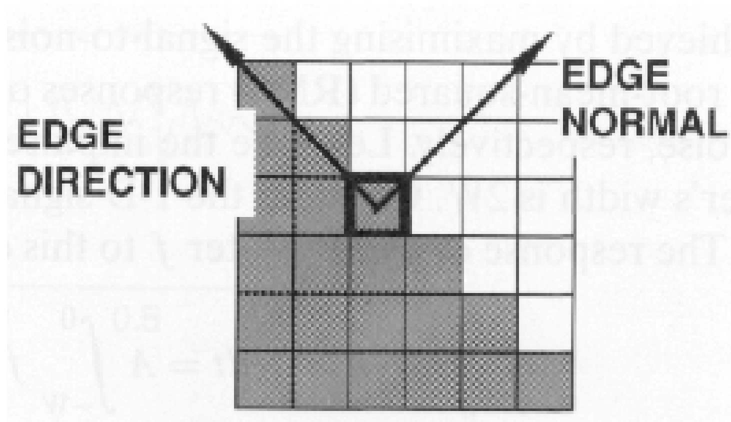
Examples



Robert Collins
CSE486, Penn State

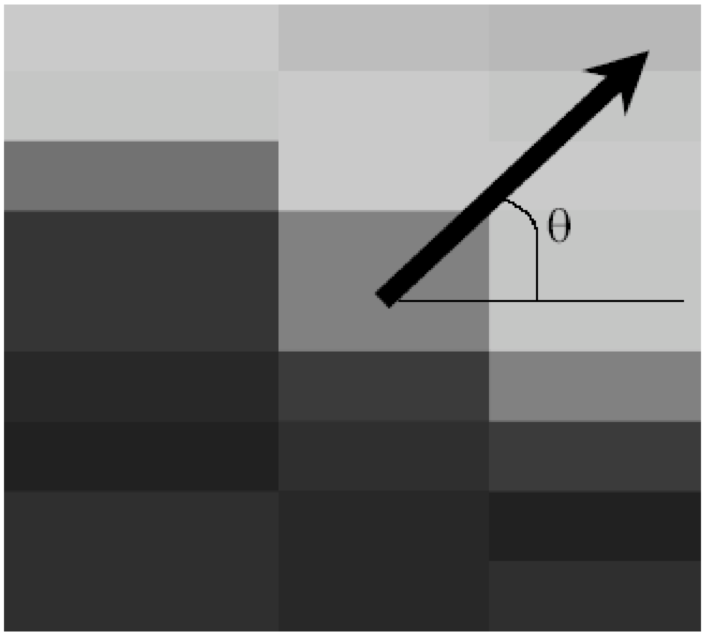
Step/Ramp Edge Terminology

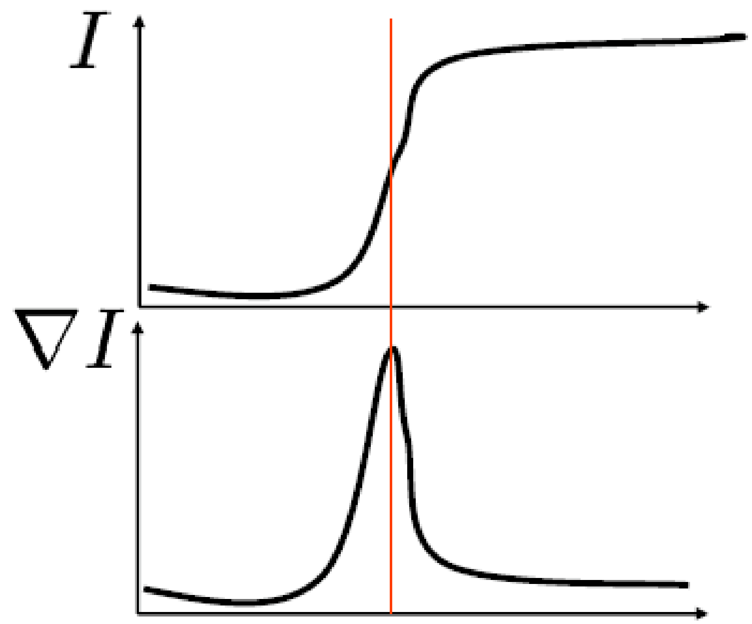
- Edge descriptors
  - Edge normal: unit vector in the direction of maximum intensity change.
  - Edge direction: unit vector along edge (perpendicular to edge normal).
  - Edge position or center: the image position at which the edge is located.
  - Edge strength or magnitude: local image contrast along the normal.



Important point: All of this information can be computed from the gradient vector field!!

Summary of Gradients





Edge pixels are at local maxima of gradient magnitude
Gradient direction is always perpendicular to edge direction

Gradient Vector: $\nabla I = [\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}]^T$

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

Magnitude:

$$\theta = atan2\left(\frac{\partial I}{\partial y}, \frac{\partial I}{\partial x}\right)$$

Orientation

Simple Edge Detection Using Gradients

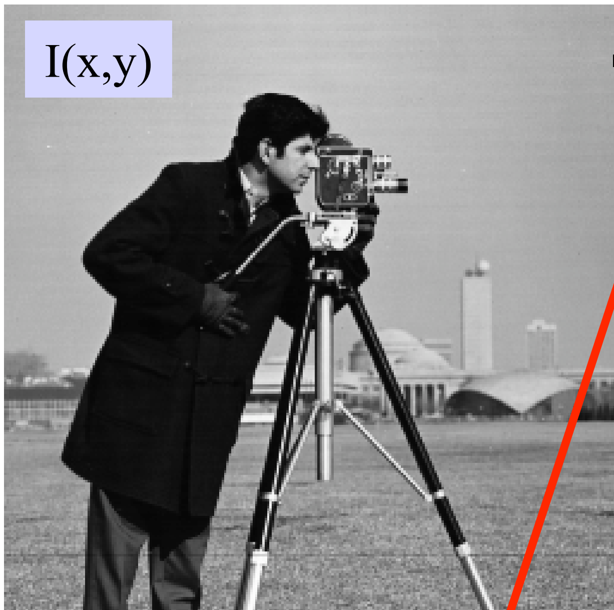
A simple edge detector using gradient magnitude

- Compute gradient vector at each pixel by convolving image with horizontal and vertical derivative filters

- Compute gradient magnitude at each pixel

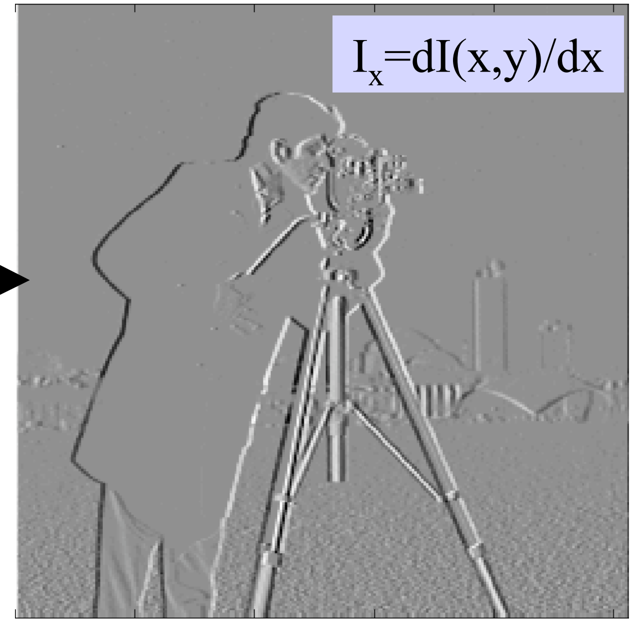
- If magnitude at a pixel exceeds a threshold, report a possible edge point.

Compute Spatial Image Gradients



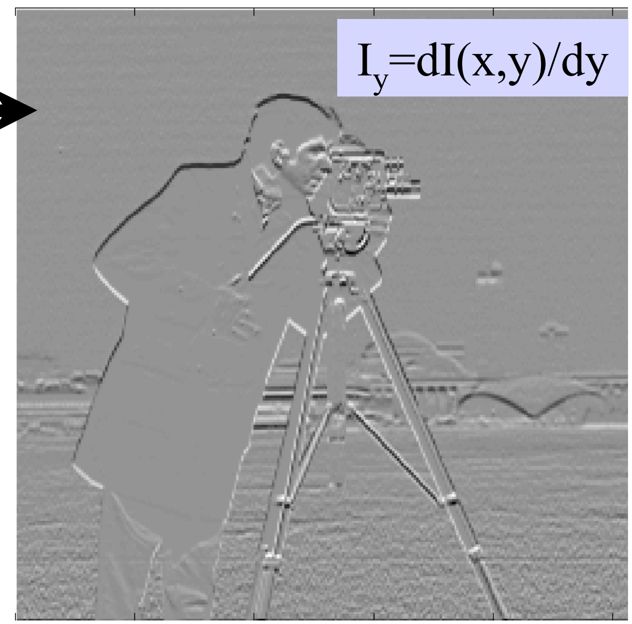
$$\frac{I(x+1,y) - I(x-1,y)}{2}$$

Partial derivative wrt x



$$\frac{I(x,y+1) - I(x,y-1)}{2}$$

Partial derivative wrt y



Replace with your favorite smoothing+derivative operator

Simple Edge Detection Using Gradients

A simple edge detector using gradient magnitude

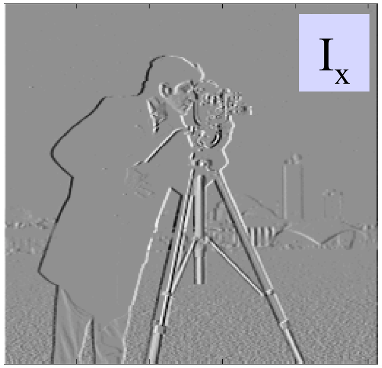
- Compute gradient vector at each pixel by convolving image with horizontal and vertical derivative filters

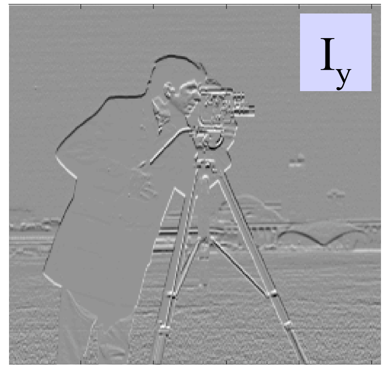
- Compute gradient magnitude at each pixel

- If magnitude at a pixel exceeds a threshold, report a possible edge point.

Robert Collins
CSE486, Penn State

Compute Gradient Magnitude





Magnitude of gradient
sqrt(Ix.^2 + Iy.^2)

Measures steepness of slope at each pixel (= edge contrast)

Simple Edge Detection Using Gradients

A simple edge detector using gradient magnitude

- Compute gradient vector at each pixel by convolving image with horizontal and vertical derivative filters

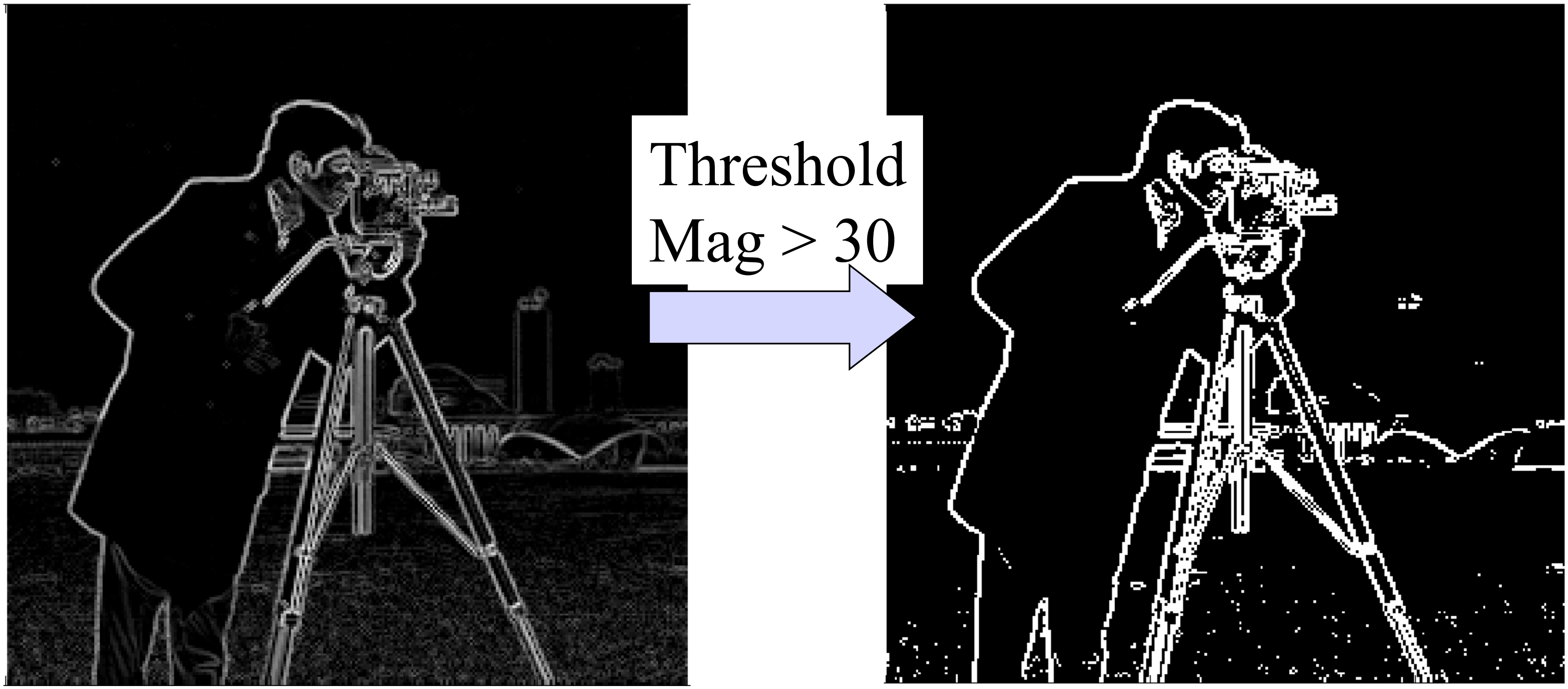
- Compute gradient magnitude at each pixel

- If magnitude at a pixel exceeds a threshold, report a possible edge point.

Threshold to Find Edge Pixels

- Example – cont.:

Binary edge image



Edge Detection Using Gradient Magnitude

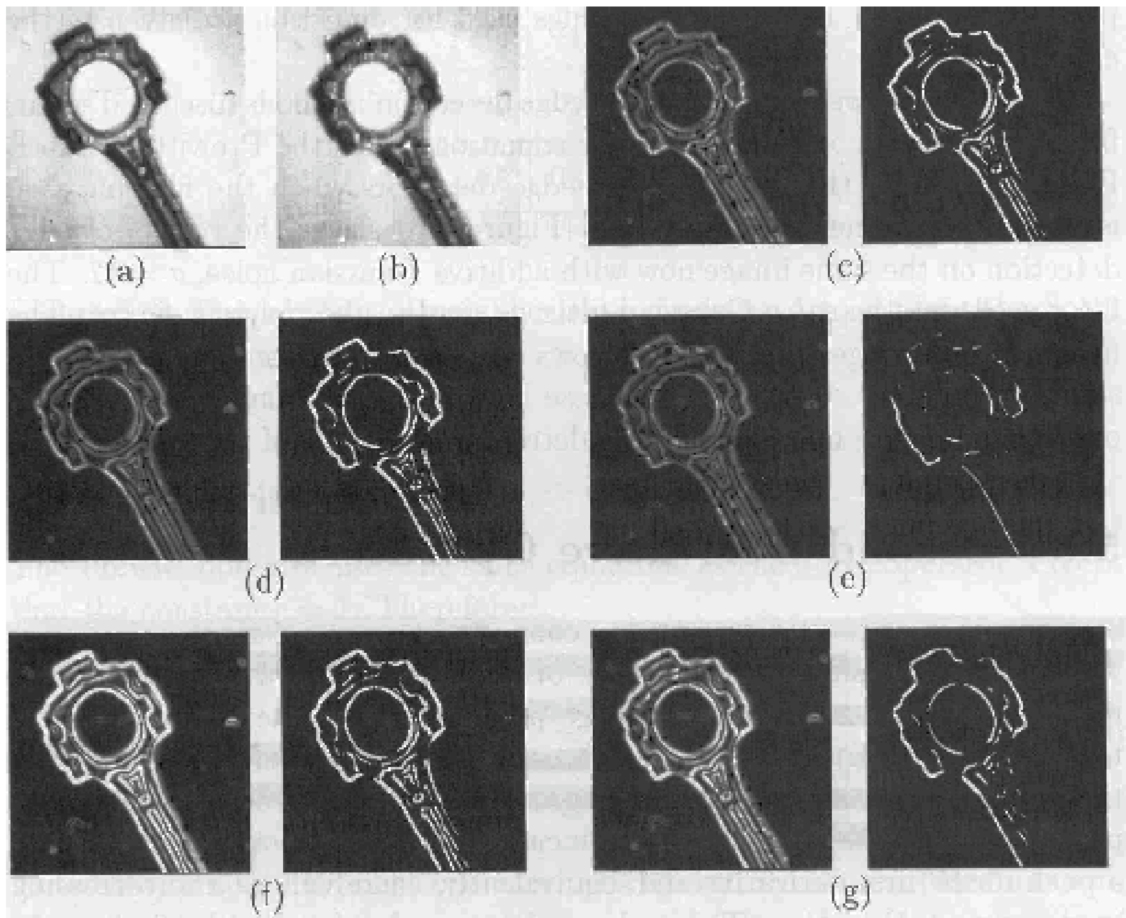
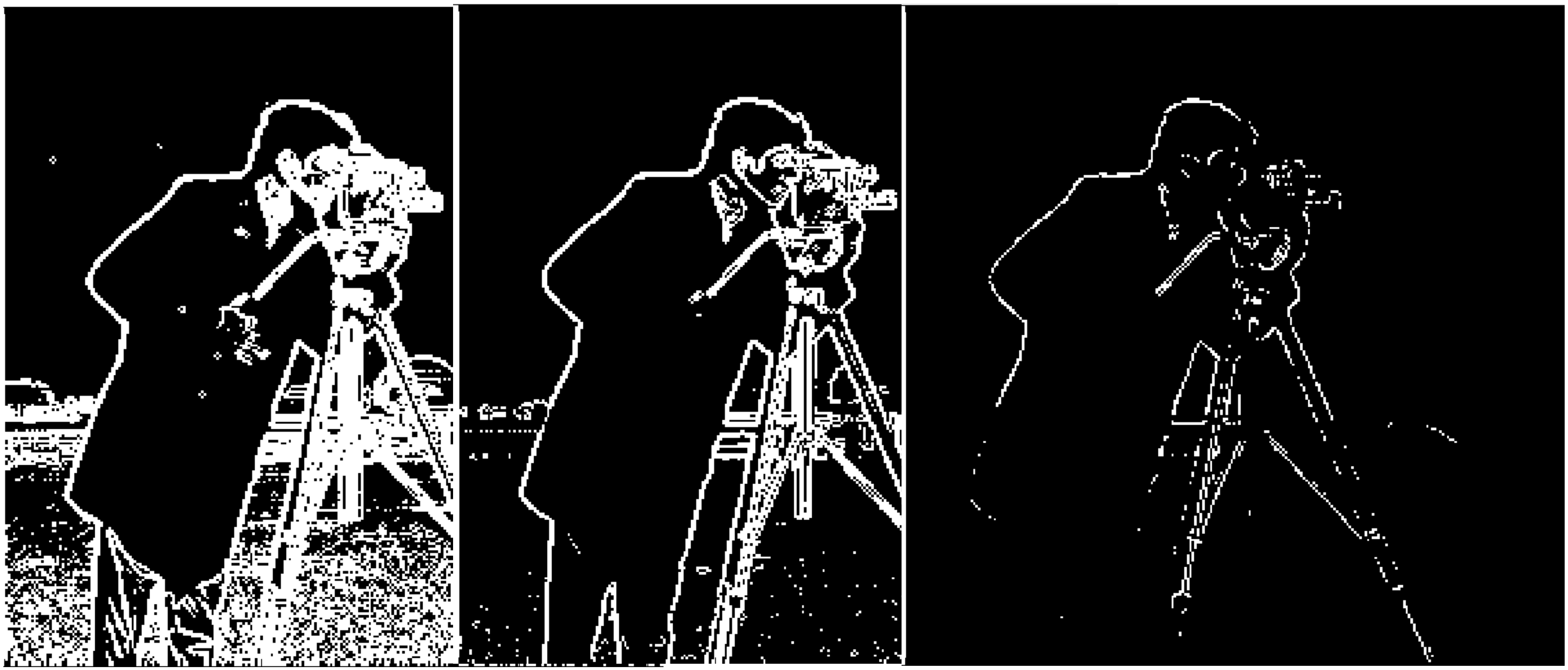


Figure 5.4: A comparison of various edge detectors. (a) Original image. (b) Filtered image. (c) Simple gradient using 1×2 and 2×1 masks, $T = 32$. (d) Gradient using 2×2 masks, $T = 64$. (e) Roberts cross operator, $T = 64$. (f) Sobel operator, $T = 225$. (g) Prewitt operator, $T = 225$.

(with noise filtering)

Issues to Address

How should we choose the threshold?



> 10 > 30 > 80

Trade-off: Smoothing vs Localization

There is ALWAYS a tradeoff between smoothing and good edge localization!

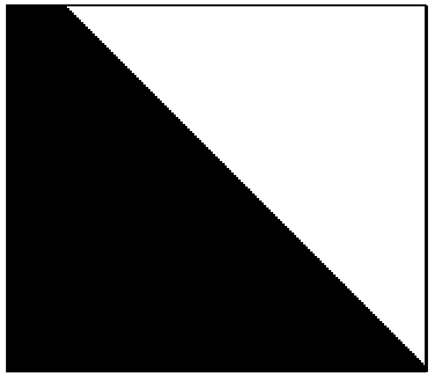
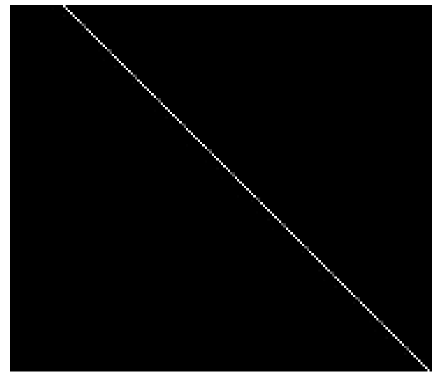


Image with Edge



Edge Location

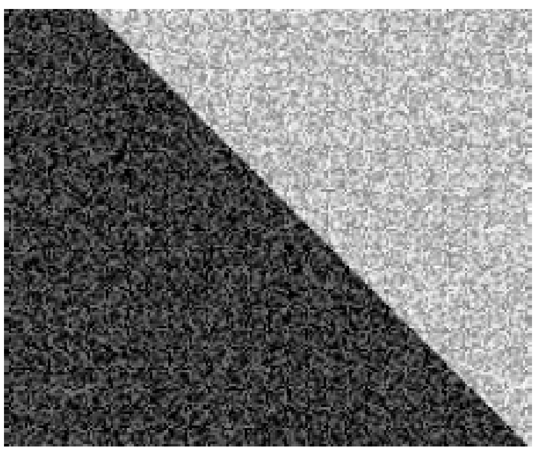
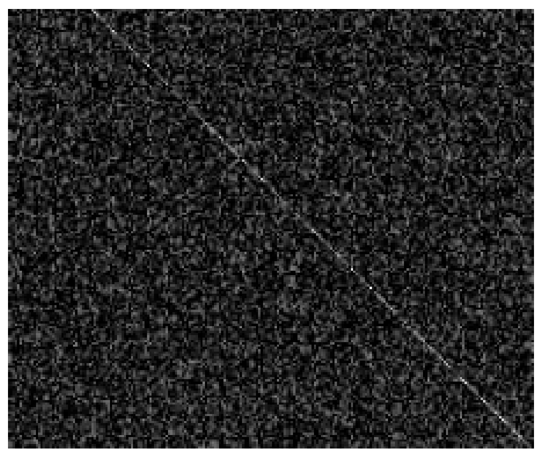


Image + Noise

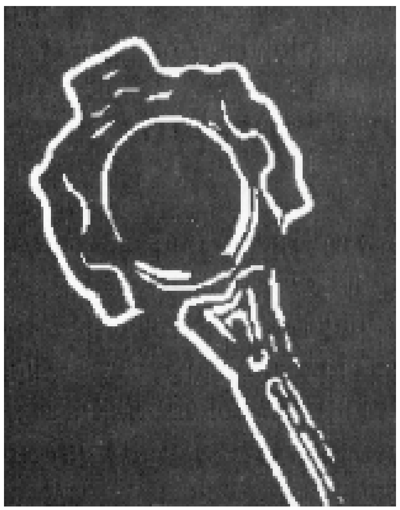


Derivatives detect edge *and* noise

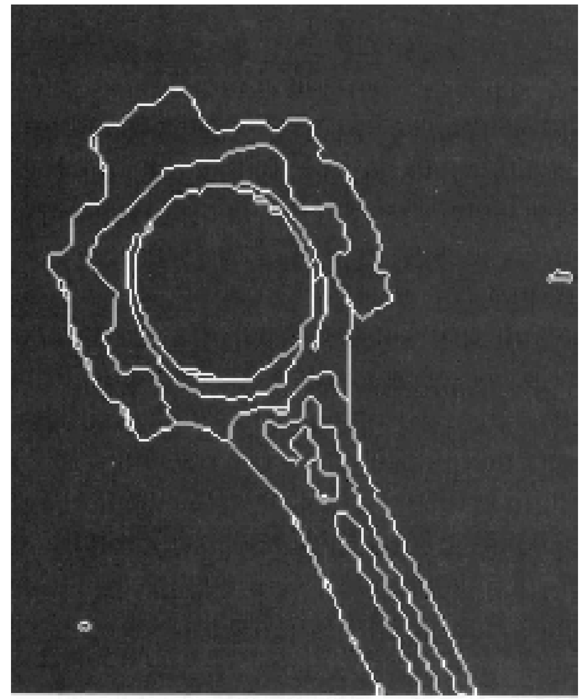
Smoothed derivative removes noise, but blurs edge

Issues to Address

<u>Edge thinning and linking</u>



smoothing+thresholding gives us a binary mask with “thick” edges



we want thin, one-pixel wide, connected contours

Canny Edge Detector

An important case study

Probably, the most used edge detection algorithm by C.V. practitioners

Experiments consistently show that it performs very well

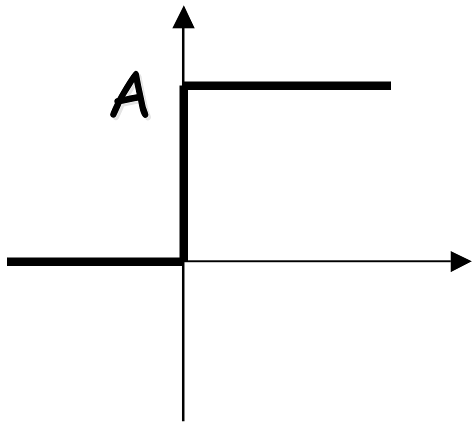
J. Canny *A Computational Approach to Edge Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 8, No. 6, Nov 1986

Formal Design of an Optimal Edge Detector

- Edge detection involves 3 steps:
  – Noise smoothing
  – Edge enhancement
  – Edge localization
- J. Canny formalized these steps to design an *optimal* edge detector

Edge Model (1D)

- An ideal edge can be modeled as an step



$$G(x) = \begin{cases} 0 & \text{if } x < 0 \\ A & \text{if } x \leq 0 \end{cases}$$

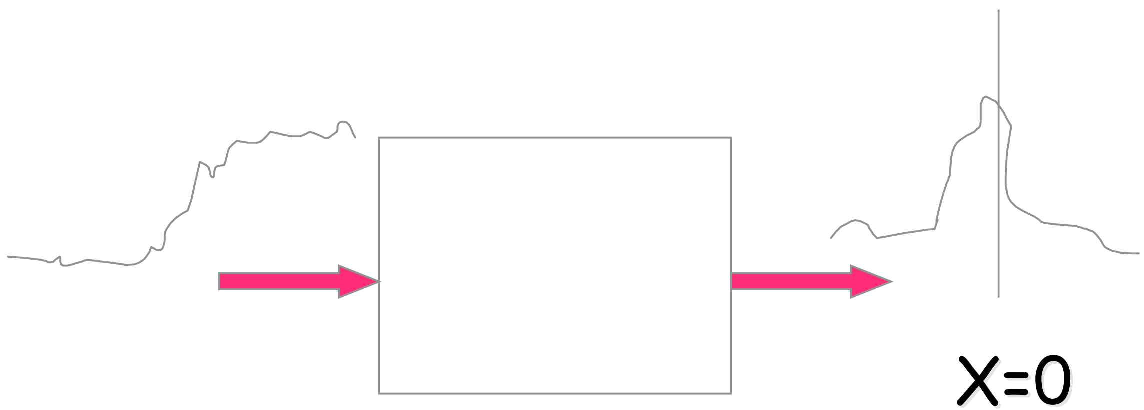
- Additive, White Gaussian Noise
  - RMS noise amplitude/unit length n_o^2

Performance Criteria (1)

- Good detection
  - The filter must have a stronger response at the edge location (x=0) than to noise

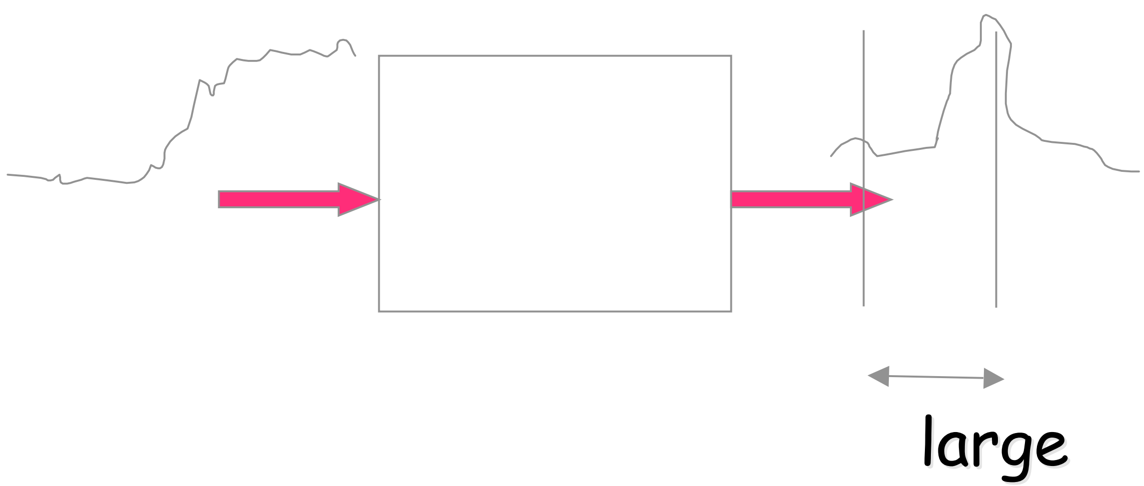
Performance Criteria (2)

- Good Localization
  - The filter response must be maximum very close to x=0



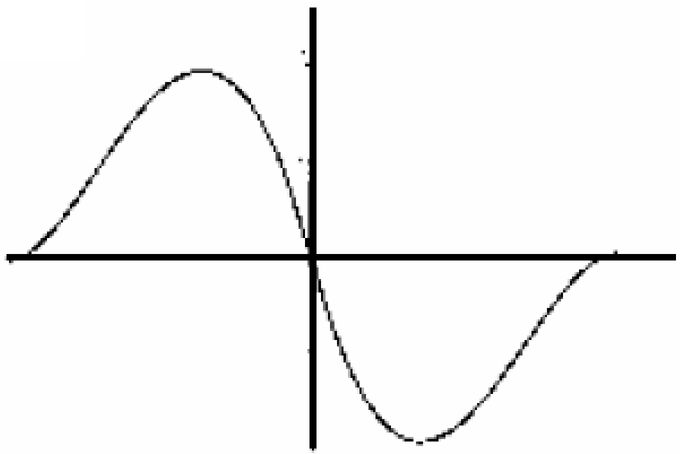
Performance Criteria (3)

- Low False Positives
  - There should be only one maximum in a reasonable neighborhood of x=0

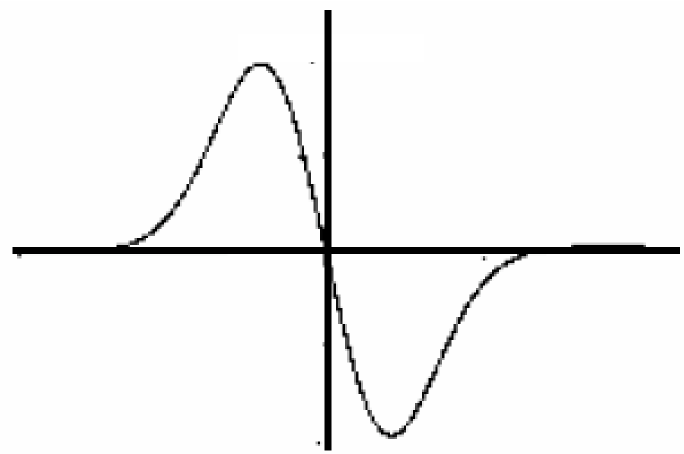


Canny Edge Detector

- Canny found a linear, continuous filter that maximized the three given criteria.
- There is no closed-form solution for the optimal filter.
- However, it looks VERY SIMILAR to the derivative of a Gaussian.



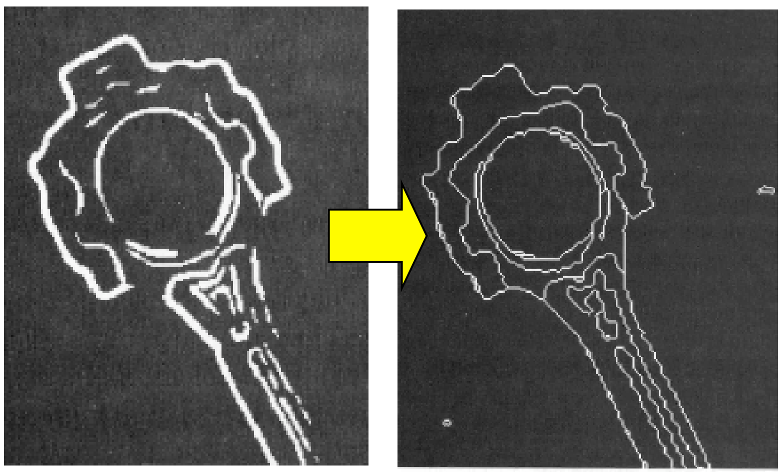
Canny



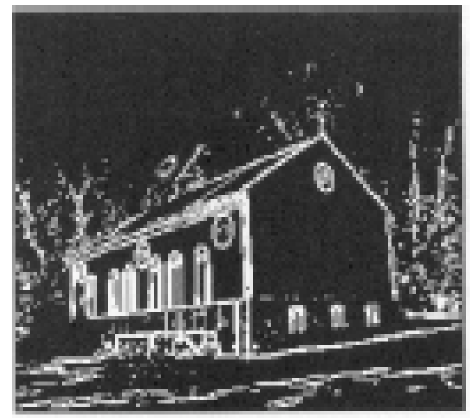
Derivative of Gaussian

Recall: Practical Issues for Edge Detection

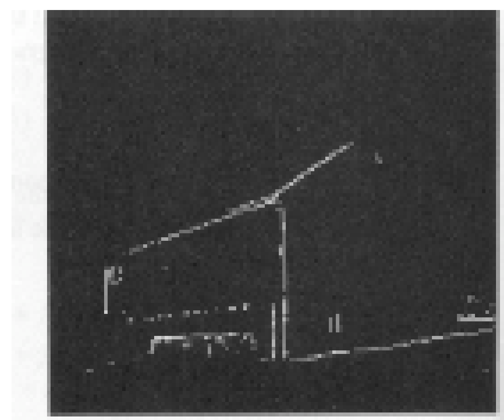
Thinning and linking



Choosing a magnitude threshold



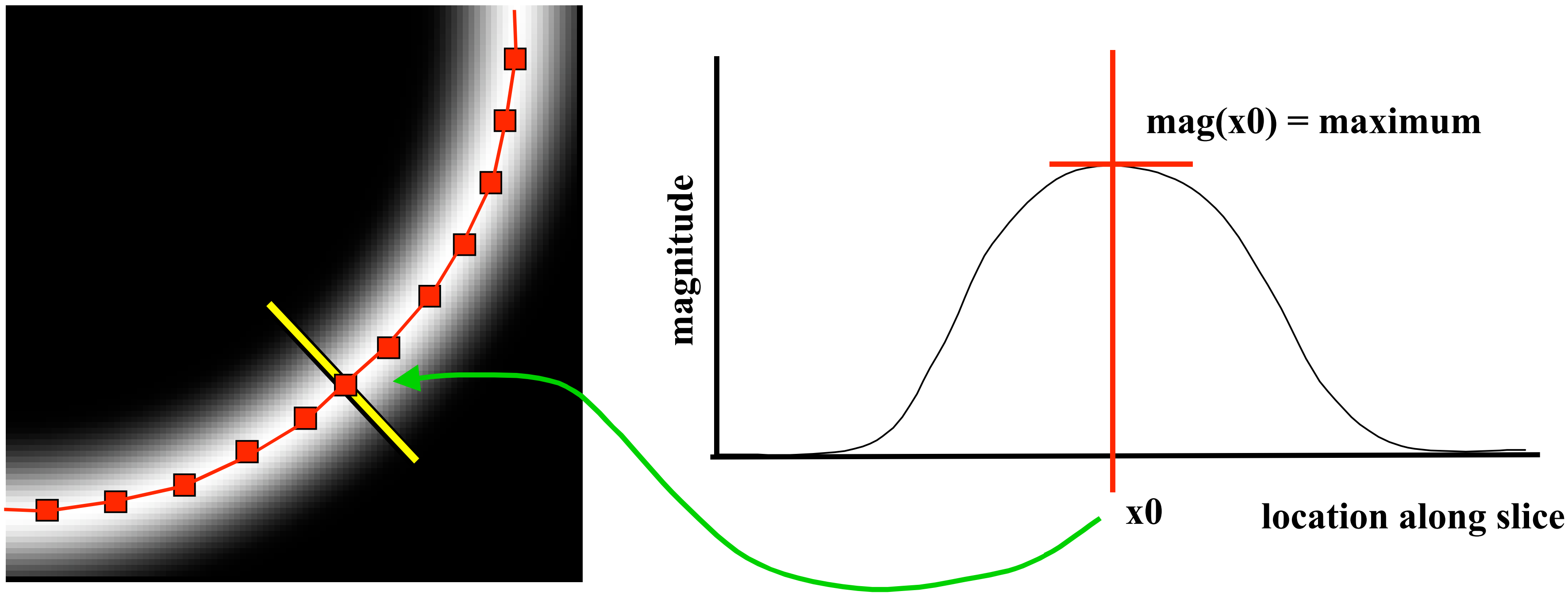
OR



Canny has good answers to all!

Thinning

note: do thinning before thresholding!

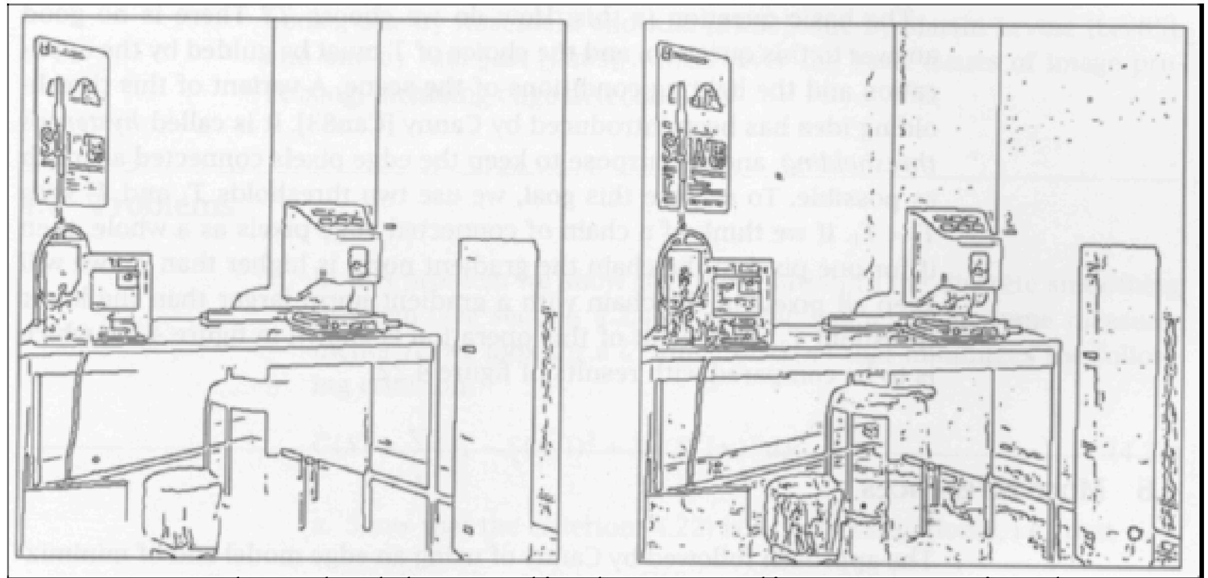


We want to mark points along curve where the magnitude is largest.

We can do this by looking for a maximum along a 1D intensity slice normal to the curve (non-maximum supression).

These points should form a one-pixel wide curve.

Which Threshold to Pick?



Two thresholds applied to gradient magnitude

T = 15 T = 5

problem:

- If the threshold is too high:
  - –Very few (none) edges
    - High MISDETECTIONS, many gaps
- If the threshold is too low:
  - –Too many (all pixels) edges
    - High FALSE POSITIVES, many extra edges

SOLUTION: Hysteresis Thresholding

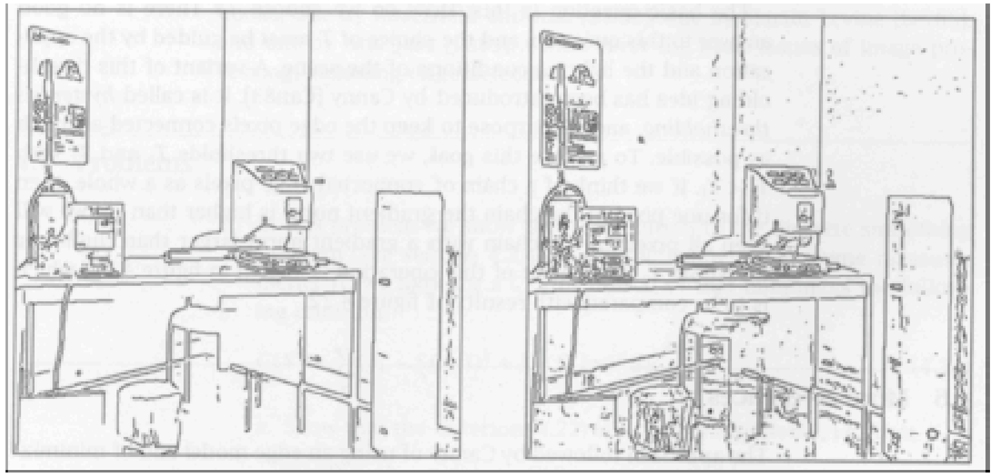
Allows us to apply both! (e.g. a "fuzzy" threshold)

- Keep both a high threshold H and a low threshold L.
- Any edges with strength < L are discarded.
- Any edge with strength > H are kept.
- An edge P with strength <u>between</u> L and H is kept only if there is a path of edges with strength > L connecting P to an edge of strength > H.

- In practice, this thresholding is combined with edge linking to get connected contours

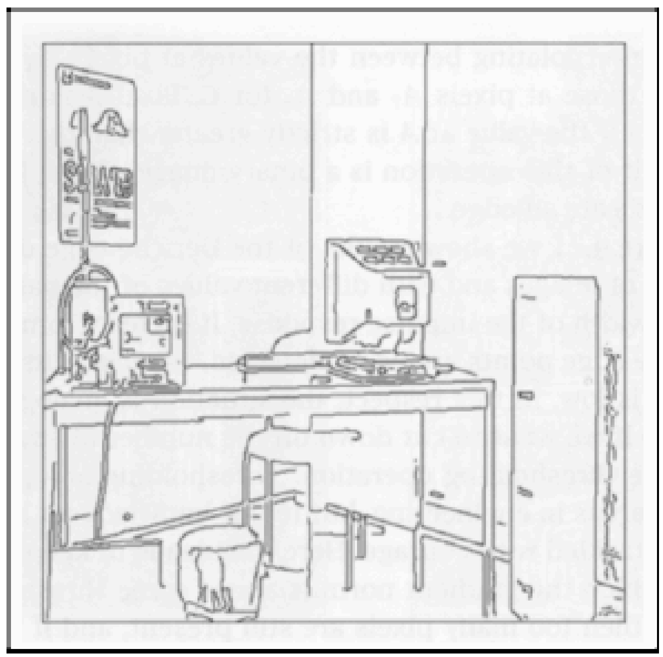
Example of Hysteresis Thresholding

T=15



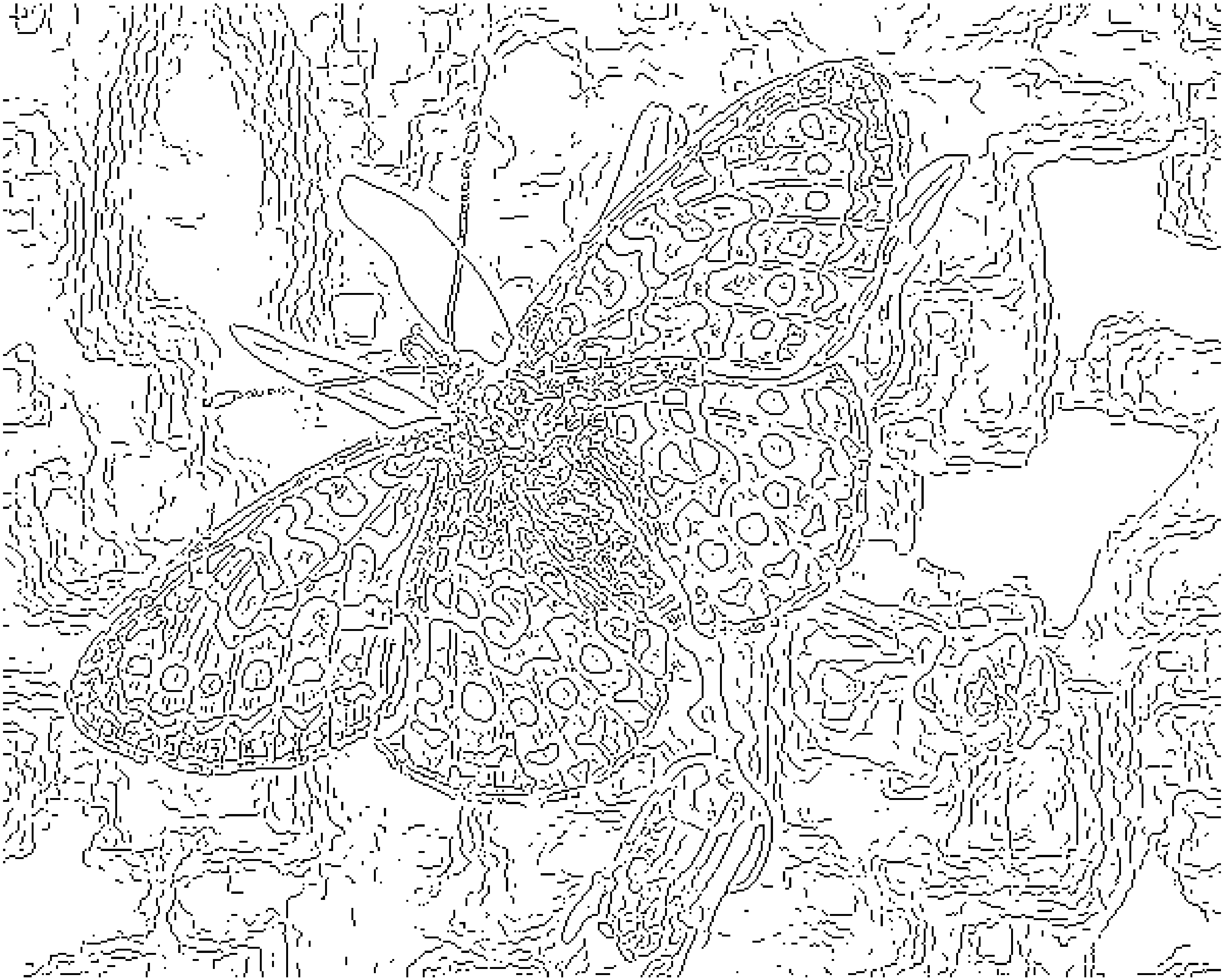
T=5

Hysteresis
T_h=15 T_l = 5



Hysteresis
thresholding

Canny Edges: Examples



fine scale
high
threshold

coarse
scale,
high
threshold

coarse
scale
low
threshold

Complete Canny Algorithm

1. Compute x and y derivatives of image

$$I_x = G_\sigma^x * I \quad I_y = G_\sigma^y * I$$

2. Compute magnitude of gradient at every pixel

$$M(x,y) = |\nabla I| = \sqrt{I_x^2 + I_y^2}$$

3. Eliminate those pixels that are not local maxima of the magnitude in the direction of the gradient

4. Hysteresis Thresholding

   - Select the pixels such that $M > T_h$ (high threshold)
   - Collect the pixels such that $M > T_l$ (low threshold) that are neighbors of already collected edge points

See textbook for more details.