Digital Image Processing

Segmentation - Edge Detection

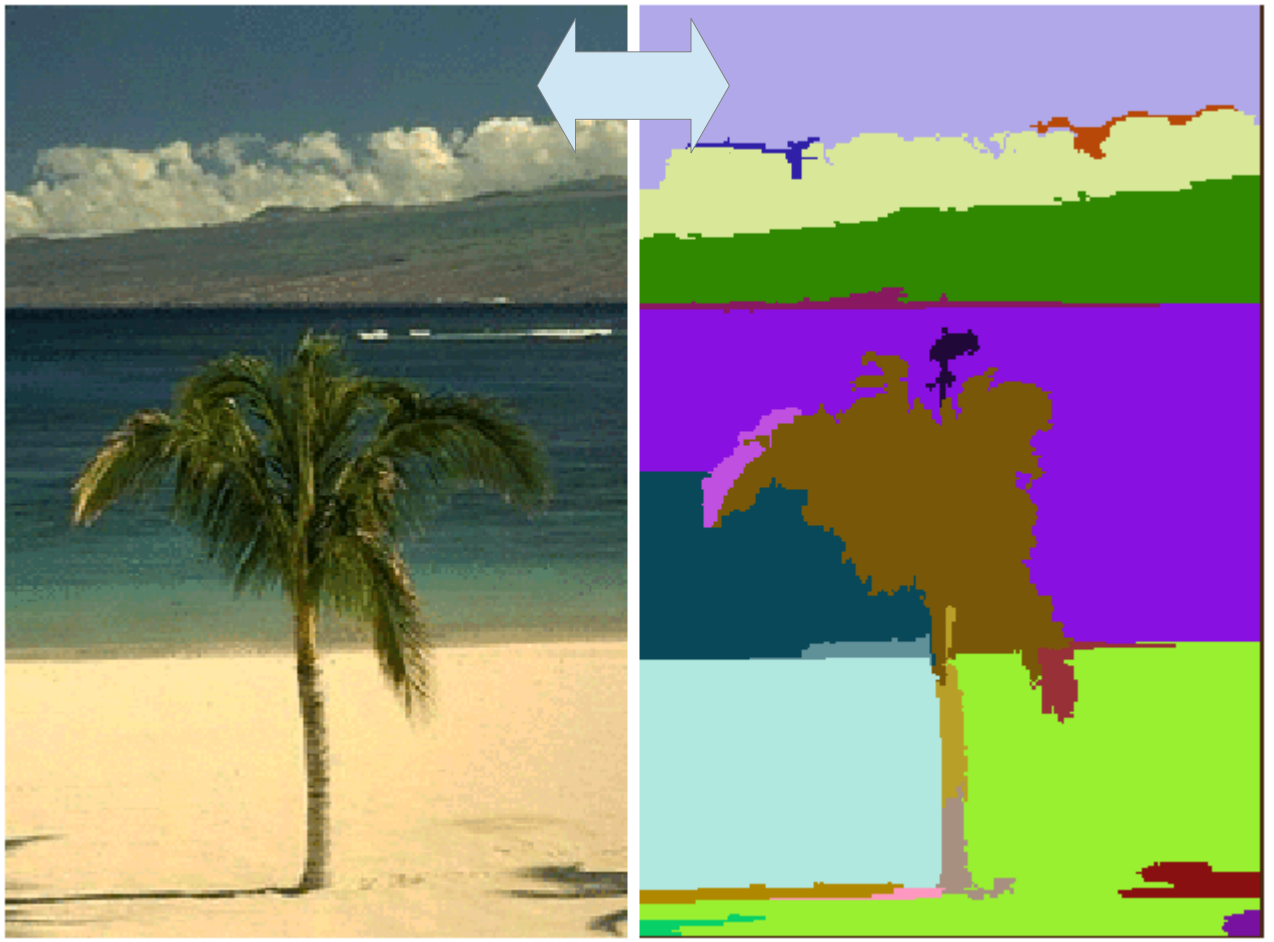
Suyash P. Awate

Segmentation

- Definition : Process of partitioning an image into parts
  - Parts are called **segments**
- Why ?
  - Simplification of information content
  - Towards higher-level understanding / interpretation
- Segmentation = **labeling**
  - Label = categorical variable
    - {A, B, C}, {1, 2, 3}, etc.
  - Assign a label to each pixel
  - All pixels with same label = one segment
- Typically, 1 segment = 1 (or more) connected regions

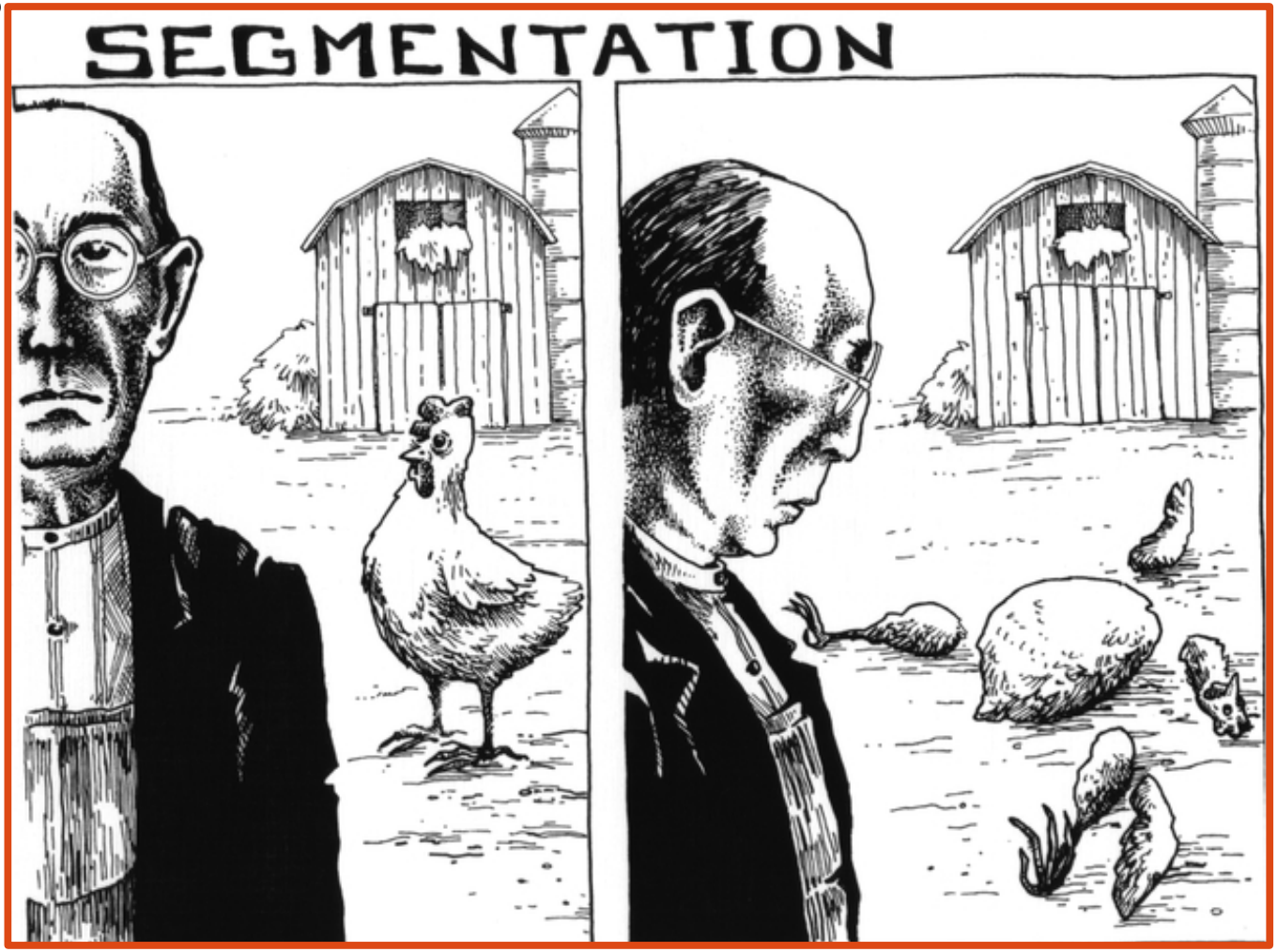
Segmentation

- Segmentation example



Segmentation

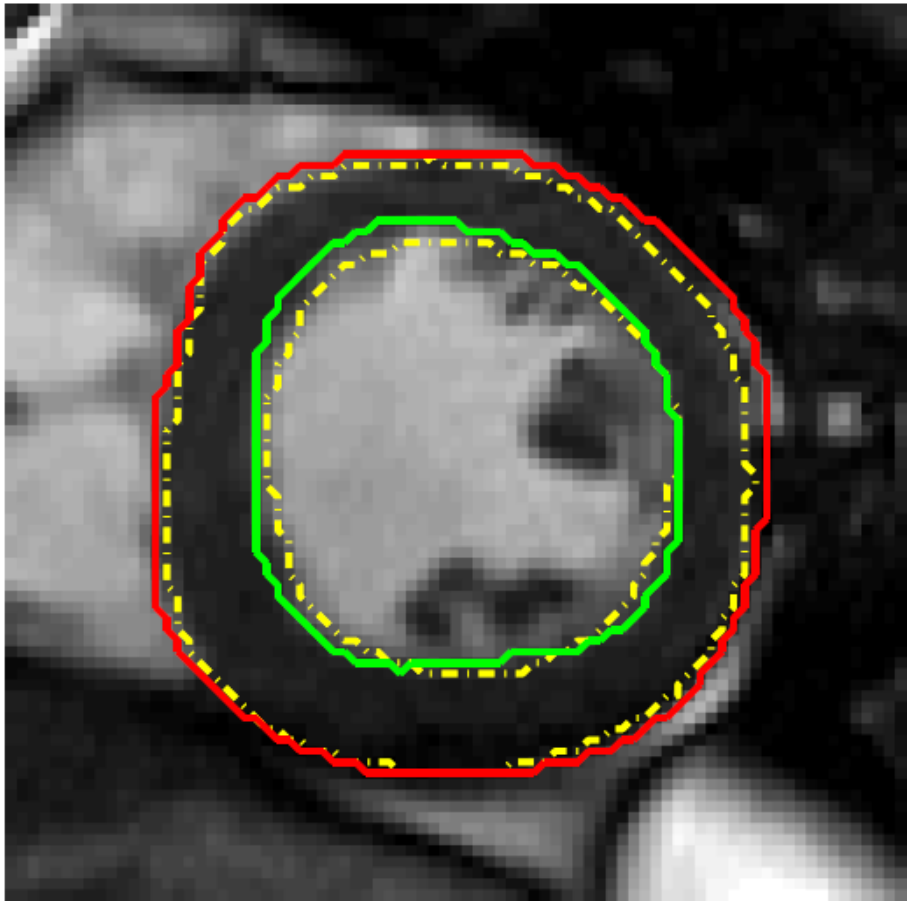
- Human perception
  - Instead of perceiving **indivisible** objects, we perceive objects in terms of their **labeled** or **categorized parts**
  - We break objects into parts that we have learned are relevant or important



cognitrn.psych.indiana.edu/rgoldsto/projects/segmentation.html

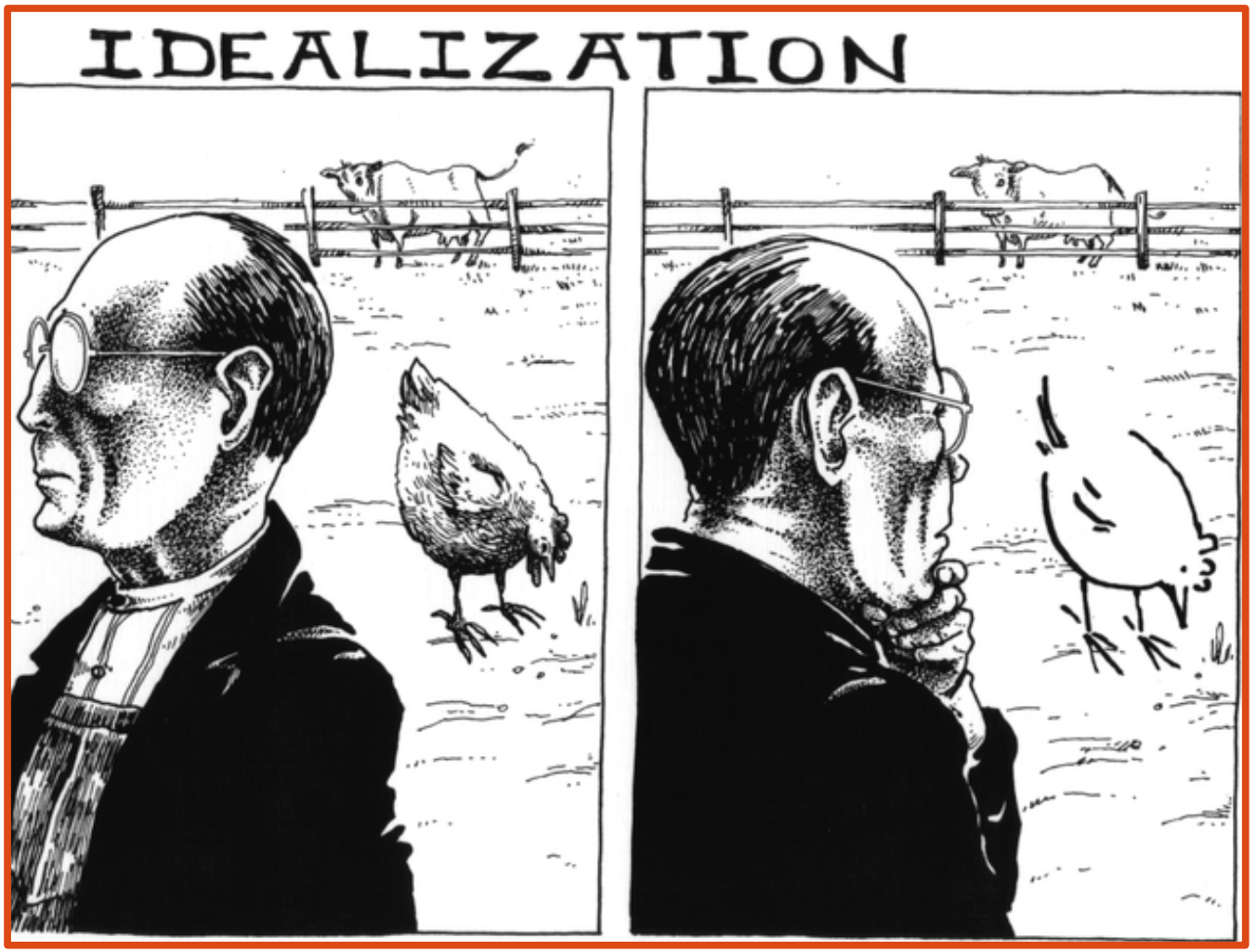
Segmentation

- **Delineation**
  - Process of determining the outline of **an object** in image



Segmentation

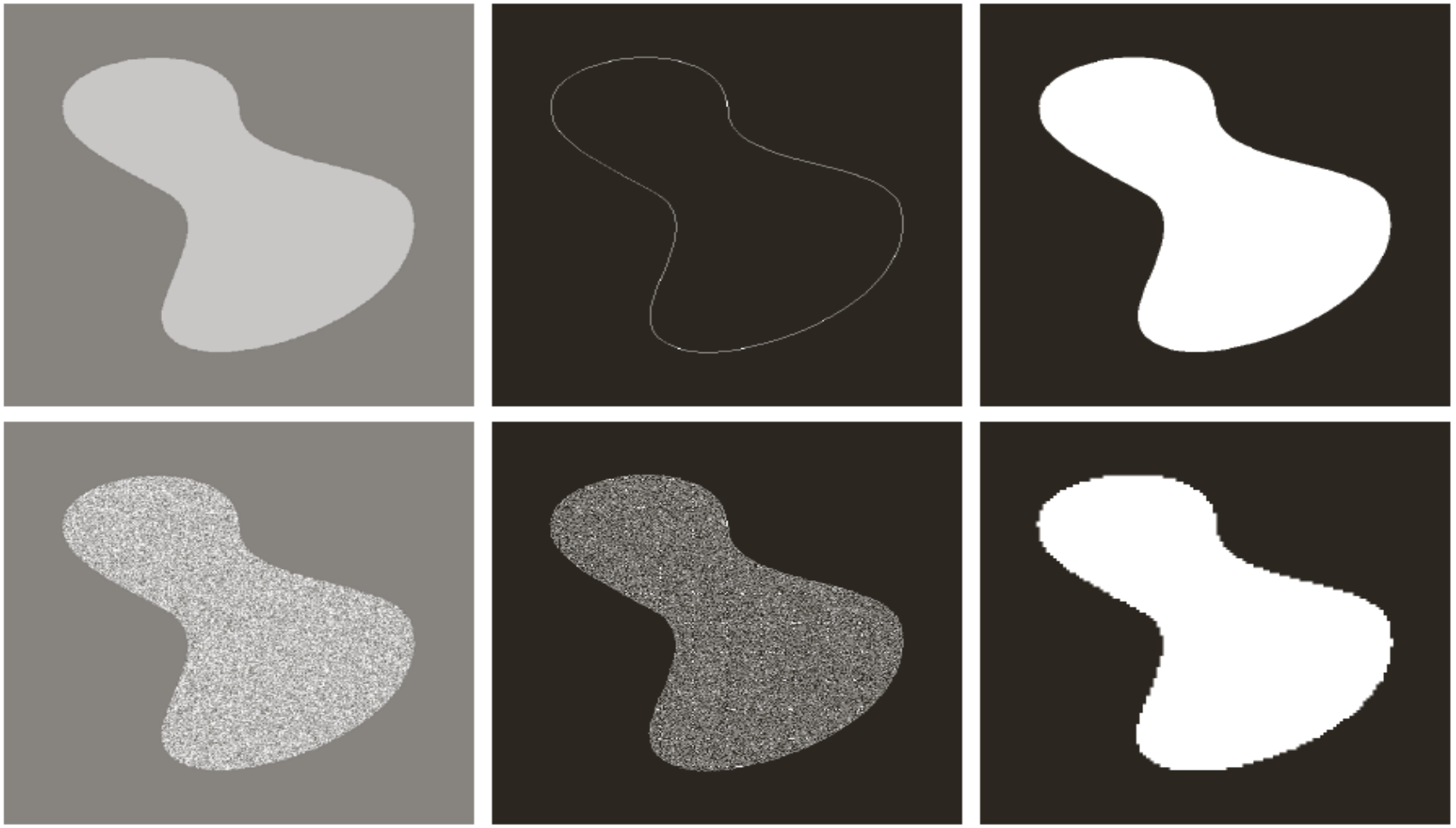
- Human perception (delineation)
  - Objects are often simplified so as to capture the **basic essence** of the **underlying concept**



cognitrn.psych.indiana.edu/rgoldsto/projects/idealization.html

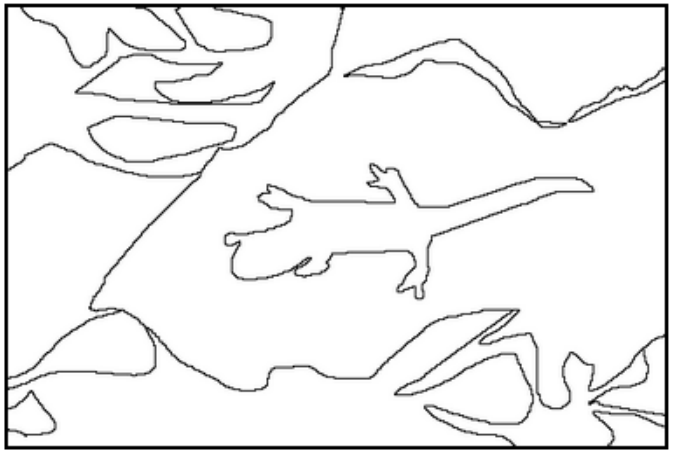
Segmentation

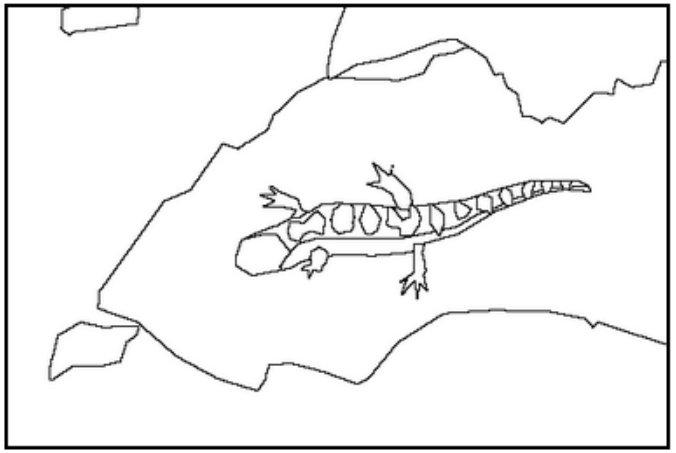
- Image segmentation
  - Top row: noiseless image, edge detection, segmentation
  - Bottom row: noisy image, edge detection, segmentation

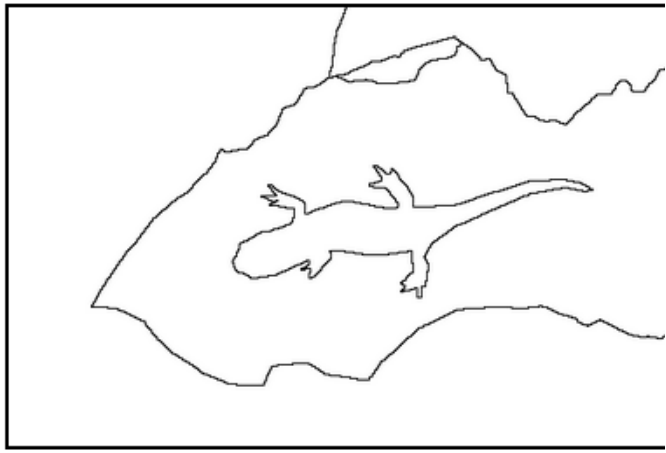


Segmentation

- Subjective
  - Segmentation is in the eye of the beholder !



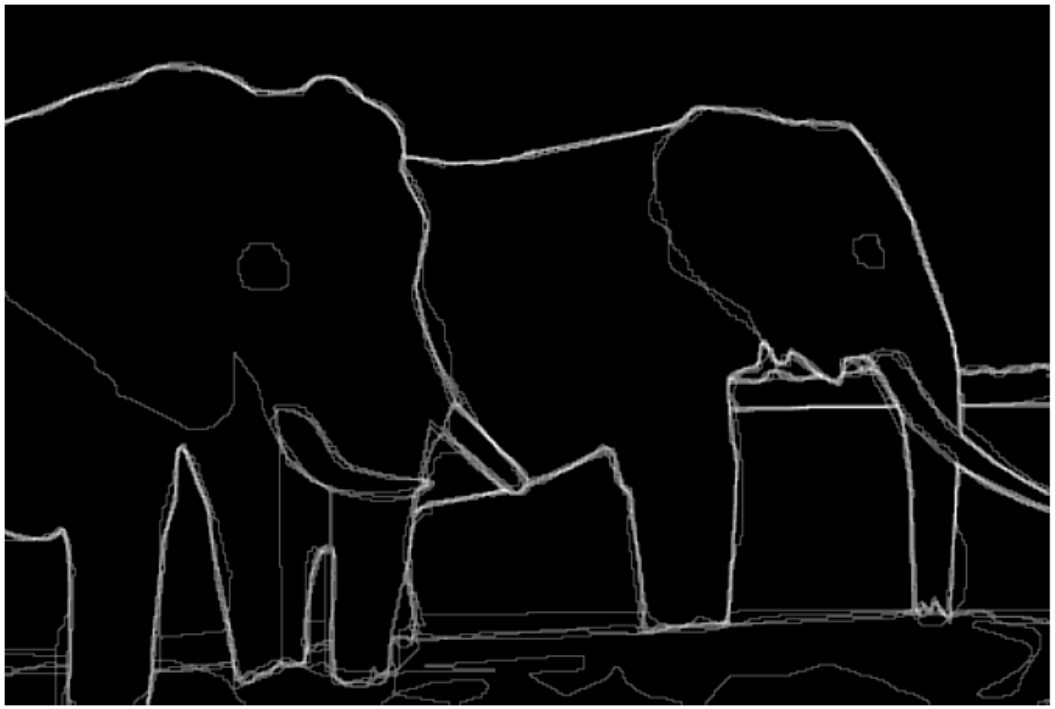




www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html

Segmentation

- Subjective
  - Segmentation is in the eye of the beholder !

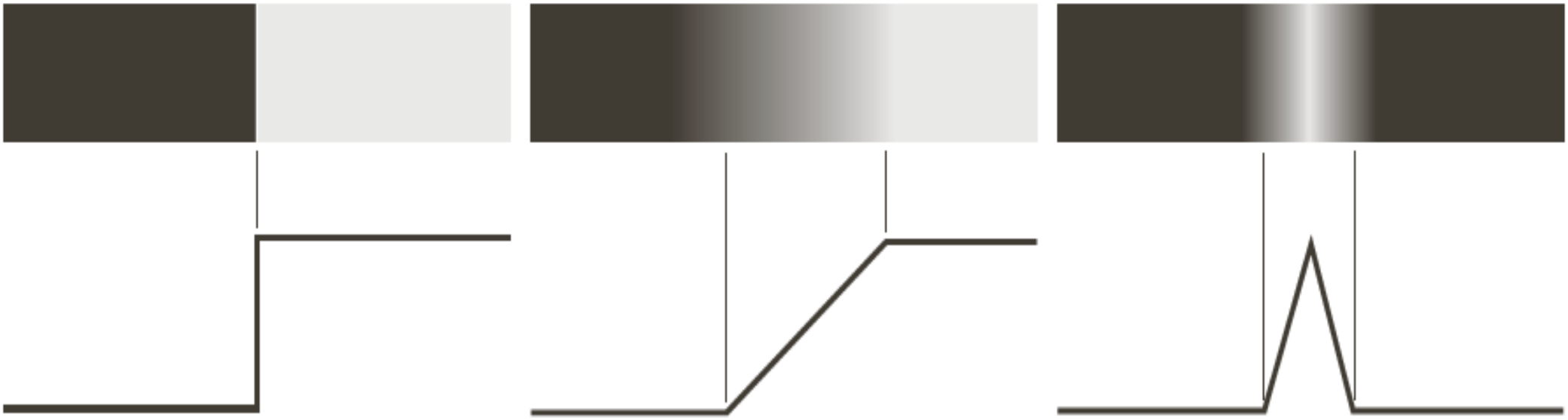


Edge Detection

- Edge = sharp change in image intensity over space
  - Large magnitude of gradient

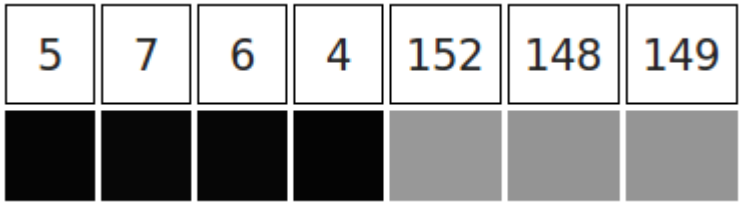
Edge Detection

- Ideal edges
  - (1) Step edge
  - (2) Ramp edge
  - (3) Double edge / roof edge

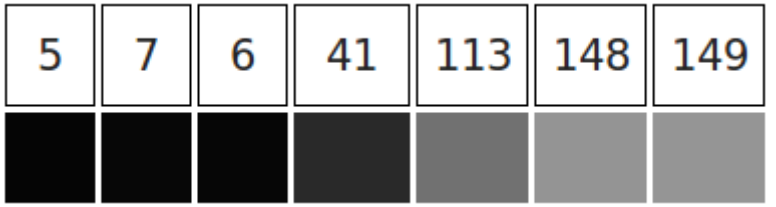


Edge Detection

- Non-trivial, ambiguous
  - Whether to call some intensity change an edge or not ?
  - 1D image with a clear edge

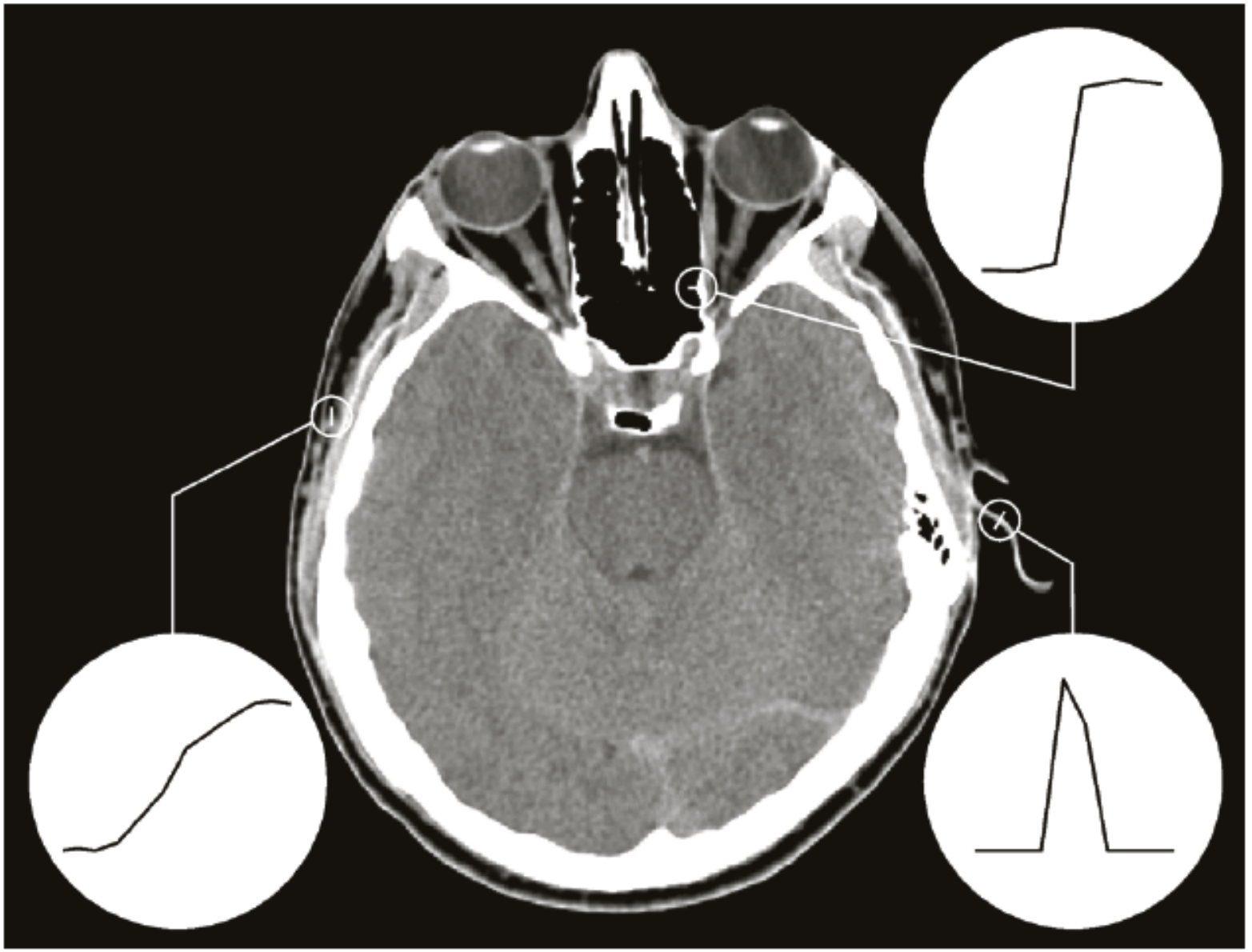


  - 1D image where edges are unclear



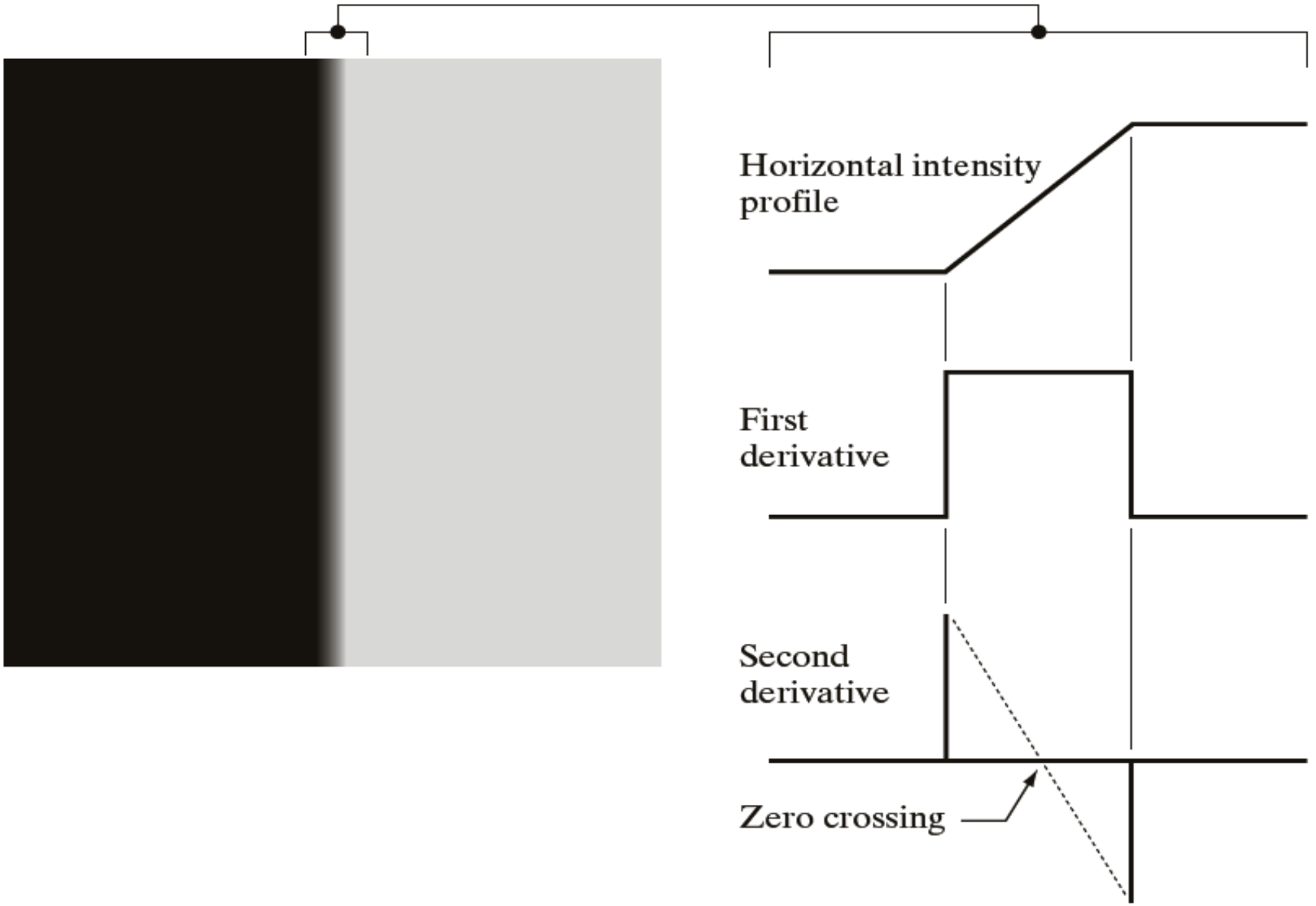
Edge Detection

- Edges in the real world → ramp edges



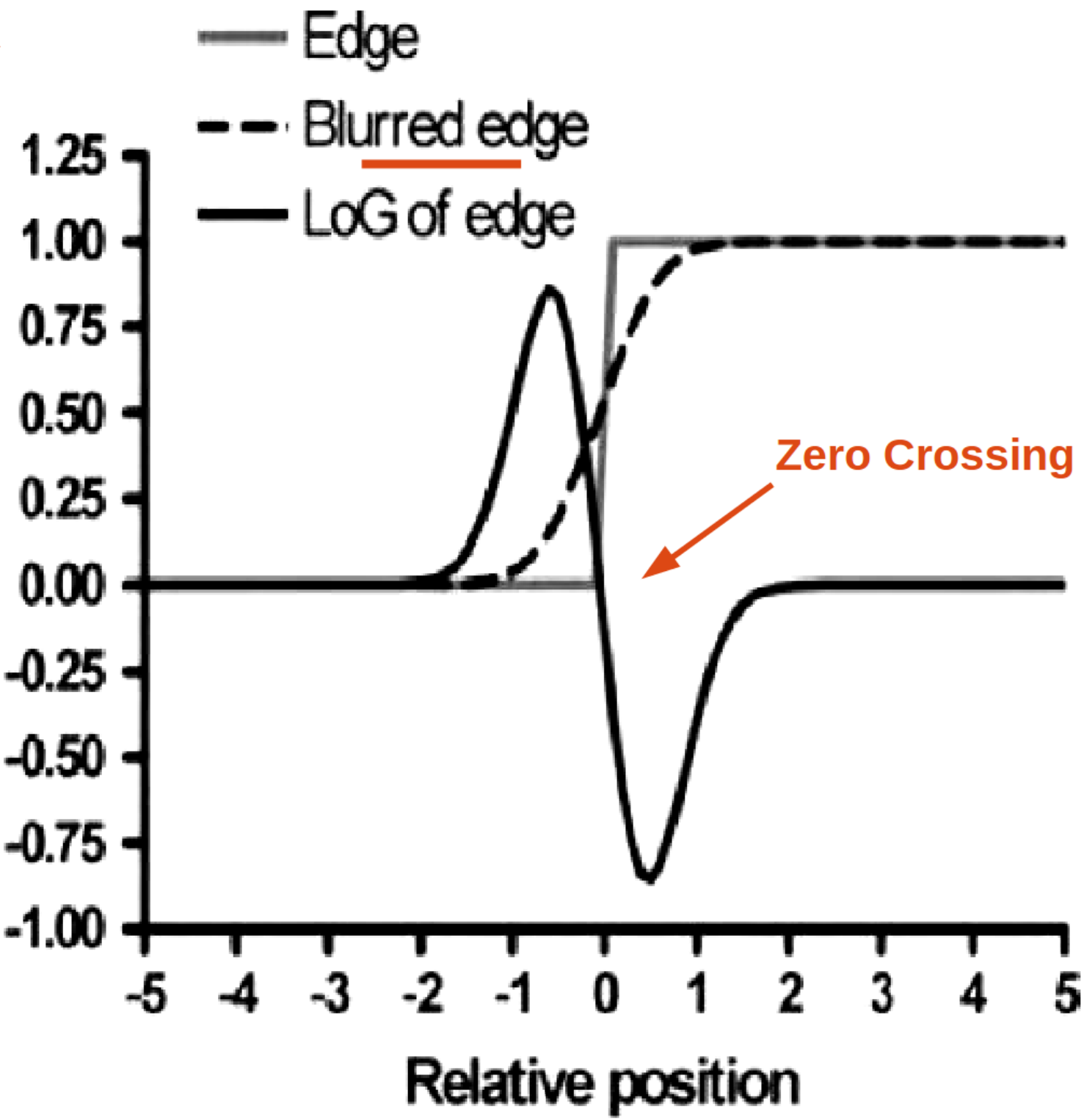
Edge Detection

- Ideal (ramp) edge



Edge Detec

- A (ramp) edge
  - **Blurred**



Edge Detection

- **“Location” of an edge** can be determined by the location of **zero-crossing of Laplacian**

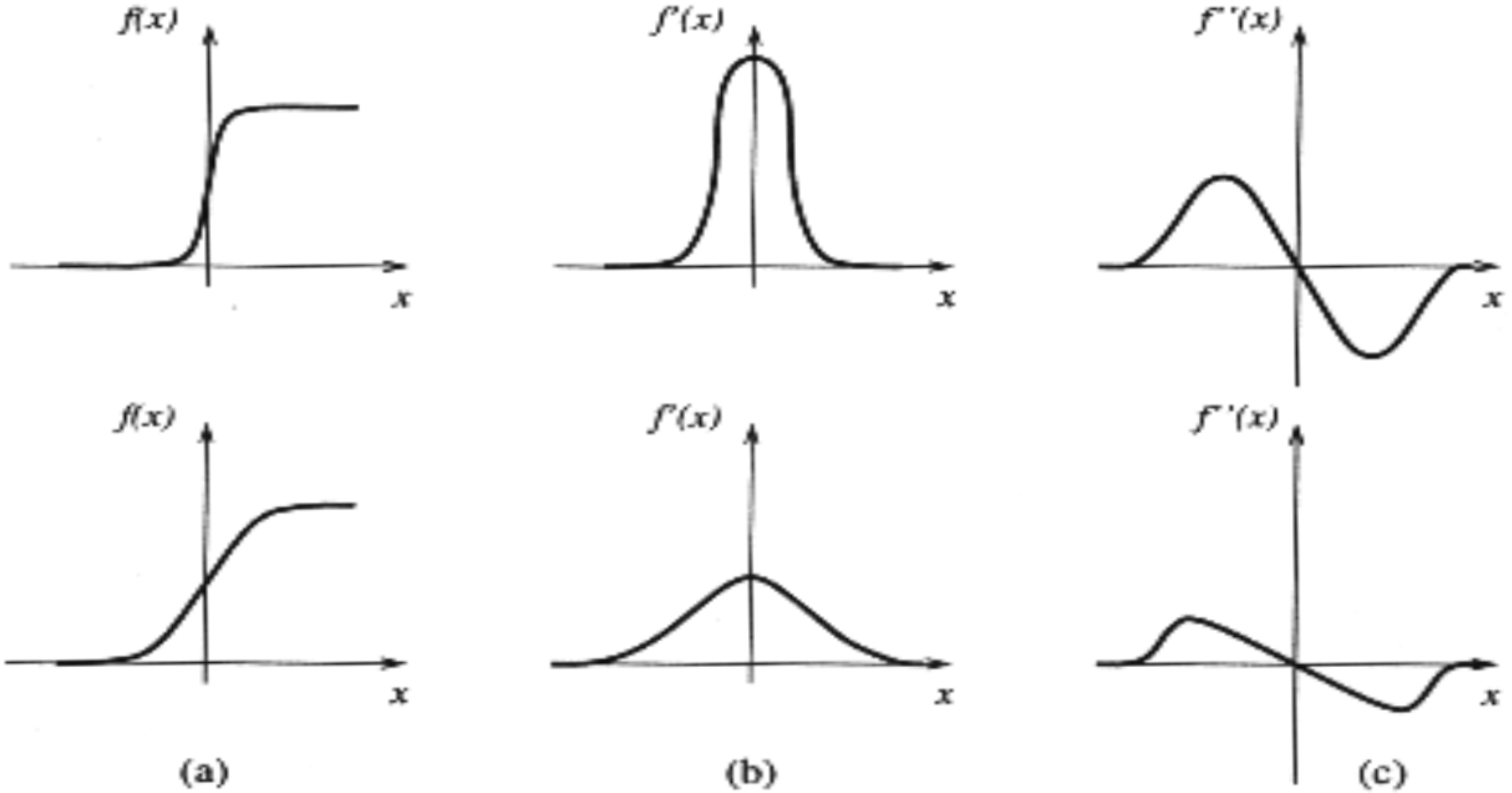
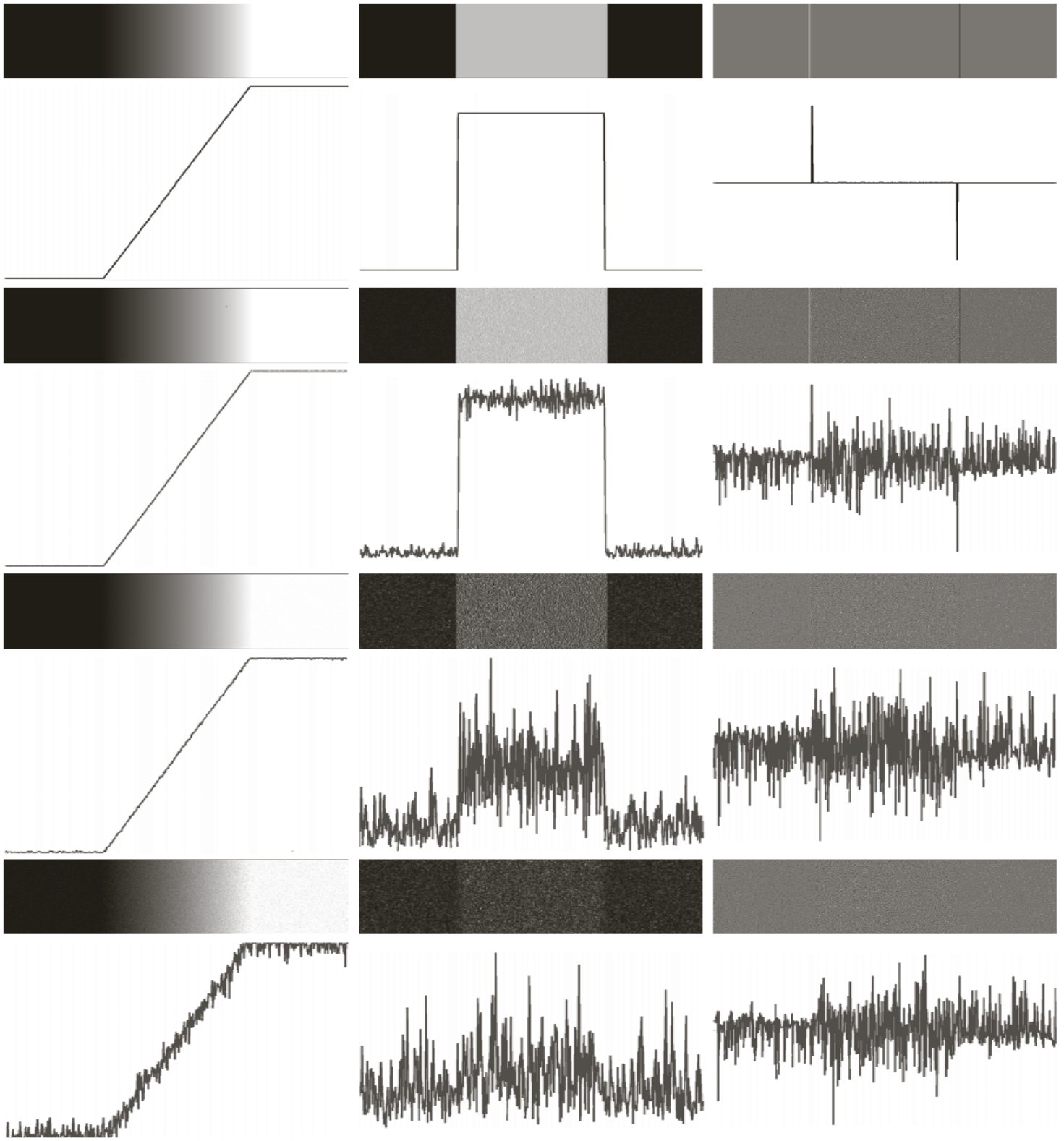


Figure 4.20: *1D edge profile of the zero-crossing.*

Edge Detec

- A (ramp) edge
  - Noisy



Edge Detection

- **Line detection**
  - Masks capturing directional derivatives along different directions

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

2	−1	−1
−1	2	−1
−1	−1	2

+45°

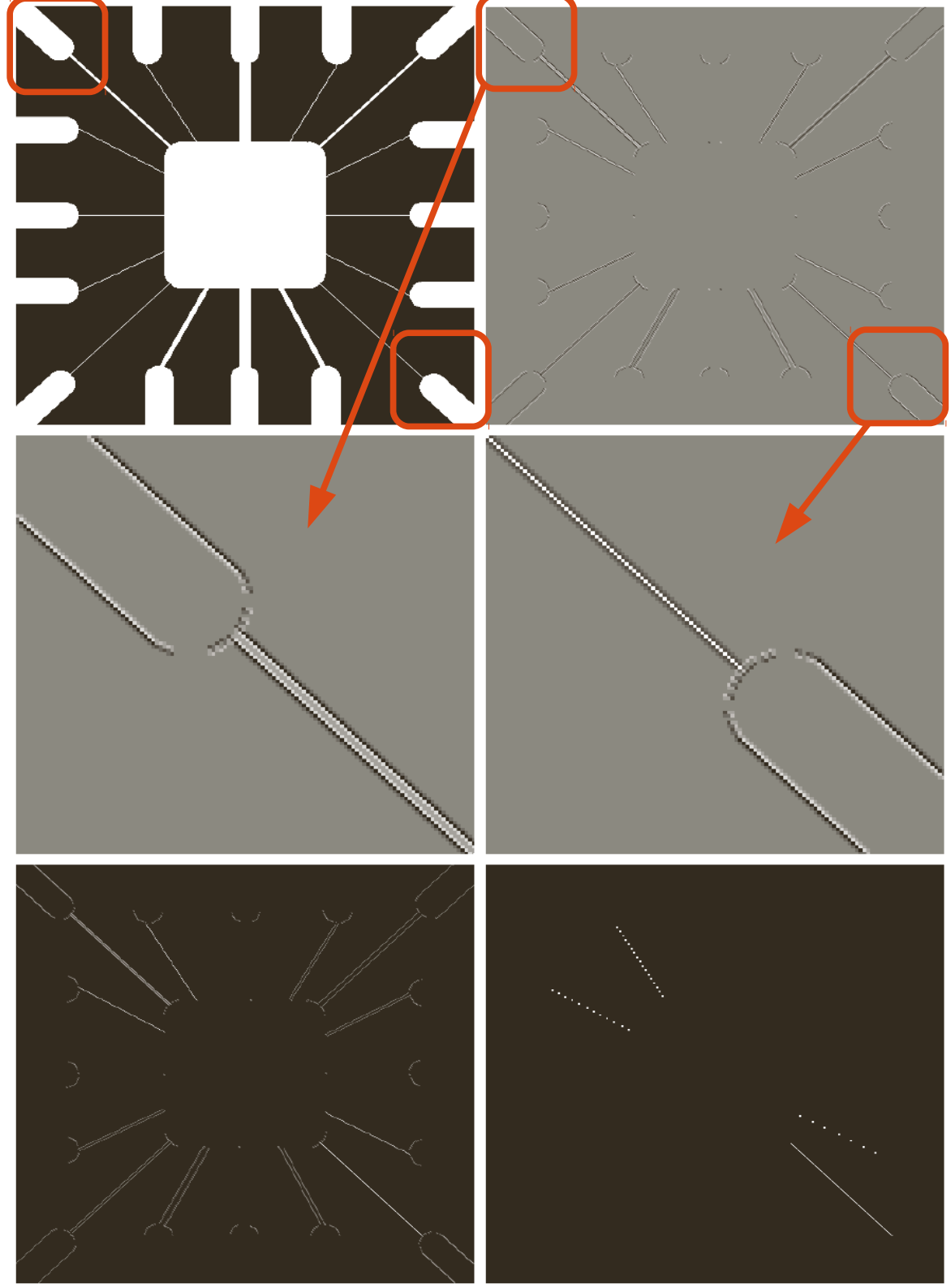
−1	2	−1
−1	2	−1
−1	2	−1

Vertical

−1	−1	2
−1	2	−1
2	−1	−1

−45°

- Line detection

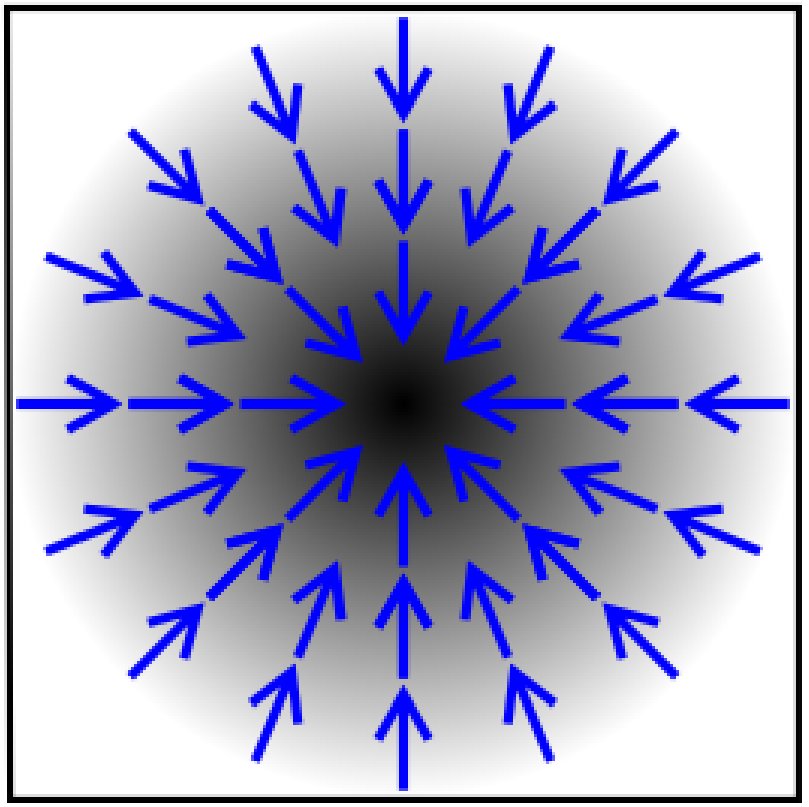


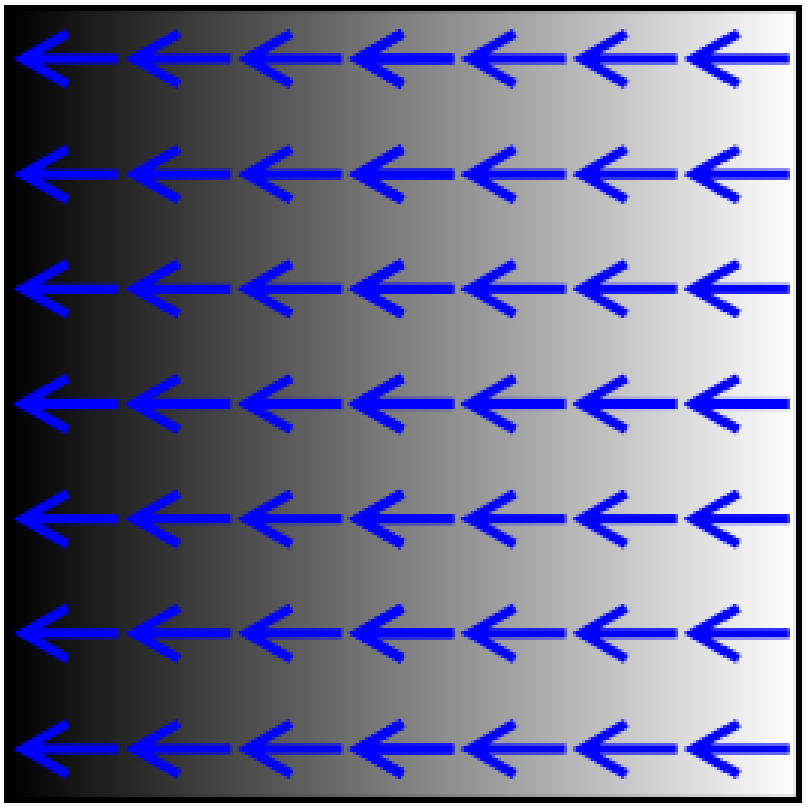
a	b
c	d
e	f

FIGURE 10.7
(a) Image of a wire-bond template. (b) Result of processing with the +45° line detector mask in Fig. 10.6. (c) Zoomed view of the top left region of (b). (d) Zoomed view of the bottom right region of (b). (e) The image in (b) with all negative values set to zero. (f) All points (in white) whose values satisfied the condition $g \geq T$, where g is the image in (e). (The points in (f) were enlarged to make them easier to see.)

Edge Detection

- **Gradient**
  - Can be used for edge detection
  - Gradient (column) vector = [df/dx df/dy]' = [Gx Gy]'





Edge Detection

- Masks for computing gradient [Gx Gy]
  - Averaging in neighborhood
    → noise reduction
  - **Prewitt** operators

$$\begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix}$$

  - Prewitt operators are separable

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

Edge Detection

- Masks for computing gradient [Gx Gy]
  - Averaging in neighborhood
    → noise reduction
  - **Sobel** operators

$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

  - Sobel operators are separable

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

Edge Detection

- Gradient (sobel)
  - Test image

  - Gradient x-component magnitude
    → |df/dx| = |Gx|

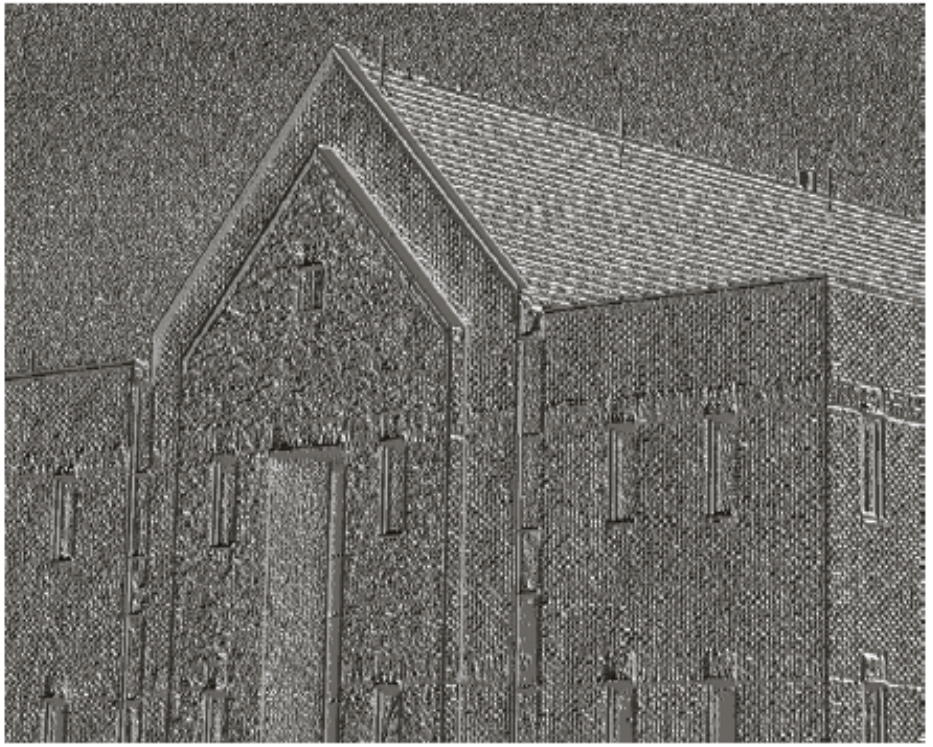
  - Gradient y-component magnitude
    → |df/dy| = |Gy|

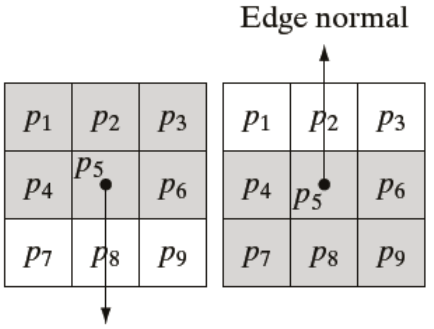
  - Gradient magnitude

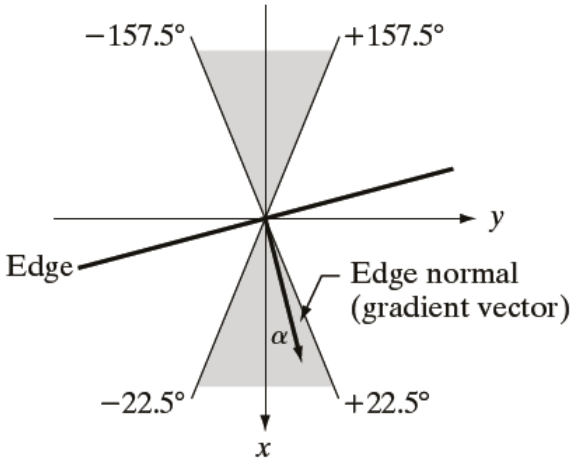
$$\mathbf{G} = \sqrt{\mathbf{G}_x^{\,2} + \mathbf{G}_y^{\,2}}$$

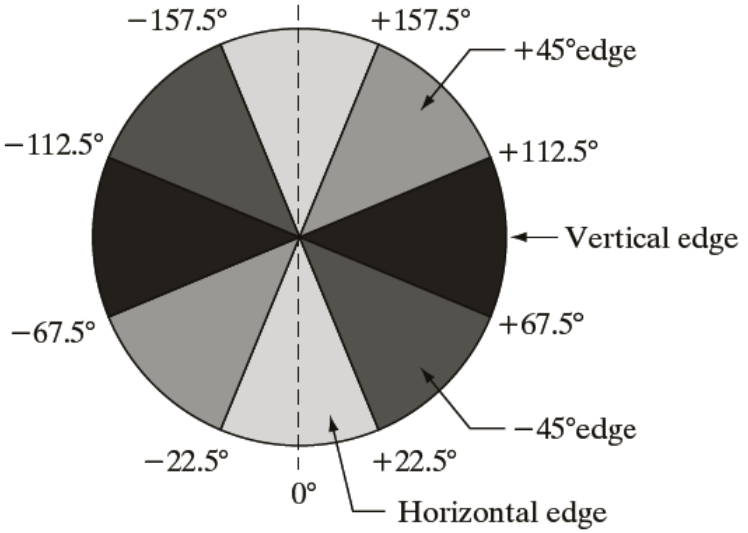
Edge Detection

- Gradient (sobel)
  - Gradient direction (angle) $\Theta = \operatorname{atan2}(\mathbf{G}_y, \mathbf{G}_x)$
    - Non-informative for edge detection
    - Gives direction of edge



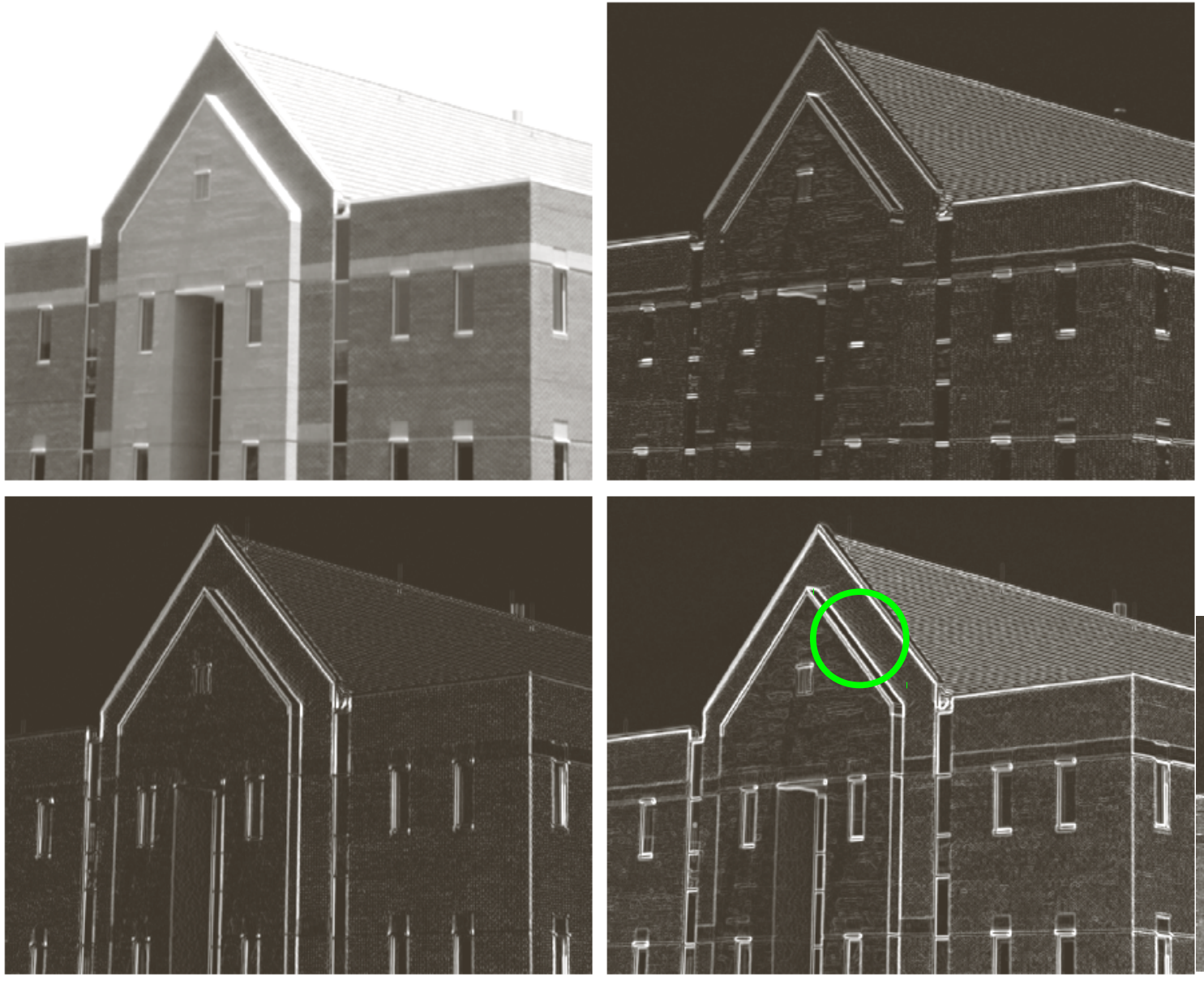






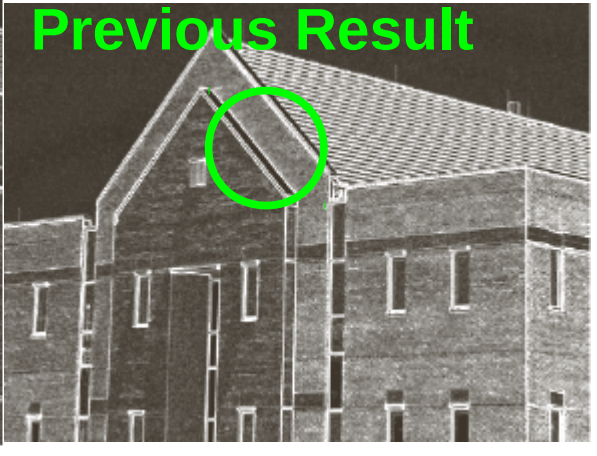
Edge Detection

- Sobel gradient after smoothing (5x5 mean filter)



a b
c d

FIGURE 10.18
Same sequence as in Fig. 10.16, but with the original image smoothed using a 5 × 5 averaging filter prior to edge detection.



Edge Detection

- **Marr-Hildreth algorithm for edge detection** (1980)

  (1/2) Convolve image with LoG

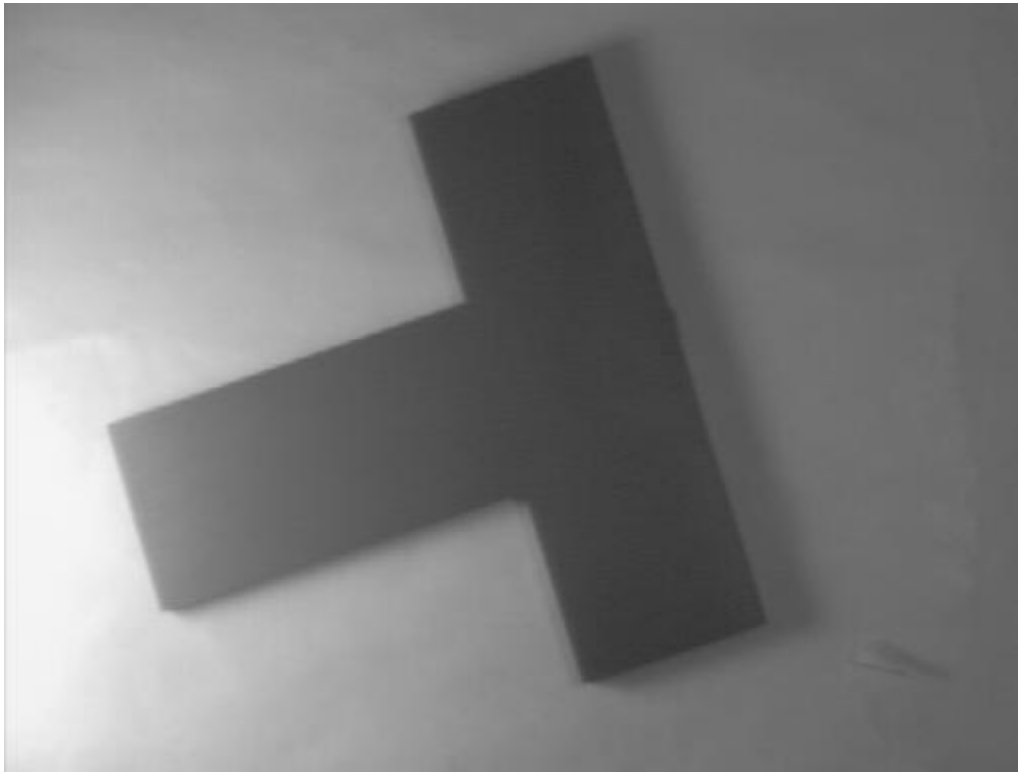
  - Approximated as DoG

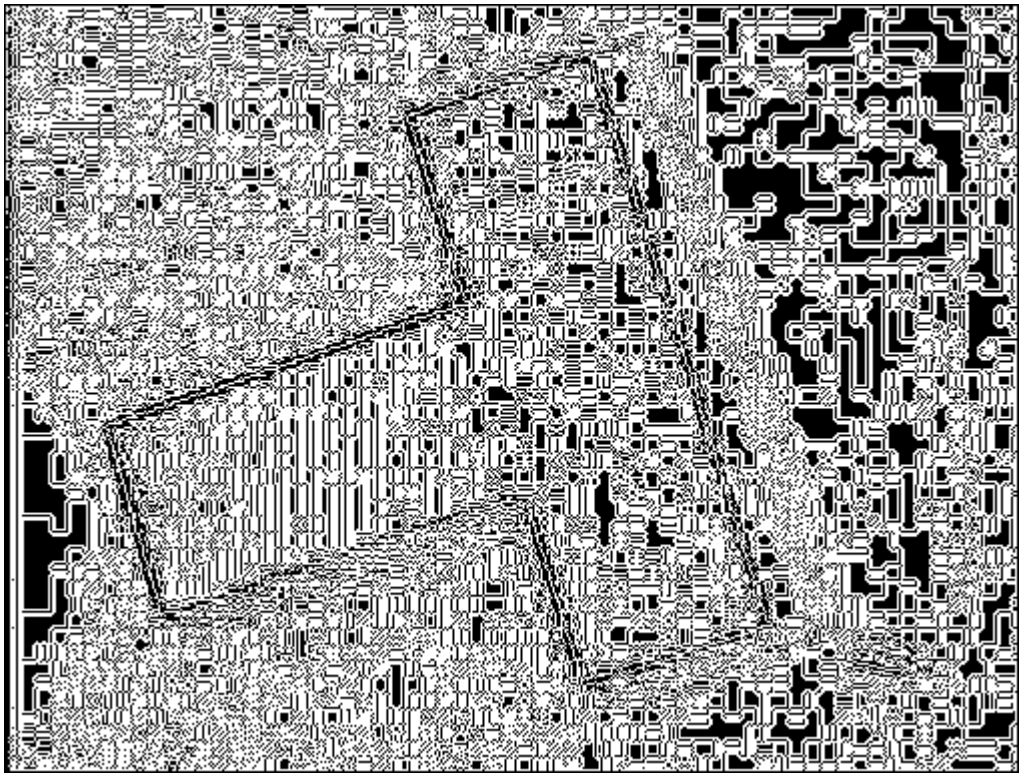
  (2/2) Detect zero crossings

  - Compare signs of pixel intensities of neighbors
    - Left versus right
    - Up versus down
    - NW versus SE
    - NE versus SW
  - If **signs different** and **magnitude of difference > threshold**, then, edge found

Edge Detection

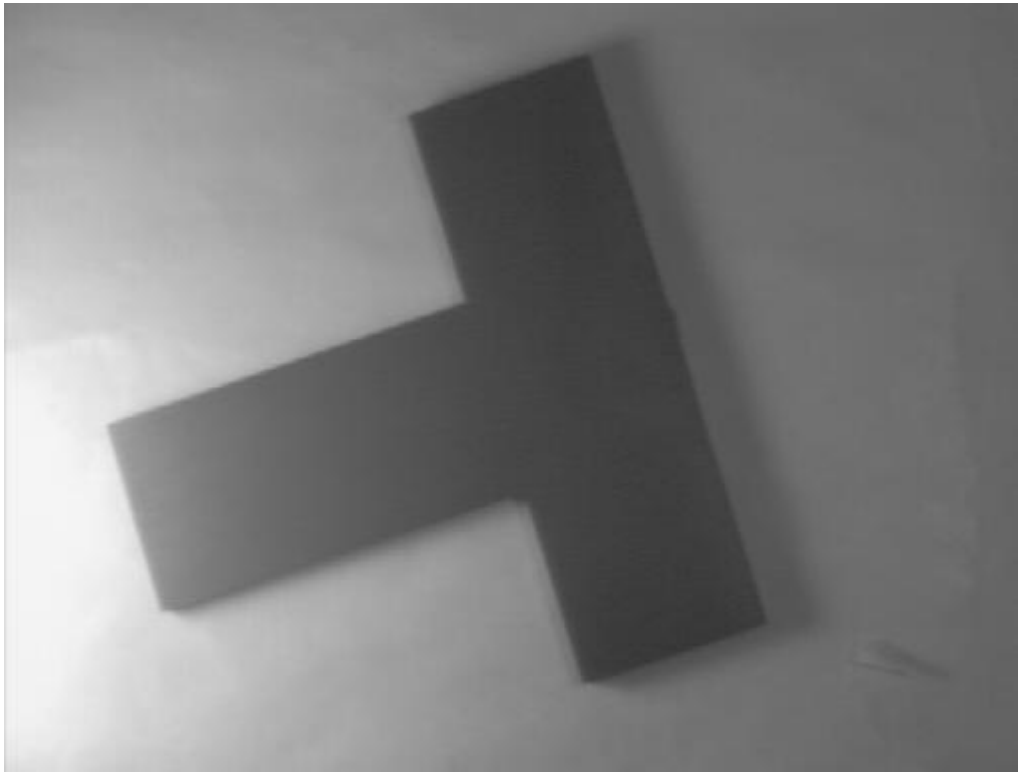
- Marr-Hildreth algorithm for edge detection
  - Gaussian variance = 0
  - Threshold = 0

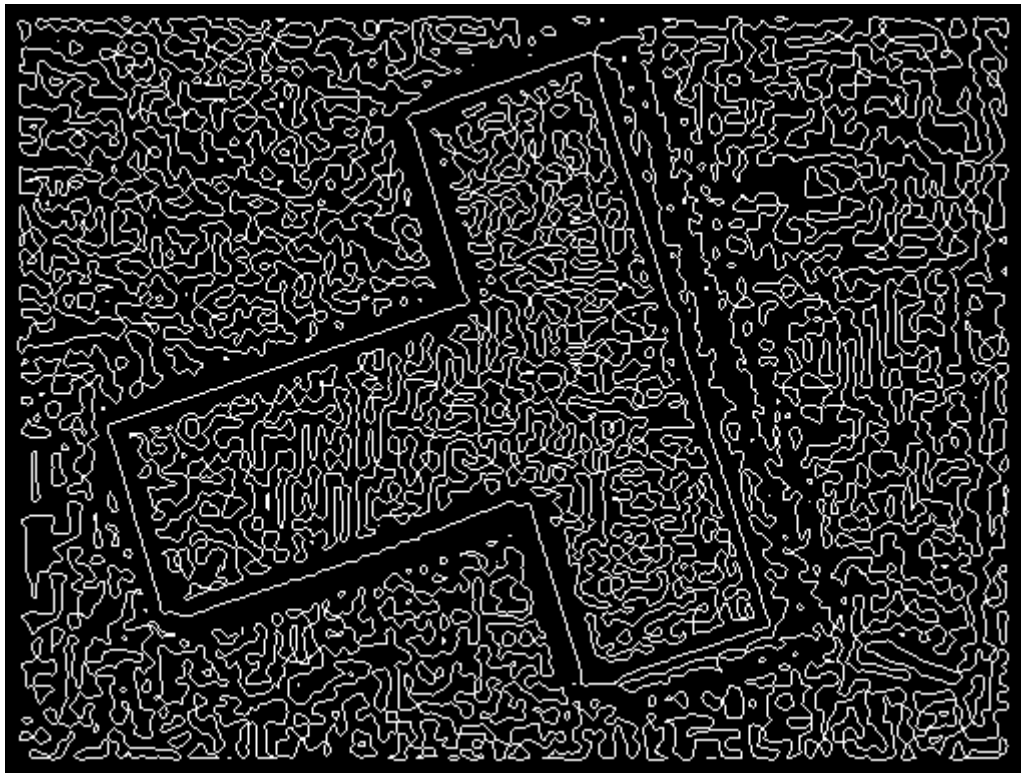




Edge Detection

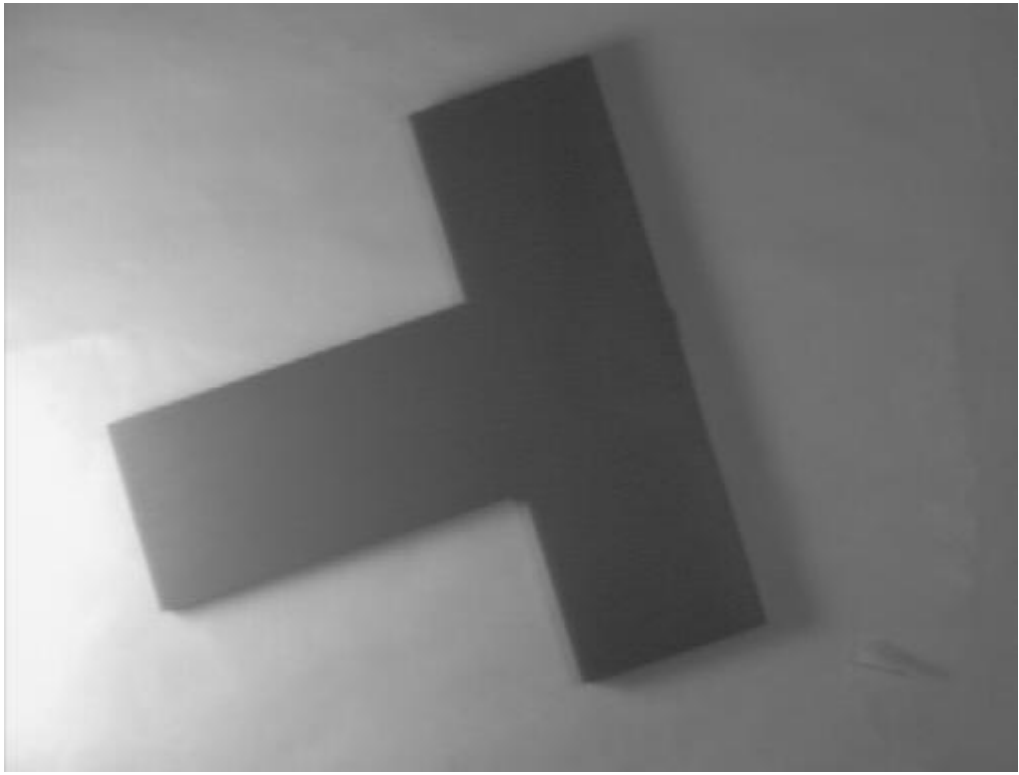
- Marr-Hildreth algorithm for edge detection
  - Gaussian variance > 0
  - Threshold = 0

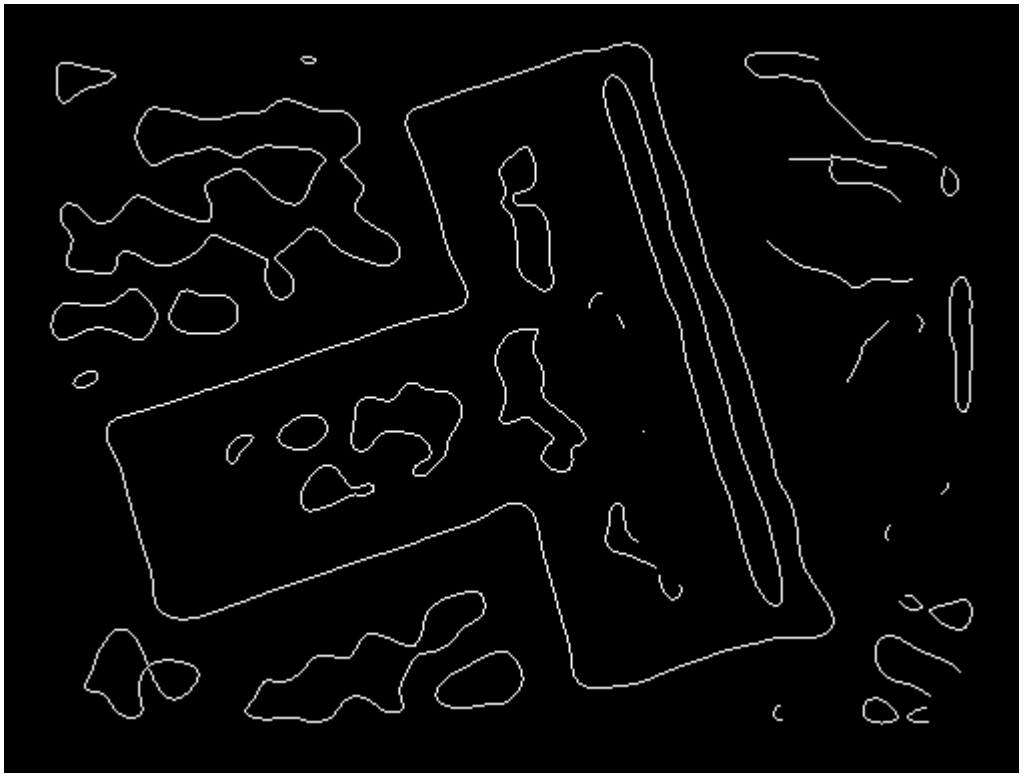




Edge Detection

- Marr-Hildreth algorithm for edge detection
  - Gaussian variance > 0
  - Threshold > 0

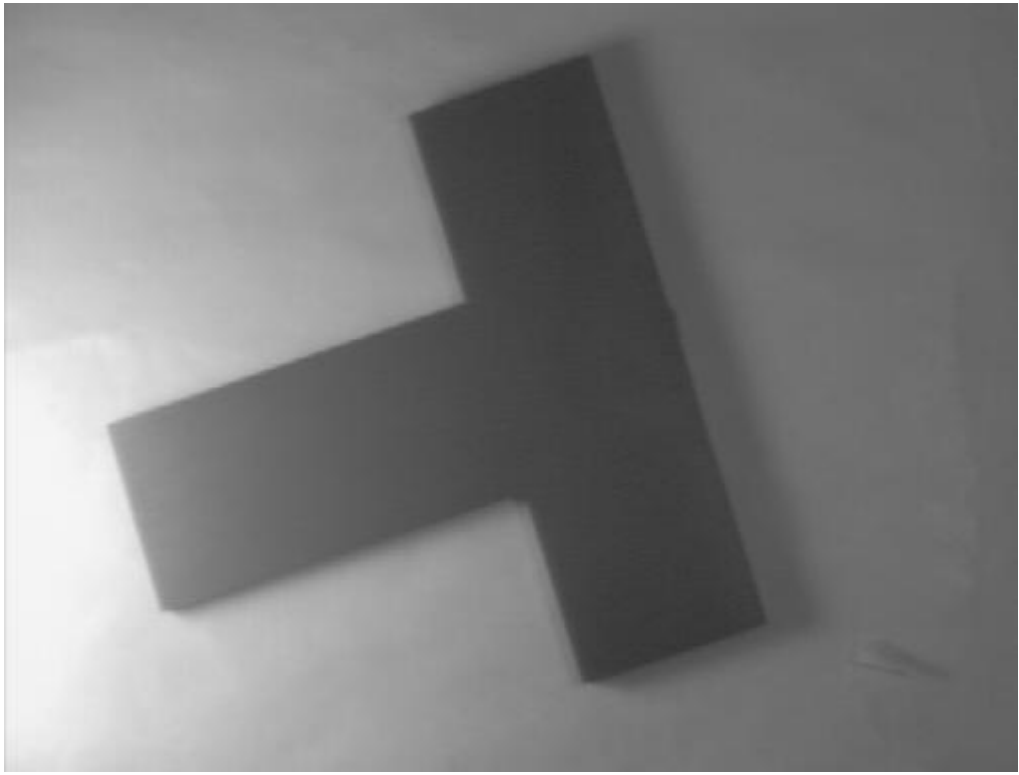


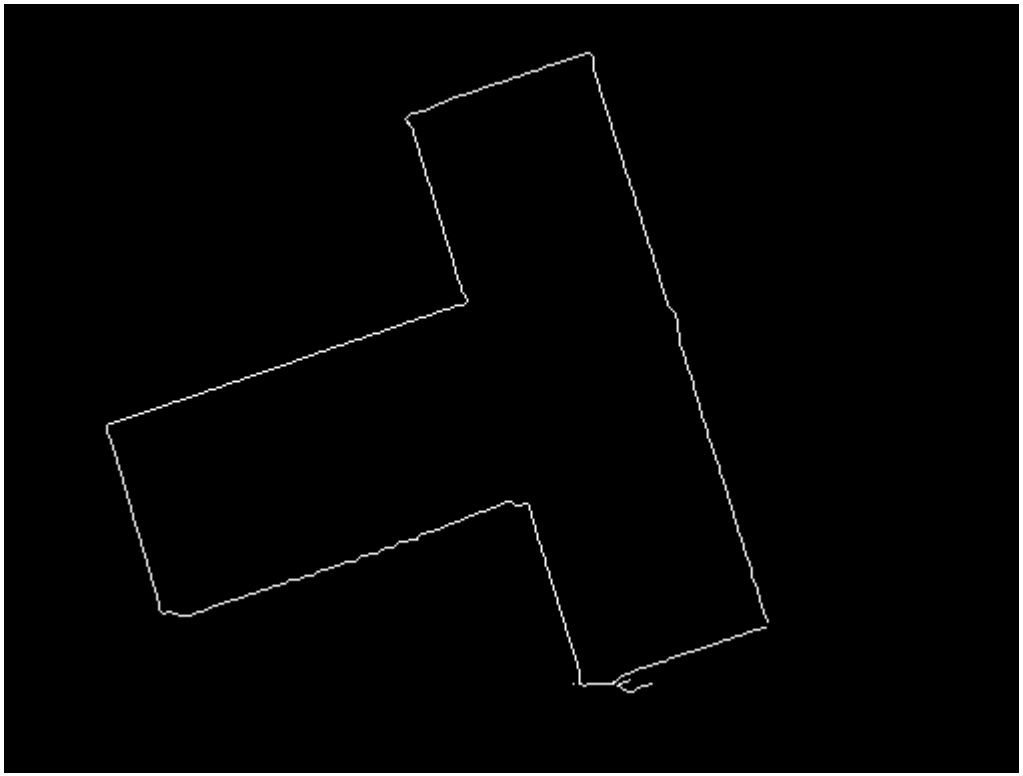


Parameter Tuning :
Sub-optimal

Edge Detection

- Marr-Hildreth algorithm for edge detection
  - Gaussian variance > 0
  - Threshold > 0

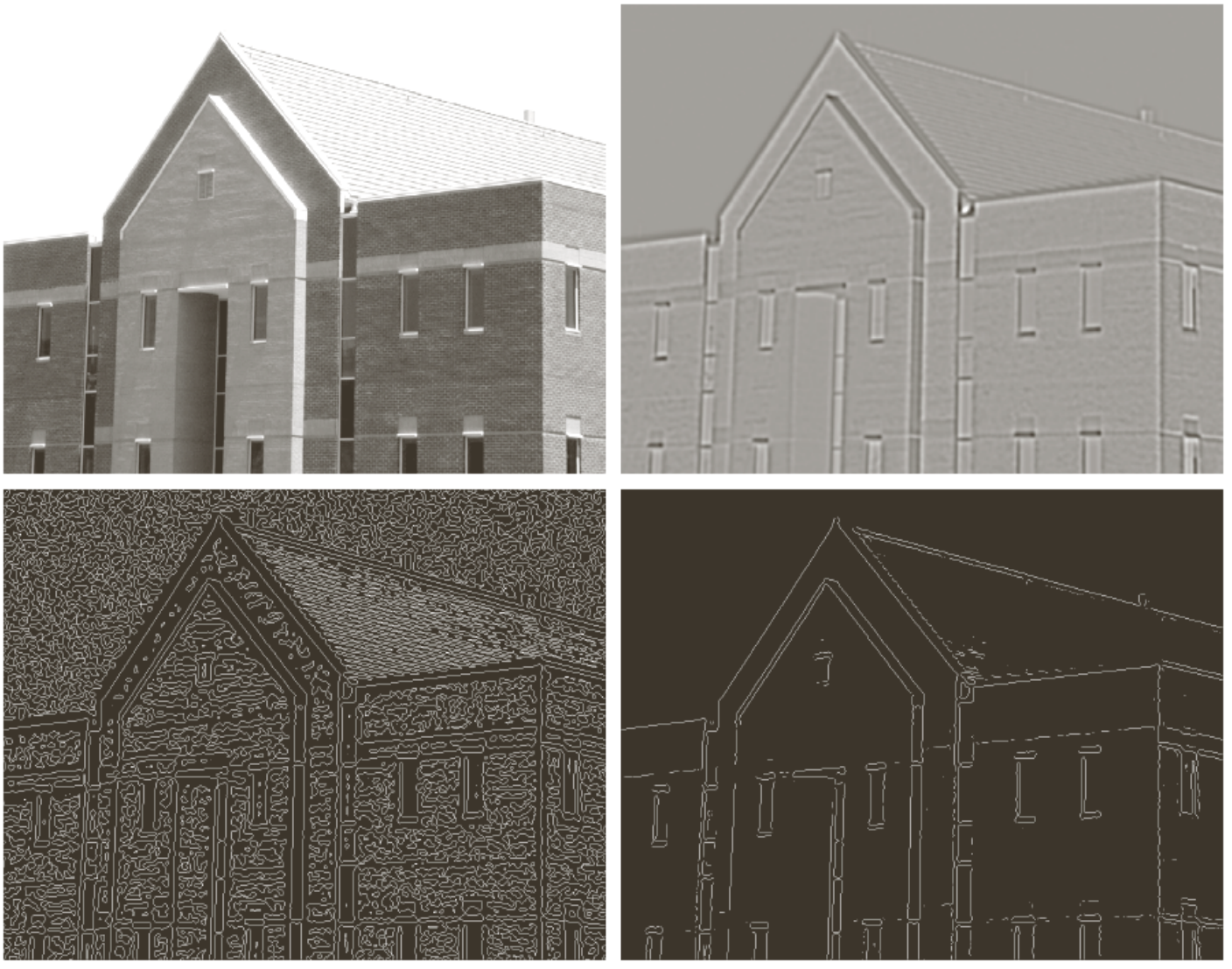




Parameter Tuning : Good

Edge Detection

- Marr-Hildreth algorithm for edge detection (1980)
  - Gives 1-pixel thick edges
  - Top row: image, LoG
  - Bottom row: detected edges with two different thresholds



Edge Detection

- David Marr
  - Neuroscientist, psychologist
  - CS work in artificial intelligence, image processing
  - After PhD in physiology, he got interested in visual processing
  - Marr Prize
    - One of the most prestigious awards in computer vision

Edge Detection

- Canny edge detection
  - John Canny. 1986.

    (1/4) Gaussian smoothing to reduce noise

    (2/4) Gradient computation at each pixel $\mathbf{G} = \sqrt{\mathbf{G}_x{}^2 + \mathbf{G}_y{}^2}$
    - Positive and negative directions equivalent $\Theta = \operatorname{atan2}(\mathbf{G}_y, \mathbf{G}_x)$

    (3/4) Edge **thinning** by **non-maximum suppression**
    - See next ...

    (4/4) Edge **tracing** via **hysteresis thresholding**
    - See next ...

Edge Detection

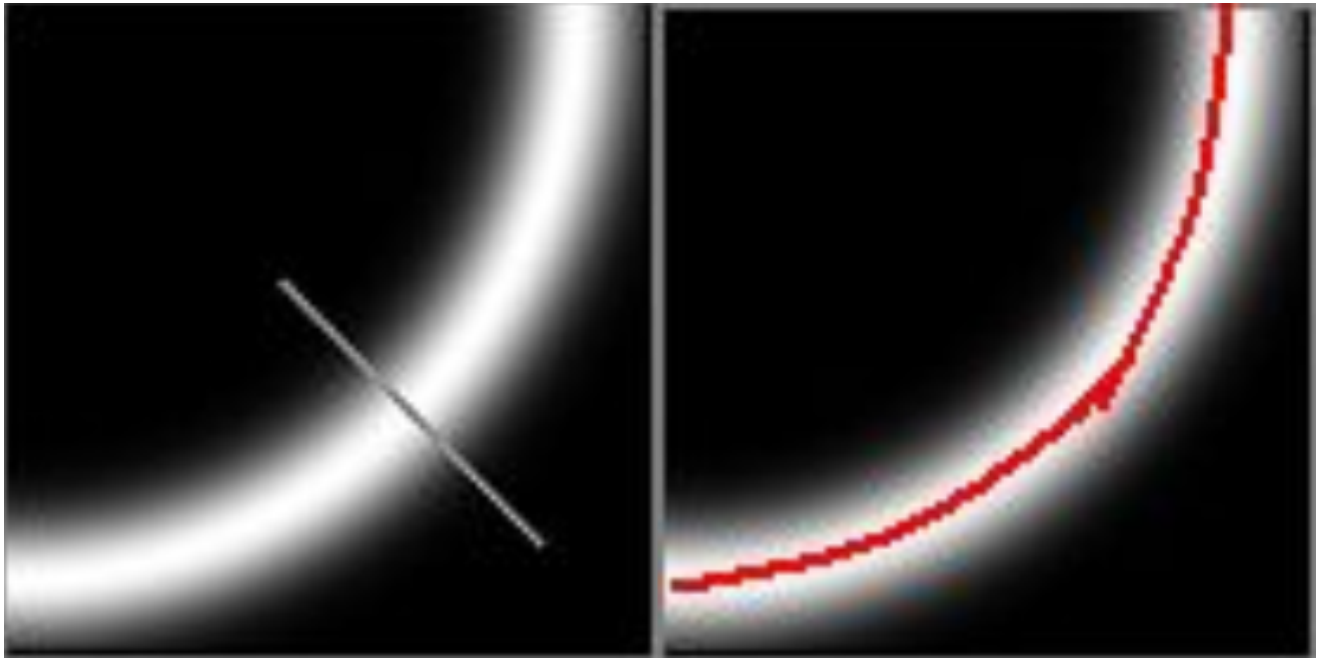
- Canny edge detection

  (1) Gaussian smoothing to reduce noise

  (2) Gradient computation at each pixel

  (3) Edge **thinning** by **non-maximum suppression**

  - Mark pixels where gradient magnitude is >= that of neighbors along the gradient direction (across the edge)
  - This may require interpolation



Edge Detection

- Canny edge detection
  - (1) Gaussian smoothing to reduce noise
  - (2) Gradient computation at each pixel
  - (3) Edge thinning by non-maximum suppression
  - (4) Edge **tracing** via **hysteresis thresholding**
    - Difficult to define one threshold on gradient magnitude to detect pixels on an edge
    - Assume: edges are continuous curves
    - Define 2 thresholds: thresholdHigh and thresholdLow
    - First, select pixels where gradient magnitude > thresholdHigh
    - Then, find connected path starting from selected pixels where all pixels have gradient magnitude > thresholdLow

Edge Detection

- Canny edge detection
  - Hysteresis thresholding makes following assumptions

    (1) PixelValue >= thresholdHigh → pixels inside object

    (2) PixelValue <= thresholdLow → pixels outside object

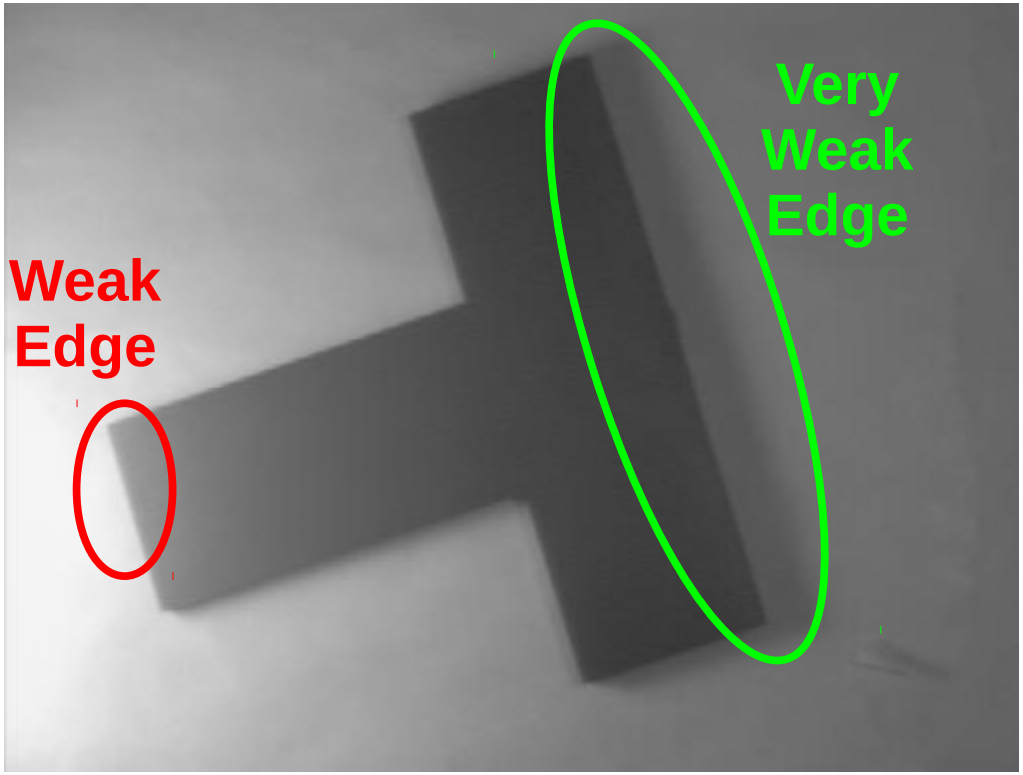
    (3) thresholdLow < PixelValue < thresholdHigh → pixel inside object if connected path exists to *object pixel* such that all pixels in path have values > thresholdLow

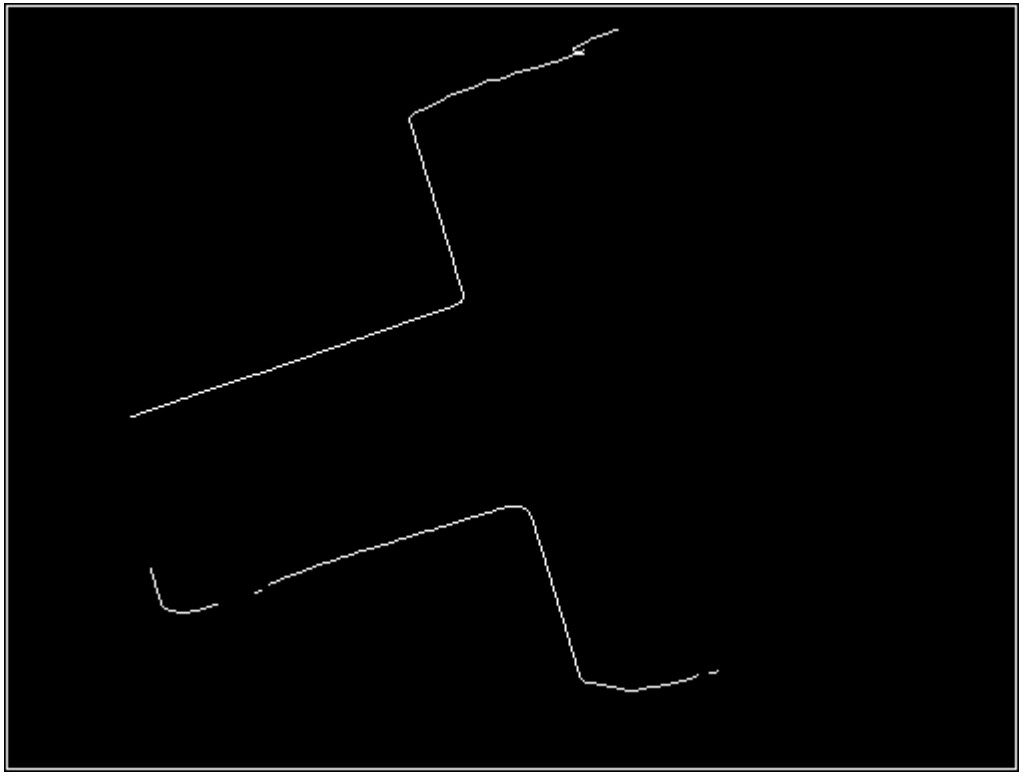
Edge Detection

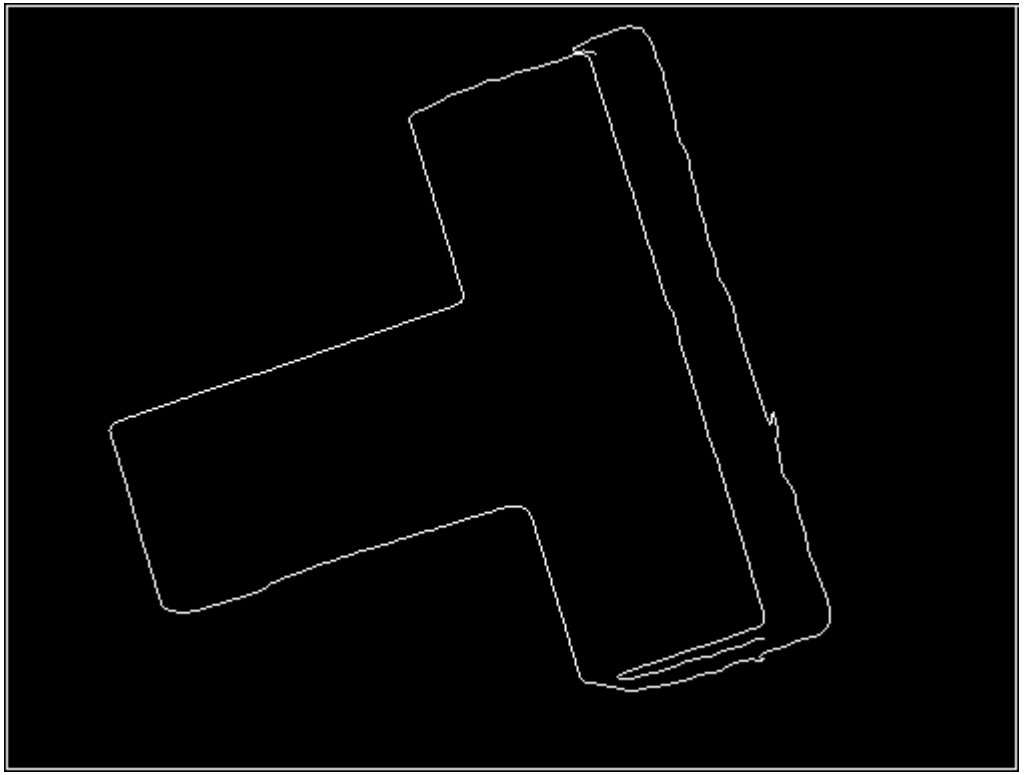
- Canny edge detection

  (1), (2), (3) ...

  (4) Simple thresholding vs. Hysteresis thresholding





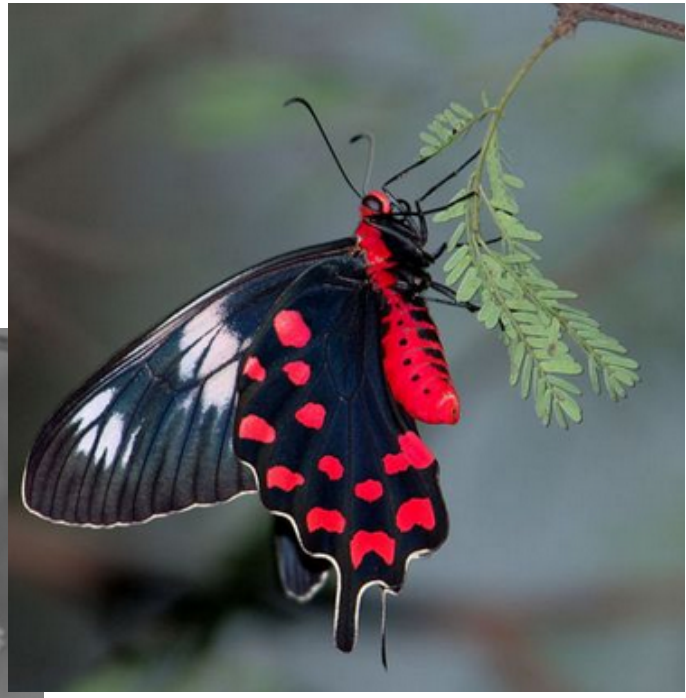


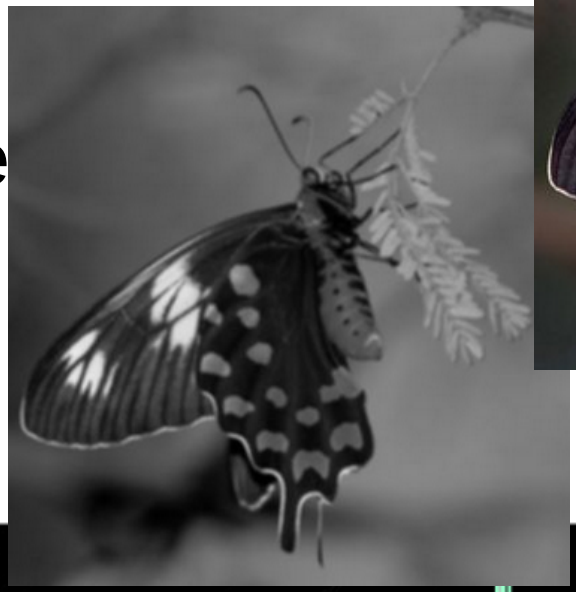
Edge Detection

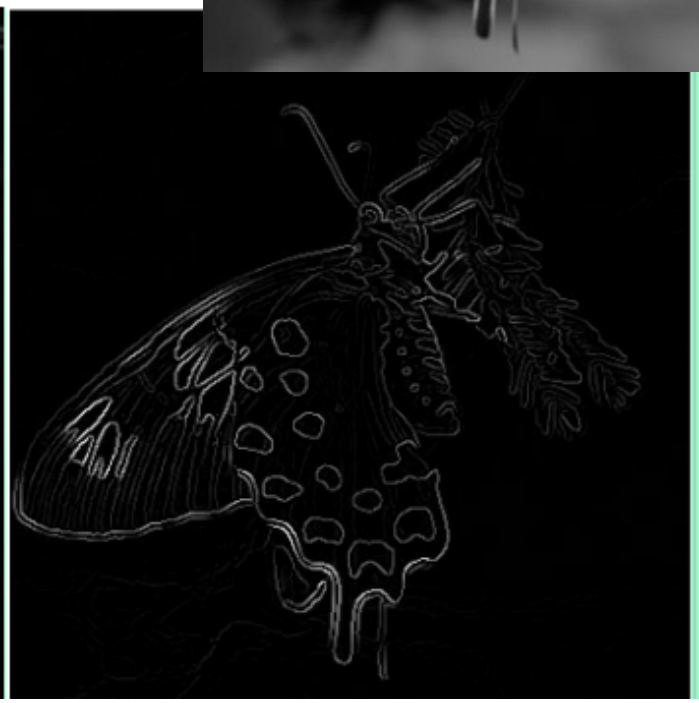
- Canny edge detection
  - Parameters
    - Variance of Gaussian used for smoothing
    - High threshold on gradient magnitude
    - Low threshold on gradient magnitude

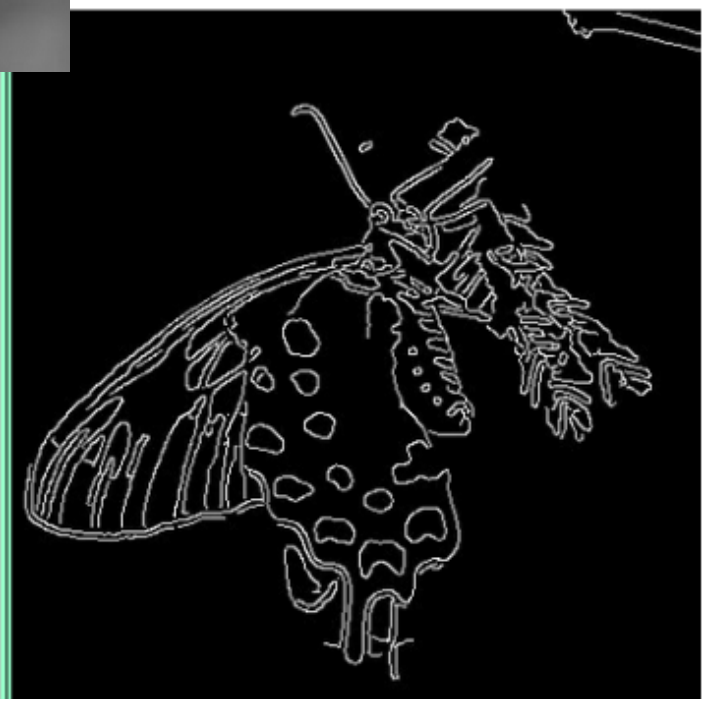
Edge Detection

- Canny edge detection
  - Smoothed gradient
  - Non-maximum suppre
  - Hysteresis threshold









www-scf.usc.edu/~boqinggo/Canny.htm

Edge Detection

- Canny edge detection
  - Smoothed gradient
  - Non-maximum suppression
  - Hysteresis threshold

Edge Detection

- Canny edge detection
  - Smoothed gradient
  - Non-maximum suppression
  - Hysteresis threshold

Edge Detection

- Canny edge detection
  - Smoothed gradient
  - Non-maximum suppression
  - Hysteresis threshold

Canny Edge Detection.

Sigma= 1, Low= 0.4, High= 0.8

Sigma= 2, Low= 0.4, High= 0.8

Sigma= 1, Low= 0.3, High= 0.7

Edge Detection

- **Corner detection**
  - **Moravec corner detection** algorithm. 1980.
  - Consider a pixel “p” as a corner of an object when:
    - Low similarity between:
      (1) a patch around “p” and
      (2) another patch around any neighbor “q” of “p”
    - Threshold on similarity
    - Similarity captured via
      (1) sum of squared differences → dissimilarity OR
      (2) (normalized) cross correlation → similarity

Edge Detection

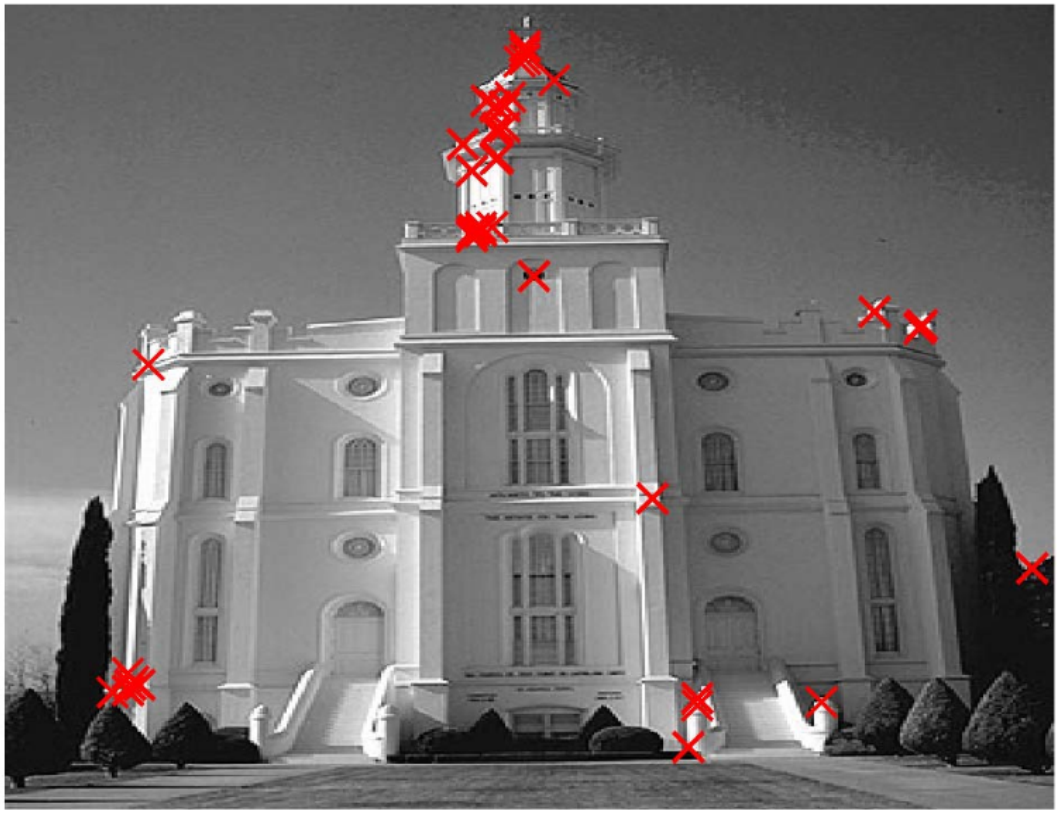
- Moravec corner detection
  - Step 1/3 → At each pixel, compute lowest similarity

Edge Detection

- Moravec corner detection
  - Step 2/3 → Non-maximal suppression
    - Ignore pixels having any 8-neighbor with lower similarity
    - Set their values to white

Edge Detection

- Moravec corner detection
  - Step 3/3 → Threshold



Edge Detection

- Limitations of Moravec corner detection
  - Only 8 shifts of patches are considered at 45-degree angle increments
    - Left, right
    - Up, down
    - NW, NE, SW, SE
  - Patches are rectangular (not isotropic)

Edge Detection

- **Harris corner detection**. 1988.
  - Consider
    “all” (infinitely many)
    small (infinitesimally small) shifts
  - Use Taylor-series expansion of image function
  - Dissimilarity measured via sum of squared differences
  - **Dissimilarity** between (i) patch over pixels (u,v) and (ii) another patch over pixels shifted by (x,y)

$$S(x,y) = \sum_u \sum_v w(u,v)\ (I(u+x, v+y) - I(u,v))^2$$

where w(u,v) = weights over pixels within patch

- Can be used to make patch isotropic

Edge Detection

- Harris corner detection
  - Taylor-series expansion of image function :

$$I(u+x, v+y) \approx I(u,v) + I_x(u,v)x + I_y(u,v)y$$

  - Thus, approximated patch dissimilarity is :

$$S(x,y) \approx \sum_u \sum_v w(u,v)\ \left(I_x(u,v)x + I_y(u,v)y\right)^2$$

  - This is a quadratic form over shift variables x, y

Edge Detection

- Harris corner detection
  - Patch dissimilarity at any shift (x,y) is :

$$S(x,y) \approx \begin{pmatrix} x & y \end{pmatrix} A \begin{pmatrix} x \\ y \end{pmatrix}$$

where “A” = **structure tensor**

$$A = \sum_u \sum_v w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}_{(u,v)}$$

  - “A” gives averaged local differential structure
  - “A” is symmetric, positive definite
    - Because each summand matrix $\begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} I_x \\ I_y \end{bmatrix} \cdot \begin{bmatrix} I_x & I_y \end{bmatrix}$

Edge Detec $S(x,y) \approx (x \quad y) A \begin{pmatrix} x \\ y \end{pmatrix}$

- Harris corner detection
  - A has an eigen decomposition : A = Q D Q^{-1} = V D V'
    - Eigenvalues → non-negative
    - Eigenvectors → orthogonal
  - For a patch's center pixel to be at an object's corner, **dissimilarity** S(x,y) should be **large** for **all shifts (x,y)**
  - Thus, "A" must have **2 large** (positive) eigenvalues
    - Why ?
    - If A has an eigenvalue = 0,
      then for shift (x,y) = corresponding eigenvector "v",
      v' A v = 0

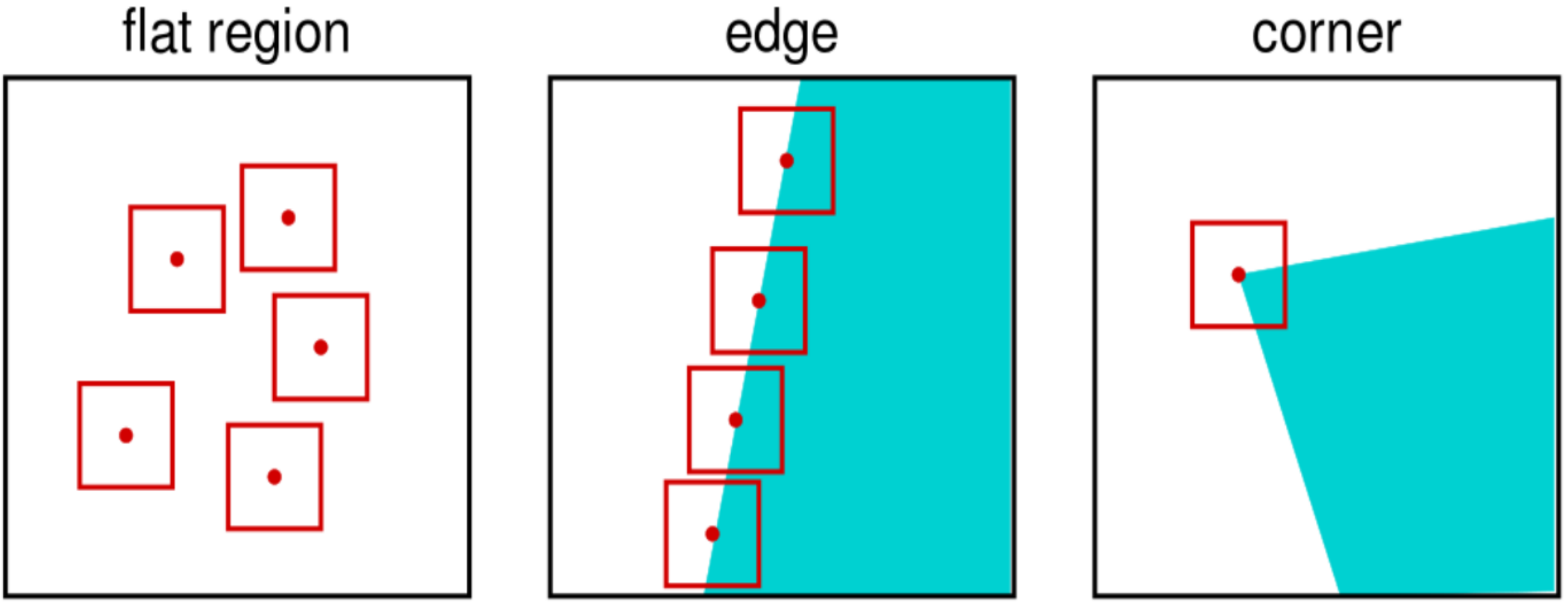
Edge Detection

- Harris corner detection

  (1) Constant region : both eigenvalues = 0

  (2) Edge : one eigenvalue large, other → 0

  (3) Corner : both eigenvalues large



Edge Detection

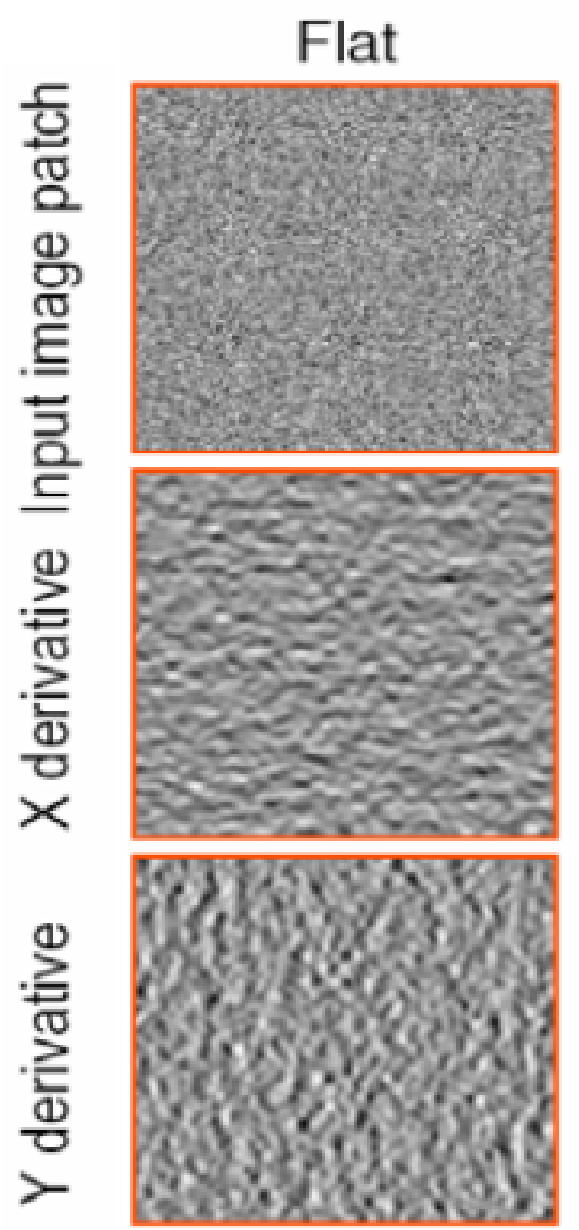
- Harris corner detection
  - Corner-ness measure
    - Designed to avoid eigen decomposition
  - $C = \text{Determinant}(A) - k\,(\text{Trace}(A))^2$
    - Determinant (A) = product of eigenvalues
    - Trace (A) = sum of eigenvalues = sum of diagonal elements
    - k = empirically-tuned constant ($0 < k < 0.25$) such that
      - If both eigenvalues $\rightarrow 0$, then $C \rightarrow 0$
      - If one eigenvalue $\rightarrow 0$, then $C < 0$
      - If both eigenvalues large, then $C > 0$
    - Rotation invariant
      - Because eigenvalues don't change when image gets rotated,
        i.e., gradient vectors get rotated,
        i.e., coordinate frame <Ix,Iy> gets rotated

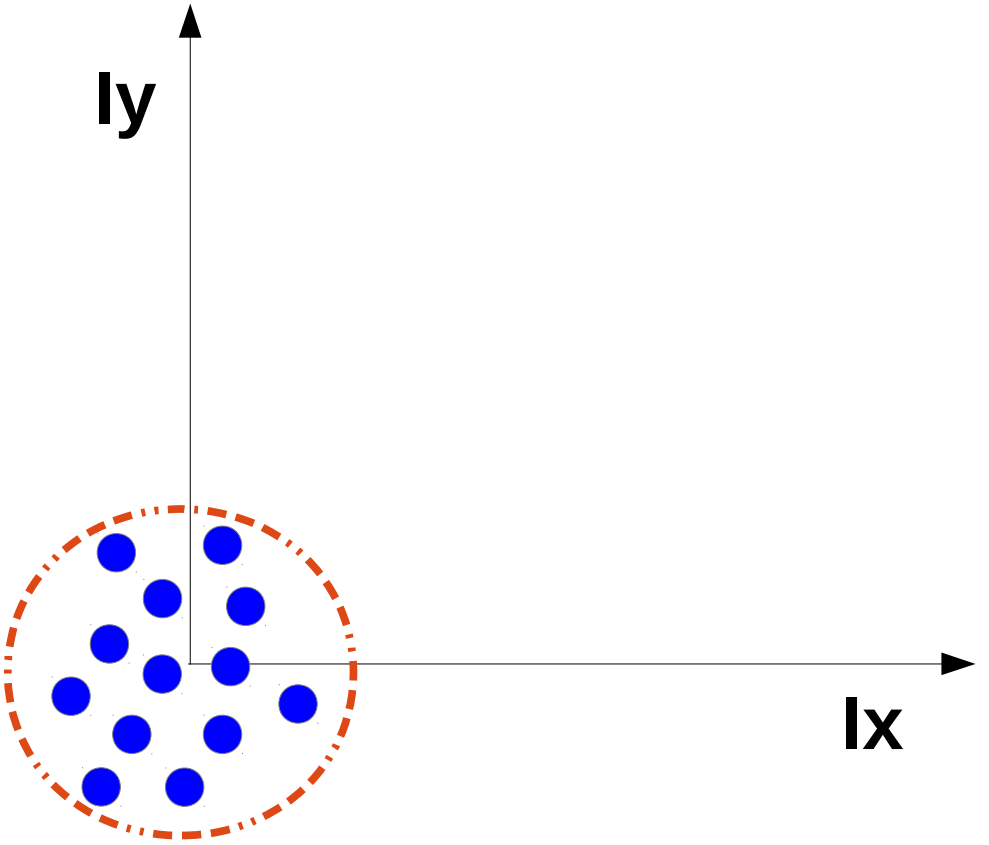
Edge Detection

- Harris corner detection
  - Another interpretation of “A”
  - Ignoring weights w(u,v),
    “A” = 2nd-moment matrix of gradient vectors over patch
    - Not covariance matrix, because no mean subtraction

Edge Detection

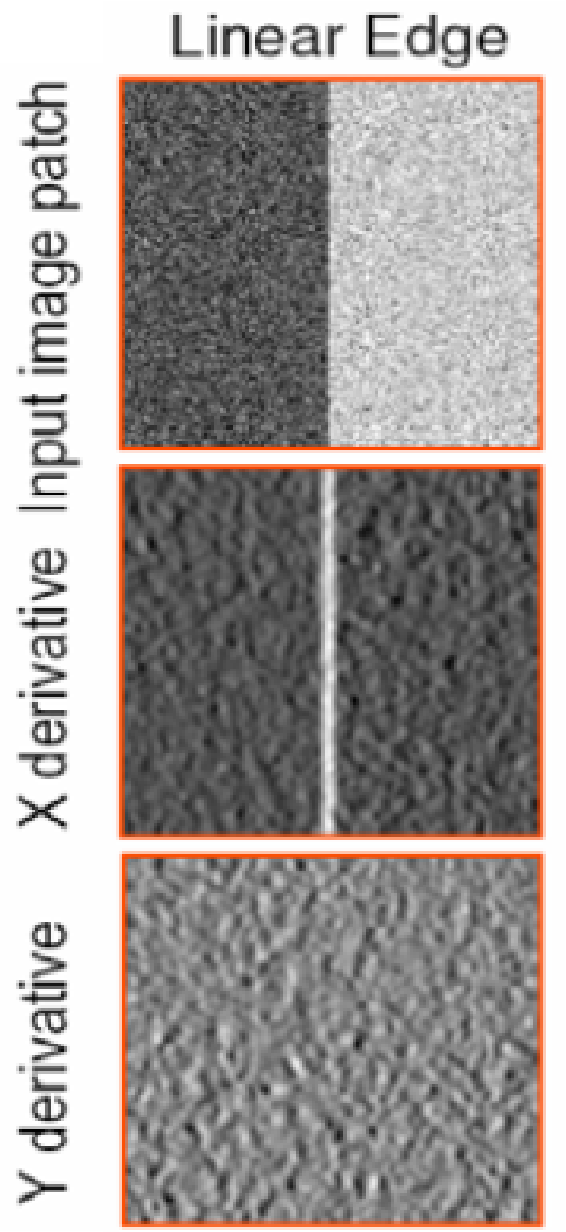
- Harris corner detection

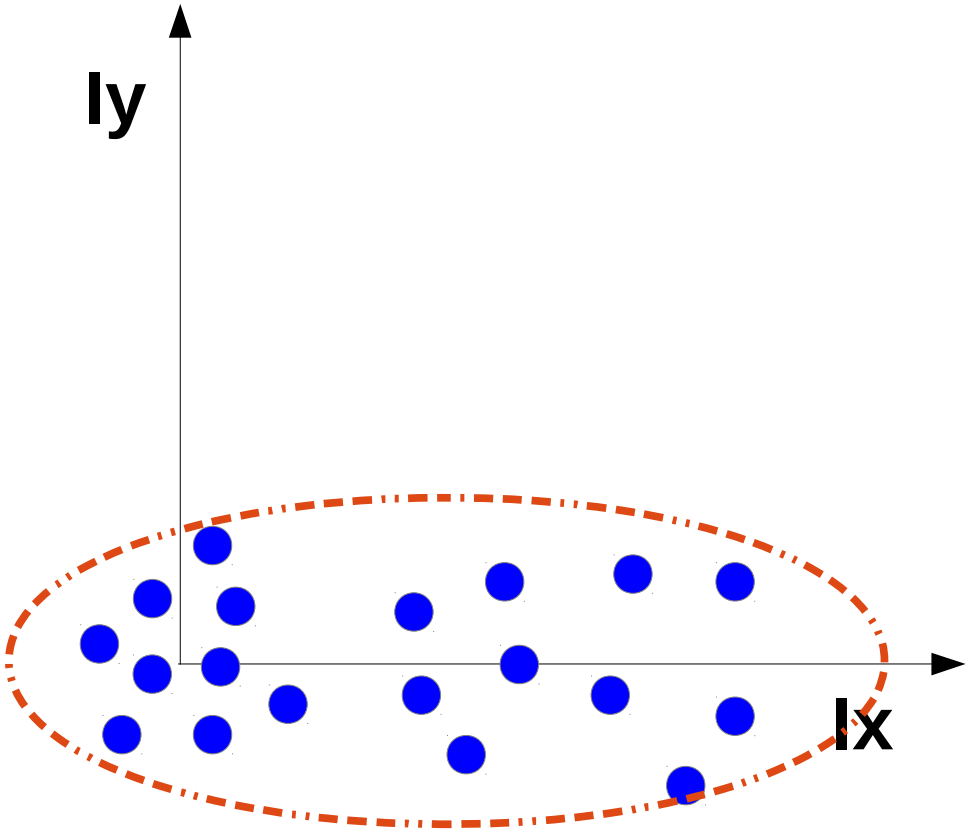




Edge Detection

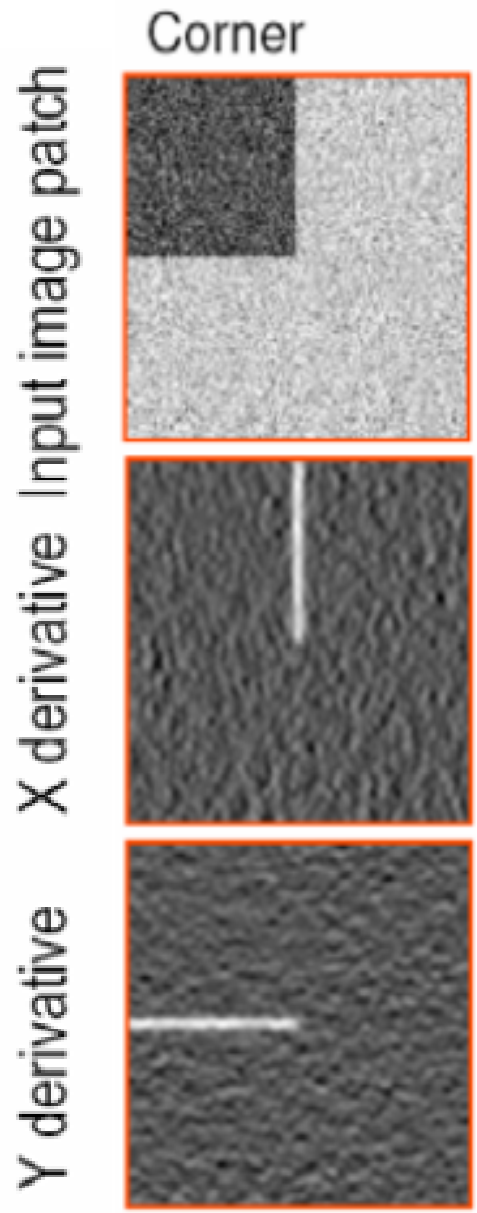
- Harris corner detection

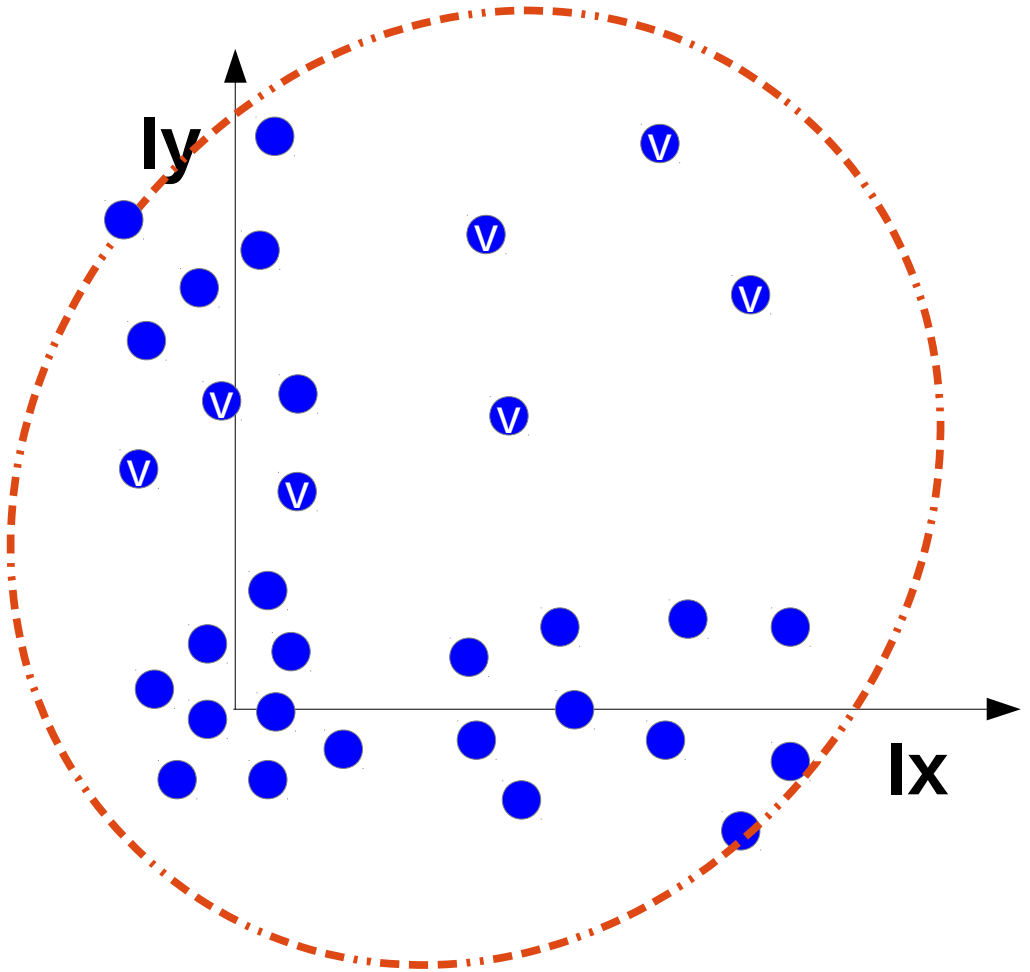




Edge Detection

- Harris corner detection





Edge Detection

- Harris corner detection

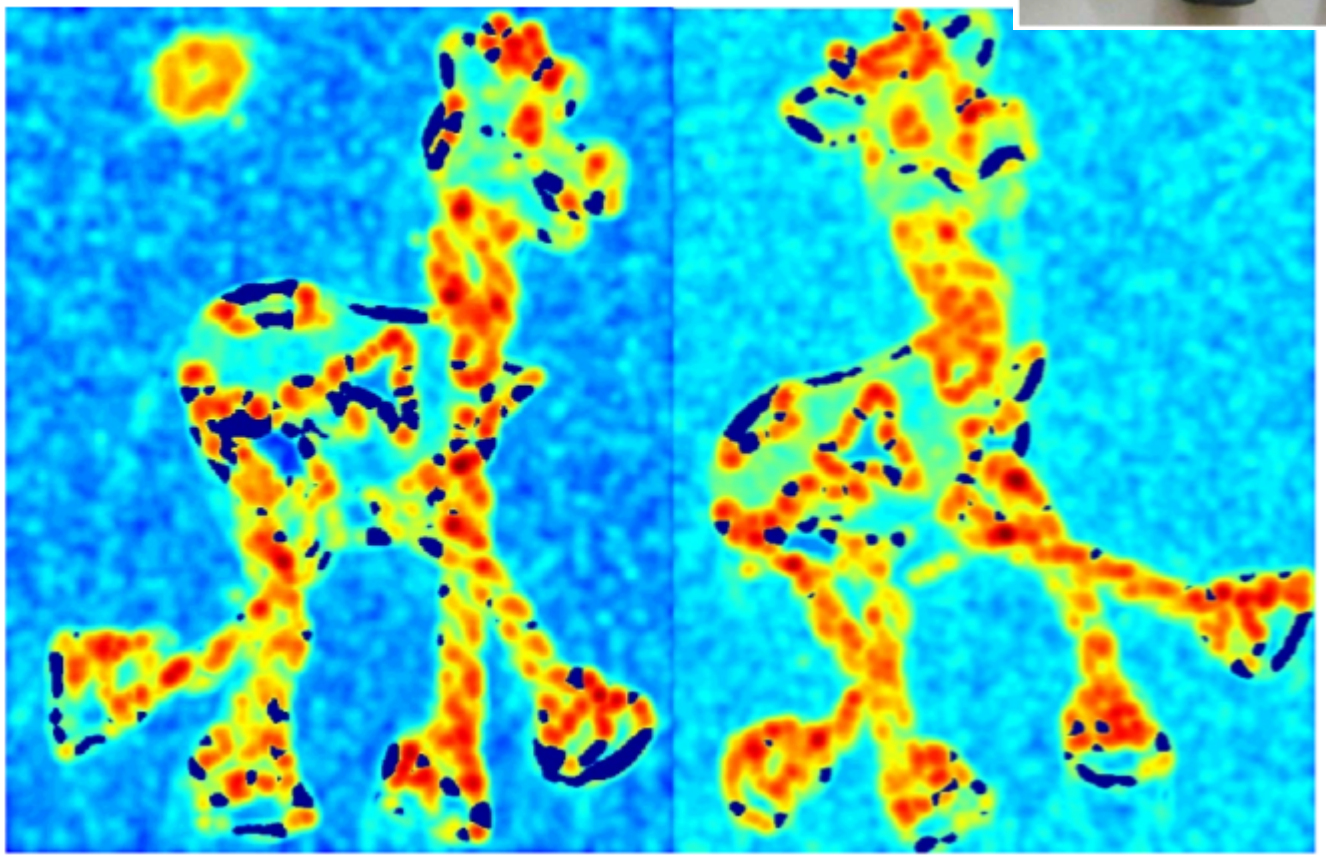
  (1) Compute cornerness measure

  (2) Non-maximum suppression (in neighborhoods)

www.csie.ntu.edu.tw/~cyy/courses/vfx/07spring/lectures/handouts/lec04_feature_4up.pdf

Edge Detection

- Harris corner detection
  - (1) Corner-ness measure



Edge D

- Harris corner detection
  - (2) Non-maximum suppression