CSCI435/CSCI935
Computer Vision: Algorithms & Systems

Edge Detection

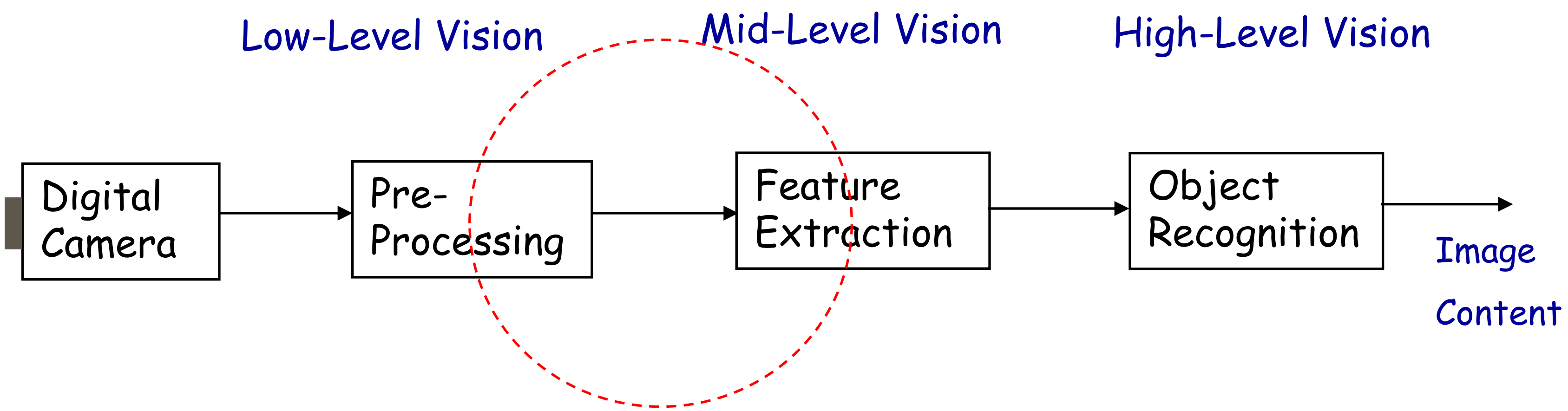
Lecturer: Prof Lei Wang
Room 3.219
Email: leiw@uow.edu.au
Web: https://scholars.uow.edu.au/lei-wang

Image Quality & Enhancement (review)

- Sharpness and Aliasing
  - Sampling, focusing, motion, exposure
- Noise
  - Sources, Statistic model
- Enhancement
  - Histogram equalization
  - Filtering (low-pass, high-pass, band-pass)
    - Spatial domain – convolution
    - Frequency/Spectrum domain – filtering (multiplication)
    - Gaussian filters

Machine Vision Concept (review)

- Machine Vision is a multistage process where each previous stage affects performance of all following stages



Visual Image Analysis

We isolate objects by analysing sudden variations of brightness or colour

- Image content is analysed through analysis of individual objects, their composition and interaction
- To analyse objects, the objects must be separated from the background

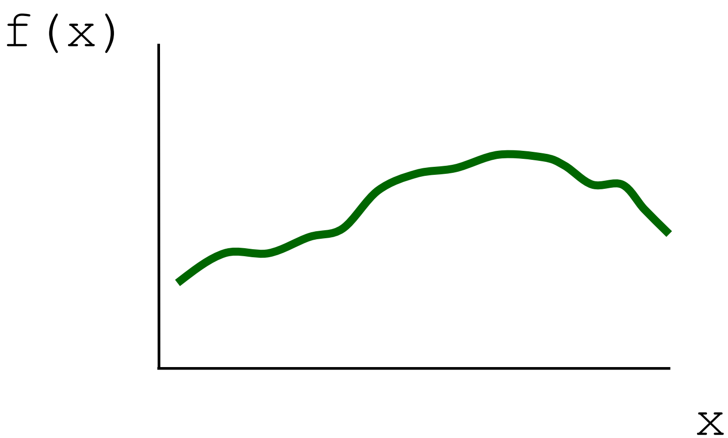
Separation of Objects

- Object separation is a complex process, but it is based on two basic principles:
  - detection of discontinuities (luminance or colour) ← The scope of this lecture
  - identification of similarity

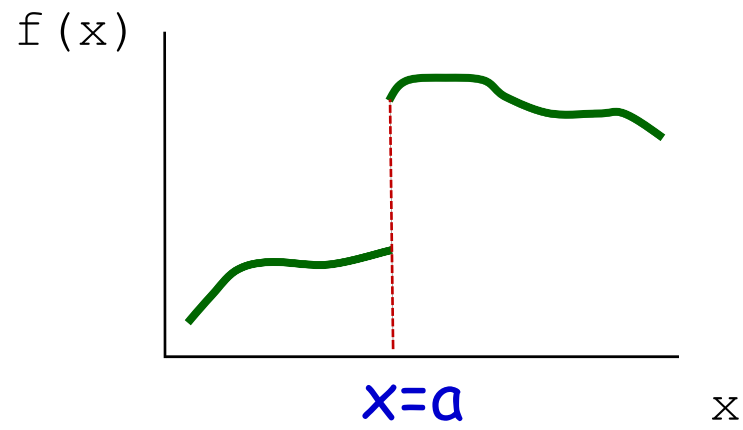
Continuous Functions

- In mathematics, function $f(x)$ is said to be continuous at the point c, if for any small ε there is Δ, that

$$c-\varepsilon < x < c+\varepsilon \Rightarrow f(c)-\Delta < f(x) < f(c)+\Delta$$



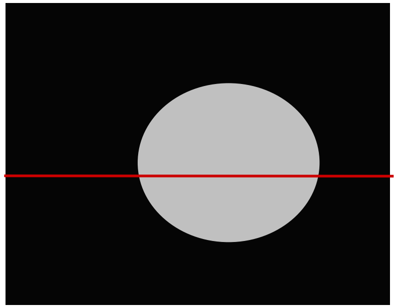
Continuous Function



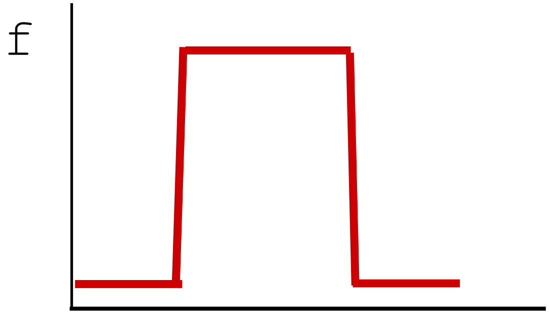
Discontinuity

Edges

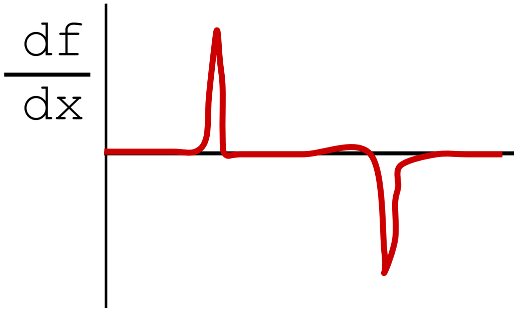
- Edges are locations in the image where image luminance has discontinuity



Image



Luminance variation along a image row



- Areas with sudden change of luminance have large magnitude of the gradient in horizontal, or vertical direction (or both)

$|g_x| > \varepsilon, \quad |g_y| > \varepsilon \quad$ where $\quad g_x = \frac{df}{dx} \quad g_y = \frac{df}{dy}$

Gradient of image *f(x,y)*

$$\nabla \mathbf{f} = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

Gradient of at (x,y)

$$\nabla f = mag(\nabla \mathbf{f}) = \left[g_x^2 + g_y^2 \right]^{1/2}$$

Magnitude

$$\approx |g_x| + |g_y|$$

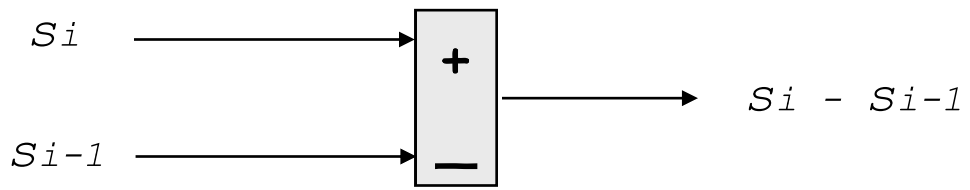
$$\alpha(x, y) = \tan^{-1}\left(\frac{g_y}{g_x} \right)$$

Direction of gradient

The direction of an edge at (x,y) is perpendicular to the direction of gradient at that point

Detection of Discountinuities (1D)

- Areas of discontinuities can be detected by measuring gradients and comparing it against a predefined threshold value ε
- Gradient is essentially a derivative calculated in one of the directions
- As images are digitised sequences the derivative can be calculated by the commonly used digital differentiator



An example of the 1D differentiator

Detection of Discountinuities (2D)

- As images are 2D sequences, the gradients can be measured by convoluting the image with 2D differentiators

-1	**0**	**1**
-1	0	1
-1	0	1

g_x

3x3 Prewitt kernels

-1	**-1**	**-1**
0	0	0
1	1	1

g_y

- Each gradient operator produces output that at each location is proportional to the derivative at that location
- Diagonal gradients affect both operators, but to a smaller degree
- Outputs from both operators are combined into a single magnitude

$$g = |g_x| + |g_y|$$

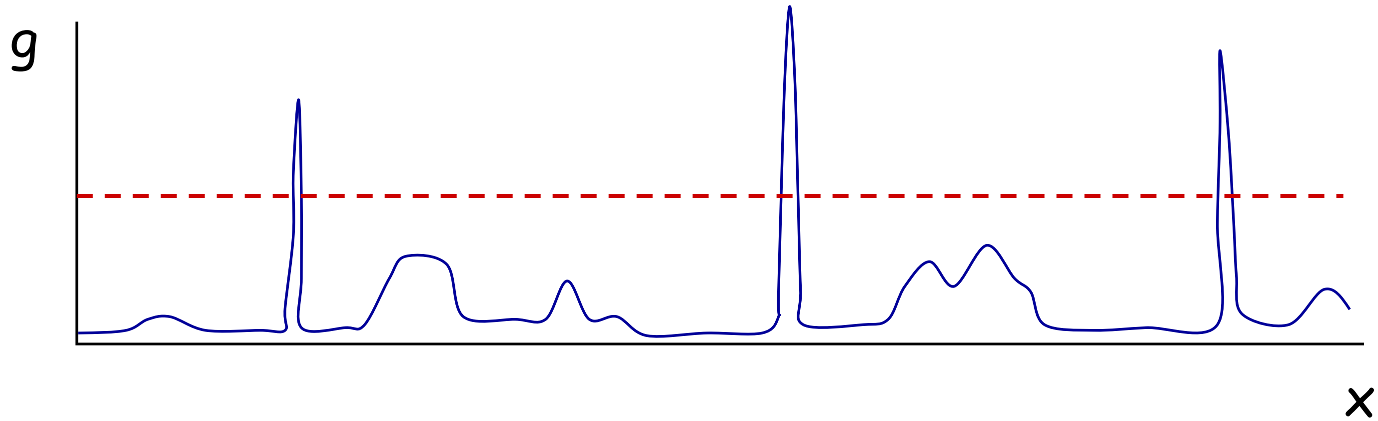
Example

Luminance component of a colour image

Magnitude of the gradient

Note on gradient magnitude

- The magnitude has largest values at object edges
- Areas with uniform luminance (regardless its absolute value) result in very low values of the magnitude
- Luminance variation inside objects can also produce large magnitudes which can be mistaken for object edges
- Magnitudes created by object luminance variation can be discarded by properly selected thresholds



Example

Magnitude of the gradient

Threshold = 200

Influence of Noise

- White noise causes random sudden variation of image luminance in smooth areas producing false dots and disconnected lines

If images are affected by noise, it should be suppressed before edge detection is carried out

Influence of Noise

- Some operators with larger kernels combine differentiation with smoothing to reduce influence of noise

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

5x5 Prewitt kernel

Other Operators

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

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

3x3 Sobel kernels

- Some research results suggest using diagonal edge detection kernels together with the vertical and horizontal ones

0	1	1
-1	0	1
-1	-1	0

Prewitt

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

Sobel

-1	-1	0
-1	0	1
0	1	1

Prewitt

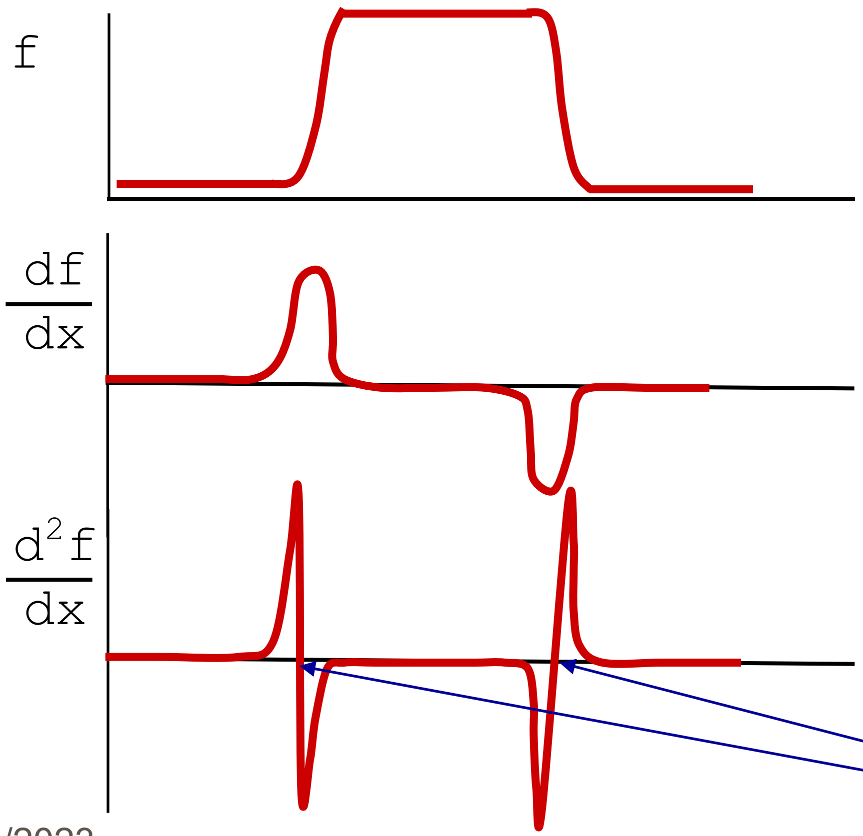
-2	-1	0
-1	0	1
0	1	2

Sobel

Zero-Crossing & Canny Edge Detection

Higher Order Derivatives

- Second order derivatives for edge detection



The second derivative has a zero crossing at the location of each edge

Higher Order Derivatives

- The second-order derivative can be approximated by the Laplacian kernel

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

Laplacian of f(x,y)

-1	2	-1
2	-4	2
-1	2	-1

Laplacian kernel

- After Laplacian filtering, zero crossings located between double picks have to be detected.

Note on Laplacian filter

- Other digital approximation of Laplacian kernel

0	−1	0		−1	−1	−1
−1	4	−1		−1	8	−1
0	−1	0		−1	−1	−1

- Laplacian kernel is unacceptably sensitive to noise
- Laplacian kernel is not sensitive to edge directions
  - i.e. it cannot detect edge directions

Laplacian of a Gaussian (LoG)

- Due to its sensitivity to noise, Laplacian is always combined with smoothing as a precursor to find edges via zero-crossings.
- A typical smoothing is Gaussian smoothing with the kernel

$$h(r) = e^{-r^2/2\sigma^2}$$

$$r^2 = x^2 + y^2$$

LoG

- Given image f(x,y)

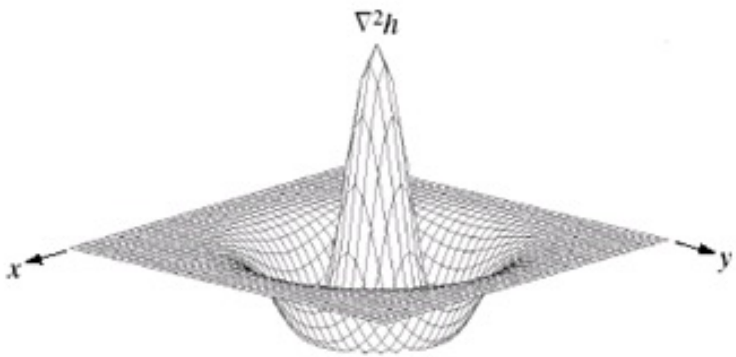
$$LoG(f) = \nabla^2 (f \otimes h)$$

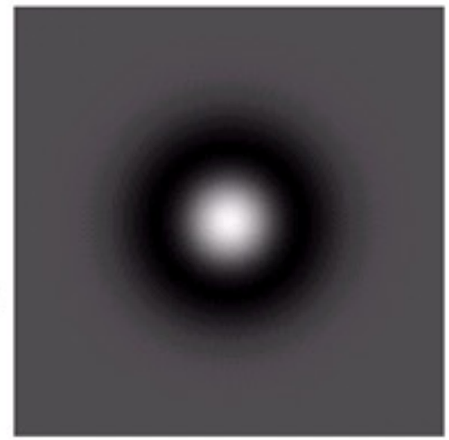
- Because the second derivative is a linear operation

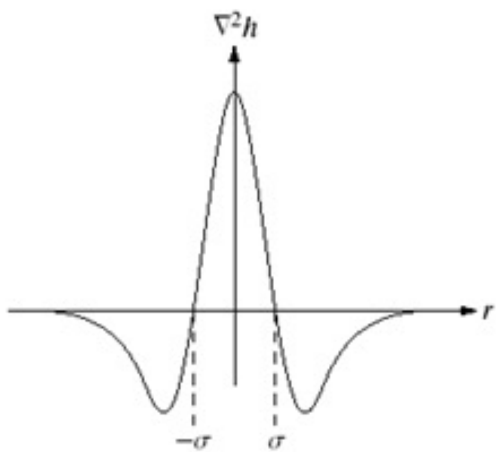
$$LoG(f) = \nabla^2 (f \otimes h) = f \otimes \nabla^2 h$$

$$\nabla^2 h(r) = -\left[\frac{r^2 - \sigma^2}{\sigma^4}\right] e^{-r^2/2\sigma^2}$$

LoG Approximation

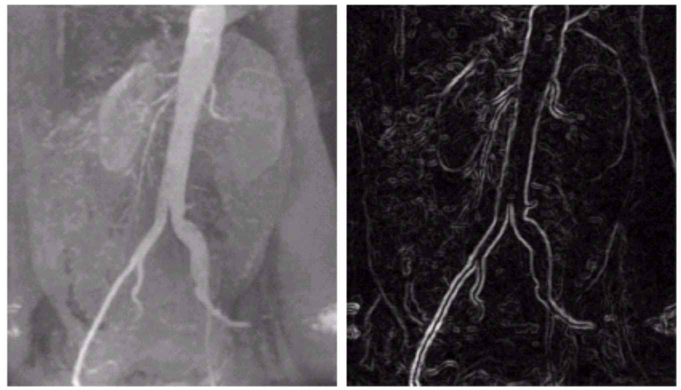


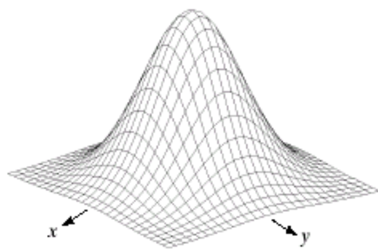




0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

Example

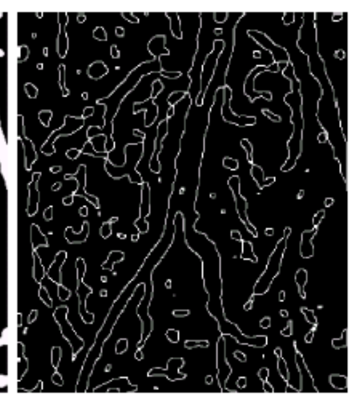




–1	–1	–1
–1	8	–1
–1	–1	–1

Log

Thresholding



Zero-crossing

Difference of Gaussian (DoG)

- Laplacian of Gaussian can be approximated by the difference between two different Gaussians

$$LoG(f,\sigma) \approx \frac{\sigma}{\Delta\sigma}\left(L(x,y,\sigma+\Delta\sigma)-L(x,y,\sigma)\right)$$

$$L(x,y,\sigma) = h(x,y,\sigma) * f(x,y)$$

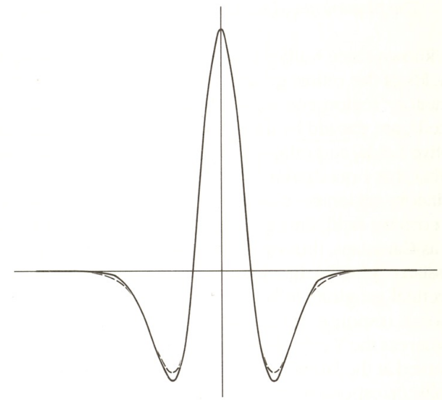


Figure 2–16. The best engineering approximation to $\nabla^2 G$ (shown by the continuous line), obtained by using the difference of two Gaussians (DOG), occurs when the ratio of the inhibitory to excitatory space constraints is about 1:1.6. The DOG is shown here dotted. The two profiles are very similar. (Reprinted by permission from D. Marr and E. Hildreth, "Theory of edge detection, " *Proc. R. Soc. Lond. B 204,* pp. 301–328.)

DoG Edge Detection

(a) $\sigma = 1$

(b) $\sigma = 2$

(b)-(a)

Edge Thinning and Linking

- Edges detected by thresholding the gradient calculated using Sobel or Prewitt operators are often
  - Multiple pixels
  - Broken due to noise

- For many CV tasks, we need to have
  - Thinning - to convert to one pixel edges
  - Tracing and linking

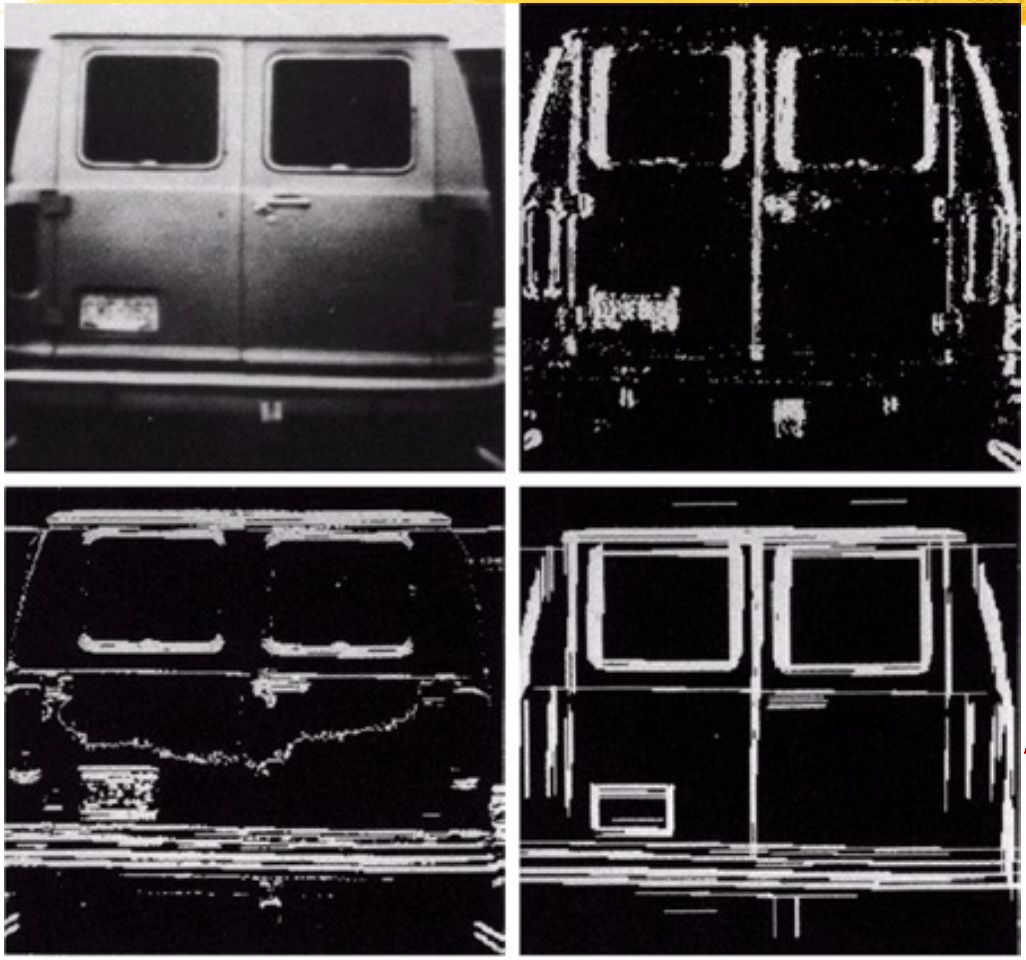
Edge Linking - Local Processing

- For every edge pixels, consider a local neighbourhood, e.g. 3x3 or 5x5
- Two edge pixels (x,y) and (x_0,y_0) should be linked if the following conditions are met

$$\left|\nabla f(x,y) - \nabla f(x_0,y_0)\right| \le E$$

$$\left|\alpha(x,y) - \alpha(x_0,y_0)\right| \le A$$

Example



Linked horizontal and vertical edge using E=25, A=15°

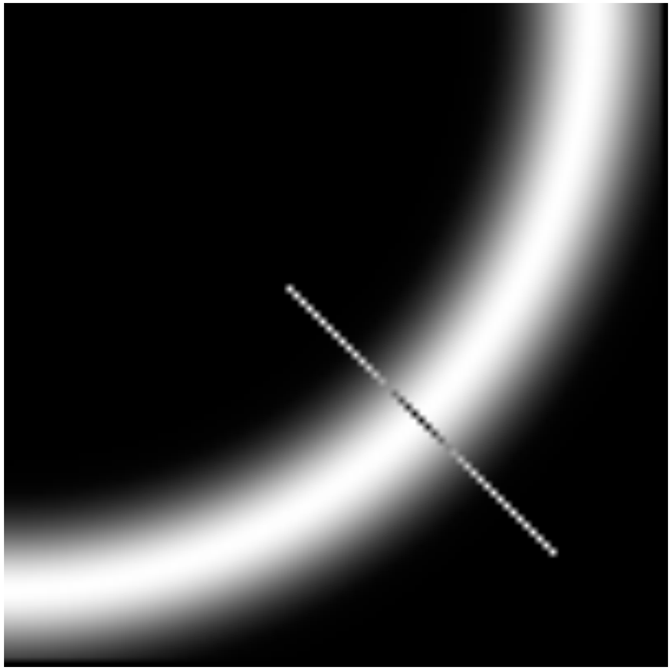
Canny Edge Detector

- Canny edge detector is one of the most commonly used edge detectors
  - Combines edge detection, thinning, tracing and linking
- Reference sources
  - Canny, J., *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679-698, 1986.
  - http://en.wikipedia.org/wiki/Canny_edge_detector

Canny Edge Detection

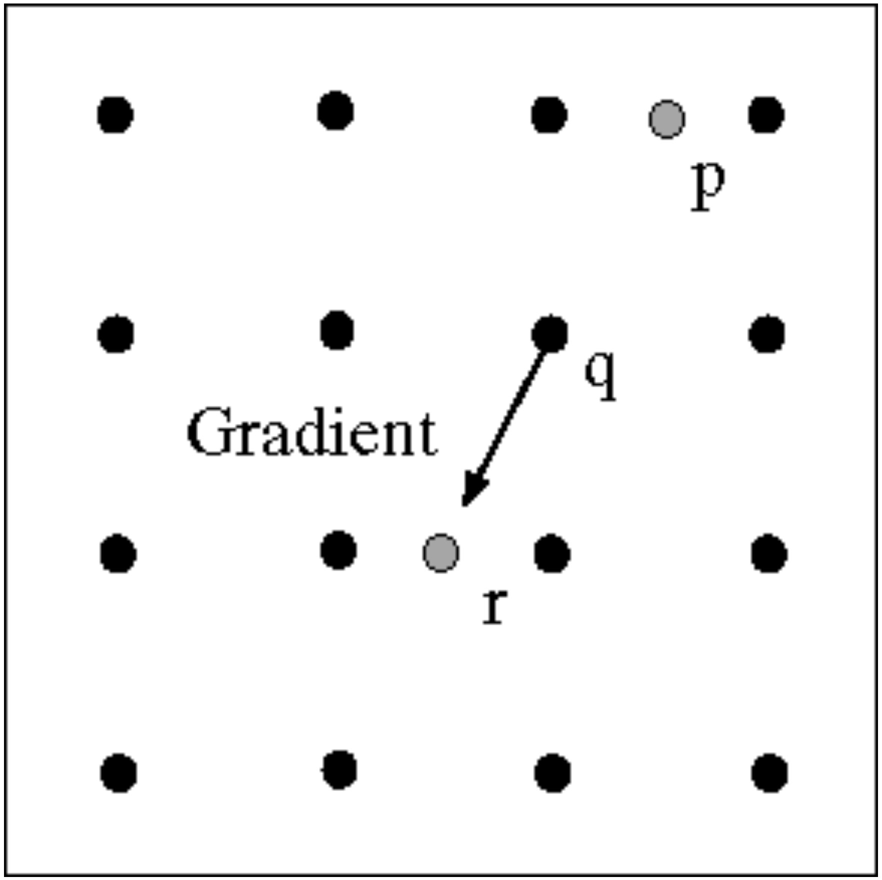
- Apply derivative of Gaussian
- Non-maximum suppression
  - Thin multi-pixel wide "ridges" down to single pixel width
- Linking and thresholding
  - Low, high edge-strength thresholds
  - Accept all edges over low threshold that are connected to edge over high threshold

Non-Maximum Supression

Select the single maximum point across the width of an edge.

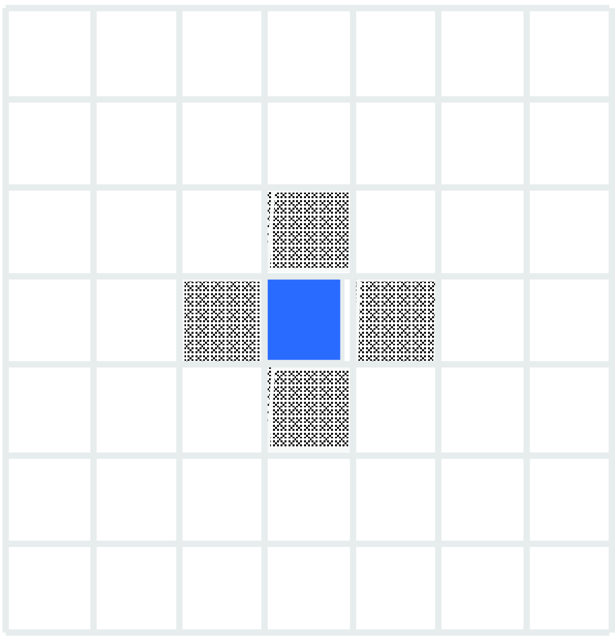
Linking to the Next Edge Point



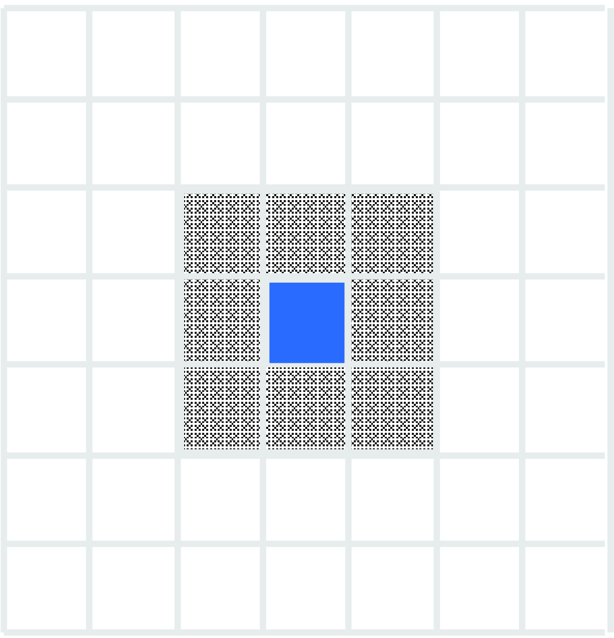
Assume the marked point q is an edge point.

Take the normal to the gradient at that point and use this to predict continuation points (either r or p).

Neighborhood for Edge Linking



Four Neighbors



Eight Neighbors

Edge Hysteresis

- Hysteresis threshold in Canny

- Idea: Maintain two thresholds k_{high} and k_{low}
  - Use k_{high} to find strong edges to start edge chain
  - Use k_{low} to find weak edges which continue edge chain

- Typical ratio of thresholds is roughly
  - $k_{high} / k_{low} = 2$

Canny Edge Detector

Original image (Lena)

Magnitude of the gradient

Canny Edge Detector

After non-maximum suppression

Canny Edge Operator

original

Canny with $\sigma = 1$

Canny with $\sigma = 2$

- The choice σ of depends on desired behavior
  - σ - large detects large scale edges
  - σ - small detects fine features

Implementation of Canny Detector

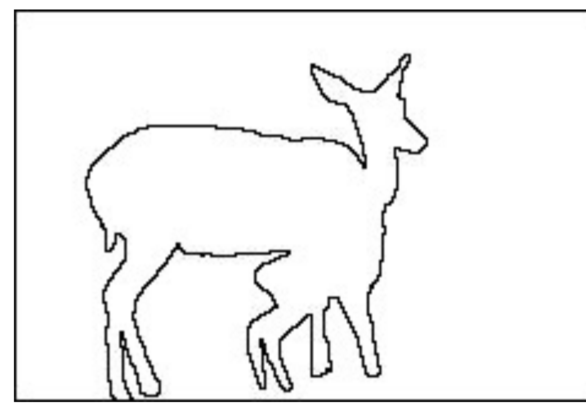
- There are many ways to implement Canny edge detector
- In openCV:

  void Canny(InputArray image, OutputArray edges, double threshold1, double threshold2, int aperture Size=3, bool L2gradient=false)

  - does not include Gaussian smoothing
  - Uses sobel operators to calculate the edge
  - Before use this function, Gaussian smoothing is often required

Edge-Based Object Extraction

Gaussian smooth σ_x, σ_y → Canny detector k_{hight}, k_{low} → Extract the contour of the object →



Suggested Reading

- E. R Davies, Computer Vision: Principles, Algorithms, Applications, Learning, Academic Press; 5th edition; 2017
  - Chapter 5
- D Forsyth, Computer Vision. A Modern Approach
  - Chapter 5

OpenCV 4.6.0

- Tutorials - python
  - https://docs.opencv.org/4.6.0/d2/d96/tutorial_py_table_of_contents_imgproc.html
  - Image Gradients
  - Canny Edge Detection
  - Contours in OpenCV