

CS 534: Computer Vision Edges

Spring 2004
Ahmed Elgammal
Dept of Computer Science
Rutgers University

CS 534 – Ahmed Elgammal – Edges - 1

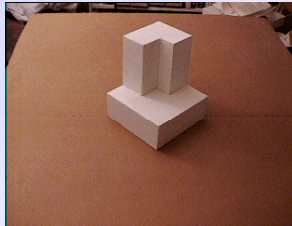
Outlines

- What are edges
- Image gradient
- Smoothing and image gradient
- Laplacian for edge detection/ Laplacian of Gaussian
- Gradient-based edge detection
- Hysteresis
- Edge detection by function fitting.

CS 534 – Ahmed Elgammal – Edges - 4

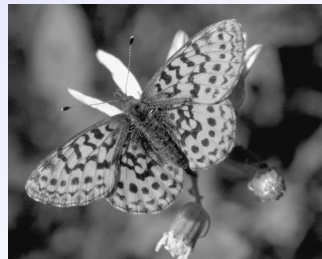
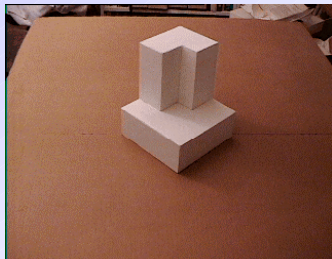
What are edges

- What is an edge? A sharp change in brightness
- What generates an edge (Where edges occur) ?
 - Boundaries between objects
 - Reflectance changes (within object)
 - Illumination changes: e.g., cast shadow boundary (within object)
 - Change in surface orientation (within object)



CS 534 – Ahmed Elgammal – Edges - 5

- Edge: A sharp change in brightness
- But which changes we would like to mark ? Meaningful changes. Hard to defined.
- How to tell a semantically meaningful edge from a nuisance edge ?
- Both low level and high level information



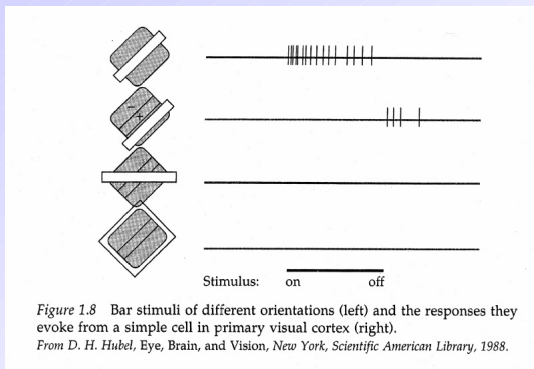
CS 534 – Ahmed Elgammal – Edges - 6



CS 534 – Ahmed Elgammal – Edges - 7

Biological

- We have seen evidence before of edge/bar detectors at different stages of our visual system.



CS 534 – Ahmed Elgammal – Edges - 8

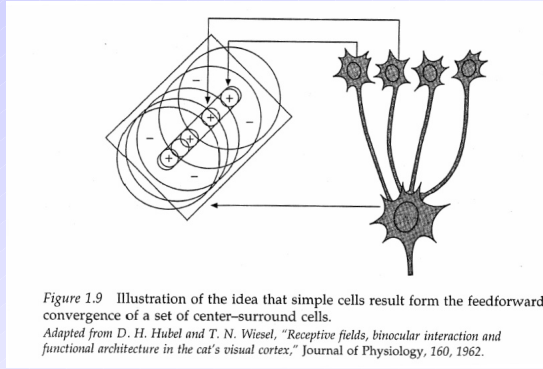


Figure 1.9 Illustration of the idea that simple cells result from the feedforward convergence of a set of center-surround cells.
Adapted from D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," Journal of Physiology, 160, 1962.

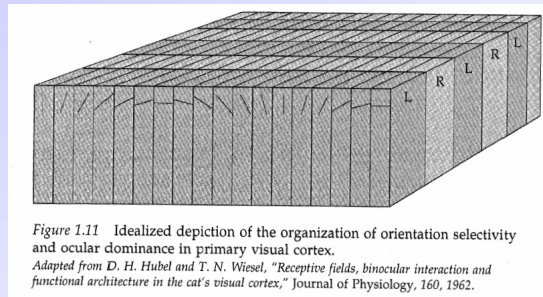
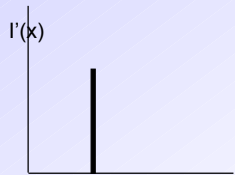
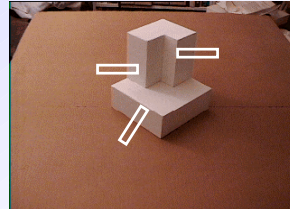
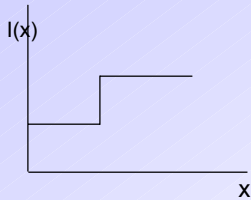


Figure 1.11 Idealized depiction of the organization of orientation selectivity and ocular dominance in primary visual cortex.
Adapted from D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," Journal of Physiology, 160, 1962.

Characteristic of an edge

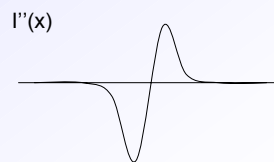
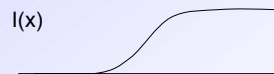
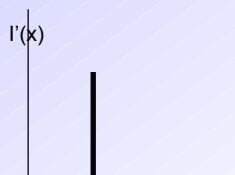
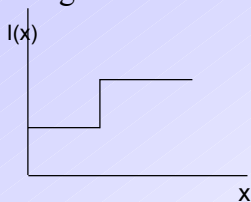
- Edge: A sharp change in brightness
- Ideal edge is a step function in certain direction.



CS 534 – Ahmed Elgammal – Edges - 11

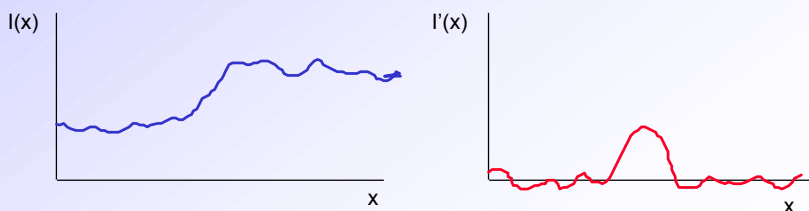
Characteristic of an edge

- Ideal edge is a step function in certain direction.
- The first derivative of $I(x)$ has a **peak** at the edge
- The second derivative of $I(x)$ has a **zero crossing** at the edge



CS 534 – Ahmed Elgammal – Edges - 12

- More realistically, image edges are blurred and the regions that meet at those edges have noise or variations in intensity.
 - blur - high first derivatives near edges
 - noise - high first derivatives within regions that meet at edges



CS 534 – Ahmed Elgammal – Edges - 13

Mathematical Edge Model

An edge is a step function of unknown height in the presence of stationary additive Gaussian noise

$$edge(x) = AU(x) + n(x)$$

Contrast \nearrow Step function \nwarrow Stationary Additive Gaussian Noise

$$U(x) = \begin{cases} 0 & x < 0 \\ 1 & x > 0 \end{cases}$$

CS 534 – Ahmed Elgammal – Edges - 14

Edge detection in 2-D: Image Gradient

- Let $f(x,y)$ be the image intensity function. It has derivatives in all directions
 - the **gradient** is a vector whose first component is the direction in which the first derivative is highest, and whose second component is the magnitude of the first derivative in that direction.
- If f is continuous and differentiable, then its gradient can be determined from the directional derivatives in any two orthogonal directions - standard to use x and y

– magnitude = $[(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2]^{1/2}$

– direction = $\tan^{-1}(\frac{\partial f / \partial y}{\partial f / \partial x})$

CS 534 – Ahmed Elgammal – Edges - 15

Partial Image derivatives

- With a digital image, the partial derivatives are replaced by finite differences:

– $\Delta_x f = f(x,y) - f(x-1, y)$

– $\Delta_y f = f(x,y) - f(x, y-1)$

$$\begin{bmatrix} -1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

- Alternatives are:

– $\Delta_{2x} f = f(x+1,y) - f(x-1,y)$

– $\Delta_{2y} f = f(x,y+1) - f(x,y-1)$

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

- Robert's gradient

– $\Delta_x f = f(x+1,y+1) - f(x,y)$

– $\Delta_y f = f(x,y+1) - f(x+1, y)$

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

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

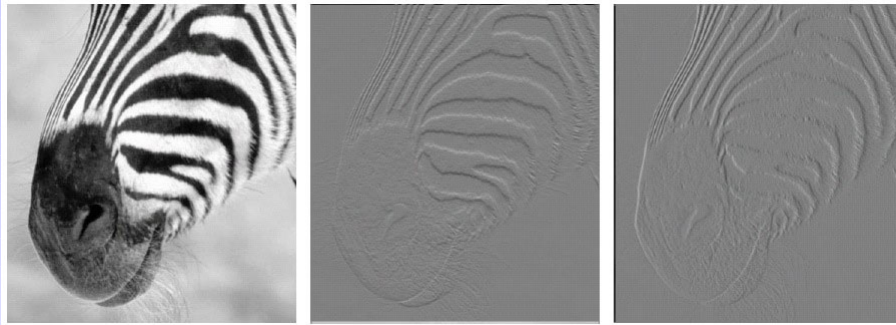
$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Prewitt

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

Sobel

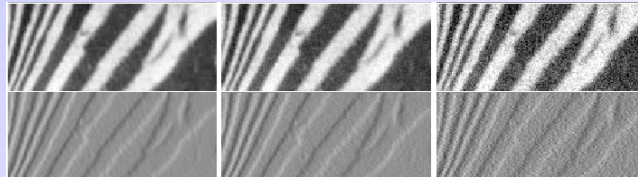
CS 534 – Ahmed Elgammal – Edges - 16



CS 534 – Ahmed Elgammal – Edges - 17

Finite differences and noise

- Finite difference filters respond strongly to noise
 - obvious reason: image noise results in pixels that look very different from their neighbors



- What is to be done?
 - intuitively, most pixels in images look quite a lot like their neighbors
 - this is true even at an edge; along the edge they're similar, across the edge they're not
 - **suggests that smoothing the image should help**, by forcing pixels different to their neighbors (=noise pixels?) to look more like neighbors

CS 534 – Ahmed Elgammal – Edges - 18

The response of a linear filter to noise (Recall)

- Recall noise model: stationary independent additive Gaussian noise with zero mean (non-zero mean is easily dealt with)
- Mean:**
 - output is a weighted sum of inputs
 - so we want mean of a weighted sum of zero mean normal random variables
 - must be zero
- Variance:**
 - recall
 - variance of a sum of random variables is sum of their variances
 - variance of constant times random variable is constant² times variance
 - then if σ^2 is noise variance and kernel is K , variance of response is

$$f_{observed}(x, y) = f(x, y) + N(0, \sigma^2)$$

$$g * f_{observed} = g * f + g * N(0, \sigma^2)$$

$$N'(0, \sigma'^2)$$

$$\sigma^2 \sum_{u,v} K_{u,v}^2$$

CS 534 – Ahmed Elgammal – Edges - 19

The response of a linear filter to noise (Recall)

$$f_{observed}(x, y) = f(x, y) + N(0, \sigma^2)$$

$$g * f_{observed} = g * f + g * N(0, \sigma^2)$$



$$N'(0, \sigma'^2)$$

$$\sigma'^2 = \sigma^2 \sum_{u,v} K_{u,v}^2$$

This can magnify or reduce the variance of the noise based on $\sum_{u,v} K_{u,v}^2$

If $\sum_{u,v} K_{u,v} = 1 \Rightarrow \sum_{u,v} K_{u,v}^2 \leq 1$ This reduces noise variance

CS 534 – Ahmed Elgammal – Edges - 20

Smoothing + Differentiation

- So smoothing should help.
- Recall: smoothing and differentiation are linear filters
- Recall also: linear filter are associative

$$K_{\partial/\partial x} * (g * I) = (K_{\partial/\partial x} * g) * I = \frac{\partial g}{\partial x} * I$$

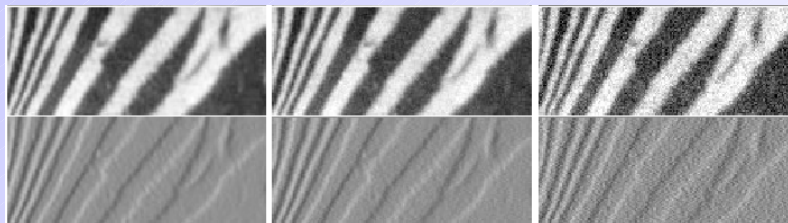
- **Smoothing then differentiation \equiv convolution with the derivative of the smoothing kernel.**
- If Gaussian is used for smoothing: We need to convolve the image with derivative of the Gaussian

$$\frac{\partial G_{\sigma}}{\partial x} * I \quad \frac{\partial G_{\sigma}}{\partial y} * I$$

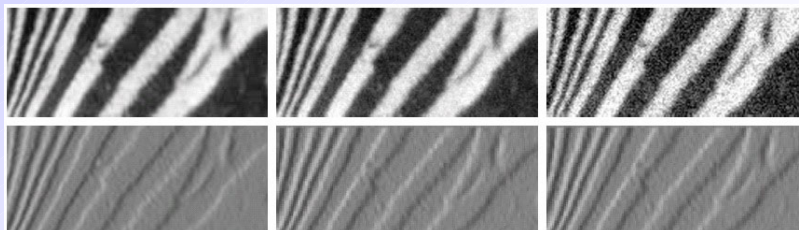
CS 534 – Ahmed Elgammal – Edges - 21

Noise $\sigma=3\%$

Noise $\sigma=9\%$



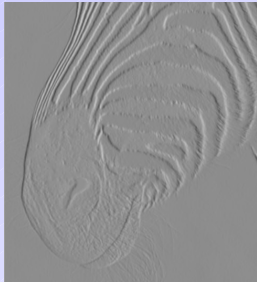
x- derivative - No smoothing



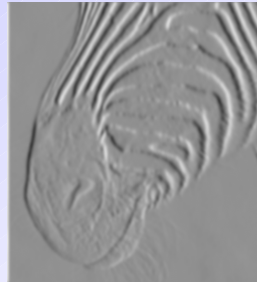
Convolution with x-derivative of a Gaussian ($\sigma=1$ pixel)

CS 534 – Ahmed Elgammal – Edges - 22

- The scale (σ) of the Gaussian has significant effects on the results



1 pixel



3 pixels



7 pixels

CS 534 – Ahmed Elgammal – Edges - 23

Detecting Edges

Approach 1: Using Laplacian

- Recall: an edge corresponds to a zero crossing at the second derivative
- Laplacian:

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

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- Its digital approximation is:

$$\nabla^2 f(x, y) = [f(x+1, y) - f(x, y)] - [f(x, y) - f(x-1, y)] + [f(x, y+1) - f(x, y)] - [f(x, y) - f(x, y-1)]$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$= [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4 f(x, y)$$

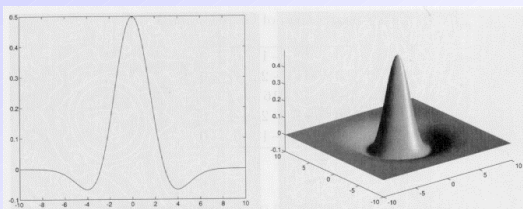
CS 534 – Ahmed Elgammal – Edges - 24

Laplacian of Gaussian

- Laplacian is a linear filter
- Bad idea to apply a Laplacian without smoothing
- If we smooth by a Gaussian before applying a Laplacian:

$$K_{\nabla^2} * (G_{\sigma} * I) = (K_{\nabla^2} * G_{\sigma}) * I = (\nabla^2 G_{\sigma}) * I$$

Laplacian of Gaussian (LoG)
"Mexican Hat"



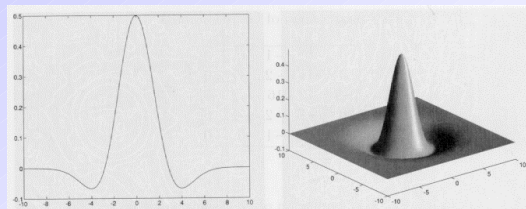
$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

CS 534 – Ahmed Elgammal – Edges - 25

Laplacian of Gaussian

- Can be approximated as difference of two Gaussians
- This is called Difference of Gaussians filter DoG

$$\nabla^2 g(x) \approx c_1 e^{-\frac{x^2}{2\sigma_1^2}} - c_2 e^{-\frac{x^2}{2\sigma_2^2}} \quad \sigma_1 < \sigma_2$$



$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

CS 534 – Ahmed Elgammal – Edges - 26

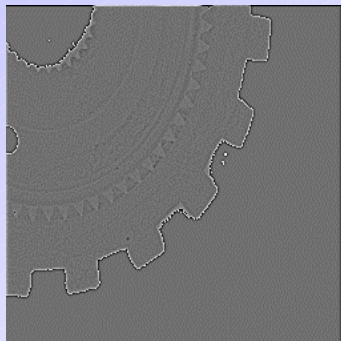
Algorithm:

- Convolve the image with a LoG
- Mark the point with zero crossings:
 - these are pixels whose LoG is positive and which have neighbor's whose LoG is negative or zero
- Check these points to ensure the gradient magnitude is large (to avoid low contrast edges)

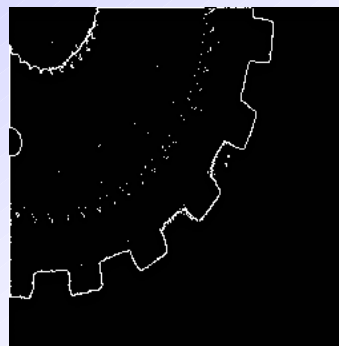
- Note : Two parameters: Gaussian scale, contrast threshold

CS 534 – Ahmed Elgammal – Edges - 27

Laplacian of Gaussian



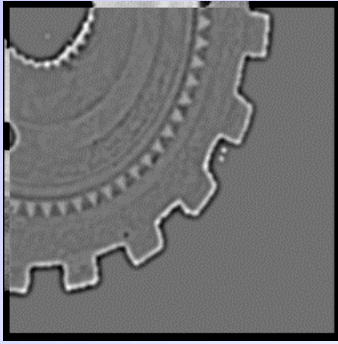
5x5 Mexican Hat - Laplacian of
Gaussian



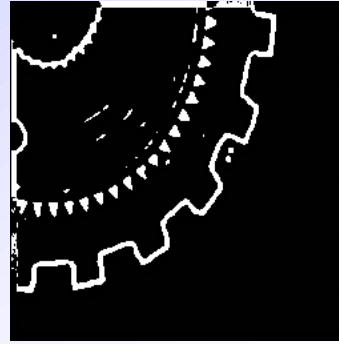
Zero crossings

CS 534 – Ahmed Elgammal – Edges - 28

Laplacian of Gaussian

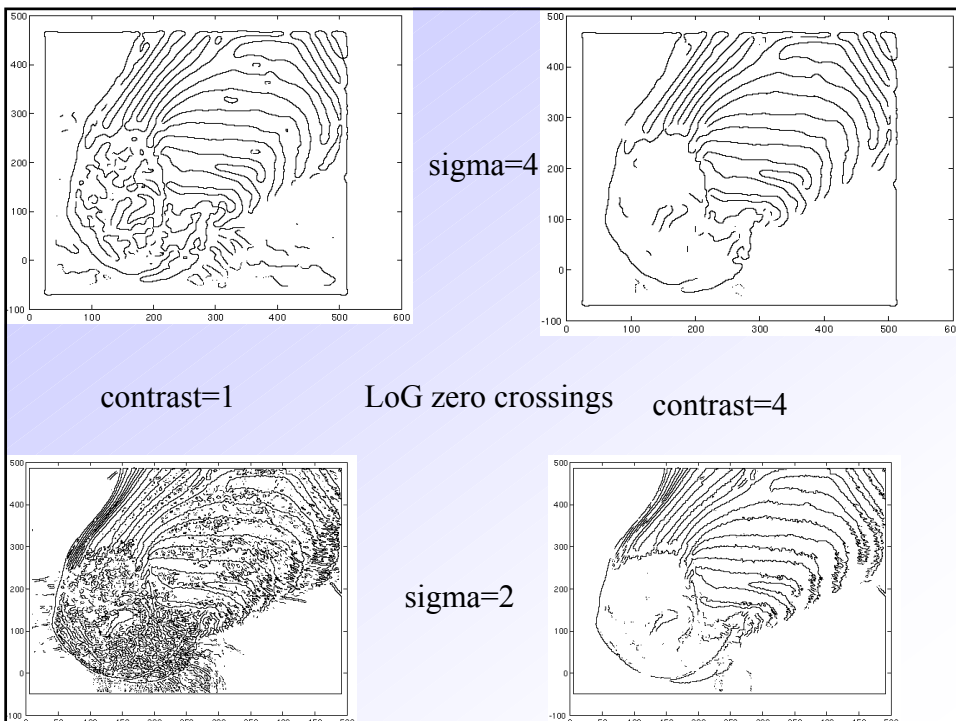


13 x 13 Mexican hat



zero crossings

CS 534 – Ahmed Elgammal – Edges - 29



contrast=1

LoG zero crossings

contrast=4

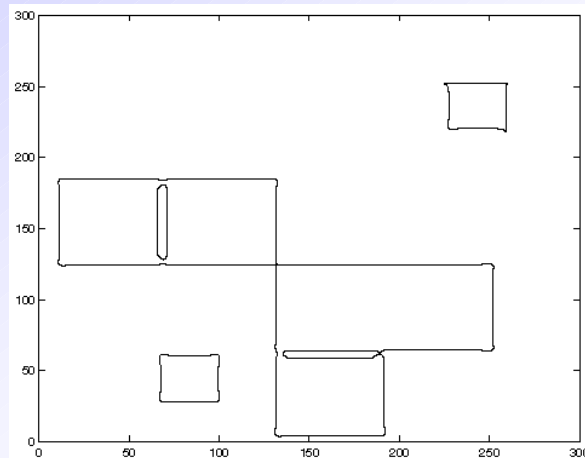
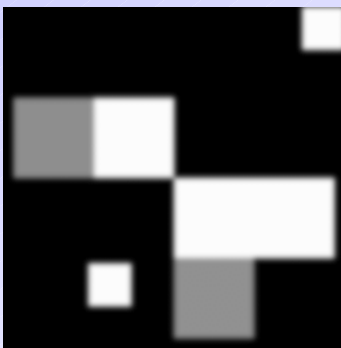
sigma=2

Problems with the Laplacian approach

- Poor behavior at corners
- Computationally: we need to compute both the LoG and the gradient.

CS 534 – Ahmed Elgammal – Edges - 31

We have unfortunate behavior
at corners



Gradient-Based Edge Detectors

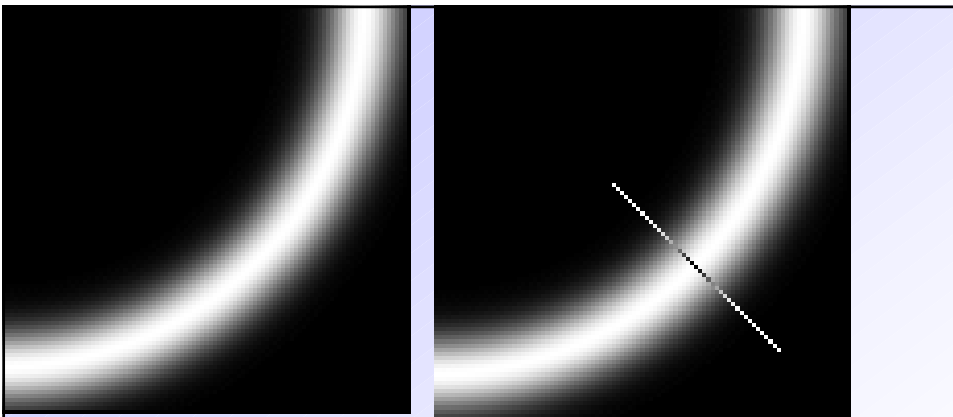
General strategy

- determine image gradient (with smoothing)
- now mark points where gradient magnitude is particularly large wrt neighbors



- The gradient magnitude is large along thick trail; how do we identify the significant points?

CS 534 – Ahmed Elgammal – Edges - 33

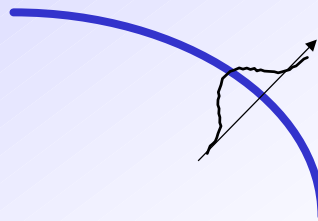


We wish to mark points along the curve where the magnitude is biggest. We can do this by looking for a maximum along a slice normal to the curve (non-maximum suppression). These points should form a curve. There are then two algorithmic issues: at which point is the maximum, and where is the next one?

CS 534 – Ahmed Elgammal – Edges - 34

Non-maxima suppression

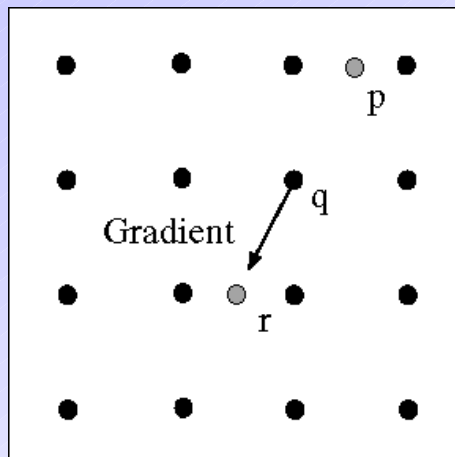
- Non-maxima suppression - Retain a point as an edge point if:
 - its gradient magnitude is higher than a threshold
 - its gradient magnitude is a local maxima in the gradient direction



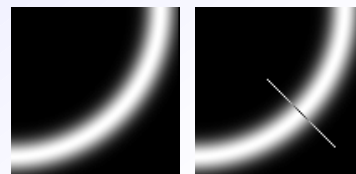
simple thresholding will
compute thick edges

CS 534 – Ahmed Elgammal – Edges - 35

Non-maximum suppression



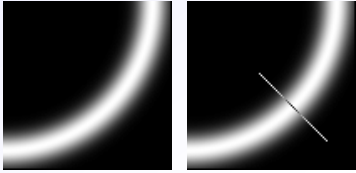
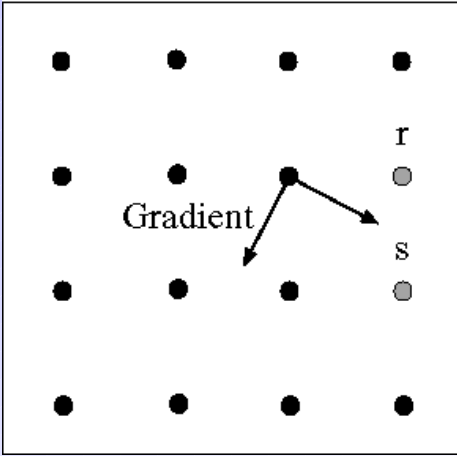
At q , we have a maximum if
the value is larger than
those at both p and at r .
Interpolate to get these
values.



CS 534 – Ahmed Elgammal – Edges - 36

Predicting the next edge point

Assume the marked point is an edge point. Then we construct the tangent to the edge curve (which is normal to the gradient at that point) and use this to predict the next points (here either r or s).



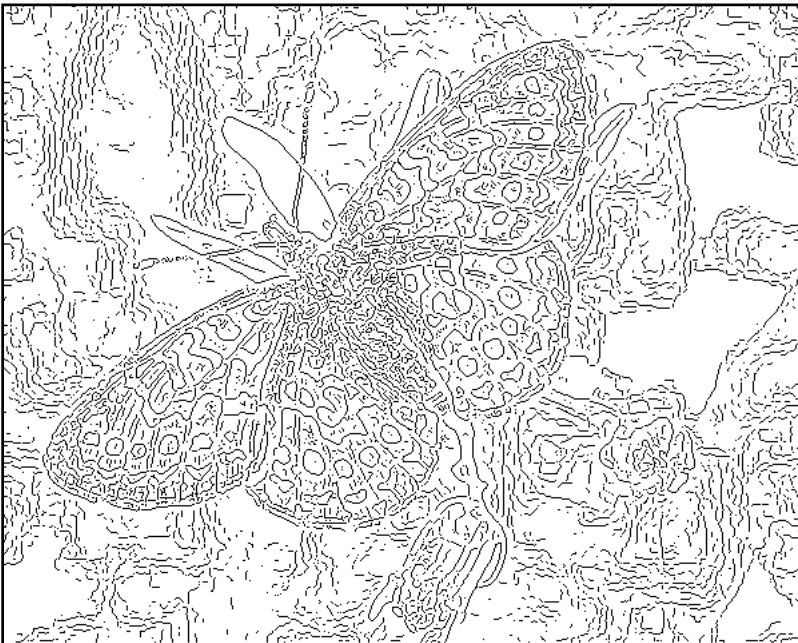
CS 534 – Ahmed Elgammal – Edges - 37

Problem of scale and threshold

- Usually, any single choice of σ does not produce a good edge map
 - a large σ will produce edges from only the largest objects, and they will not accurately delineate the object because the smoothing reduces shape detail
 - a small σ will produce many edges and very jagged boundaries of many objects.
- Scale-space approaches
 - detect edges at a range of scales $[\sigma_1, \sigma_2]$
 - combine the resulting edge maps
 - trace edges detected using large σ down through scale space to obtain more accurate spatial localization.

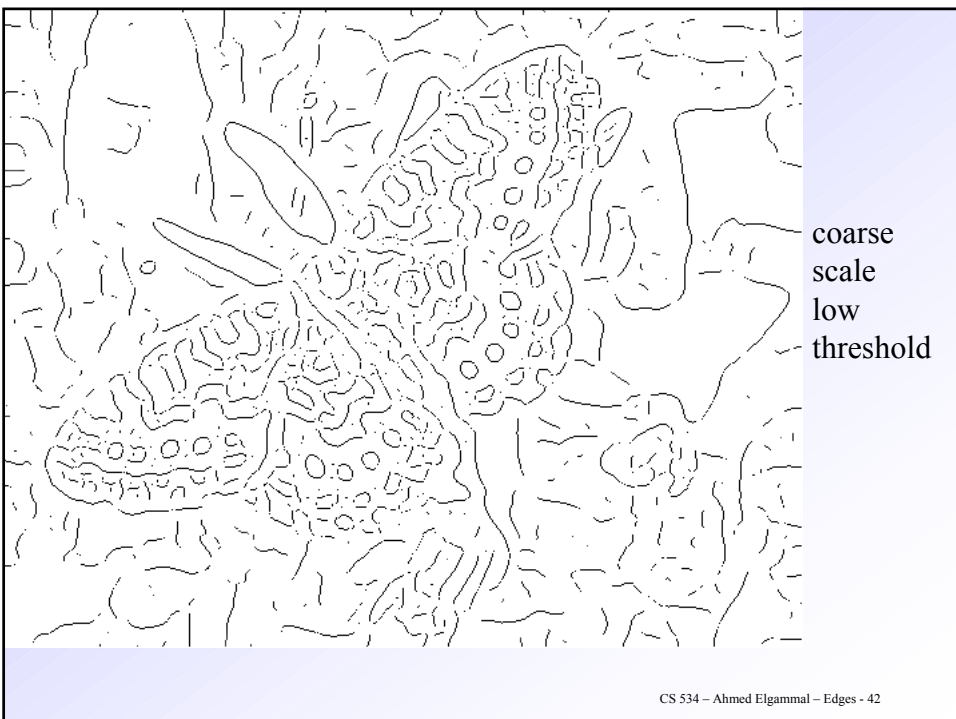
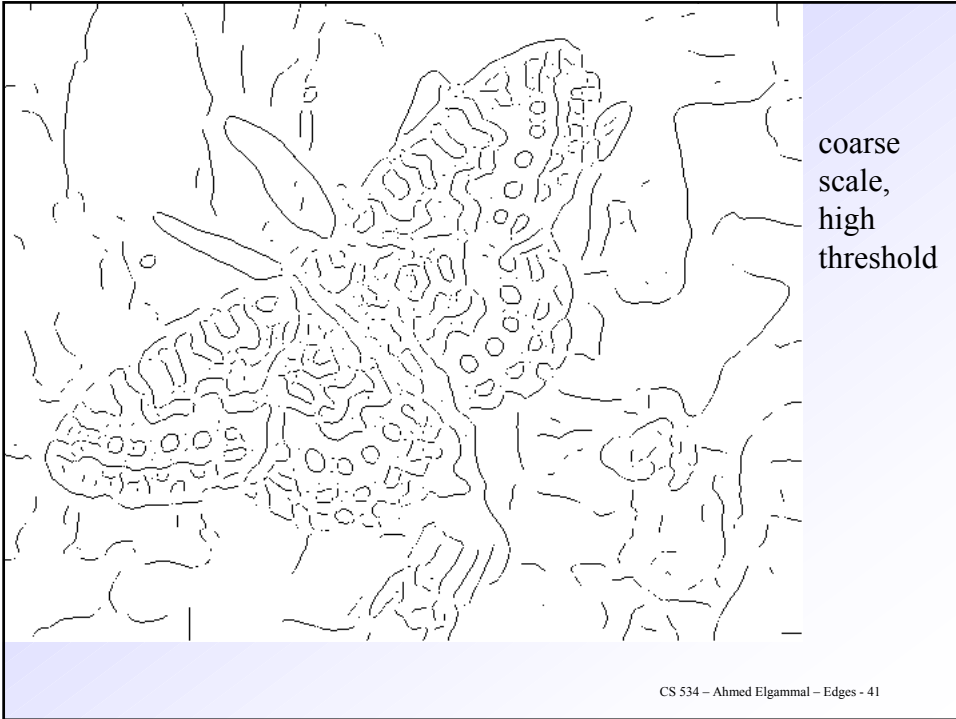


CS 534 – Ahmed Elgammal – Edges - 39



fine scale
high
threshold

CS 534 – Ahmed Elgammal – Edges - 40



Hysteresis

- Which threshold we should use ?
- Fine scale: fine details.
- Coarser scale: fine details disappear.
- Low threshold : low contrast edges. a variety of new edge points of dubious significance are introduced.
- High threshold: loose low contrast edges \Rightarrow broken edges
- Solution: use two thresholds
 - Larger threshold: more certain edge, use to start an edge chain
 - Smaller threshold: use to follow the edge chain

CS 534 – Ahmed Elgammal – Edges - 43

Edge detection by function fitting

- General approach
 - fit a function to each neighborhood of the image
 - use the gradient of the function as the digital gradient of the image neighborhood

CS 534 – Ahmed Elgammal – Edges - 44

Edge detection by function fitting

- Example: fit a plane to a 2x2 neighborhood
 - $z = ax + by + c$; z is gray level - need to determine a, b, c
 - gradient is then $(a^2 + b^2)^{1/2}$
 - neighborhood points are $f(x, y)$, $f(x+1, y)$, $f(x, y+1)$ and $f(x+1, y+1)$
- Need to minimize

$$E(a, b, c) = \sum_{i=0}^1 \sum_{j=0}^1 [a(x+i) + b(y+j) + c - f(x+i, y+j)]^2$$

- Solve this and similar problems by:
 - differentiating with respect to a, b, c , setting results to 0, and
 - solving for a, b, c in resulting system of equations

CS 534 – Ahmed Elgammal – Edges - 45

Edge detection by function fitting

- $\partial E / \partial a = \sum \sum 2[a(x+i) + b(y+j) + c - f(x+i, y+j)](x+i)$
- $\partial E / \partial b = \sum \sum 2[a(x+i) + b(y+j) + c - f(x+i, y+j)](y+j)$
- $\partial E / \partial c = \sum \sum 2[a(x+i) + b(y+j) + c - f(x+i, y+j)]$
- It is easy to verify that
 - $a = [f(x+1, y) + f(x+1, y+1) - f(x, y) - f(x, y+1)]/2$
 - $b = [f(x, y+1) + f(x+1, y+1) - f(x, y) - f(x+1, y)]/2$
- a and b are the x and y partial derivatives

$$a = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

CS 534 – Ahmed Elgammal – Edges - 46

Edge detection by function fitting

- Could also fit a higher order surface than a plane
 - with a second order surface we could find the (linear) combination of pixel values that corresponds to the higher order derivatives, which can also be used for edge detection
- Would ordinarily use a neighborhood larger than 2x2
 - better fit
 - for high degree functions need more points for the fit to be reliable.

CS 534 – Ahmed Elgammal – Edges - 47

Edge linking and following

- Group edge pixels into **chains** and chains into large pieces of object boundary.
 - can use the shapes of long edge chains in recognition
 - slopes
 - curvature
 - Corners
- Basic steps
 - thin connected components of edges to one pixel thick
 - find simply connected paths
 - link them at corners into a graph model of image contours
 - compute local and global properties of contours and corners

CS 534 – Ahmed Elgammal – Edges - 48

Sources

- Forsyth and Ponce, Computer Vision a Modern approach: chapter 8.
- L. G. Shapiro and G. C. Stockman “Computer Vision”, Prentice Hall 2001.
- R. Gonzalez and R.E. Woods, “Digital Image Processing”, 2002.
- Slides by
 - D. Forsyth @ Berkeley
 - L.S. Davis @UMD
 - G.C. Stockman @MSU