

Introduction to Computer Vision

CS / ECE 181B
Thursday, April 22, 2004

Today

→ Edge detection (HO #5)

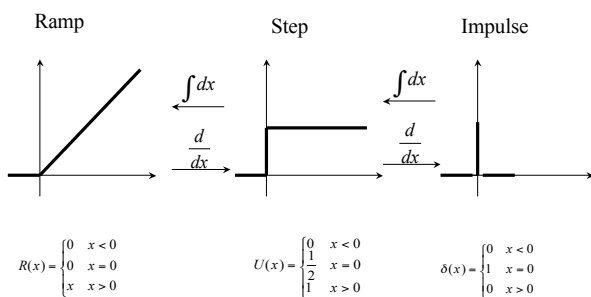
- HW#3 due, next week
- No office hours today

Edge Detection

- Edge detection is a local area operator that seeks to find significant, meaningful changes in image intensity (color?) that correspond to
 - Boundaries of objects and patterns
 - Texture
 - Changes in object color or brightness
 - Highlights
 - Occlusions
 - Etc.



Useful Mathematics Funcs.

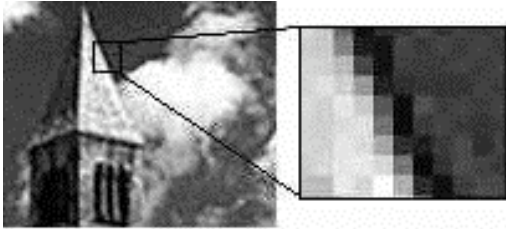


The bad news

- Unfortunately, it's very hard to tell *significant* edges from *bogus* edges!
 - Noise is a big problem!
- An edge detector is basically a high-frequency filter, since sharp intensity changes are high-frequency events
- But image noise is also high-frequency, so edge detectors tend to accentuate noise!
- Some things to do:
 - Smooth before edge detection (hoping to get rid of noise but not edges!)
 - Look for edges at multiple scales (pyramids!)
 - Use an adaptive edge threshold

Caveats

- In reality, low light levels and random noise lead to high fluctuations in individual pixel values, leading to bad estimations.



Edge detection history

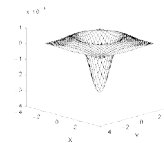
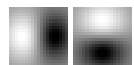
- Edge detection has a long history and a huge literature
 - Edge modeling: Step edges, roof edges, impulse edges...
 - Biological modeling: How does human vision do it?
 - Elegant and complex mathematical models
 - Simple and computationally cheap edge detectors
 - Etc., etc., etc....
- Typical usage:
 - Detect “edge points” in the image (filter then threshold)
 - ♦ Edges may have magnitude **and** orientation
 - Throw away “bad” ones (isolated points)
 - Link edge points together to make edge segments
 - Merge segments into lines, corners, junctions, etc.
 - Interpret these higher-level features in the context of the problem

Edge detection

- The bottom line:
 - It doesn't work!
 - At least, now how we'd like it to:
 - ♦ Too many false positives (noise)
 - ♦ Too many omissions (little or no local signal)
- Still, edge detection is often the first step in a computer vision program
 - We have to learn to live with imperfection

Edge detectors

- Gradient-based edge detectors
 - Approximate a spatial derivative
 - X and Y directions, or at various orientations
 - Fundamentally high-pass (accentuates noise)
- Laplacian and other band-pass edge detectors



- Roberts, Sobel, Prewitt, Canny,....

Digital Implementations

- 1st order operator - 1x2 or 2x1 mask
 - simple
 - unbalanced (forward differencing)
 - sensitive to noise

$$\begin{bmatrix} E_{i,j} & E_{i+1,j} \end{bmatrix}$$

$$\frac{\partial E}{\partial x} \approx E_{i+1,j} - E_{i,j}$$

$$\begin{bmatrix} E_{i,j+1} \\ E_{i,j} \end{bmatrix}$$

$$\frac{\partial E}{\partial y} \approx E_{i,j+1} - E_{i,j}$$

Another Implementation

- 1st order operator - 2x2 mask
 - simple
 - unbalanced (forward differencing)
 - more resistive to noise

$$\begin{bmatrix} E_{i,j+1} & E_{i+1,j+1} \\ E_{i,j} & E_{i+1,j} \end{bmatrix}$$

$$\frac{\partial E}{\partial x} \approx \frac{1}{2} ((E_{i+1,j+1} - E_{i,j+1}) + (E_{i+1,j} - E_{i,j}))$$

$$\begin{bmatrix} E_{i,j+1} & E_{i+1,j+1} \\ E_{i,j} & E_{i+1,j} \end{bmatrix}$$

$$\frac{\partial E}{\partial y} \approx \frac{1}{2} ((E_{i+1,j+1} - E_{i+1,j}) + (E_{i,j+1} - E_{i,j}))$$

Robert's detector

$$\begin{bmatrix} 1 & \text{diagonal} \\ \text{diagonal} & -1 \end{bmatrix} \quad E_x$$

- Compute the X- and Y- derivatives using the above masks
- Compute the magnitude of the gradient
- Compute the gradient direction

$$\begin{bmatrix} \text{diagonal} & -1 \\ 1 & \text{diagonal} \end{bmatrix} \quad E_y$$

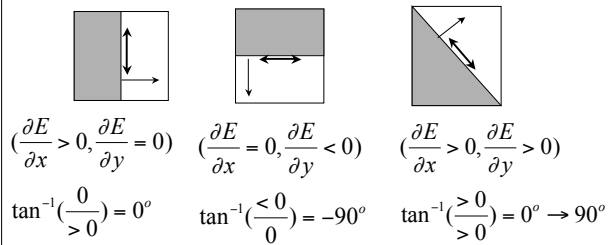
Observation in 2D

- 2D 1st order edge operator
 - A magnitude
 - A direction, but ...
 - Edge direction: iso-brightness direction
 - Gradient direction: largest brightness change direction

$$magnitude = \sqrt{\left(\frac{\partial E}{\partial x}\right)^2 + \left(\frac{\partial E}{\partial y}\right)^2}$$

$$direction = \tan^{-1}\left(\frac{\frac{\partial E}{\partial y}}{\frac{\partial E}{\partial x}}\right)$$

Gradient vs. Iso-brightness dirs



More Edge detectors

- Sobel detector

G_x			G_y		
-1	0	1	-1	-2	-1
-2	0	2	0	0	0
-1	0	1	1	2	1

$$|G| = G_x^2 + G_y^2$$

$$\angle G = \text{atan } G_y/G_x$$

- Prewitt detector

G_x			G_y		
-1	0	1	-1	-1	-1
-1	0	1	0	0	0
-1	0	1	1	1	1

Lena



Edge detectors: second order operators

- Laplacian detectors

$$\frac{\partial^2 E}{\partial x^2} = \frac{\partial E}{\partial x} - \frac{\partial E}{\partial x}$$

$$= (E_{i+1,j} - E_{i,j}) - (E_{i,j} - E_{i-1,j})$$

$$= E_{i+1,j} - 2E_{i,j} + E_{i-1,j}$$

$$\frac{\partial^2 E}{\partial y^2} = E_{i,j+1} - 2E_{i,j} + E_{i,j-1}$$

$$\text{Laplacian: } \frac{\partial^2 E}{\partial x^2} + \frac{\partial^2 E}{\partial y^2} = E_{i+1,j} + E_{i-1,j} + E_{i,j+1} + E_{i,j-1} - 4E_{i,j}$$

$E_{i-1,j+1}$	$E_{i,j+1}$	$E_{i+1,j+1}$
$E_{i-1,j}$	$E_{i,j}$	$E_{i+1,j}$
$E_{i-1,j-1}$	$E_{i,j-1}$	$E_{i+1,j-1}$

	1	
1	-4	1
	1	

Edge detectors are not limited to 3x3 kernels

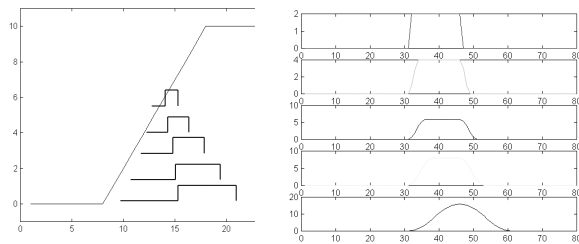
2nd order operator (cont.)

- Zero-crossing
 - if pixel $> t$ and one of its neighbor $< -t$, or
 - if pixel $< -t$ and one of its neighbor $> t$

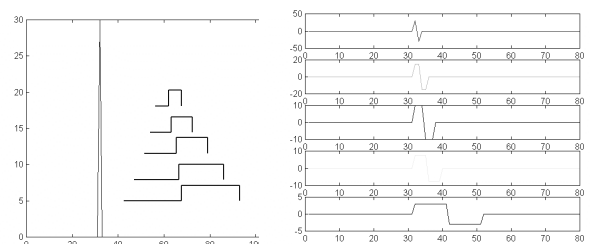
Smoothing Filter

- Differentiation enhances noise (as well as edges).
- Smooth the image before edge detection
 - Helps in minimizing false positives.
 - Edges at different scales.
- Gaussian smoothing: Why?
 - $G * G$ is also a Gaussian
 - ♦ Efficient multi-scale convolutions
 - Central limit theorem—smooth many times == Gaussian smoothing with an appropriate sigma.
 - Gaussians are separable==good for implementation.

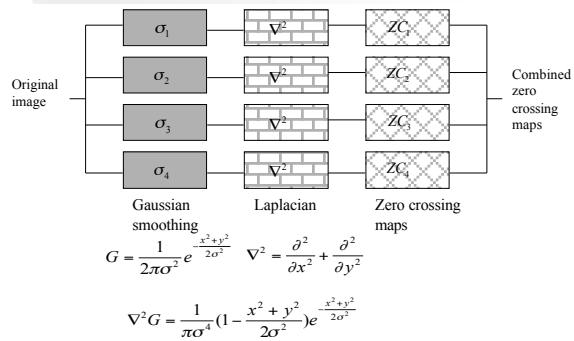
Multiple Scales



Multiple Scales



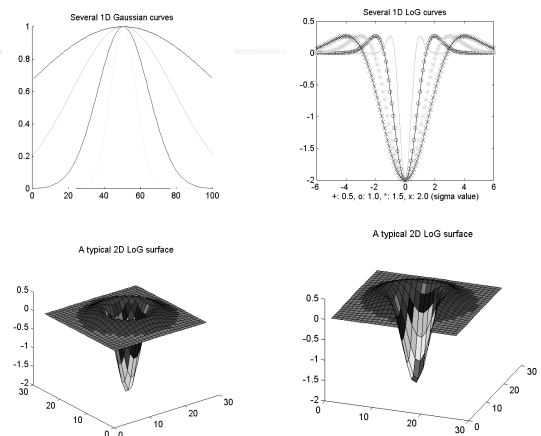
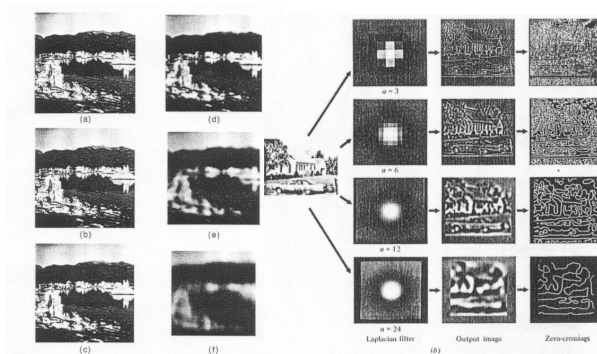
Multiple Scales

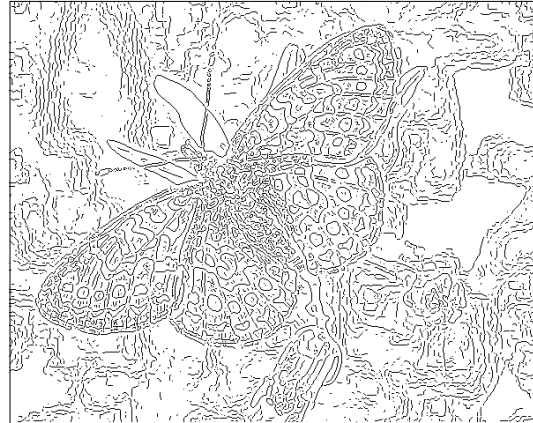


Laplacian of the Gaussian (LOG)

- Smooth with the Gaussian + Laplacian + zero-crossing detector==gives edges
- Equivalent to convolving the image with the Laplacian of the Gaussian kernel.
 - Note: differentiation is linear and shift invariant.
 - Convolution is associative.
- Can be well approximated by the Difference of the Gaussians (DoG)
- Marr-Hildreth operator.

Multiple Scales





fine scale

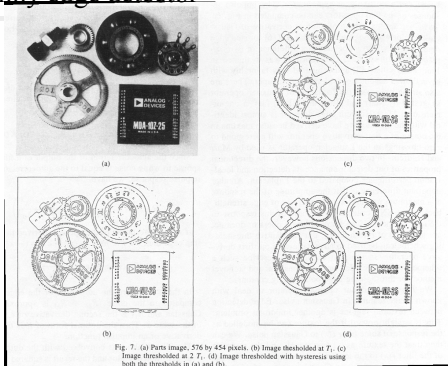


coarse scale

Other popular edge detectors

- Canny edge detector
 - Formulate the problem as an optimization problem
 - What is the criteria of *good* edge detection?
 - ♦ Good detection
 - Low probability of misses
 - Low probability of false alarms
 - ♦ Good localization
 - Closeness of marked edge points to true edge points
 - ♦ Single response criteria
 - Not *too many* strong responses of a single edge in a *small* neighborhood
- Beyond the scope of this course...read the handout for additional information.

Canny edge detector



Summary of Linear Filtering (HO#4) and Edge Detection (HO #5)

- Convolution (correlation) defines a shift-invariant linear filter
- The Fourier transform is a linear operation that exposes the spatial frequency composition of an image
- Sampling and aliasing are directly related to spatial frequency issues
 - E.g., image pyramids
- Correlation can be viewed as template matching (or pattern matching)
- If the template is a gradient/derivative operator, correlation implements edge detection
- Edge detection by itself doesn't work very well, although it can be useful if its limitations are understood
 - Noise, missing edges, complex scenes, ...