Lecture 11:
LoG and DoG Filters

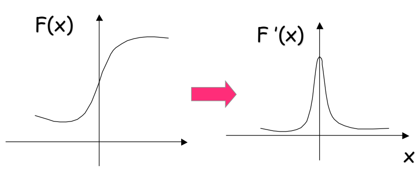
Today's Topics

Laplacian of Gaussian (LoG) Filter
- useful for finding edges
- also useful for finding blobs!

approximation using Difference of Gaussian (DoG)

Recall: First Derivative Filters

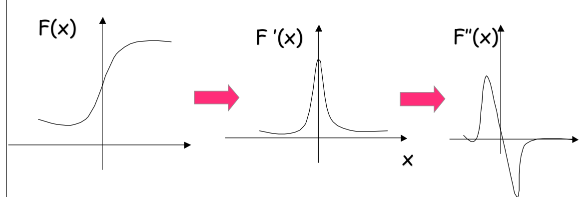
- Sharp changes in gray level of the input image correspond to "peaks or valleys" of the first-derivative of the input signal.



(1D example)

Second-Derivative Filters

- Peaks or valleys of the first-derivative of the input signal, correspond to "zero-crossings" of the second-derivative of the input signal.



Numerical Derivatives

See also T&V, Appendix A.2

Taylor Series expansion

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(x) + \frac{1}{3!}h^3 f'''(x) + O(h^4)$$

add

$$+ \left[f(x-h) = f(x) - hf'(x) + \frac{1}{2}h^2 f''(x) - \frac{1}{3!}h^3 f'''(x) + O(h^4) \right]$$

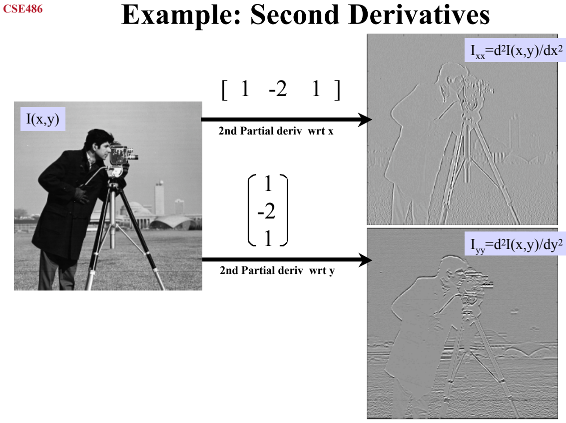
$$f(x+h) + f(x-h) = 2f(x) + h^2 f''(x) + O(h^4)$$

$$\frac{f(x-h) - 2f(x) + f(x+h)}{h^2} = f''(x) + O(h^2)$$

1	-2	1

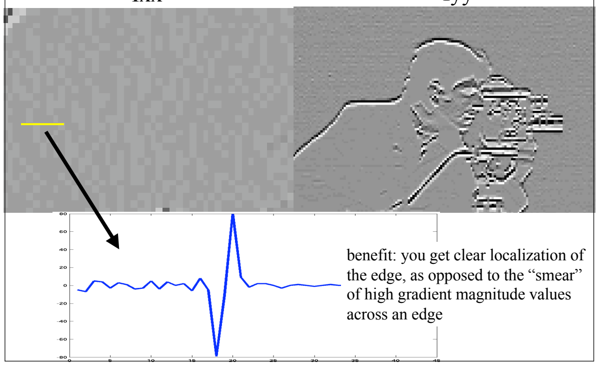
Central difference approx to second derivative

Example: Second Derivatives



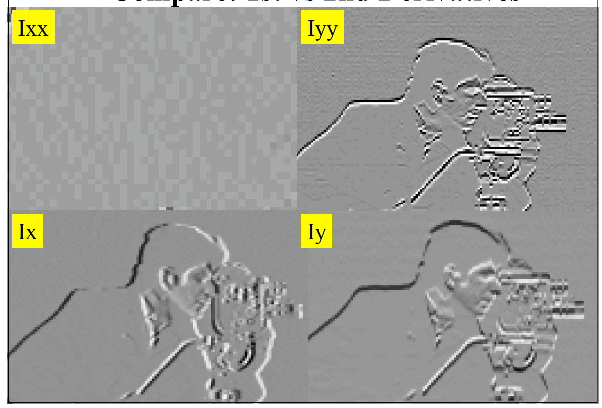
Example: Second Derivatives

Ixx Iyy



benefit: you get clear localization of the edge, as opposed to the "smear" of high gradient magnitude values across an edge

Compare: 1st vs 2nd Derivatives



Finding Zero-Crossings

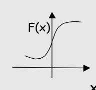
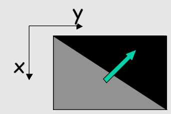
An alternative approx to finding edges as peaks in first deriv is to find zero-crossings in second deriv.

In 1D, convolve with [1 -2 1] and look for pixels where response is (nearly) zero?

Problem: when first deriv is zero, so is second. I.e. the filter [1 -2 1] also produces zero when convolved with regions of constant intensity.

So, in 1D, convolve with [1 -2 1] and look for pixels where response is nearly zero AND magnitude of first derivative is "large enough".

Edge Detection Summary

	1D	2D
step edge	I(x) 	I(x,y) 
1st deriv	$\left\| \frac{dI(x)}{dx} \right\| > Th$	$\|\nabla I(x,y)\| = (I_x^2(x,y) + I_y^2(x,y))^{1/2} > Th$ $\tan\theta = I_x(x,y) / I_y(x,y)$
2nd deriv	$\frac{d^2I(x)}{dx^2} = 0$	$\nabla^2 I(x,y) = I_{xx}(x,y) + I_{yy}(x,y) = 0$ Laplacian

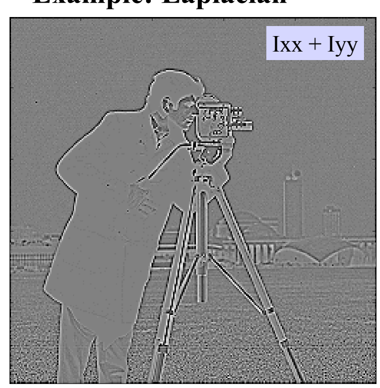
Finite Difference Laplacian

$$I_{xx} + I_{yy} = \left(\begin{bmatrix} 1 & -2 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \right) * I$$

$$= \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{\text{Laplacian filter } \nabla^2 I(x,y)} * I$$

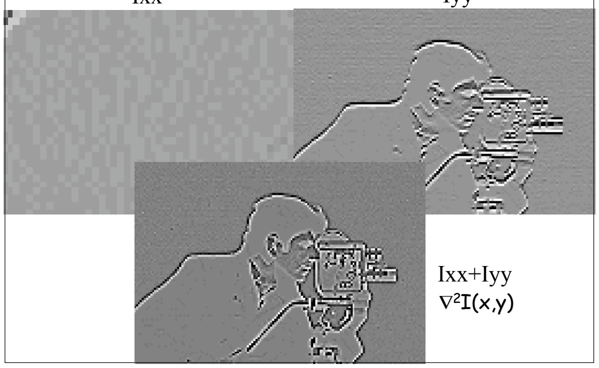
Example: Laplacian

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



Example: Laplacian

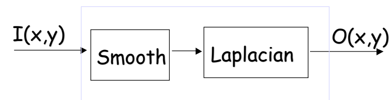
Ixx Iyy



Ixx+Iyy
$\nabla^2 I(x,y)$

Notes about the Laplacian:

- $\nabla^2 I(x,y)$ is a SCALAR
  - ↑ Can be found using a SINGLE mask
  - ↓ Orientation information is lost
- $\nabla^2 I(x,y)$ is the sum of SECOND-order derivatives
  - But taking derivatives increases noise
  - Very noise sensitive!
- It is always combined with a smoothing operation:



LoG Filter

- First smooth (Gaussian filter),
- Then, find zero-crossings (Laplacian filter):
  - $O(x,y) = \nabla^2(I(x,y) * G(x,y))$

Just another linear filter.

$$\underbrace{\nabla^2(f(x,y)\otimes G(x,y))}_{\text{Laplacian of Gaussian-filtered image}} = \underbrace{\nabla^2 G(x,y)\otimes f(x,y)}_{\text{Laplacian of Gaussian (LoG) -filtered image}}$$

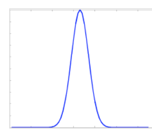
Laplacian of Gaussian-filtered image

Laplacian of Gaussian (LoG) -filtered image

Do you see the distinction?

1D Gaussian and Derivatives

$$g(x) = e^{-\frac{x^2}{2\sigma^2}}$$

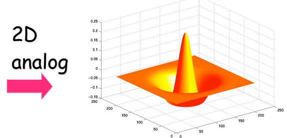


$$g'(x) = -\frac{1}{2\sigma^2}2xe^{-\frac{x^2}{2\sigma^2}} = -\frac{x}{\sigma^2}e^{-\frac{x^2}{2\sigma^2}}$$

$$g''(x) = (\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2})e^{-\frac{x^2}{2\sigma^2}}$$

Second Derivative of a Gaussian

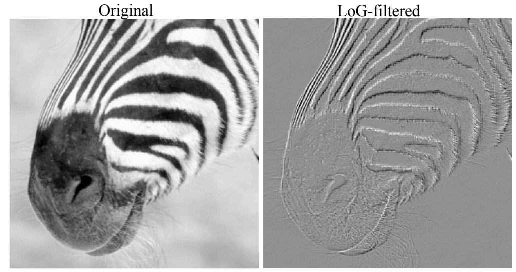
$$g''(x) = (\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2})e^{-\frac{x^2}{2\sigma^2}}$$



LoG "Mexican Hat"

Effect of LoG Operator

Original LoG-filtered

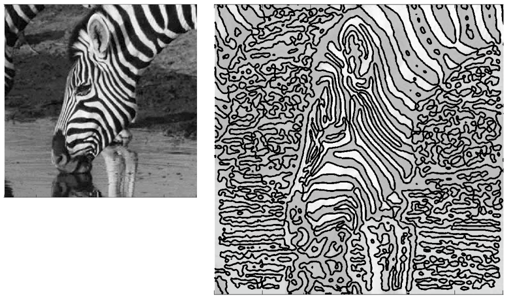


Band-Pass Filter (suppresses both high and low frequencies)
Why? Easier to explain in a moment.

Robert Collins
CSE486

Zero-Crossings as an Edge Detector

Raw zero-crossings (no contrast thresholding)

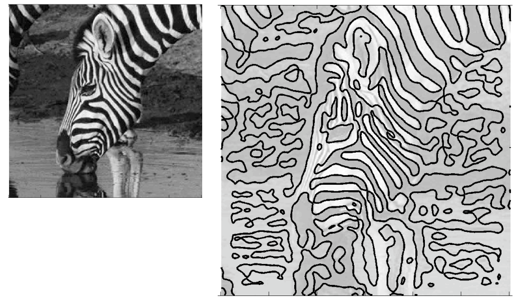


LoG sigma = 2, zero-crossing

Robert Collins
CSE486

Zero-Crossings as an Edge Detector

Raw zero-crossings (no contrast thresholding)

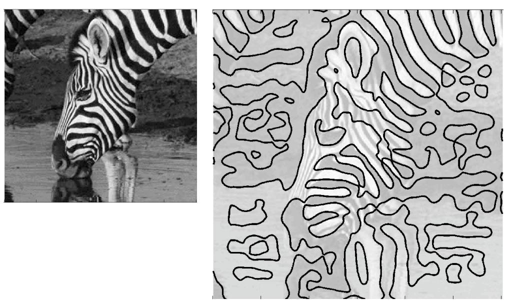


LoG sigma = 4, zero-crossing

Robert Collins
CSE486

Zero-Crossings as an Edge Detector

Raw zero-crossings (no contrast thresholding)



LoG sigma = 8, zero-crossing

Robert Collins
CSE486

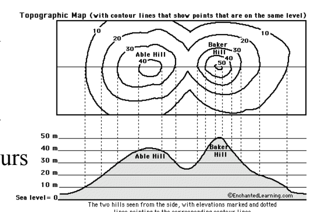
Note: Closed Contours

You may have noticed that zero-crossings form closed contours. It is easy to see why…

Think of equal-elevation contours on a topo map.

Each is a closed contour.

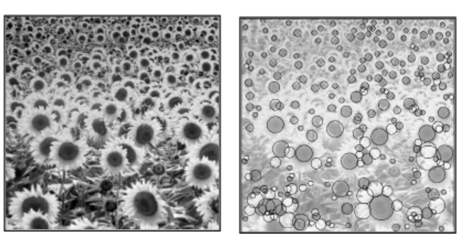
Zero-crossings are contours at elevation = 0 .



remember that in our case, the height map is of a LoG filtered image - a surface with both positive and negative "elevations"

Robert Collins
CSE486

Other uses of LoG: Blob Detection



Lindeberg: ``Feature detection with automatic scale selection''. International Journal of Computer Vision, vol 30, number 2, pp. 77--116, 1998.

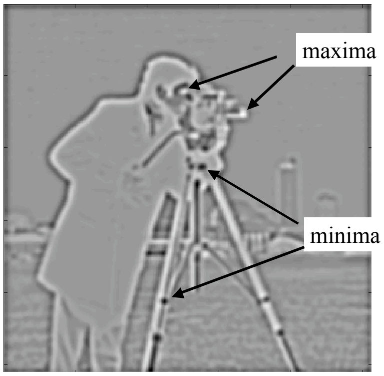
Robert Collins
CSE486

Pause to Think for a Moment:

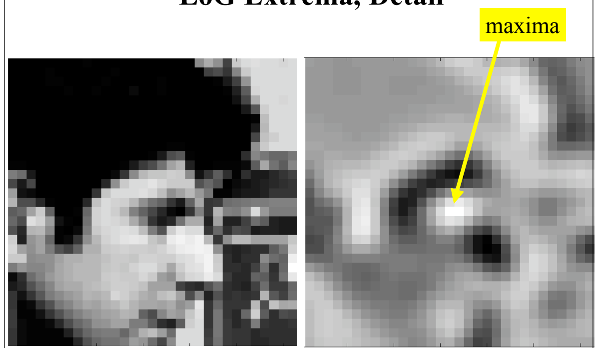
How can an edge finder also be used to find blobs in an image?

Example: LoG Extrema

LoG
sigma = 2



LoG Extrema, Detail

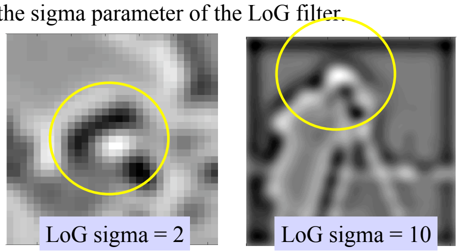


LoG sigma = 2

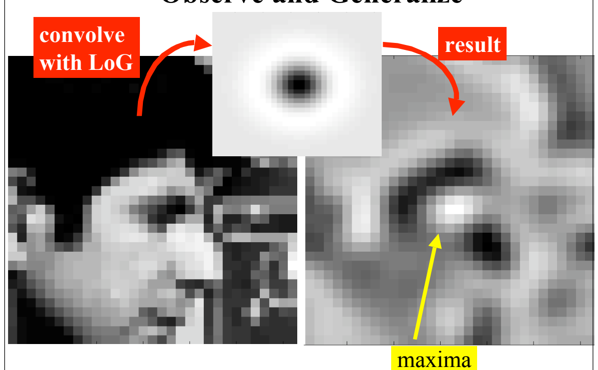
LoG Blob Finding

LoG filter extrema locates "blobs"
- maxima = dark blobs on light background
- minima = light blobs on dark background

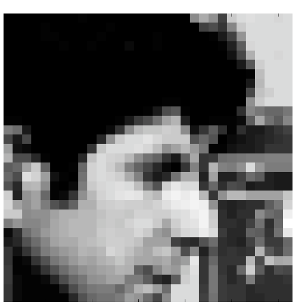
Scale of blob (size ; radius in pixels) is determined by the sigma parameter of the LoG filter.



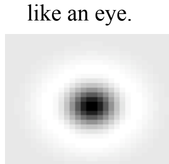
Observe and Generalize



Observe and Generalize



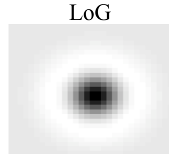
LoG looks a bit like an eye.



and it responds maximally in the eye region!

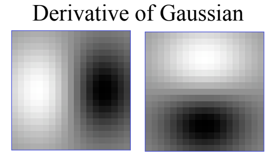
Observe and Generalize

LoG



Looks like dark blob on light background

Derivative of Gaussian

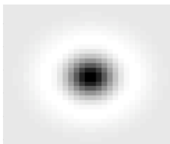


Looks like vertical and horizontal step edges

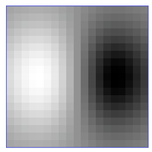
Recall: Convolution (and cross correlation) with a filter can be viewed as comparing a little "picture" of what you want to find against all local regions in the mage.

Observe and Generalize

Key idea: Cross correlation with a filter can be viewed as comparing a little "picture" of what you want to find against all local regions in the image.



Maximum response:
dark blob on light background
Minimum response:
light blob on dark background



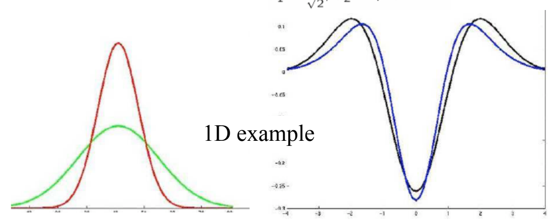
Maximum response:
vertical edge; lighter on left
Minimum response:
vertical edge; lighter on right

Efficient Implementation
Approximating LoG with DoG

LoG can be approximate by a Difference of two Gaussians (DoG) at different scales

$$\nabla^2 G_\sigma \approx G_{\sigma_1} - G_{\sigma_2}$$

Best approximation when:
$\sigma_1 = \frac{\sigma}{\sqrt{2}}, \sigma_2 = \sqrt{2}\sigma$



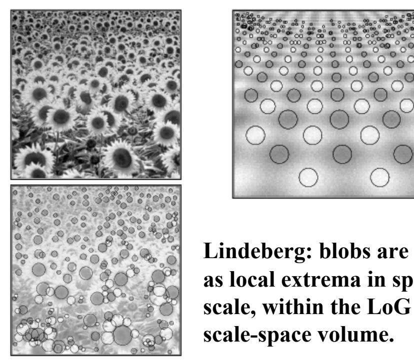
Efficient Implementation

LoG can be approximate by a Difference of two Gaussians (DoG) at different scales.

Separability of and cascadability of Gaussians applies to the DoG, so we can achieve efficient implementation of the LoG operator.

DoG approx also explains bandpass filtering of LoG (think about it. Hint: Gaussian is a low-pass filter)

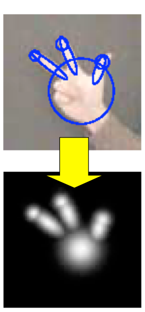
Back to Blob Detection

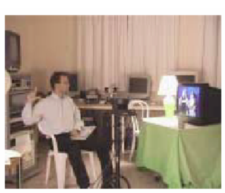


Lindeberg: blobs are detected as local extrema in space and scale, within the LoG (or DoG) scale-space volume.

Other uses of LoG:
Blob Detection

- Bretzner, Laptev, Lindeberg ``Hand-gesture recognition using multi-scale colour features, hierarchical features and particle filtering'', *Proc. Face and Gesture 2002*, Washington DC, pages 423--428. IEEE Computer Society Press. (PDF 159 kb)

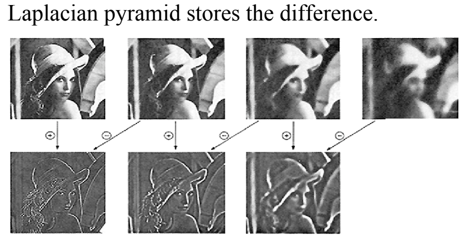




Gesture recognition for the ultimate couch potato

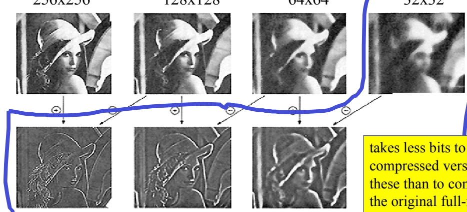
Other uses for LOG:
Image Coding

- Coarse layer of the Gaussian pyramid predicts the appearance of the next finer layer.
- The prediction is not exact, but means that it is **not** necessary to store all of the next fine scale layer.
- Laplacian pyramid stores the difference.



Robert Collins
CSE486

Other uses for LOG: Image Coding



takes less bits to store compressed versions of these than to compress the original full-res greyscale image

The Laplacian Pyramid as a Compact Image Code Burt, P., and Adelson, E. H., IEEE Transactions on Communication, COM-31:532-540 (1983).