Filtering - I

-Noise removal

-Edge Detection

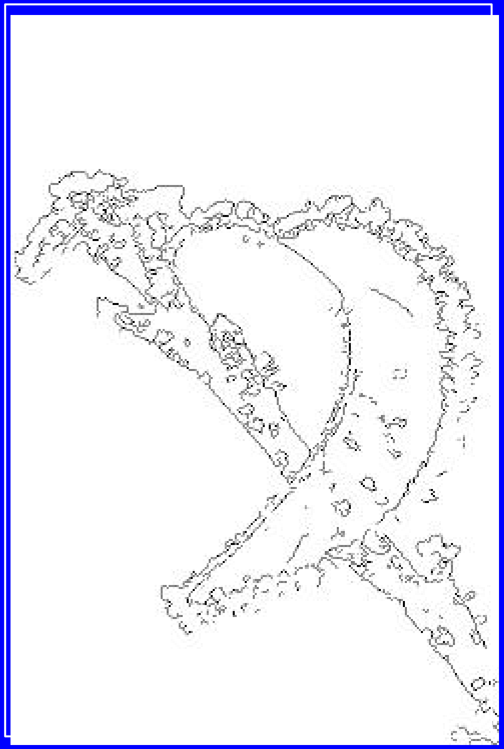
Edge Detection

Edge Detection

- Find edges in the image
- Edges are locations where intensity changes the most
- Edges can be used to represent a shape of an object

Edge Detection in Images

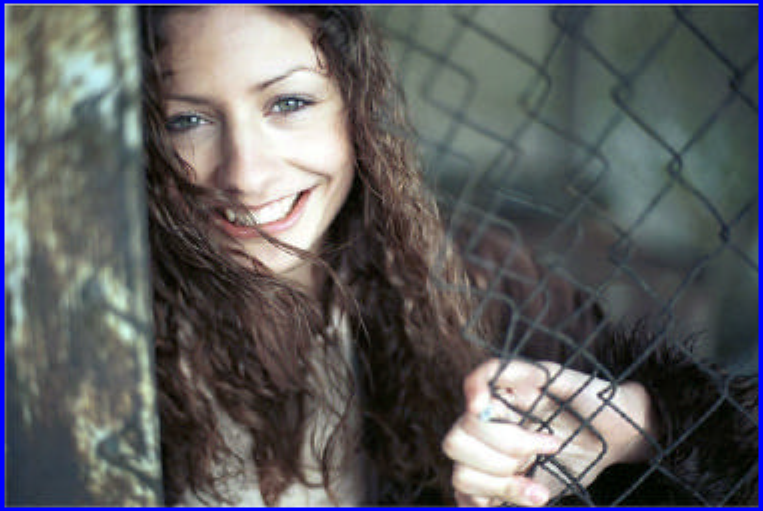
- Finding the contour of objects in a scene

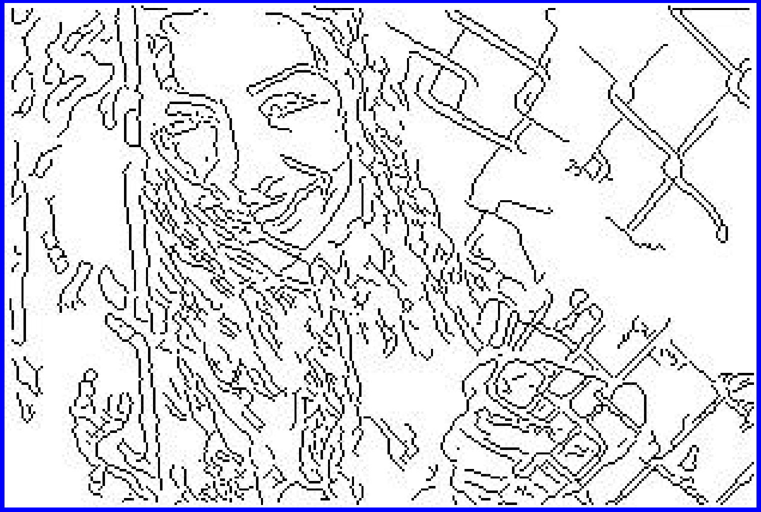


Edge Detection in Images

- What is an object?

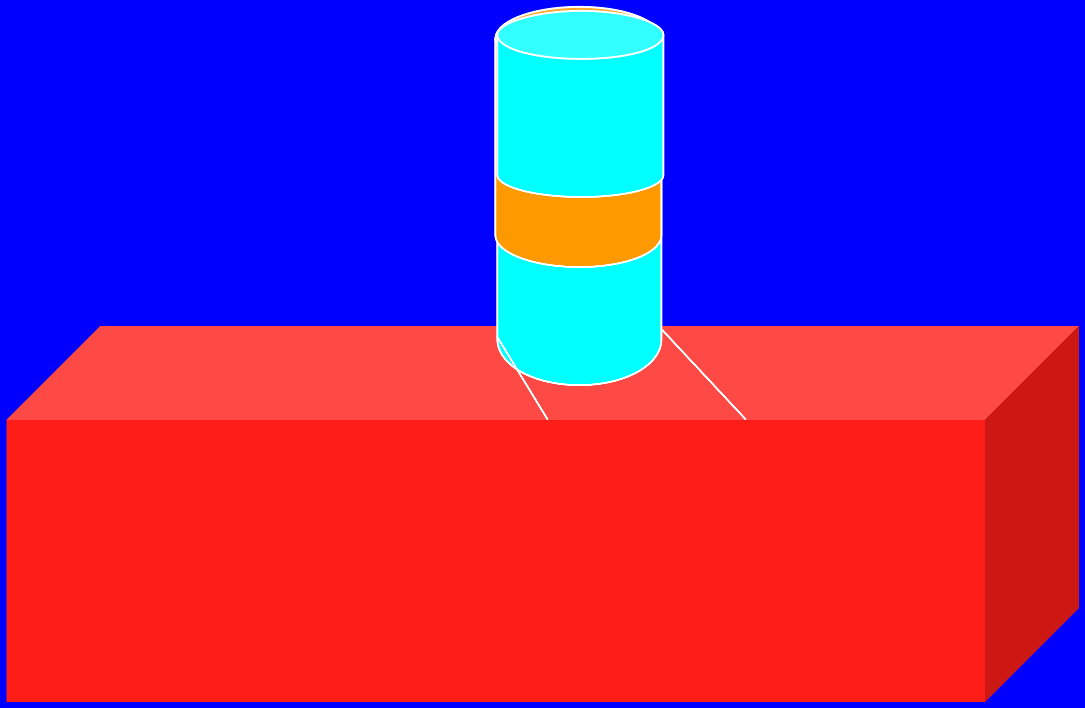
  It is one of the goals of computer vision to identify objects in scenes.





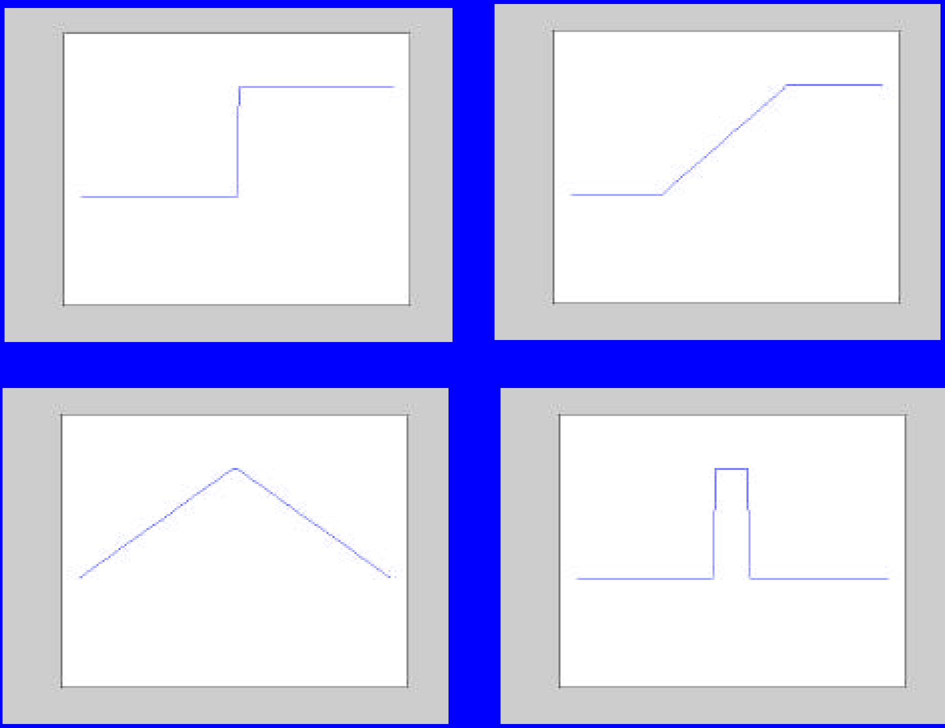
Edge Detection in Images

- Edges have different sources.



What is an Edge

- Lets define an edge to be a discontinuity in image intensity function.
- Edge Models
  - Step Edge
  - Ramp Edge
  - Roof Edge
  - Spike Edge



Detecting Discontinuities

- Discontinuities in signal can be detected by computing the derivative of the signal.
- If the signal is constant (over space), its derivative will be zero
- If there is a sharp difference in signal , then it will produce a high derivative value.

Differentiation and convolution

- Recall

$$\frac{\partial f}{\partial x} = \lim_{e \to 0} \left(\frac{f(x+e) - f(x)}{e} \right)$$

- Now this is linear and shift invariant, so must be the result of a convolution.

- We could approximate this as

$$\frac{\partial f}{\partial x} \approx \frac{f(x_{n+1}) - f(x)}{\Delta x}$$

(which is obviously a convolution with Kernel $\begin{bmatrix} 1 & -1 \end{bmatrix}$; it's not a very good way to do things, as we shall see)

Finite Difference in 2D

$$\frac{\partial f(x,y)}{\partial x} = \lim_{e \to 0} \left(\frac{f(x+e,y) - f(x,y)}{e} \right)$$

$$\frac{\partial f(x,y)}{\partial y} = \lim_{e \to 0} \left(\frac{f(x,y+e) - f(x,y)}{e} \right)$$

Definition

$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x_{n+1}, y_m) - f(x_n, y_m)}{\Delta x}$$

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

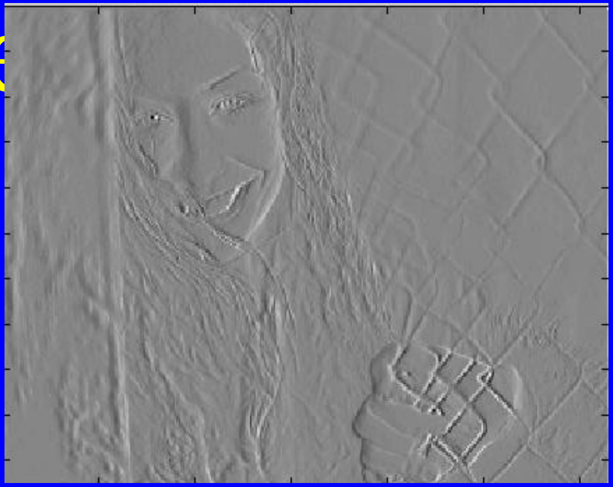
$$\frac{\partial f(x,y)}{\partial y} \approx \frac{f(x_n, y_{m+1}) - f(x_n, y_m)}{\Delta x}$$

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

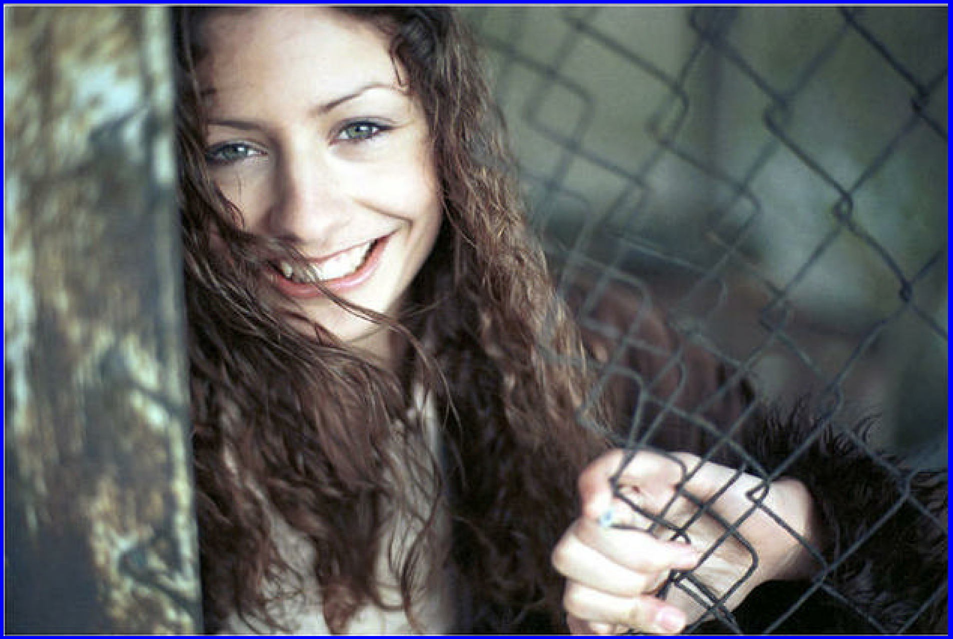
Discrete Approximation

Convolution Kernels

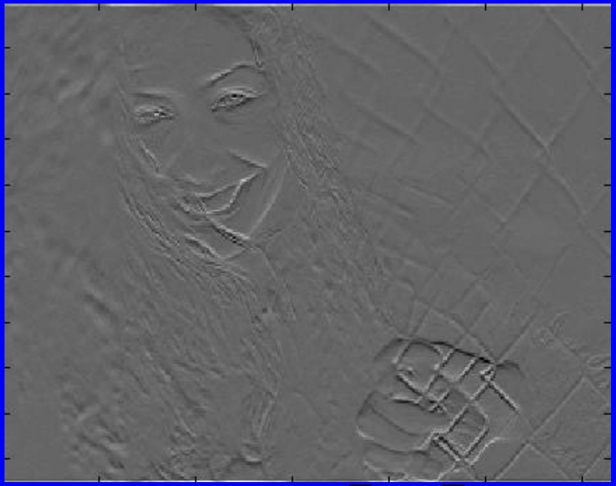
Finite differe



$$I_x = I * \begin{bmatrix} 1 & -1 \end{bmatrix}$$



I



$$I_y = I * \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Edge Detectors

- Prewit
- Sobel
- Roberts
- Marr-Hildreth (Laplacian of Gaussian)
- Canny (Gradient of Gaussian)
- Haralick (Facet Model)

Discrete Derivative

$$\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x)$$

$$\frac{df}{dx} = \frac{f(x) - f(x-1)}{1} = f'(x)$$

$$\frac{df}{dx} = f(x) - f(x-1) = f'(x)$$

(Finite Difference)

Discrete Derivative

$$\frac{df}{dx} = f(x) - f(x-1) = f'(x)$$ Left difference

$$\frac{df}{dx} = f(x) - f(x+1) = f'(x)$$ Right difference

$$\frac{df}{dx} = f(x+1) - f(x-1) = f'(x)$$ Center difference

Derivatives in Two Dimensions

$$f(x,y)$$

(partial Derivatives)

$$\frac{\partial f}{\partial x} = f_x = \lim_{\Delta x \to 0} \frac{f(x,y) - f(x-\Delta x, y)}{\Delta x}$$

$$\frac{\partial f}{\partial y} = f_y = \lim_{\Delta y \to 0} \frac{f(x,y) - f(x, y-\Delta y)}{\Delta y}$$

(f_x, f_y) Gradient Vector

$$\text{magnitude} = \sqrt{(f_x^2 + f_y^2)}$$

$$\text{direction} = q = \tan^{-1} \frac{f_y}{f_x}$$

$$\Delta^2 f = f_{xx} + f_{yy} = \text{Laplacian}$$

Derivatives of an Image(along x)

Derivative & average

$$\begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix} \qquad \begin{matrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{matrix}$$

$$f_x \qquad\qquad f_y$$

Prewit

$$I(x,y) = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix} \qquad I_x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 30 & 30 & 0 & 0 \\ 0 & 30 & 30 & 0 & 0 \\ 0 & 30 & 30 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Derivatives of an Image (along y)

$$I(x,y) = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix} \quad I_y = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Derivatives of an Image

P
Sobel

$$\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \qquad \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$$

$$f_x \qquad\qquad f_y$$

$$\begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix} \qquad \begin{matrix} 1 & 0 \\ 0 & -1 \end{matrix}$$

Roberts

$$f_x \qquad\qquad f_y$$

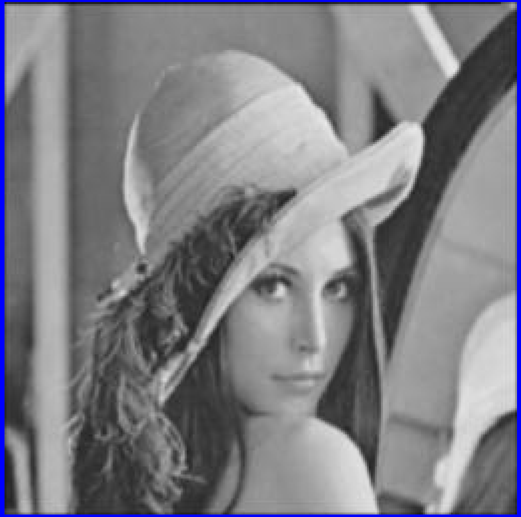
Detecting Edges in Image

- Sobel Edge Detector

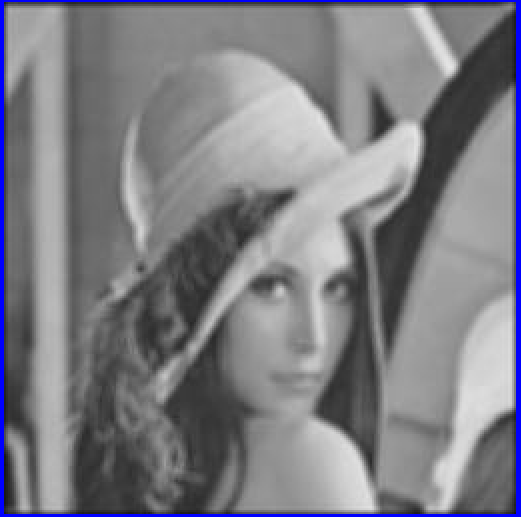
$$\begin{tikzcd}
 & \begin{bmatrix}1 & 0 & -1\\ 2 & 0 & -2\\ 1 & 0 & -1\end{bmatrix} \arrow[dr, "\frac{d}{dx}I"] & & & \\
\text{Image } I \arrow[ur, "*"] \arrow[dr, "*"'] & & \bigcirc \arrow[r, "\sqrt{\left(\frac{d}{dx}I\right)^2+\left(\frac{d}{dy}I\right)^2}"] & \boxed{\text{Threshold}} \arrow[r, "\text{Edges}"] & {} \\
 & \begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix} \arrow[ur, "\frac{d}{dy}I"'] & & &
\end{tikzcd}$$

Image Filter

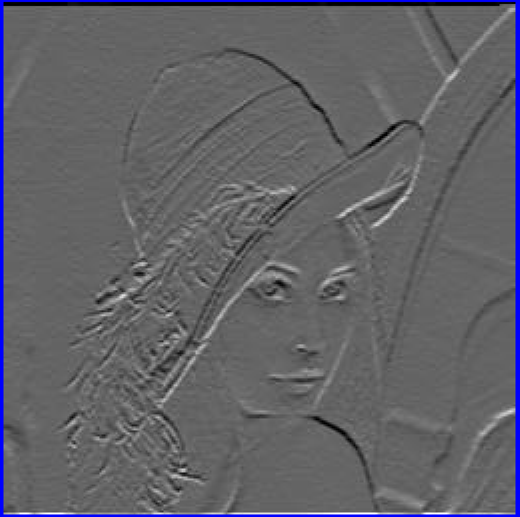
Original Image


Gaussian filter

$$\begin{bmatrix} 0.0113 & 0.0838 & 0.0113 \\ 0.0838 & 0.6193 & 0.0838 \\ 0.0113 & 0.0838 & 0.0113 \end{bmatrix}$$


Average filter

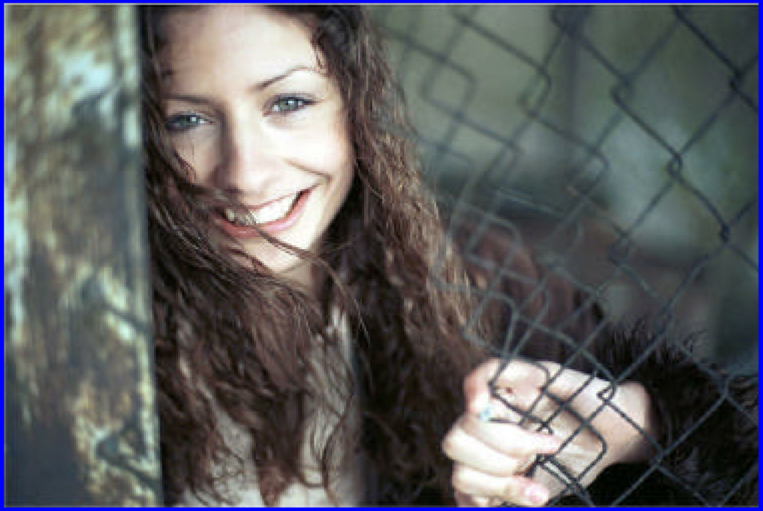
$$\tfrac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$


Sobel filter

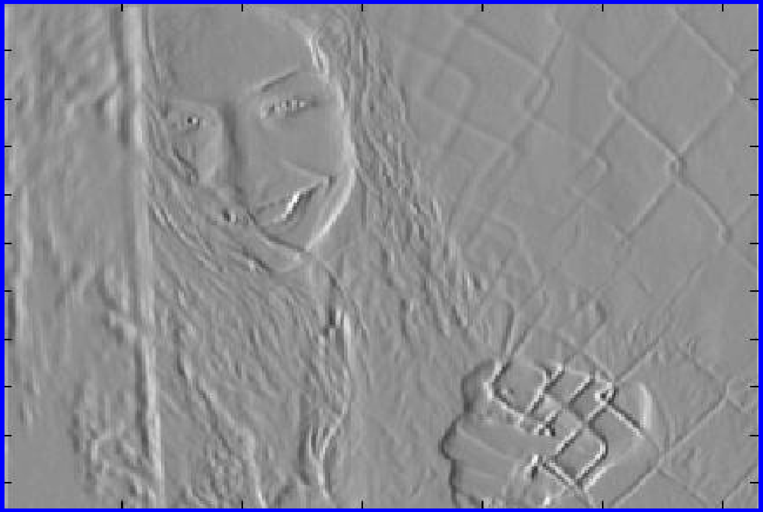
$$\tfrac{1}{8}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Sobel Edge Detector

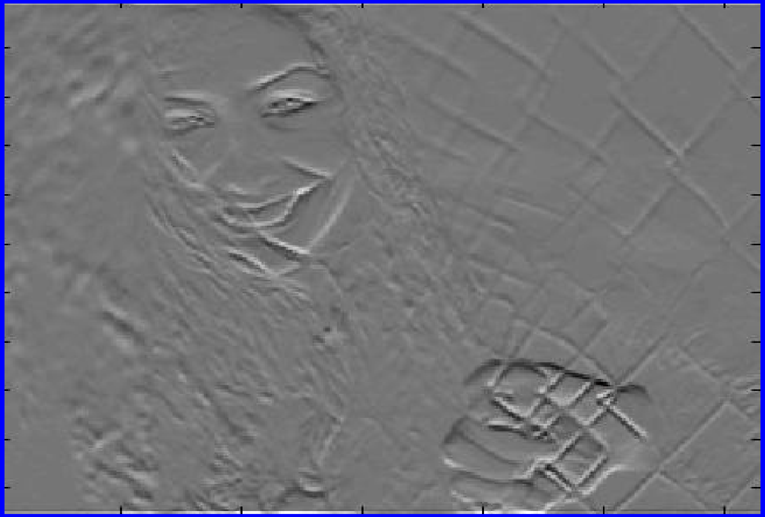
I



$\frac{d}{dx}I$

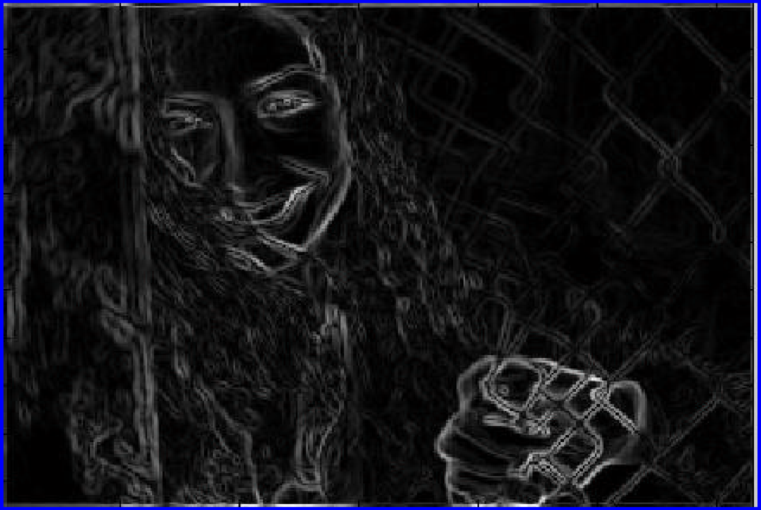


$\frac{d}{dy}I$

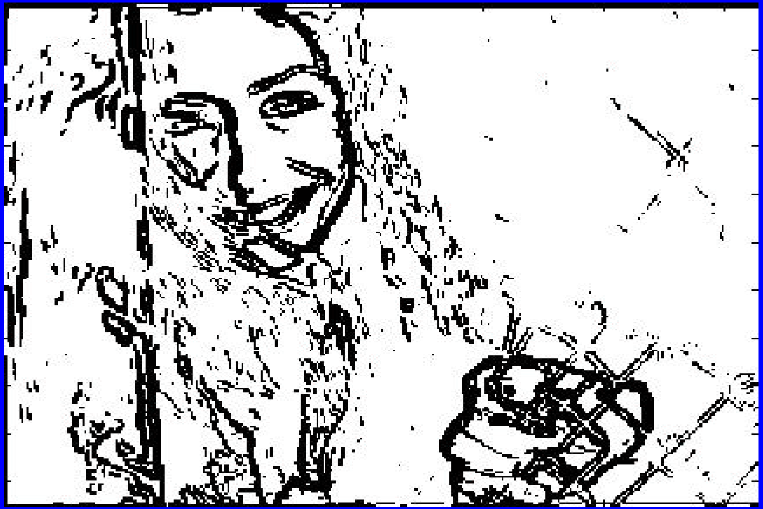


Sobel Edge Detector

$$\Delta = \sqrt{\left(\frac{d}{dx}I\right)^2 + \left(\frac{d}{dy}I\right)^2}$$

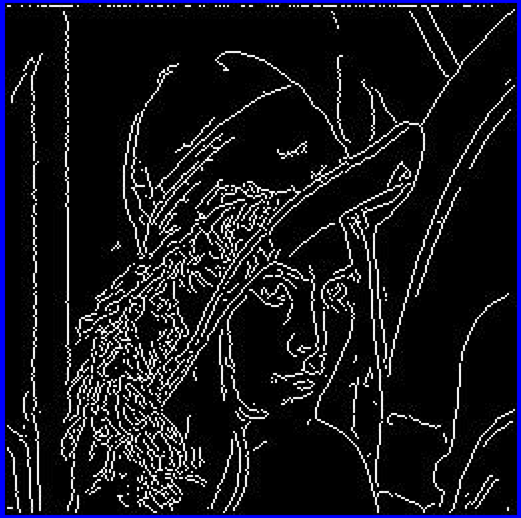


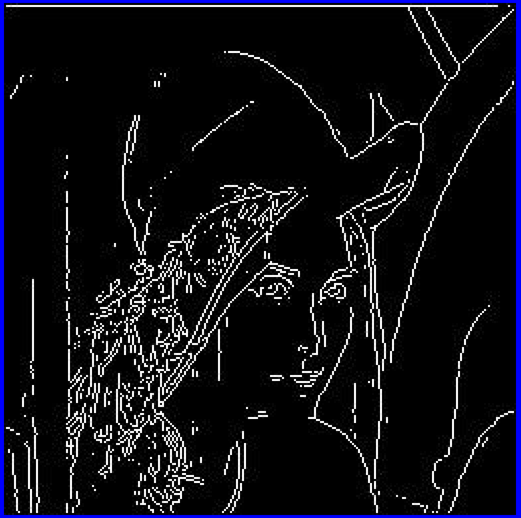
I



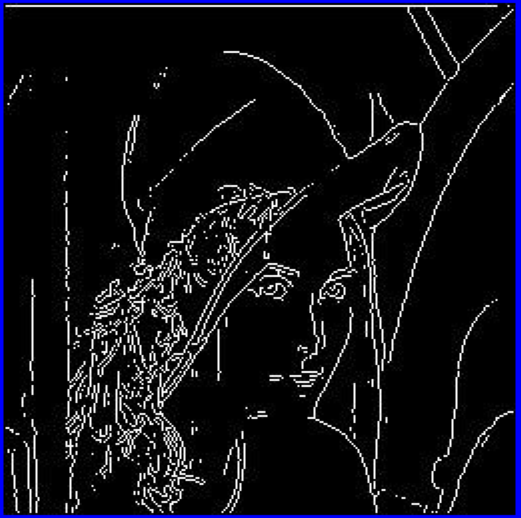
$\Delta \geq Threshold = 100$

Edge Detector


Canny edge detector using gaussian filter


Prewitt edge detector using Prewitt filter

$$\frac{1}{6}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$


Sobel edge detector using Sobel filter

$$\frac{1}{8}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

High-boost Filtering

An image with sharp features implies that there will be high frequency component, which are ignored by averaging filter(A lowpass filter – which allows only low frequencies to go through).

Highpass = Original – lowpass

High-boost = A (original) – lowpass

= (A – 1) original + original – lowpass

= (A – 1) original + highpass

A can be chosen as 1.1, 1.15, ….. 1.2 (beyond that results no good)

Second Order Derivative operators

The laplacian is a second spatial derivative of an image.

It is given by

$$\Delta^2 f = f_{xx} + f_{yy} = \text{Laplacian}$$

Laplacian Operator

One of the masks can be used to compute the Laplacian

$$\begin{array}{ccc} 0 & -1 & 0 \\ -1 & 4 & -1 \\ -1 & -1 & -1 \end{array} \qquad \begin{array}{ccc} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{array}$$

Higher order derivatives are more sensitive to noise.

Edge Detection

- To detect edges compute derivative of an image (gradient)
- If gradient magnitude is high at pixel, intensity change is maximum, that is an edge pixel
- If at a pixel the first derivative is maximum, the Laplacian (second derivative) would be zero and that point can be declared an edge pixel.

Edge Detection in Noisy Images

- Images contain noise, need to remove noise by averaging, or weighted averaging
- Filter the image by weighted averaging (Gaussian)
- Find Laplacian of image
- Detect zero-crossings

Laplacian of Gaussian

- Filter the image by weighted averaging (Gaussian)
- Find Laplacian of image
- The image can be convolved with Laplacian of Gaussian .
- Detect zero-crossings
- Verify that gradient Magnitudes are large here.

Marr and Hildreth Edge Operator

- Smooth by Gaussian

$$S = G_s * I \qquad G_s = \frac{1}{\sqrt{2p}\, s} e^{-\frac{x^2+y^2}{2s^2}}$$

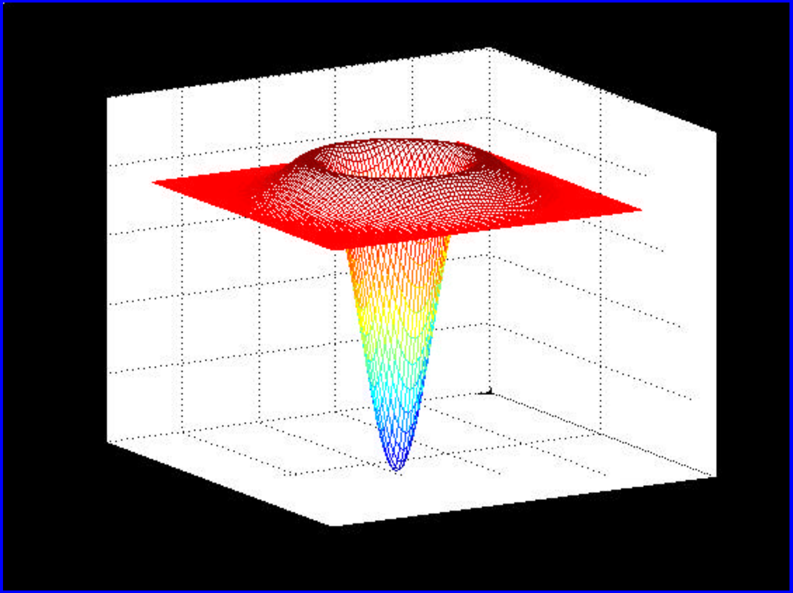
- Use Laplacian to find derivatives

$$\Delta^2 S = \frac{\partial^2}{\partial x^2} S + \frac{\partial^2}{\partial y^2} S$$

Marr and Hildreth Edge Operator

$$\Delta^2 S = \Delta^2 \left(G_s * I \right) = \Delta^2 G_s * I$$

$$\Delta^2 G_s = -\frac{1}{\sqrt{2p}\, s^3} \left(2 - \frac{x^2 + y^2}{s^2} \right) e^{\frac{x^2 + y^2}{2s^2}}$$



Marr and Hildreth Edge Operator

$$\Delta^2 G_s = -\frac{1}{\sqrt{2p}\, s^3}\left(2 - \frac{x^2 + y^2}{s^2}\right) e^{-\frac{x^2+y^2}{2s^2}}$$

Y

X

0.0008	0.0066	0.0215	0.031	0.0215	0.0066	0.0008
0.0066	0.0438	0.0982	0.108	0.0982	0.0438	0.0066
0.0215	0.0982	0	-0.242	0	0.0982	0.0215
0.031	0.108	-0.242	-0.7979	-0.242	0.108	0.031
0.0215	0.0982	0	-0.242	0	0.0982	0.0215
0.0066	0.0438	0.0982	0.108	0.0982	0.0438	0.0066
0.0008	0.0066	0.0215	0.031	0.0215	0.0066	0.0008

Response of L-o-G is positve on one side of an edge and negative on another.

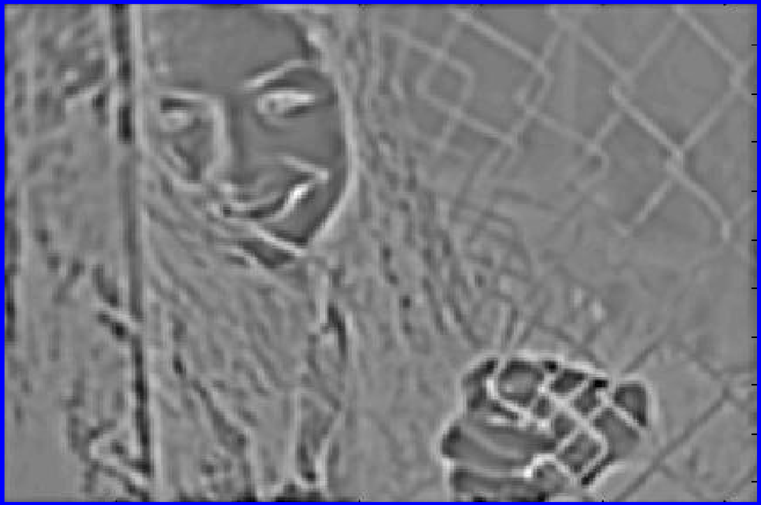
Adding some percentage of this response to the original image yields a picture with sharpened edges.

See also the following link:

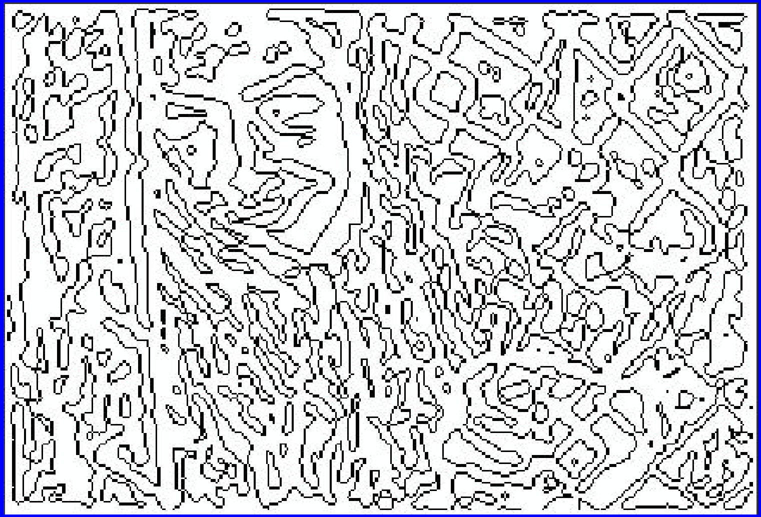
http://www.cse.secs.oakland.edu/isethi/visual/coursenotes_files/edge.pdf

Marr and Hildreth Edge Operator

$$\begin{tikzcd}[column sep=large]
\text{Image } I \arrow[r] & \boxed{* \Delta^2 G_s} \arrow[r, "\Delta^2 G_s * I"] & \boxed{\text{Zero Crossings Detection}} \arrow[r] & \text{Edge Image}
\end{tikzcd}$$



$\Delta^2 G_s * I$



Zero Crossings

Scale Space

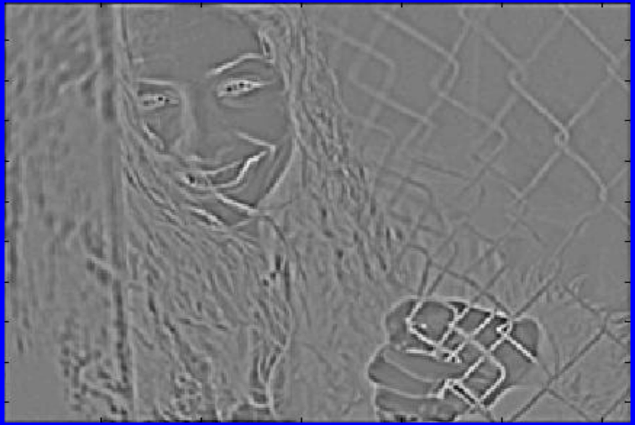
The term scale refers to the width of the Gaussian function.

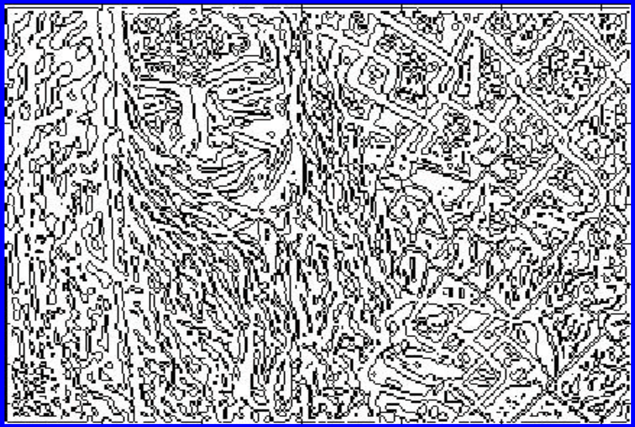
By changing the width

we can control the smoothing

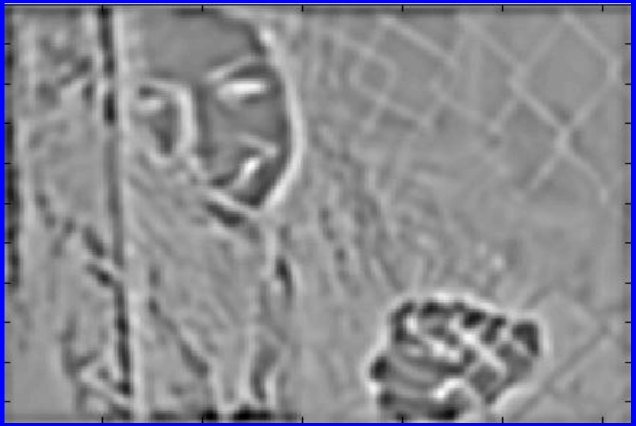
and hence the edge detection scale.

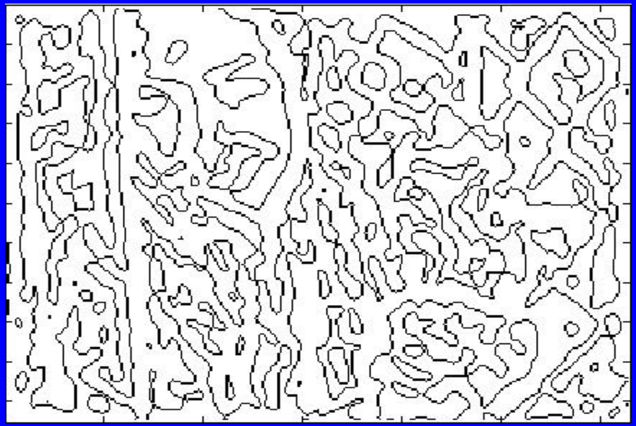
$s = 1$



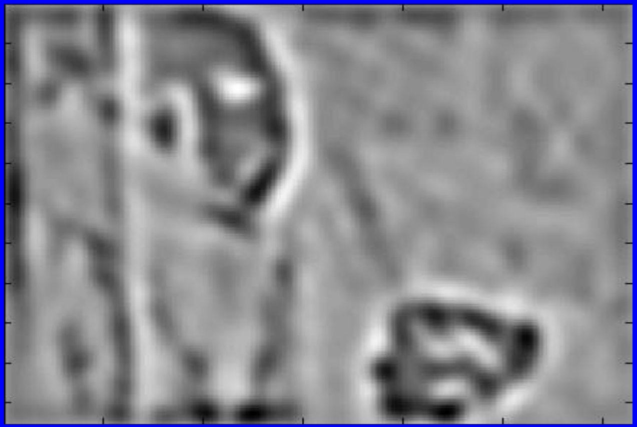


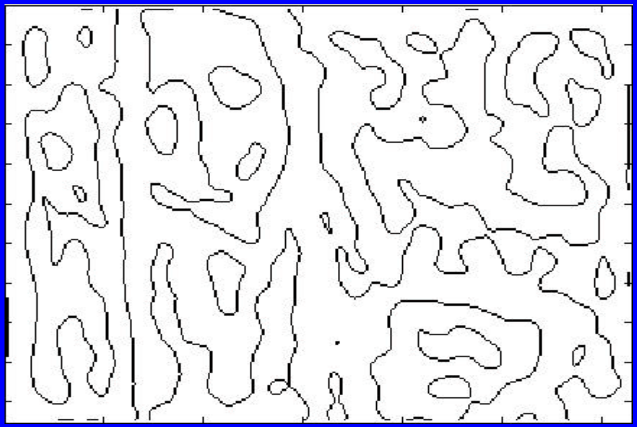
$s = 3$





$s = 6$





Separability of Gaussian

$$h(x, y) = f(x, y) * g(x, y)$$

Requires n^2 multiplications for a n by n mask, for each pixel.

$$h(x, y) = (f(x, y) * g(x)) * g(y)$$

This requires $2n$ multiplications for a n by n mask, for each pixel.

Separability of Laplacian of Gaussian

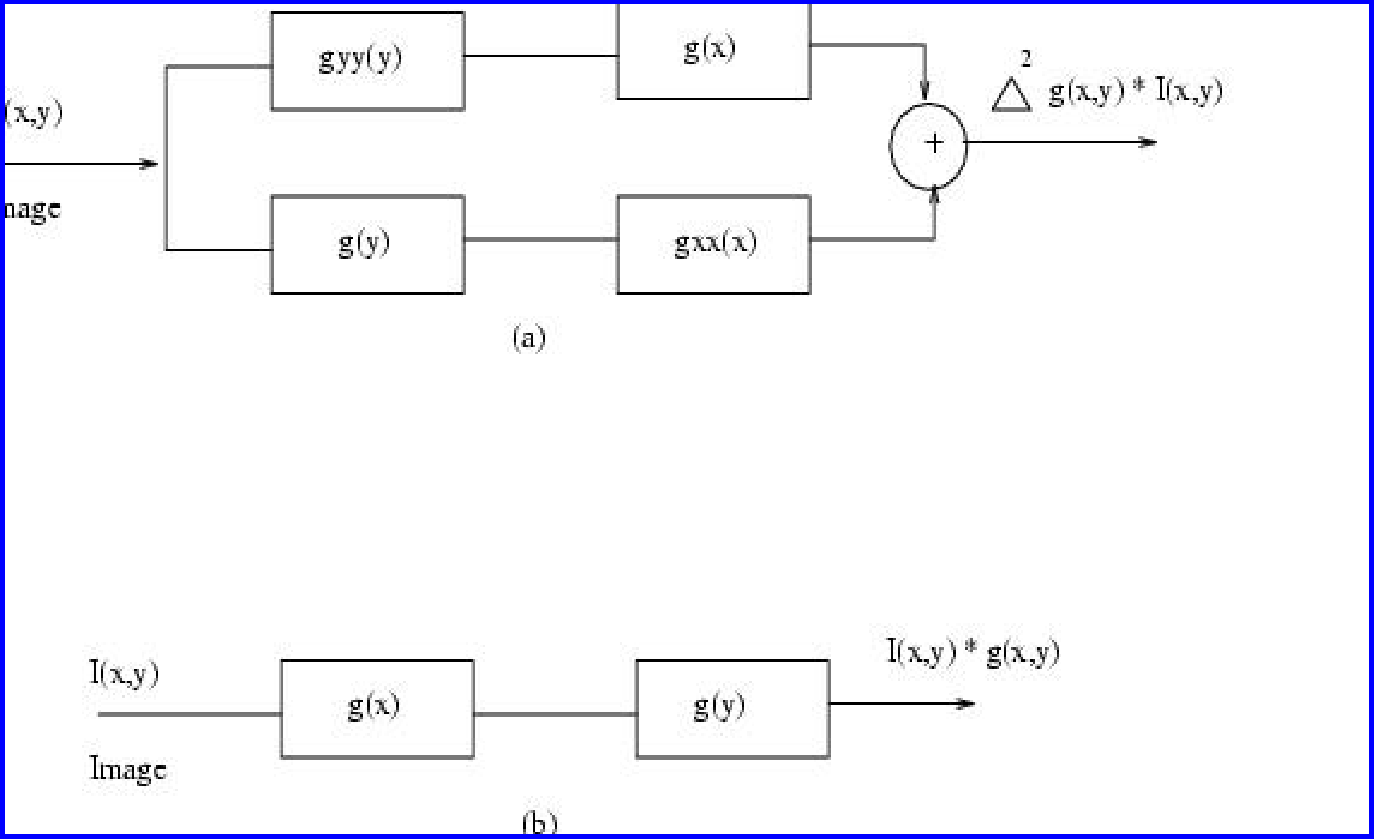
$$h(x,y) = f(x,y) * \Delta^2 g(x,y)$$

Requires n^2 multiplications for a n by n mask, for each pixel.

$$h(x,y) = (f(x,y) * g_{xx}(x)) * g(y) + (f(x,y) * g_{yy}(y)) * g(x)$$

This requires $4n$ multiplications for a n by n mask, for each pixel.

Separability



Decomposition of LG into four 1-D convolutions

- Convolve the image with a second derivative of Gaussian mask $g_{yy}(y)$ along each column
- Convolve the resultant image from step (1) by a Gaussian mask *g(x)* along each row. Call the resultant image I^x.
- Convolve the original image with a Gaussian mask, $g(y)$ along each column
- Convolve the resultant image from step (3) by a second derivative of Gaussian mask $g_{xx}(x)$ along each row. Call the resultant image I^y.
- Add I^x and I^y.

Laplacian and the second Directional Derivative and the direction of Gradient

$$\Delta^2 f = f_{xx} + f_{yy} = f_q^{''} + f_n^{''}$$

$$f_q^{'} = f_x \cos q + f_y \sin q$$

$$f_q^{''} = (f_{xx} \cos q + f_{yx} \sin q)\cos q + (f_{xy} \cos q + f_{yy} \sin q)\sin q$$

$$f_q^{''} = f_{xx} \cos^2 q + f_{yy} \sin^2 q + 2 f_{xy} \cos q \sin q$$

$$f_n^{''} = f_{xx} \cos^2 n + f_{yy} \sin^2 n + 2 f_{xy} \cos n \sin n$$

$$f_n^{''} = f_{xx} \cos^2(q+90) + f_{yy} \sin^2(q+90) + 2 f_{xy} \cos(q+90)\sin(q+90)$$

$$f_n^{''} = f_{xx} \sin^2 q + f_{yy} \cos^2 q - 2 f_{xy} \cos q \sin q$$

Laplacian and the second Directional Derivative and the direction of Gradient

$$f_q^{''} = f_{xx}\cos^2 q + f_{yy}\sin^2 q + 2 f_{xy}\cos q \sin q$$

$$f_n^{''} = f_{xx}\sin^2 q + f_{yy}\cos^2 q - 2 f_{xy}\cos q \sin q$$

$$\Delta^2 f = f_{xx} + f_{yy} = f_q^{''} + f_n^{''}$$