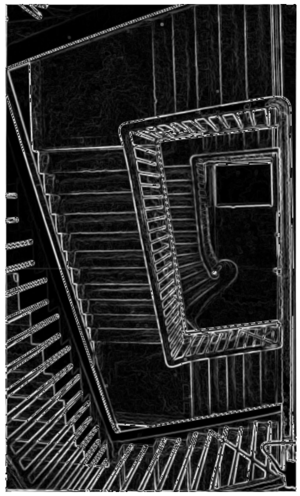
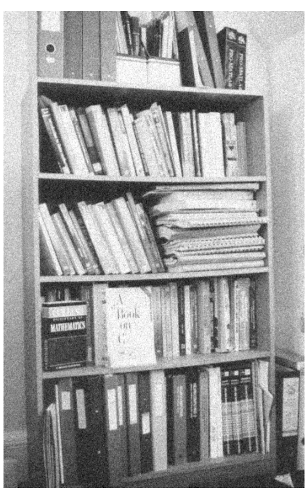
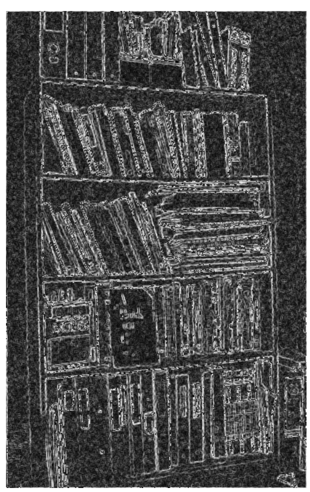


Edge magnitude

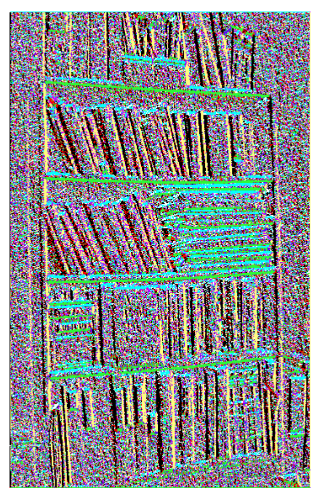
Edge orientation

Effect of presence of noise





Edge magnitude



Edge orientation

Edge Detection: boxcar operators

- Edge detector results are not as good when noise is added to the image
- To mitigate noise effects, preprocessing the image with mean spatial filters can be done
- Additionally, the size of the edge detection masks can be extended for noise mitigation
- An example of this method is to extend the Prewitt edge mask as follows:

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

- Can be rotated and used like the Prewitt for edge magnitude and direction
- Called *boxcar operators* and can be extended: 7x7, 9x9 and 11x11 are typical

Apply the Prewitt mask, with and without boxcar, to the image previously corrupted by noise.

Caiani 20/04/2012

- The Sobel operator can be extended in a similar manner:

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

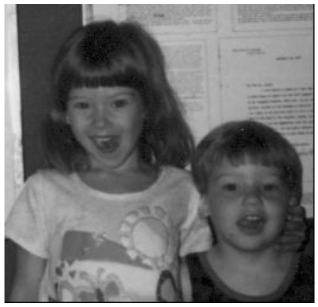
- Can be rotated and used for edge magnitude and direction as 3x3 Sobel

- *Truncated pyramid* operator can be obtained by approximating a linear distribution:

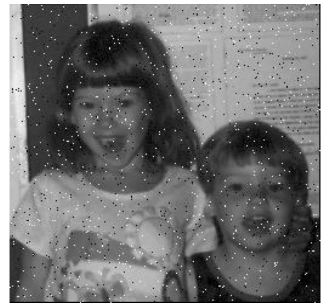
$$\begin{bmatrix} 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 2 & 2 & 0 & -2 & -2 & -1 \\ 1 & 2 & 3 & 0 & -3 & -2 & -1 \\ 1 & 2 & 3 & 0 & -3 & -2 & -1 \\ 1 & 2 & 3 & 0 & -3 & -2 & -1 \\ 1 & 2 & 2 & 0 & -2 & -2 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \end{bmatrix}$$

- This operator provides weights that decrease from the center pixel, which will smooth the result in a more natural manner
- Used like Sobel for magnitude and direction

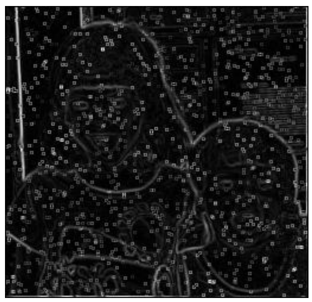
Caiani 20/04/2012



a) Original image



b) Image with added noise



e) Prewitt with a 3x3 mask

f) Prewitt with a 7x7 mask

Caiani 20/04/2012

Edge Detection: Laplacian operator

These are two-dimensional discrete approximations to the second derivative, implemented by applying *one* of the following convolution masks:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$$

The Laplacian masks are *rotationally symmetric*, which means edges at all orientations contribute to the result

Applied by selecting *one* mask and convolving it with the image

The sign of the result (positive or negative) from two adjacent pixel locations provides directional information, and tells us which side of the edge is brighter

If we increase the center coefficient by one it is equivalent to adding the original image to the edge detected image

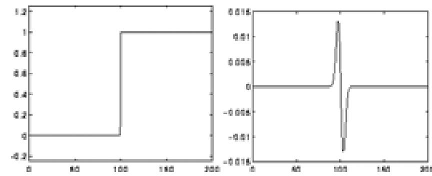
h = **fspecial**('laplacian',ALPHA) [0.2]

ALPHA (between 0 and 1) regulates the shape of the Laplacian, that is anyway 3x3.

Caiani 20/04/2012

BW = **edge**(I,'zerocross',THRESH,H)

specifies the zero-cross method to be applied using the mask H
The zero crossing detector looks for places in the Laplacian of an image where the value of the Laplacian passes through zero. Such points often occur at `edges' in images, but they also occur at places that are not as easy to associate with edges.
Zero crossings always lie on closed contours, and so the output from the zero crossing detector is usually a binary image with single pixel thickness lines showing the positions of the zero crossing points.



1) Load an image. Compare the performance of a chosen ***derivative filter*** *with that of the* ***Laplacian filter****.*
2) Corrupt the image with additive gaussian noise (0 mean, variance 128) and repeat the previous action.
3) Filter the image with a mean filter of 11x11, then re-apply the two filters and observe the effects on the contour detection.

Caiani 20/04/2012

Advanced Edge Detection

These edge detectors are considered to be advanced because they are algorithmic in nature

They are based on the idea that most edge detectors are too sensitive to noise and thus, by blurring the image prior to edge detection, it is possible to mitigate these noise effects.

Marr-Hildreth algorithm

Blur the image by convolving with a Gaussian mask:

$$Step\ 1: I' = G * I \qquad G(x,y) = e^{-\frac{x^2+y^2}{2\pi\sigma^2}}$$

Compute the second derivative:

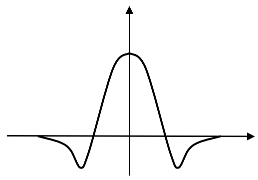
$$I'' = \nabla^2 I' = \nabla^2 (G * I) = (\nabla^2 G) * I$$

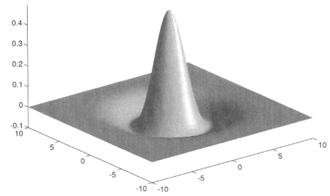
More Efficient (LoG)

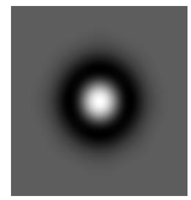
Threshold the image
Find the zero crossings as transition from black to white

Caiani 20/04/2012

Laplacian of a Gaussian (LoG)







The LoG as an image with white representing positive numbers, black negative numbers, and gray representing zero

BW = **edge**(I,'log',THRESH,SIGMA)

If THRESH=0, the contour is closed. SIGMA is the standard deviation in pixels (default=2);
filter dimensions are: N=ceil(SIGMA*3)*2+1

Filter the previous image with this filter, applying different parameters

Caiani 20/04/2012

- A common 5x5 mask that approximates the combination of the Gaussian and Laplacian into one convolution mask is as follows:

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

- An interesting aspect of the *LoG* operator is that it is believed to closely model biological vision systems
- G is smooth and localized in both the spatial and frequency domain, and wipes out all structures much smaller than the standard deviation

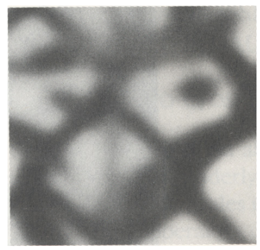
17X17 Mask

0	0	0	0	0	0	−1	−1	−1	−1	−1	0	0	0	0	0	0
0	0	0	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	0	0	0
0	0	−1	−1	−1	−2	−3	−3	−3	−3	−3	−2	−1	−1	−1	0	0
0	0	−1	−1	−2	−3	−3	−3	−3	−3	−3	−3	−2	−1	−1	0	0
0	−1	−1	−2	−3	−3	−3	−2	−3	−2	−3	−3	−3	−2	−1	−1	0
0	−1	−2	−3	−3	−3	0	2	4	2	0	−3	−3	−3	−2	−1	0
−1	−1	−3	−3	−3	0	4	10	12	10	4	0	−3	−3	−3	−1	−1
−1	−1	−3	−3	−2	2	10	18	21	18	10	2	−2	−3	−3	−1	−1
−1	−1	−3	−3	−3	4	12	21	24	21	12	4	−3	−3	−3	−1	−1
−1	−1	−3	−3	−2	2	10	18	21	18	10	2	−2	−3	−3	−1	−1
−1	−1	−3	−3	−3	0	4	10	12	10	4	0	−3	−3	−3	−1	−1
0	−1	−2	−3	−3	−3	0	2	4	2	0	−3	−3	−3	−2	−1	0
0	−1	−1	−2	−3	−3	−3	−2	−3	−2	−3	−3	−3	−2	−1	−1	0
0	0	−1	−1	−2	−3	−3	−3	−3	−3	−3	−3	−2	−1	−1	0	0
0	0	−1	−1	−1	−2	−3	−3	−3	−3	−3	−2	−1	−1	−1	0	0
0	0	0	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	0	0	0
0	0	0	0	0	0	−1	−1	−1	−1	−1	0	0	0	0	0	0

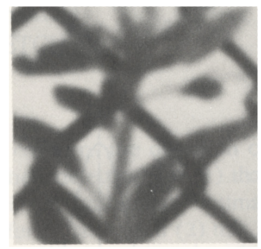
Zero-crossing detector

In the Marr-Hildreth algorithm, the starting point for the zero crossing detector is an image which has been filtered using the Laplacian of Gaussian filter.
The zero crossings that result are strongly influenced by the size of the Gaussian used for the smoothing stage of this operator. As the smoothing is increased then fewer and fewer zero crossing contours will be found, and those that do remain will correspond to features of larger and larger scale in the image.

Image



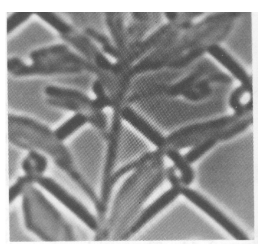
$\sigma=8$



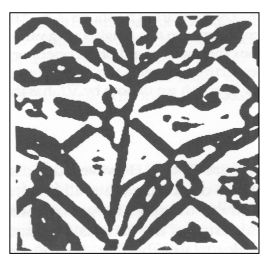
$\sigma=4$

Zero-crossing detector

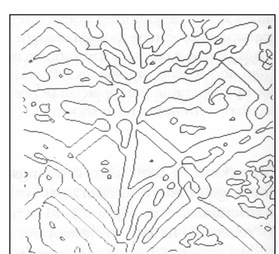
Image



After Convolution



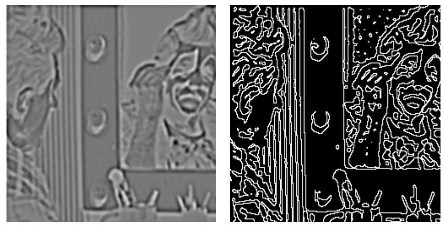
After Thresholding



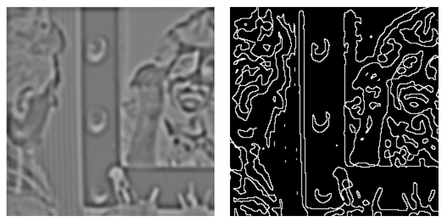
Zero Crossings

LoG filter with SD=1 (offset 128)

LoG filter with SD=2 (offset 128)



LoG filter with SD=3 (offset 128)



Advanced Edge Detection

CANNY

- Smooth the image with Gaussian Filter
- Compute the gradient magnitude and orientation using finite difference for partial derivatives (Roberts,Prewitt, Sobel)
- non-maxima suppression: a search is carried out to determine if the gradient magnitude assumes a local maximum in the gradient direction. Es: if the rounded gradient angle is zero degrees (i.e. north-south direction) the point will be considered to be on the edge if its intensity is greater than the intensities in the **west and east** directions, otherwise will be suppressed. A set of edge points ("thin edges“), in the form of a binary image, is obtained.

Original image (Lena) | Norm of the gradient | Thinning (non-maximum suppression)

- Use hysteresis thresholding (two thresholds – high and low) to detect and link edges.
- We begin by applying a high threshold. This marks out the edges we can be fairly sure are genuine. Starting from these, using the directional information derived earlier, edges can be traced through the image. While tracing an edge, we apply the lower threshold, allowing us to trace faint sections of edges as long as we find a starting point.

Canny guided by three parameters:

1. The width of the Gaussian mask used for smoothing
2. Two thresholds (T_1, T_2) are used for the tracking process
   - $T_1>T_2$
   - Tracking starts if ridge is higher than T_1.
   - Tracking continues until ridge falls below T_2.

Caiani 20/04/2012

Canny: smoothing with the Gaussian

- Performed at the beginning
- If the width of the filter is large
  - Sensitivity to noise is less
  - Fine details might get lost
  - Localization error of detected edges is higher

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

Original

Canny with σ=1

Canny with σ=2

BW = **edge**(I,'canny',THRESH,SIGMA)

where THRESH : [T1 T2], with T1<T2. If a scalar S is given: [0.4*S S]

Try Canny filter on a specified ROI, focusing on a specific goal for detection.

Caiani 20/04/2012

Edge Detector Performance

- Objective and subjective evaluations can be useful
- Objective metrics allow us to compare different techniques with fixed analytical methods
- Subjective methods often have unpredictable results

- To develop a performance metric for edge detection operators, we need to consider the types of errors that can occur and define what constitutes success.
- A successfull edge detector should have:
  - Good detection: responds to edge, not noise.
  - Good localization: detected edge near true edge.
  - Single response: one per edge.

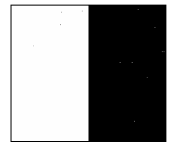
Edge Detector Performance

Possible parameters for metric

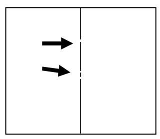
- Probability of false edges
- Probability of missing edges
- Error in estimation of edge angle
- Mean square distance of the edge estimate from true edge
- Tolerance of distorted edges and other features (corners, junctions, etc.)

Pratt's Figure of Merit (FOM):

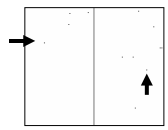
- Pratt first considered the types of errors that can occur with edge detection
- Types of errors:
  - *missing valid edge points*
  - *classifying noise as valid edge points*
  - *smearing of edges*



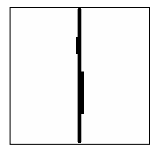
a) Original image



b) Missed edge points



c) Noise misclassified as edge points



d) Smeared edge

- If these errors do not occur, we can say that we have achieved success

Caiani 20/04/2012

The Pratt FOM, is defined as follows: $$FOM = \frac{1}{\max(I_A, I_I)} \sum_{i=1}^{I_A} \frac{1}{1+d_i \alpha^2}$$

I_A: number of detected edges in the image

I_I: number of actual edges in the image

d: distance between the ideal and detected edges

α: a constant to penalize displace edges

1. *City block distance*, four connectivity: $d = |r_1 - r_2| + |c_1 - c_2|$
2. *Chessboard distance,* 8-connectivity: $d = \max(|r_1 - r_2|, |c_1 - c_2|)$
3. *Euclidean distance*, physical distance: $d = \sqrt{(r_1 - r_2)^2 + (c_1 - c_2)^2}$

- For this metric, FOM will be 1 for a perfect edge
- Normalizing to the maximum of the ideal and found edge points guarantees a penalty for smeared edges or missing edge points
- In general, this metric assigns a better rating to smeared edges than to offset or missing edges

Caiani 20/04/2012

Given the following image array, find the Figure of Merit for the following found edge points, designated by 1's, in a), b), and c). Let $\alpha = 0.5$, and use the city block distance measure. We assume that actual edge in the locations where the line appears, that is, at the 100's.

$$\text{Image Array} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 100 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{a)} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \text{b)} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \text{c)} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{a) } FOM = \frac{1}{I_N} \sum_{i=1}^{I_F} \frac{1}{1+\alpha d_i^2} = \frac{1}{3}\left[\frac{1}{1+0.5(0)^2} + \frac{1}{1+0.5(0)^2} + \frac{1}{1+0.5(0)^2}\right] = 1$$

$$\text{b) } FOM = \frac{1}{I_N} \sum_{i=1}^{I_F} \frac{1}{1+\alpha d_i^2} = \frac{1}{6}\left[\frac{1}{1+0.5(0)^2} + \frac{1}{1+0.5(0)^2} + \frac{1}{1+0.5(0)^2} + \frac{1}{1+0.5(1)^2} + \frac{1}{1+0.5(1)^2} + \frac{1}{1+0.5(1)^2}\right] \approx 0.8333$$

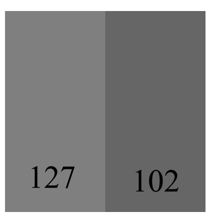
$$\text{c) } FOM = \frac{1}{I_N} \sum_{i=1}^{I_F} \frac{1}{1+\alpha d_i^2} = \frac{1}{1+0.5(1)^2} + \frac{1}{1+0.5(1)^2} + \frac{1}{1+0.5(1)^2} + \frac{1}{1+0.5(2)^2}\Bigg] \approx 0.5833$$

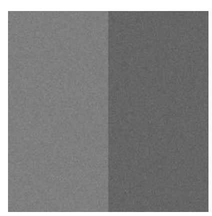
With result a), we find a perfect edge. In result b), we see that a smeared edge provides us with about 83%, and an offset edge in c) gives us about 58%. Note that the α parameter can be adjusted to determine the penalty for offset edges

Caiani 20/04/2012

Pratt's Figure of Merit: noise dependence

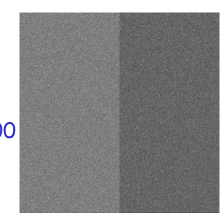
Original test image



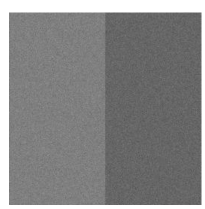


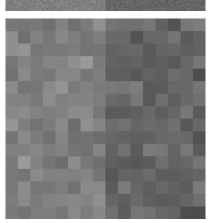
Test image with added Gaussian noise with a variance of 25

Test image with added Gaussian noise, variance 100



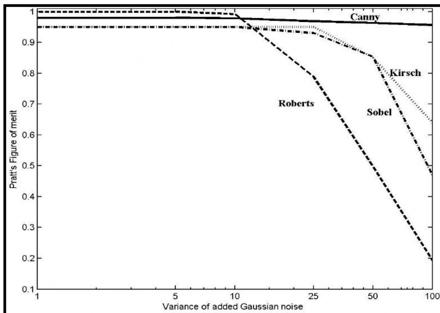
Gaussian noise variance of 50





A 16x16 subimage to show that the edge is not as easy to find at the pixel level

Caiani 20/04/2012



As the noise variance increases the Canny has the best performance