

Edge Detection

Outline

Part 1

- **Introduction**
- **Local Edge Operators**
 - Gradient of image intensity
 - Robert, Sobel and Prewitt operators
 - Second Derivative Operators
 - Laplacian of Gaussian

Part 2

- **Marr-Hildreth Edge Detector**
- **Multiscale Processing**
- **Canny Edge Detector**
- **Edge Detection Performance**

References:

- D. Huttenlocher, “Edge Detection” (<http://www.cs.wisc.edu/~cs766-1/readings/edge-detection-huttenlocher.ps>)
- R. Gonzalez, R. Woods, “Digital Image Processing”, Addison Wesley, 1993.
- D. Marr, E. Hildreth, “Theory of Edge detection”, Proc. R. Soc. Lond., B. 207, 187-217, 1980.

Edge Detection

Introduction

Initial considerations

Edges are significant local intensity changes in the image and are important features to analyse an image. They are important clues to separate regions within an object or to identify changes in illumination or colour. They are an important feature in the early vision stages in the human eye.

There are many different geometric and optical physical properties in the scene that can originate an edge in an image.

Geometric:

- Object boundary: *eg* a discontinuity in depth and/or surface colour or texture
- Surface boundary: *eg* a discontinuity in surface orientation and/or colour or texture
- Occlusion boundary: *eg* a discontinuity in depth and/or surface colour or texture

Optical:

- Specularity: *eg* direct reflection of light, such as a polished metallic surface
- Shadows: from other objects or part of the same object
- Interreflections: from other objects or part of the same object
- Surface markings, texture or colour changes

Edge Detection

Introduction

Goals of edge detection:

Primary

To extract information about the two-dimensional projection of a 3D scene.

Secondary

Image segmentation, region separation, object description and recognition, hand/eye tasks, ...

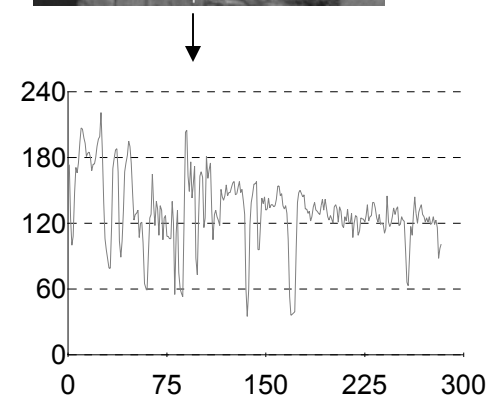
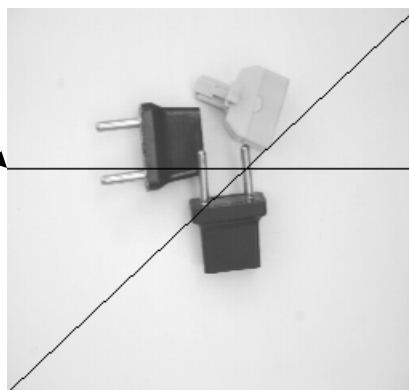
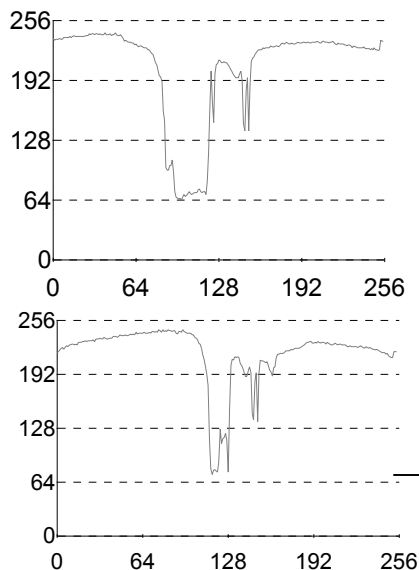
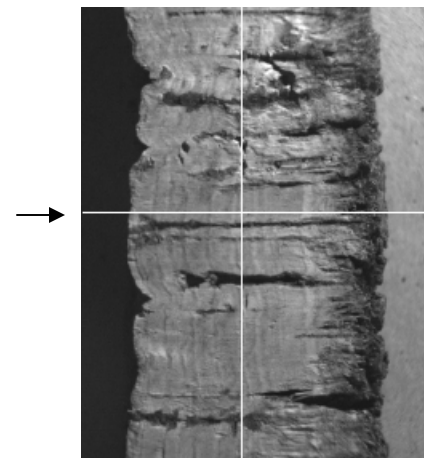
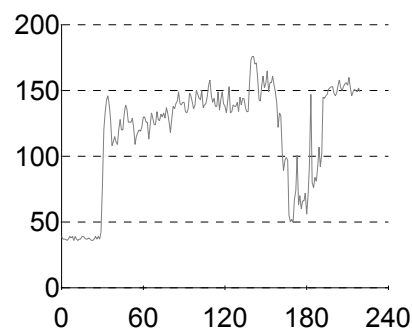
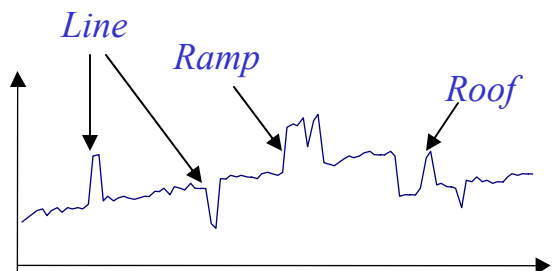
Types of Edges



Edge Detection

Introduction

Types of Edges



Edge Detection

Introduction

Definitions

Edge point : Point in an image with coordinates $[i, j]$ at the location of a significant local intensity change in the image.

Edge fragment : a small line segment about the size of a pixel, or as a point with an orientation attribute. The term *edge* is commonly used either for edge points or edge fragments.

Edge detector : Algorithm that produces a set of edges (edge points or edge fragments) from an image. Some edge detectors can also produce a direction that is the predominant tangent direction of the arc that passes through the pixel.

Contour: List of edges of the mathematical curve that models the list of edges.

Edge linking: Process of form an ordered list of edges from an unordered list.

Edge following: Process of searching the edge image (image with pixels labeled as edge) to determine contours.

Edge Detection

Local Edge Operators

Edge detection is essential the operation of detecting intensity variations. 1st and 2nd derivative operators give the information of this intensity changes. For instance, the first derivative of a step edge has a maximum located of the position of the edge.

The gradient of the image intensity is the vector: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]^t = [G_x, G_y]^t$

The magnitude and direction of the gradient are: $G = \sqrt{G_x^2 + G_y^2}$ $\theta = \tan^{-1} \frac{G_y}{G_x}$

Is common practice to approximate by absolute values:

$$G \approx |G_x| + |G_y| \quad \text{or} \quad G \approx \max(|G_x| + |G_y|)$$

Numerical approximations:

$$G_x \approx f[i, j+1] - f[i, j]$$

$$G_y \approx f[i, j] - f[i+1, j]$$

The corresponding convolution masks are:

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

↓

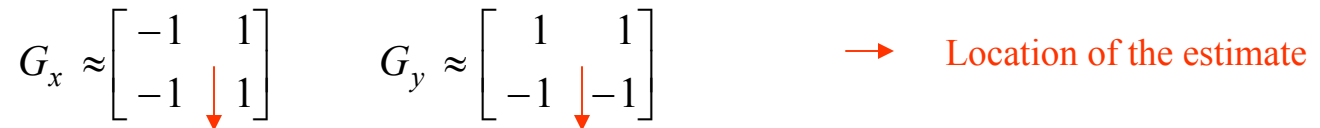
$$G_y \approx \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

→

Edge Detection

Local Edge Operators

When computing an approximation to the gradient, it is important the both directional derivatives produce a result at the same location. The unidimensional approximations, produce a result at different position. For this reason, square kernels are preferred, as:

$$G_x \approx \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \quad G_y \approx \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \quad \rightarrow \text{Location of the estimate}$$


Main steps for Edge detection

- **Filtering:** gradient operators are very sensitive to noise. It is important to improve the signal to noise ratio, by filtering previously the image. However, there is a trade-off between edge strength and noise reduction.
- **Enhancement:** To emphasize pixels with a significant change in local intensity, using a gradient operator.
- **Detection:** Label the edge points. Thresholding the gradient magnitude is a common technique.
- **Localization:** The location of the edge can be estimated with subpixel resolution.

Edge Detection

Local Edge Operators

Roberts Operator

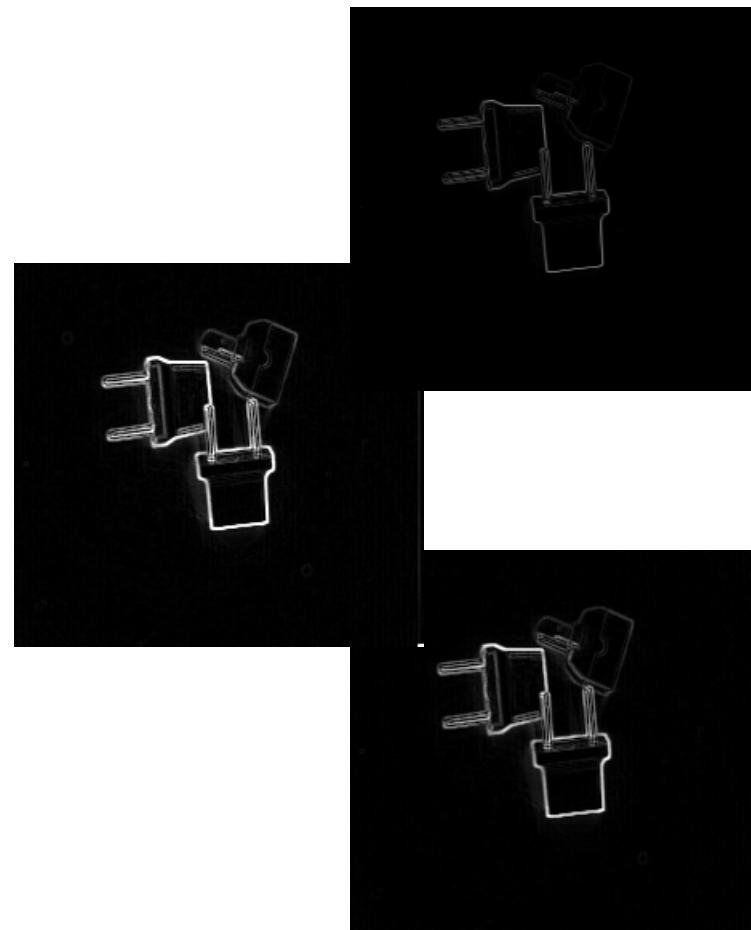
$$G_x \approx \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad G_y \approx \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Sobel Operator

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

Prewitt Operator

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

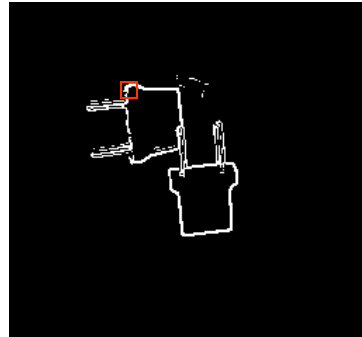


Edge Detection

Local Edge Operators

Sobel Operator

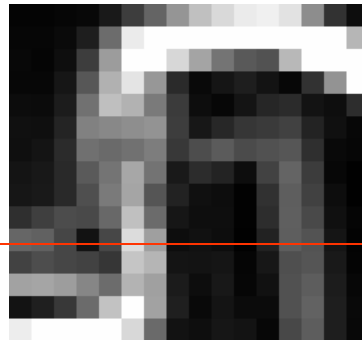
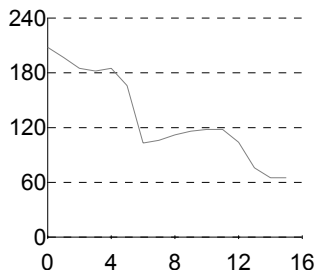
Detection



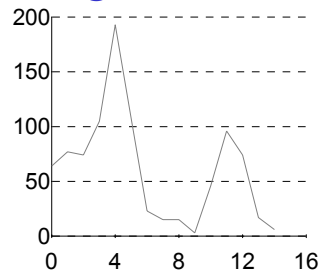
The result from the Sobel operator is thresholded to obtain the map of edge pixels. A simple threshold gives rise to discontinuities and several pixels wide. A better approach would be to find only the points with local maxima in gradient values.



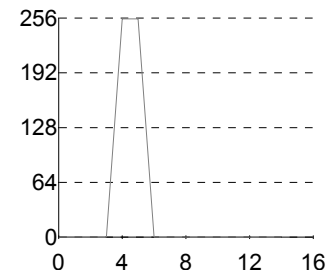
Input



Edge Enhancement

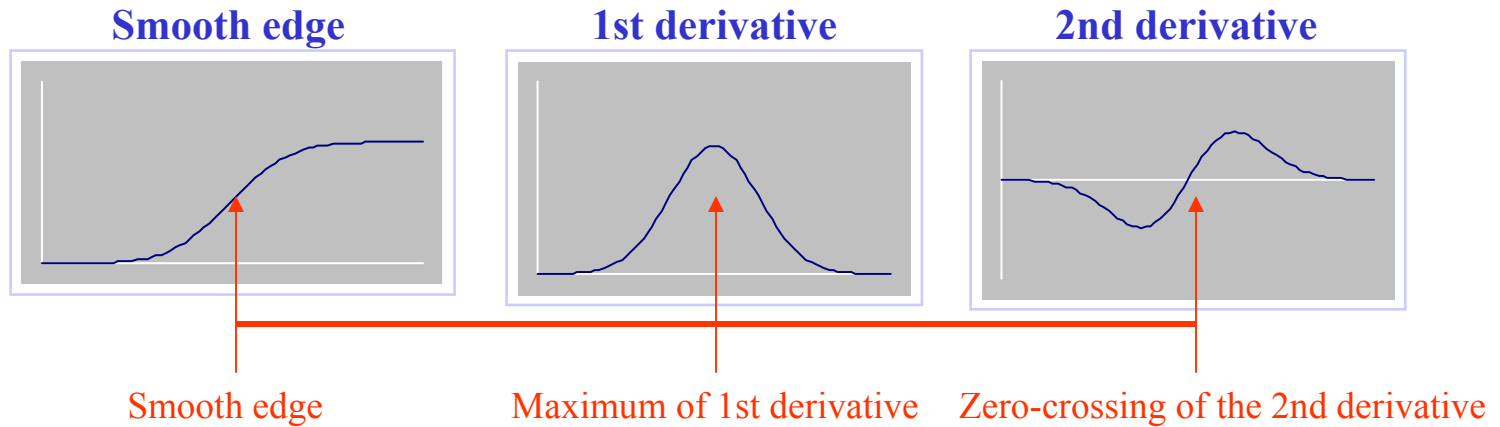


Detection



But the local maxima of the gradient is the zero crossing in the second derivative. To find the zero crossing of the second derivative of image intensity is another approach for edge detection.

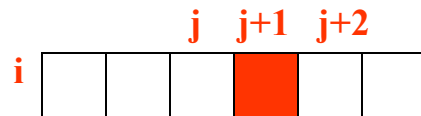
Laplacian operator



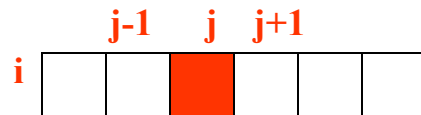
The **Laplacian** is the 2-D equivalent of the second derivative

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

1D numerical approximations:



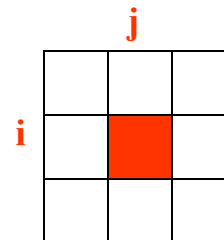
or



$$\begin{aligned} \frac{\partial^2 f}{\partial x^2} &= \frac{\partial}{\partial x} G_x = \frac{\partial}{\partial x} (f[i, j+1] - f[i, j]) = \frac{\partial}{\partial x} f[i, j+1] - \frac{\partial}{\partial x} f[i, j] = \\ &= (f[i, j+2] - f[i, j+1]) - (f[i, j+1] - f[i, j]) \\ &= f[i, j+2] - 2f[i, j+1] + f[i, j] \end{aligned}$$

$$\frac{\partial^2 f}{\partial x^2} = f[i, j+1] - 2f[i, j] + f[i, j-1]$$

2D numerical approximations:



$$\nabla^2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Edge Detection

Marr and Hildreth Edge Detector

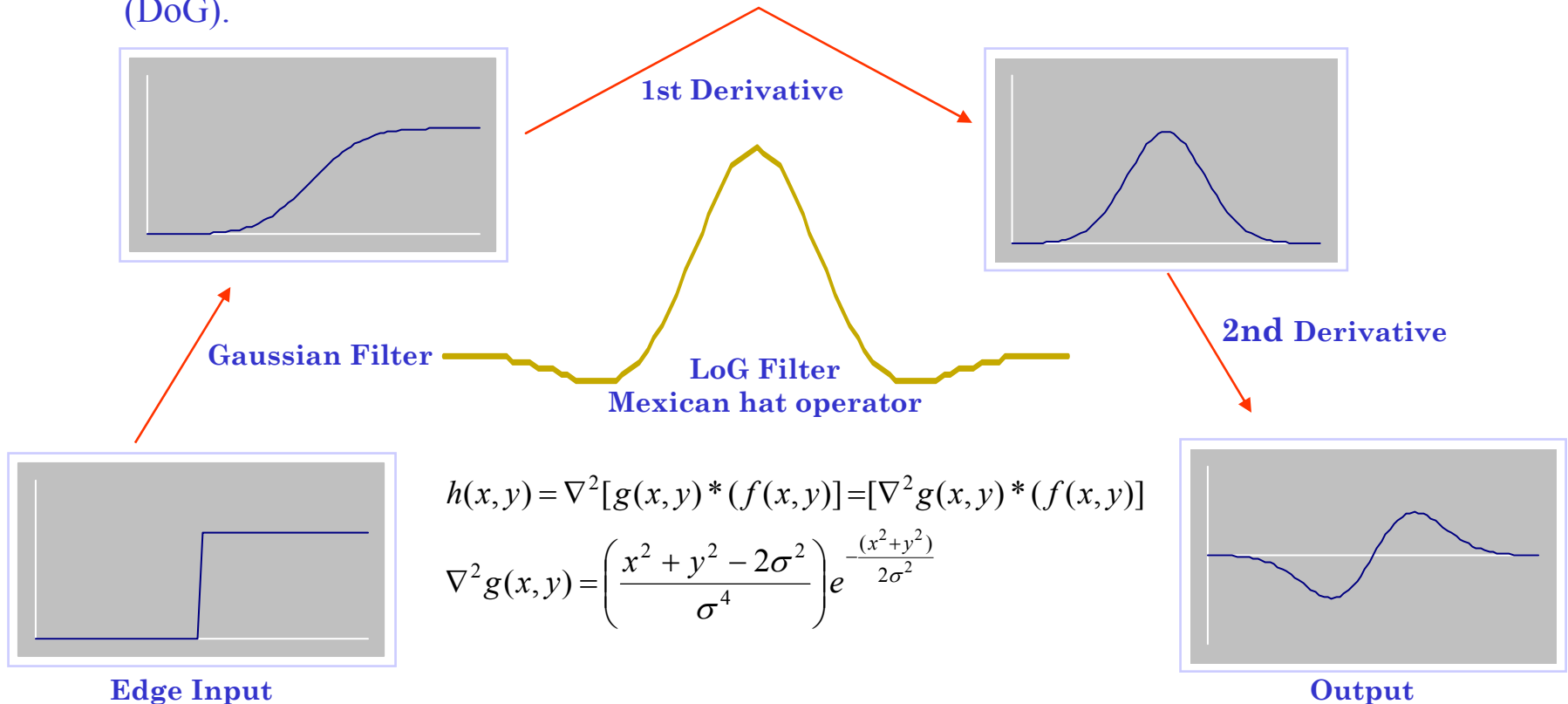
The derivative operators presented so far are not very useful because they are very sensitive to noise. To filter the noise before enhancement, Marr and Hildreth proposed a Gaussian Filter, combined with the Laplacian for edge detection. Is the Laplacian of Gaussian (LoG). The fundamental characteristics of LoG edge detector are:

- The smooth filter is Gaussian, in order to remove high frequency noise.
- The enhancement step is the Laplacian.
- The detection criteria is the presence of the zero crossing in the 2nd derivative, combined with a corresponding large peak in the 1st derivative. Here, only the zero crossings whose corresponding 1st derivative is above a specified threshold are considered.
- The edge location can be estimated using sub-pixel resolution by interpolation.

Edge Detection

Marr and Hildreth Edge Detector

The different phases of the LoG filter (or the Mexican hat operator) is illustrated below. The final result can be obtained either by first convolving with the Gaussian filter, and compute the Laplacian, or convolve directly with Laplacian of the Gaussian filter (kernels in the next slide). The LoG filter can be approximated by a difference of two Gaussians (DoG).



Edge Detection

Scale space

5 x 5 LoG filter

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

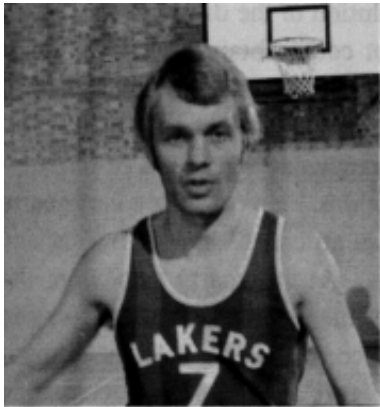
17 x 17 LoG filter

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

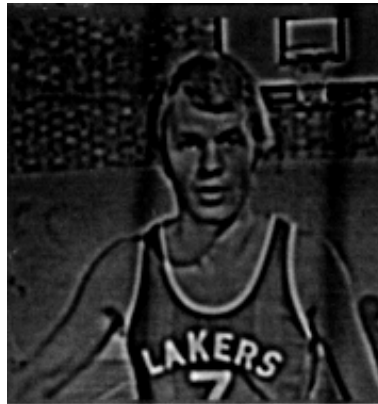
Scale (σ)

Edge Detection

Scale Space



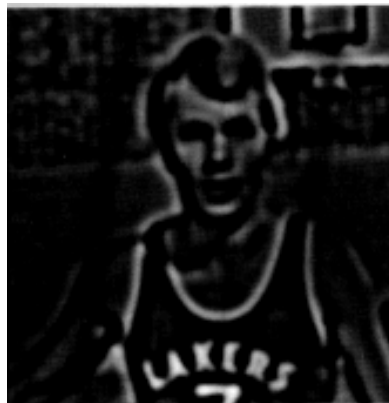
Original Image



LoG Filter



Zero Crossings



Scale (σ)



Edge Detection

Scale Space

The Gaussian filter removes noise but at the same time smooths the edges and other sharp intensity discontinuities within the images. The amount of blurring depends on the standard deviation. A larger σ corresponds to better noise filtering but the edge blurring increases. As a consequence, there is a trade-off between noise removal (better for larger σ , eg large kernels), and edge enhancement (better for smaller σ , eg small kernels). On the other hand, small filters result in too many noise points and large filters tend to dislocate the edges.

One approach, followed by Marr and Hildreth is to analyse the behavior of edges at **different scales of filtering**. To combine the information from different scales, the authors assume *spatial coincidence*:

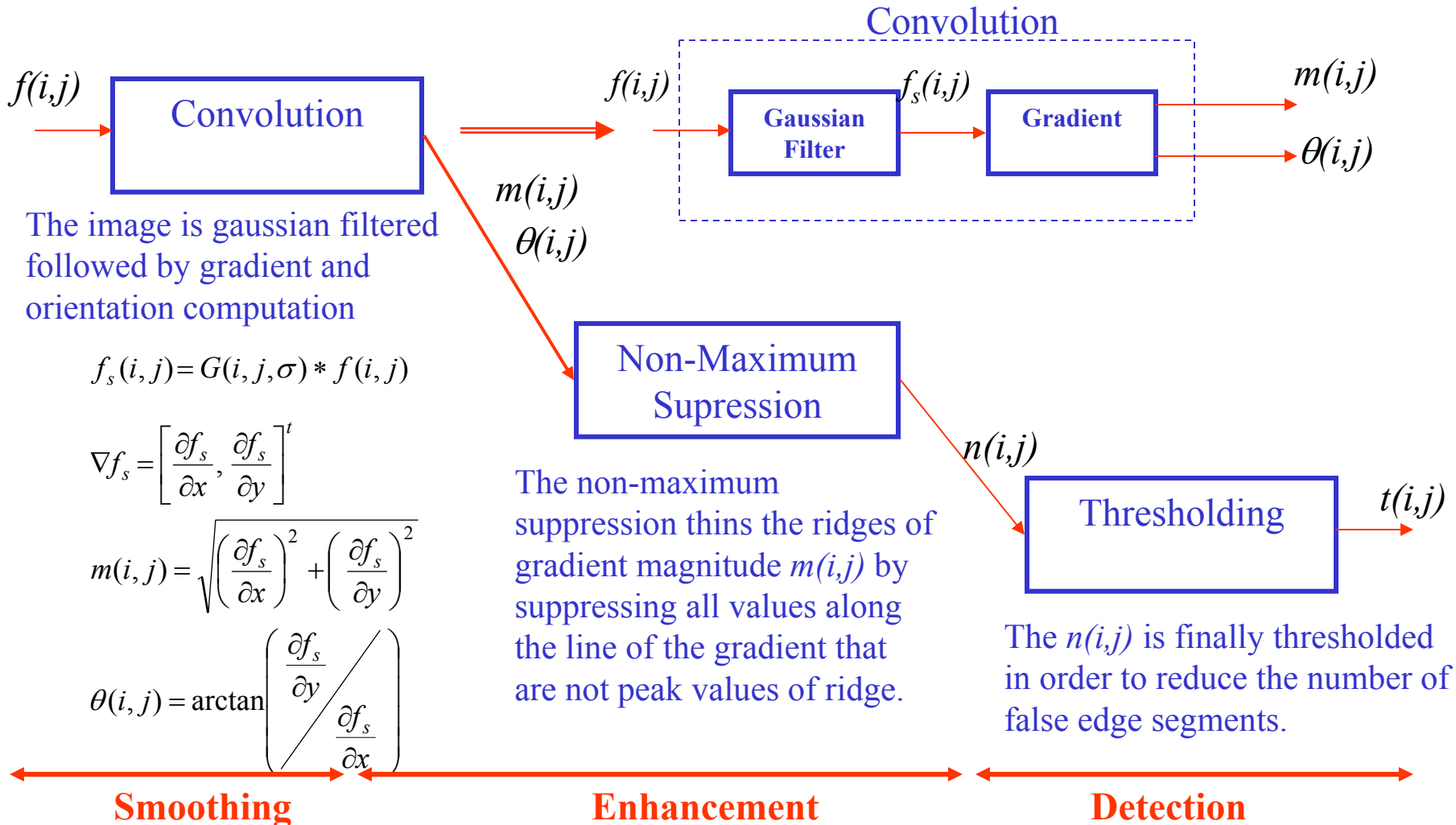
“If a zero-crossing segment is present in a set of independent $\nabla^2 G$ channels (scales) over a continuous range of sizes and the segment has the same position and orientation in each channel, then the set of such zero crossing segments may be taken to indicate the presence of an intensity change in the image that is due to single physical phenomenon (a change in reflectance, illumination, dept or surface orientation”.

Other authors stated other assumptions

- *Zero crossings connected at adjacent scales correspond to the same event, eg no new zero crossing are introduced as σ increases.*
- *The location of a zero crossing in the Gaussian filtered image tends to its true location as $\sigma \rightarrow 0$.*

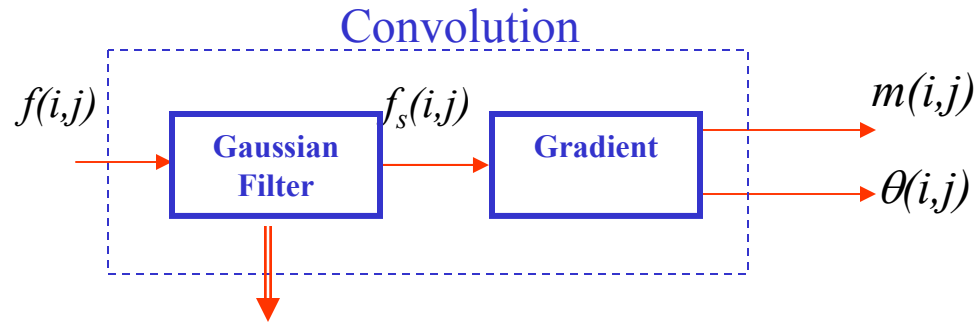
Edge Detection

Canny Edge Detector



Edge Detection

Canny Edge Detector. Gaussian Filtering



$$f_s(i, j) = G(i, j, \sigma) * f(i, j) = [G_h(i, \sigma) * f(i, j)] * G_v(j, \sigma) = ([a_1 \ a_2 \ \cdots \ a_n] * f(i, j)) * \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

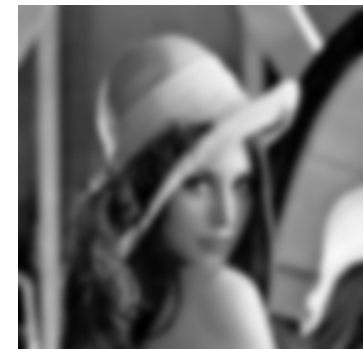
$\sigma = 1$



$\sigma = 2$



$\sigma = 3$



Kernels
(h1)

[4 60 272 450 272 60 4]

[4 19 60 146 272 397 450
397 272 146 60 19 4]

[1 4 12 29 60 112 185 272 360 425 450
425 360 272 185 112 60 29 12 4 1]

Canny Edge Detection

Gradient , non-maximum supression and thesholding

Step 1: magnitude and orientation computation

$$m(i, j) = \sqrt{\left(\frac{\partial f_s}{\partial x}\right)^2 + \left(\frac{\partial f_s}{\partial y}\right)^2} \quad \theta(i, j) = \arctan\left(\frac{\frac{\partial f_s}{\partial y}}{\frac{\partial f_s}{\partial x}}\right)$$

Step 2: Partition of angle orientations

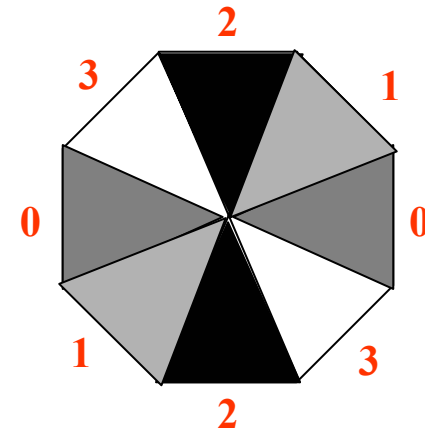
$$\mathcal{G}(i, j) = \text{sector}(\theta(i, j))$$

Step 3: At each pixel DO:

$$n(i, j) = m(i, j);$$

IF $m(i, j) \leq$ the neighbors along the
gradient sector

THEN $n(i, j) = 0$;



Step 4: Double Thresholding:

Create two thresholded images $t_1(i, j)$ and $t_2(i, j)$, using two thresholds T_1 and T_2 , with $T_1 \approx 0,4 T_2$.

This double threshold method allow to add weaker edges (those above T_1) if they are neighbors of stronger edges (those above T_2). So the threshold image is formed by $t_2(i, j)$ including some of the edges in $t_1(i, j)$

Canny Edge Detection

Results

$m(i,j)$

$\sigma=2$

$\theta(i,j)$



$m(i,j)$

$\sigma=3$

$\theta(i,j)$



$t(i,j)$



$\sigma=1$

$t(i,j)$



$\sigma=2$

$t(i,j)$



$\sigma=3$

Edge Detector Performance

Criteria for evaluating the performance of edge detectors:

- Probability of false edges
- Probability of missing edges
- Error in estimation of the edge angle
- Mean square distance of the edge estimate from the true edge
- Tolerance to distorted edges and other features such as corners and junctions.

Figure of merit (W. Pratt)

$$FM = \frac{1}{\max(I_A, I_I)} \sum_{i=1}^{I_A} \frac{1}{1 + d_i^2}$$

I_A - Image with the detected edges

I_I - Image with the ideal edges

d - distance between the actual and the ideal edges

a - design constant to penalize displaced edges.

For evaluating the performance we may generate images with known edge locations and then use FM. It is a common practice to evaluate the performance for synthetic images by introducing noise, and plot FM against the signal to noise ratio, to evaluate the degradation in the performance.