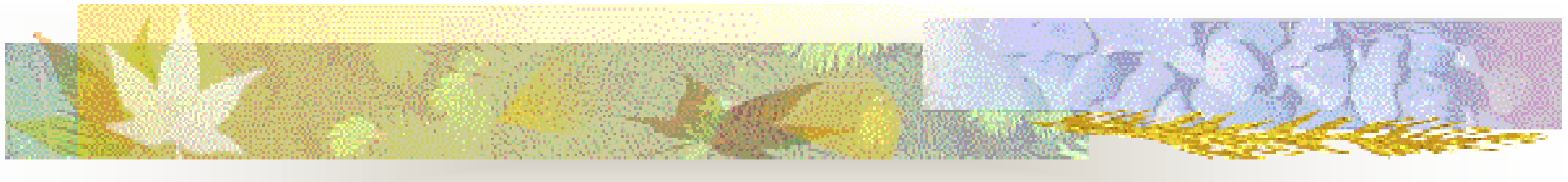
Gradient based Edge Detection



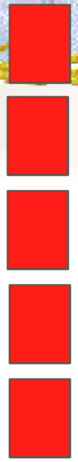
Canny Edge Detector

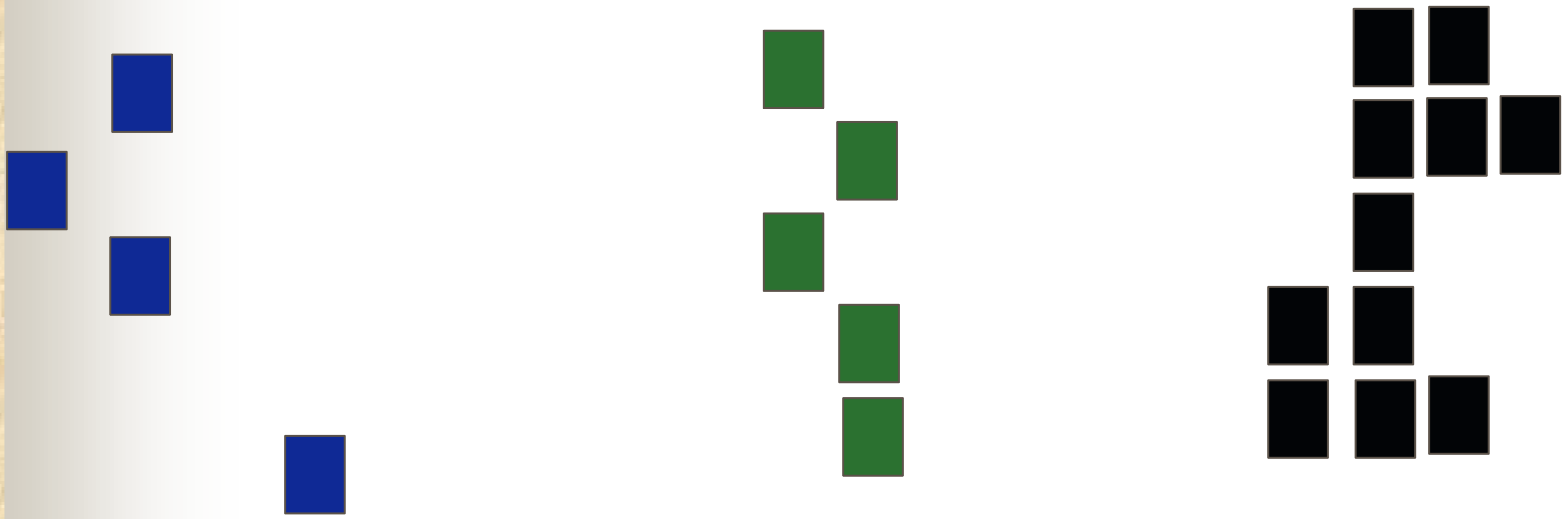
Suggested Reading

- Chapter 8, David A. Forsyth and Jean Ponce, "Computer Vision: A Modern Approach“
- Chapter 4, Emanuele Trucco, Alessandro Verri, "Introductory Techniques for 3-D Computer Vision"
- Chapter 2, Mubarak Shah, “Fundamentals of Computer Vision”

Quality of an Edge Detector

- Robustness to Noise
- Localization
- Too Many/Too less Responses





Optimal Edge Detector

- Criterion 1: Good Detection:

must minimize the probability of false positives (spurious noisy edges)

- must minimize the probability of missing real edge points

- Criterion 2: Good Localization: The edges detected must be as close as possible to the true edges.
- Problem: If signal to noise ratio is good then it has poor localization.
- Noisy edge may contain many local maxima

- Criterion 3:
- Single Response Constraint: The detector must return <u>one point only</u> for each edge point
- (to minimize number of local maxima around the true edge)

Canny Edge Detector

- Difficult to find closed-form solutions.

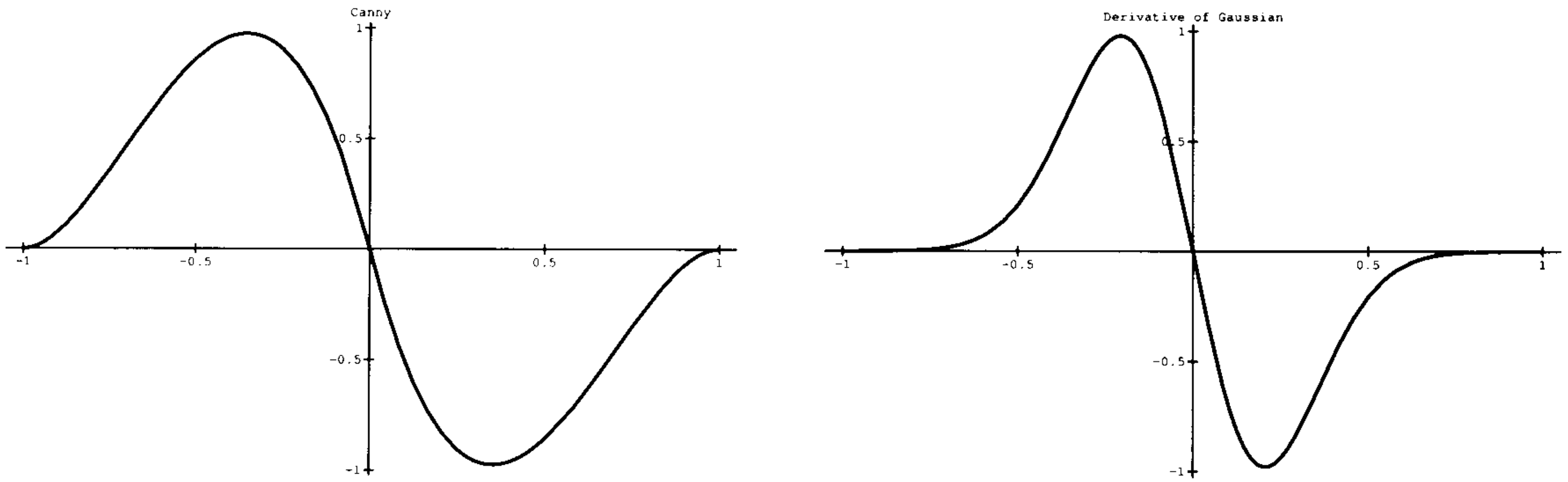


Figure 4.15 A comparison between the Canny operator and the first derivative of a Gaussian.

Canny Edge Detector

- Convolution with derivative of Gaussian
- Non-maximum Suppression
- Hysteresis Thresholding

Canny Edge Detector

- Smooth by Gaussian

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

- Compute *x* and *y* derivatives

$$\Delta S = \left[\frac{\partial}{\partial x} S \quad \frac{\partial}{\partial y} S \right]^T = \left[S_x \quad S_y \right]^T$$

- Compute gradient magnitude and orientation

$$|\Delta S| = \sqrt{S_x^2 + S_y^2}$$

$$q = \tan^{-1} \frac{S_y}{S_x}$$

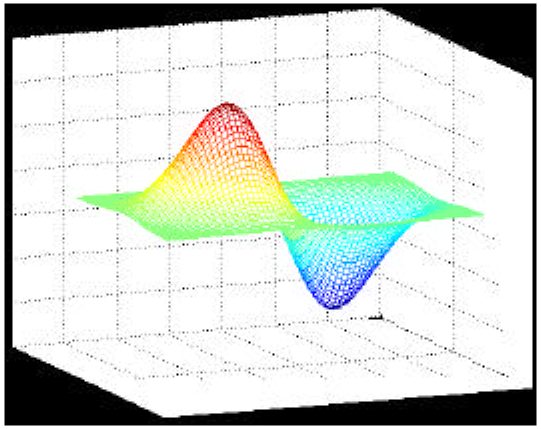
Canny Edge Operator

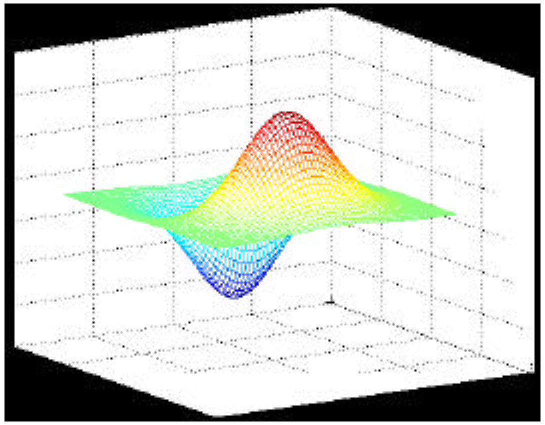
More Efficient Implementation:

$$\Delta S = \Delta(G_s * I) = \Delta G_s * I$$

$$\Delta G_s = \begin{bmatrix} \frac{\partial G_s}{\partial x} & \frac{\partial G_s}{\partial y} \end{bmatrix}^T$$

$$\Delta S = \begin{bmatrix} \frac{\partial G_s}{\partial x} * I & \frac{\partial G_s}{\partial y} * I \end{bmatrix}^T$$

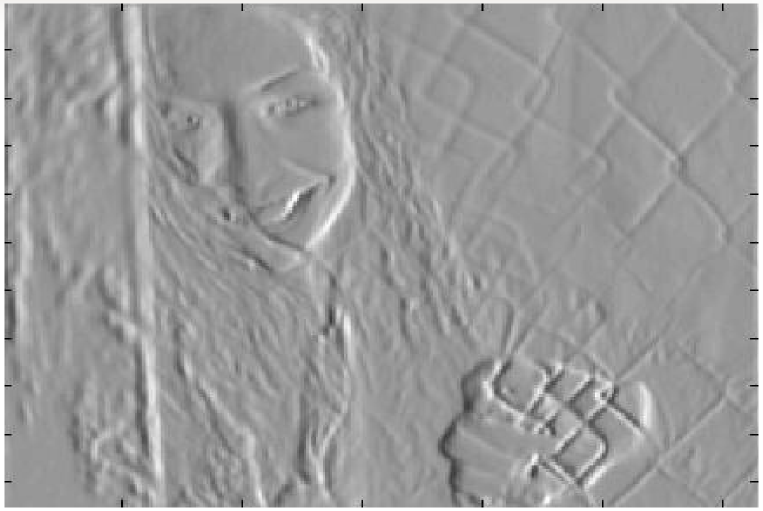




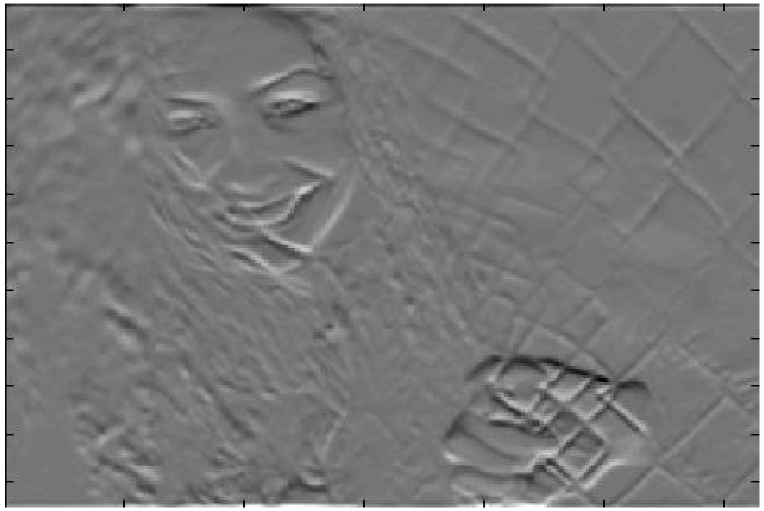
Canny Edge Detector

I

S_x

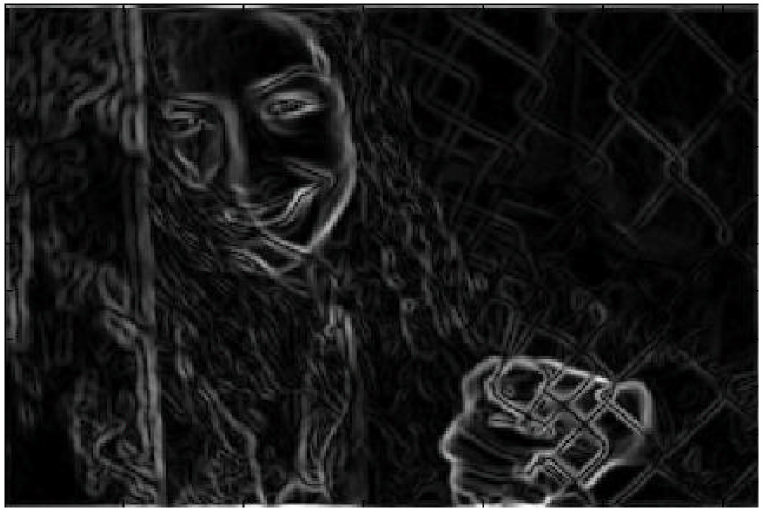


S_y

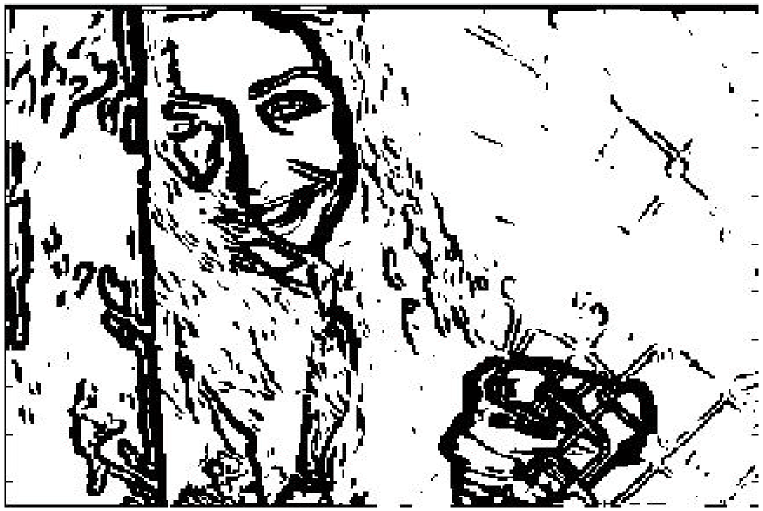


Canny Edge Detector

$$|\Delta S| = \sqrt{S_x^2 + S_y^2}$$



I

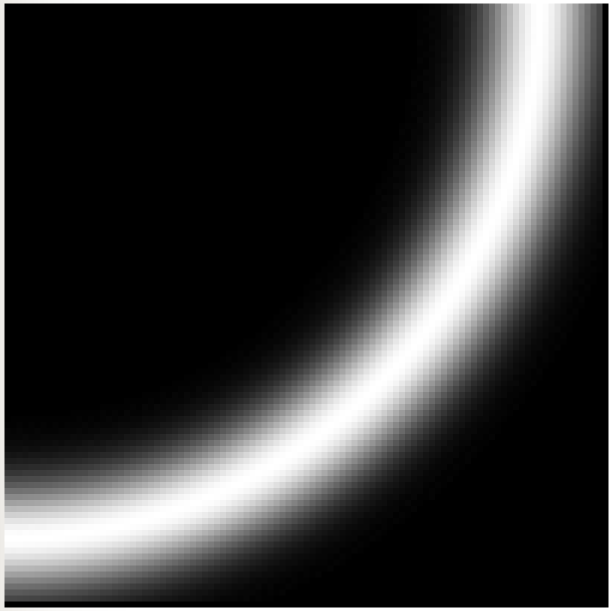


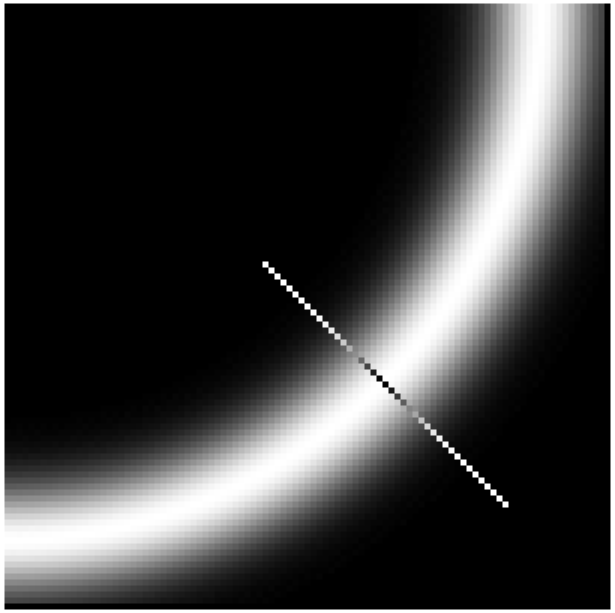
$$|\Delta S| \geq Threshold = 25$$

Non-Maximum Suppression

- Gradient Magnitudes – like chain of low hills
- Slice the gradient magnitude along the gradient direction (Perpendicular to edge)
- Mark points along the slice where magnitude is maximal.
- Discard non-maximal pts. (suppress)

Non-Maximum Suppression



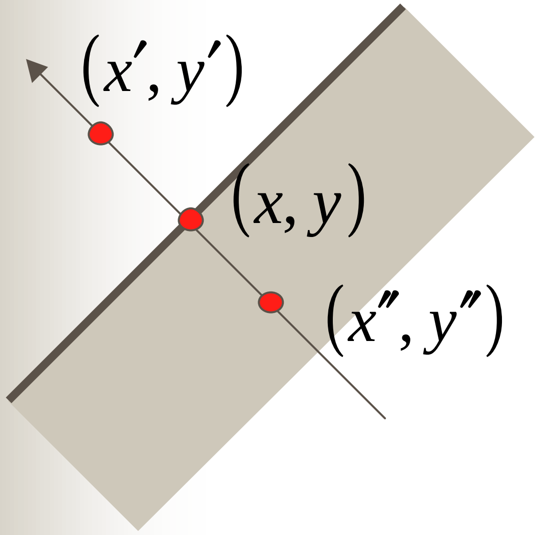


We wish to mark points along the curve where the magnitude is biggest. We can do this by looking for a maximum along a slice normal to the curve (non-maximum suppression). These points should form a curve. There are then two algorithmic issues: at which point is the maximum, and where is the next one?

Non-Maximum Suppression

- Suppress the pixels in ‘Gradient Magnitude Image’ which are not local maximum

$$M(x,y)=\begin{cases} |\Delta S|(x,y) & \text{if } |\Delta S|(x,y)>|\Delta S|(x',y') \\ & \& \, |\Delta S|(x,y)>|\Delta S|(x'',y'') \\ 0 & \text{otherwise} \end{cases}$$



(x',y') and (x'',y'') are the neighbors of (x,y) in $|\Delta S|$ along the direction normal to an edge

Direction?

- Consider the four directions 0^0, 45^0 , 90^0 ,135^0
- Find direction which best approximates the edge orientation
- If at a pixel gradient magnitude is smaller than at least one of its two neighbors along that direction, assign I (x,y) = 0
- Detect edge points by passing through a threshold.

Non-Maximum Suppression

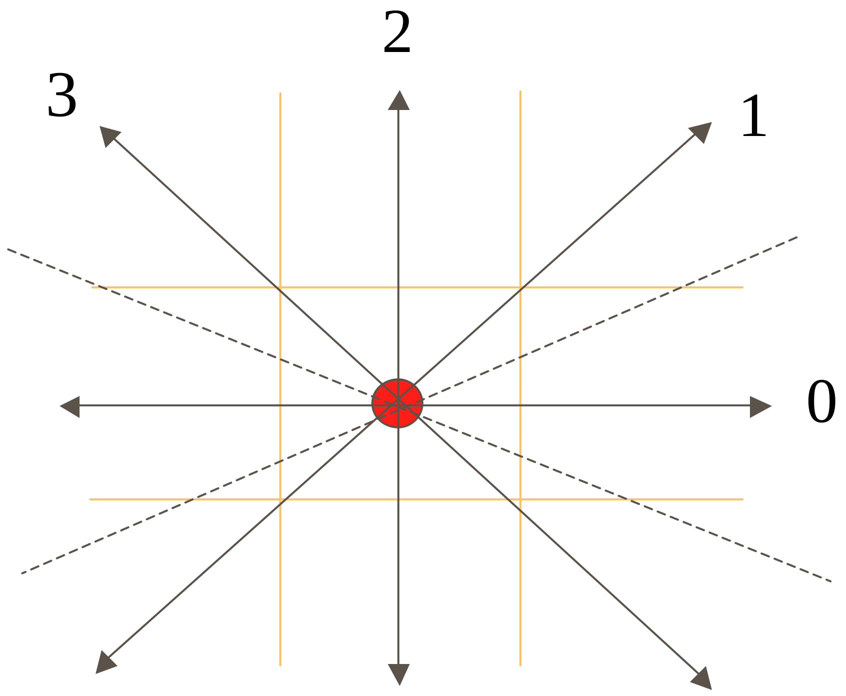
$$\tan ? = \frac{S_y}{S_x}$$

$$0: -0.4142 < \tan ? \leq 0.4142$$

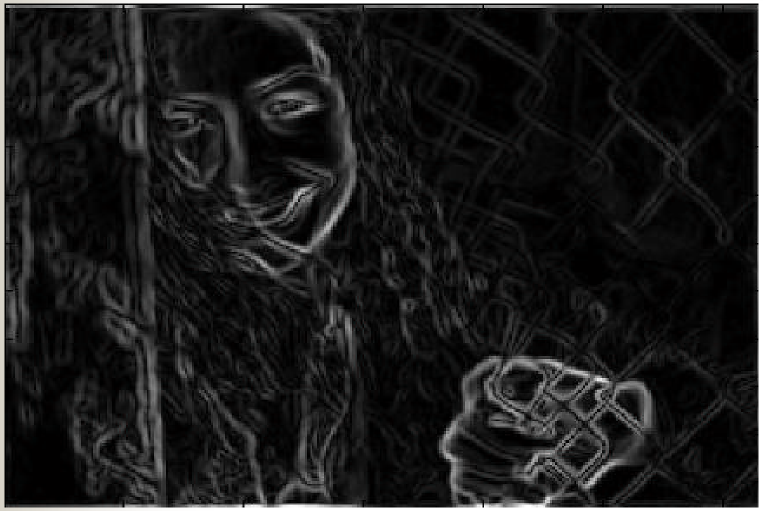
$$1: 0.4142 < \tan ? < 2.4142$$

$$2: |\tan ?| \geq 2.4142$$

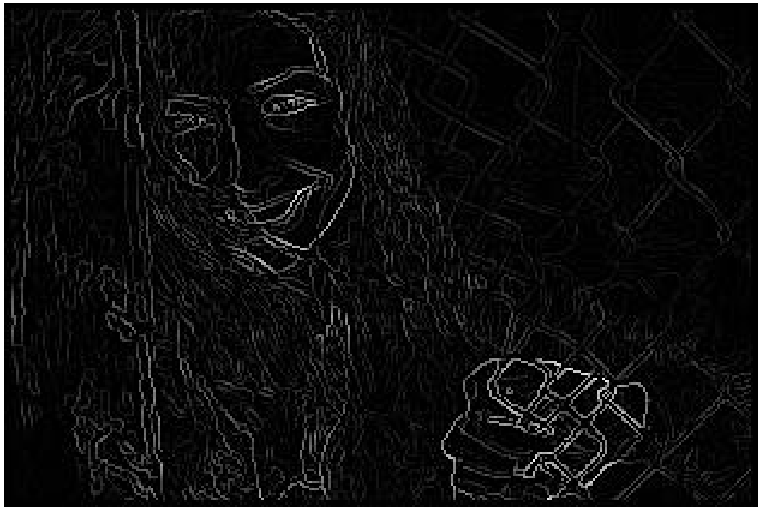
$$3: -2.4142 < \tan ? \leq -0.4142$$



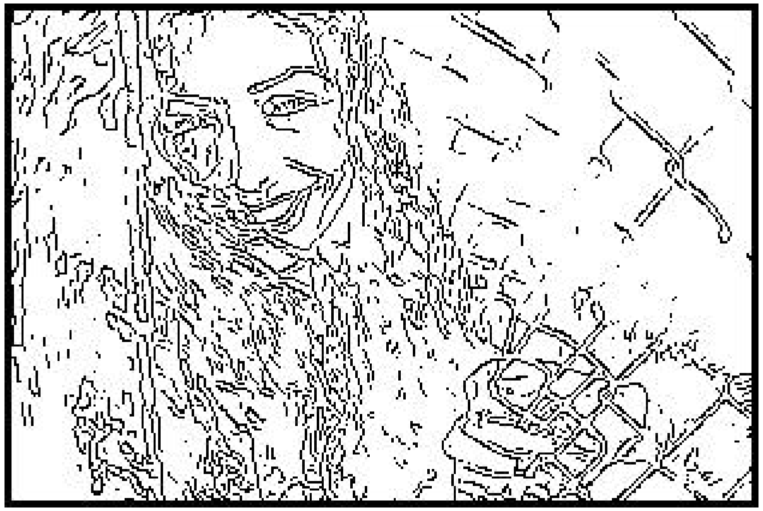
Non-Maximum Suppression



$$|\Delta S| = \sqrt{S_x^2 + S_y^2}$$



M

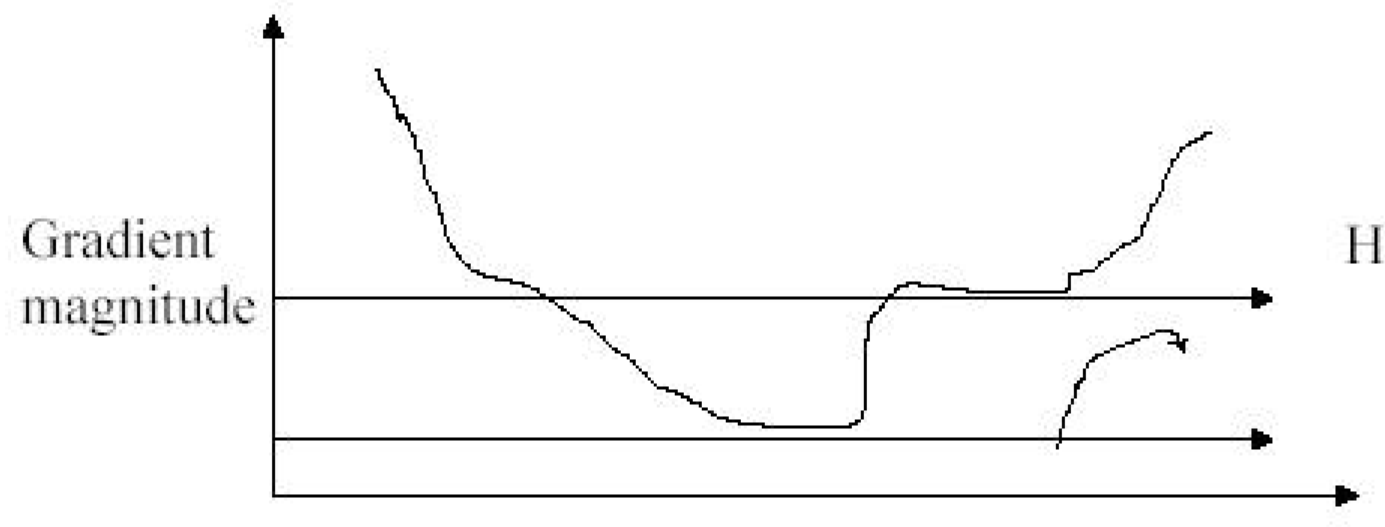


$$M \geq Threshold = 25$$

Selection from Maxima edges

- Too many maxima edge curves are obtained. Now time to look at how large the maximas are.
- Single Threshold does not work
  - Low threshold: we get false contours
  - True maxima may be below/above threshold resulting in edge fragmentation (broken edges)

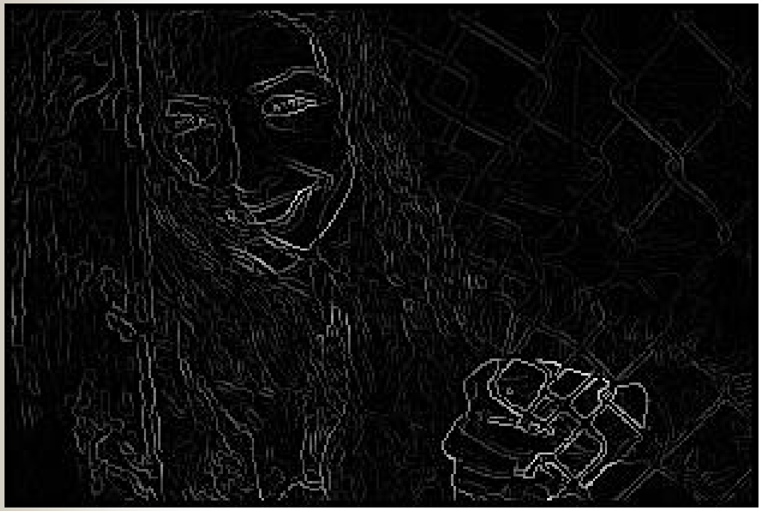
Hysteresis Thresholding



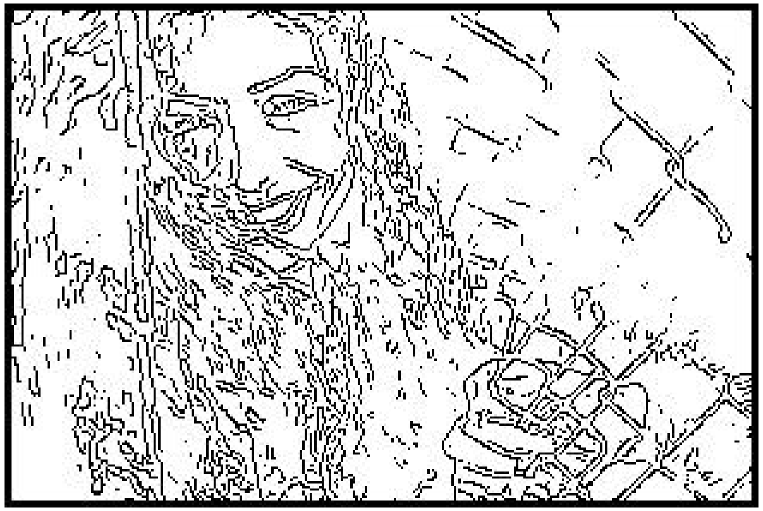
Hysteresis Thresholding

- If the gradient at a pixel is above ‘High’, declare it an ‘edge pixel’
- Starting from that pixel, follow the chains of connected local maxima in both directions (perpendicular to edge normal), as long as gradient is above ‘Low’
- Mark all visited points, and contour points.

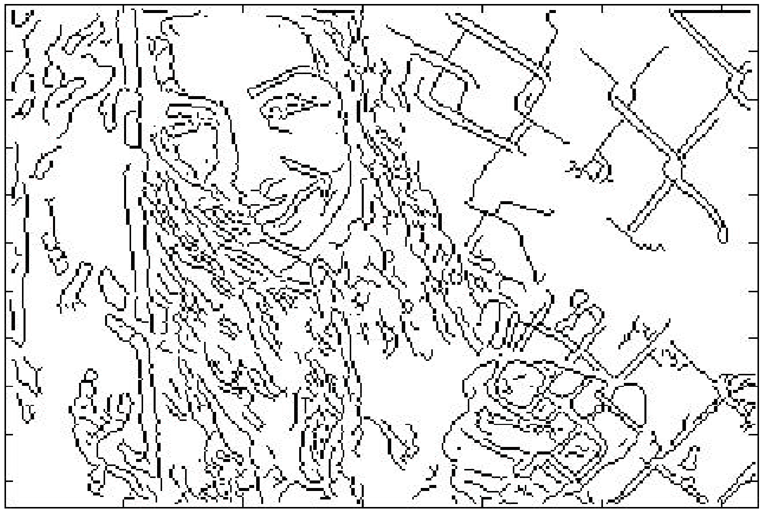
Hysteresis Thresholding



M



$M \geq Threshold = 25$



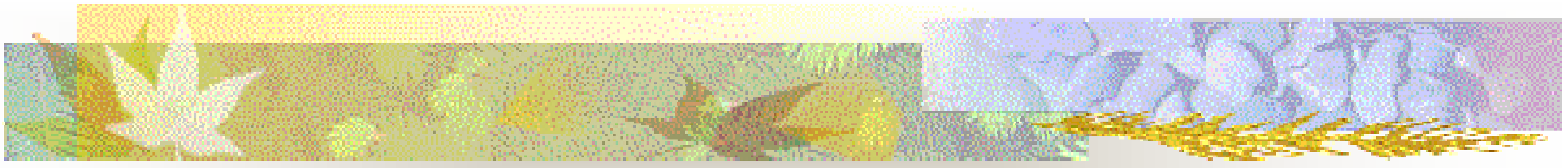
$High = 35$

$Low = 15$

Finding Connected Components

- Scan the binary image left to right top to bottom
- If there is an unlabeled pixel *p* with a value of ‘1’
  - assign a new label to it
  - Recursively check the neighbors of pixel *p* and assign the same label if they are unlabeled with a value of ‘1’.
- Stop when all the pixels with value ‘1’ have been labeled.

Haralick’s Edge Detector



Smoothing through local bi-cubic polynomial fitting

Haralick’s Edge Detector

- Fit a bi-cubic polynomial to a small neighborhood of a pixel.
- Compute analytically second and third directional derivatives in the direction of gradient.
- If the second derivative is equal to zero, and the third derivative is negative, then that point is an edge point.

Haralick’s Edge Detector

It is assumed that the intensity function at any point x,y can be represented in terms of a bicubic polynomial ,

analogous to a Talyor series expansion of a function of two variables, and going up to cubic terms

$$f(x,y) = k_1 + k_2 x + k_3 y + k_4 x^2 + k_5 xy + k_6 y^2 + k_7 x^3 + k_8 x^2 y + k_9 xy$$

If we just take the first order terms of this polynomial , the Gradient angle, defined with positive y-axis is defined as:

$$\sin\theta = \frac{k_2}{\sqrt{k_2{}^2 + k_3{}^2}},$$

$$\cos\theta = \frac{k_3}{\sqrt{k_2{}^2 + k_3{}^2}}.$$

Directional Derivatives

- The directional derivative of vector F along vector direction n is given by the dot product

  F.n

Haralick's Edge Detector

The directional derivates for a given direction vector, with the gradient angle measured with positive y-axis

$$f'_\theta(x,y) = \frac{\partial f}{\partial x}\sin\theta + \frac{\partial f}{\partial y}\cos\theta,$$

$$f''_\theta(x,y) = \frac{\partial^2 f}{\partial x^2}\sin^2\theta + \frac{\partial^2 f}{\partial y^2}\cos^2\theta + 2\frac{\partial^2 f}{\partial x \partial y}\cos\theta\sin\theta.$$

Substituting the variables x and y in polar form as

$$x = \rho\sin\theta, \; y = \rho\cos\theta$$

And substituting in the bicubic polynomial

$$f(x,y) = k_1 + k_2 x + k_3 y + k_4 x^2 + k_5 xy + k_6 y^2 + k_7 x^3 + k_8 x^2 y + k_9 xy^2 + k_{10} y^3.$$

The intensity function can be expressed as

$$f_\theta(\rho) = C_0 + C_1\rho + C_2\rho^2 + C_3\rho^3,$$

Homework

$$\begin{aligned}
C_0 &= k_1, \\
C_1 &= k_2 \sin\theta + k_3 \cos\theta, \\
C_2 &= k_4 \sin^2\theta + k_5 \sin\theta\cos\theta + k_6 \cos^2\theta, \\
C_3 &= k_7 \sin^3\theta + k_8 \sin^2\theta\cos\theta + k_9 \sin\theta\cos^2\theta + k_{10}\cos^3\theta.
\end{aligned}$$

Haralick's Edge Detector

The derivatives are obtained as follows:

$$f_\theta(\rho) = C_0 + C_1\rho + C_2\rho^2 + C_3\rho^3,$$

$$f_\theta'(\rho) = C_1 + 2C_2\rho + 3C_3\rho^2,$$

$$f_\theta''(\rho) = 2C_2 + 6C_3\rho,$$

$$f_\theta'''(\rho) = 6C_3.$$

Conditions for edge pixel

The second condition is that third derivative should be negative

$$f_\theta'''(\rho) < 0, \text{ we get } 6C_3 < 0, \text{ or } C_3 < 0.$$

The first condition is that second derivate should be zero

$$f_\theta''(\rho) = 2C_2 + 6C_3\rho = 0, \text{ we get } \left|\frac{C_2}{3C_3}\right| < \rho_0.$$

Deriving a mask instead of computing the least sq solution

We can take a first order polynomial to illustrate the idea. Consider number of points, say 9, in a small neighborhood of a pixel, and use a least square formulation to obtain the coefficients.

Haralick's Edge Detector

First order polynomial

$$f(x, y) = k_1 + k_2 x + k_3 y.$$

9 points give 9 eqs

$$\begin{aligned} f1 &= k_1 + k_2 x_1 + k_3 y_1, \\ f2 &= k_1 + k_2 x_2 + k_3 y_2, \\ &\vdots \\ f9 &= k_1 + k_2 x_9 + k_3 y_9. \end{aligned}$$

$$\begin{bmatrix} f1 \\ f2 \\ \vdots \\ f9 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \vdots & & \\ 1 & x_9 & y_9 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix},$$

$$f = Ak.$$

$$\begin{aligned} (A^T A)^{-1} A^T f &= k, \\ Bf &= k, \end{aligned}$$

Haralick’s Edge Detector

$$B = (A^T A)^{-1} A^T \text{ is a } 3 \times 9 \text{ matrix}$$

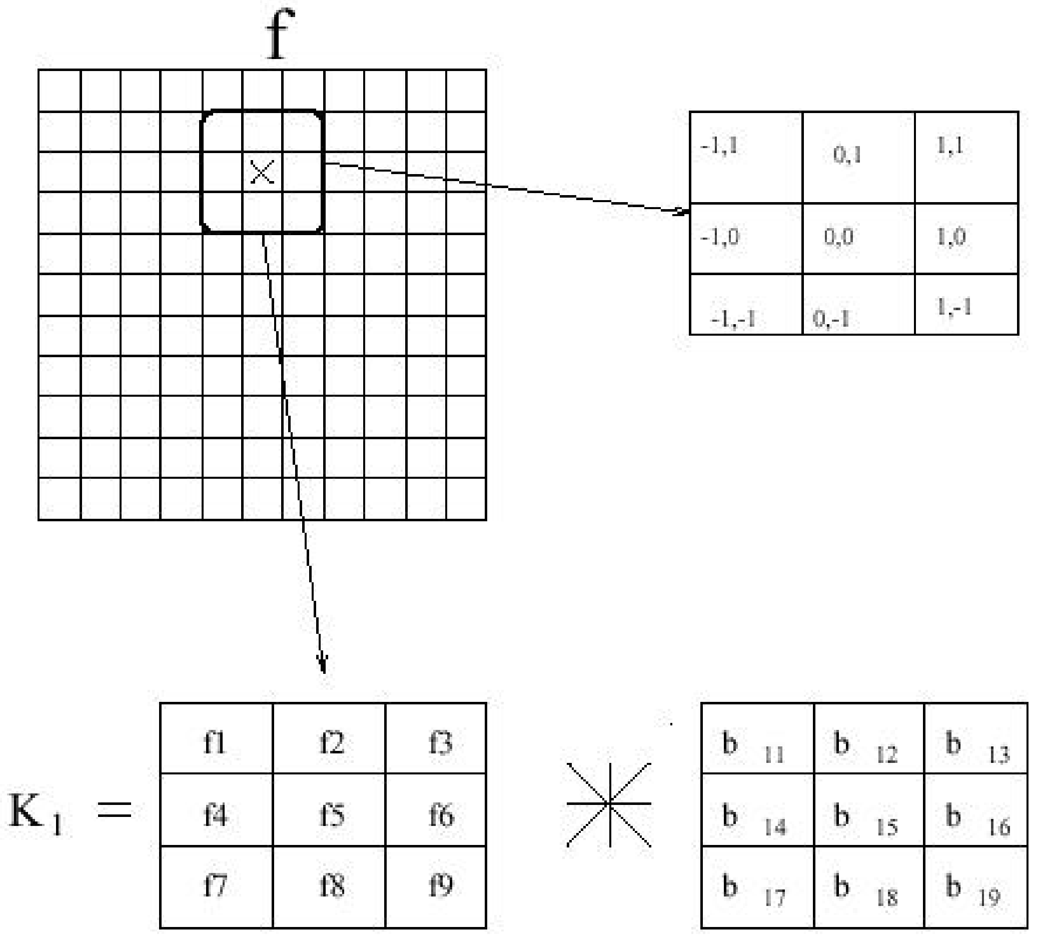
$$Bf = k.$$

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} & b_{17} & b_{18} & b_{19} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} & b_{27} & b_{28} & b_{29} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} & b_{36} & b_{37} & b_{38} & b_{39} \end{bmatrix} \begin{bmatrix} f1 \\ f2 \\ f3 \\ f4 \\ f5 \\ f6 \\ f7 \\ f8 \\ f9 \end{bmatrix} = \begin{bmatrix} k1 \\ k2 \\ k3 \end{bmatrix}$$

$$k_1 = b_{11}f1 + b_{12}f2 + b_{13}f3 + b_{14}f4 + b_{15}f5 + b_{16}f6 + b_{17}f7 + b_{18}f8 + b_{19}f9$$

$$k_1 = f * b_1.$$

Computing coefficients using convolution



$$\frac{1}{175}\begin{array}{|c|c|c|c|c|}\hline -13 & 2 & 7 & 2 & -13 \\ \hline 2 & 17 & 22 & 17 & 2 \\ \hline 7 & 22 & 27 & 22 & 7 \\ \hline 2 & 17 & 22 & 17 & 2 \\ \hline -13 & 2 & 7 & 2 & -13 \\ \hline \end{array}$$

k_1

$$\frac{1}{420}\begin{array}{|c|c|c|c|c|}\hline 31 & -5 & -17 & -5 & 31 \\ \hline -44 & -62 & -68 & -62 & -44 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 44 & 62 & 68 & 62 & 44 \\ \hline -31 & 5 & 17 & 5 & -31 \\ \hline \end{array}$$

k_2

$$\frac{1}{420}\begin{array}{|c|c|c|c|c|}\hline 31 & -44 & 0 & 44 & -31 \\ \hline -5 & -62 & 0 & 62 & 5 \\ \hline -17 & -68 & 0 & 68 & 17 \\ \hline -5 & -62 & 0 & 62 & 5 \\ \hline 31 & -44 & 0 & 44 & -31 \\ \hline \end{array}$$

k_3

$$\frac{1}{70}\begin{array}{|c|c|c|c|c|}\hline 2 & 2 & 2 & 2 & 2 \\ \hline -1 & -1 & -1 & -1 & -1 \\ \hline -2 & -2 & -2 & -2 & -2 \\ \hline -1 & -1 & -1 & -1 & -1 \\ \hline 2 & 2 & 2 & 2 & 2 \\ \hline \end{array}$$

k_4

$$\frac{1}{100}\begin{array}{|c|c|c|c|c|}\hline 4 & 2 & 0 & -2 & -4 \\ \hline 2 & 1 & 0 & -1 & -2 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline -2 & -1 & 0 & 1 & 2 \\ \hline -4 & -2 & 0 & 2 & 4 \\ \hline \end{array}$$

k_5

$$\frac{1}{70}\begin{array}{|c|c|c|c|c|}\hline 2 & -1 & -2 & -1 & 2 \\ \hline 2 & -1 & -2 & -1 & 2 \\ \hline 2 & -1 & -2 & -1 & 2 \\ \hline 2 & -1 & -2 & -1 & 2 \\ \hline 2 & -1 & -2 & -1 & 2 \\ \hline \end{array}$$

k_6

$$\frac{1}{60}\begin{array}{|c|c|c|c|c|}\hline -1 & -1 & -1 & -1 & -1 \\ \hline 2 & 2 & 2 & 2 & 2 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline -2 & -2 & -2 & -2 & -2 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

k_7

$$\frac{1}{140}\begin{array}{|c|c|c|c|c|}\hline -4 & -2 & 0 & 2 & 4 \\ \hline 2 & 1 & 0 & -1 & -2 \\ \hline 4 & 2 & 0 & -2 & -4 \\ \hline 2 & 1 & 0 & -1 & -2 \\ \hline -4 & -2 & 0 & 2 & 4 \\ \hline \end{array}$$

k_8

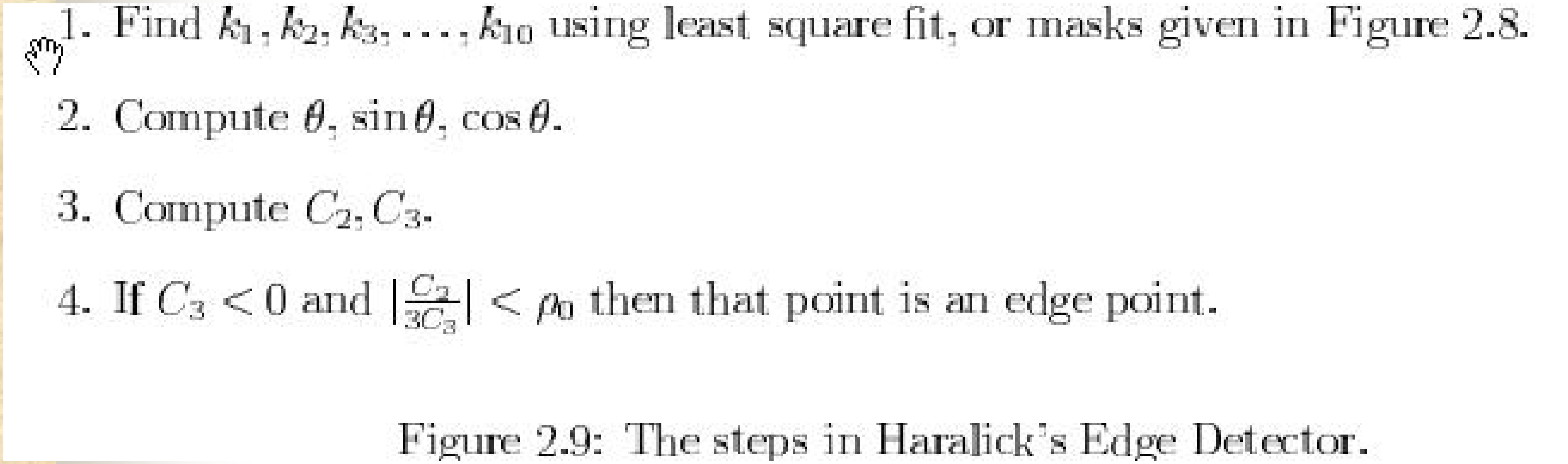
$$\frac{1}{140}\begin{array}{|c|c|c|c|c|}\hline -4 & 2 & 4 & 2 & -4 \\ \hline -2 & 1 & 2 & 1 & -2 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 2 & -1 & -2 & -1 & 2 \\ \hline 4 & -2 & -4 & -2 & 4 \\ \hline \end{array}$$

k_9

$$\frac{1}{60}\begin{array}{|c|c|c|c|c|}\hline -1 & 2 & 0 & -2 & 1 \\ \hline -1 & 2 & 0 & -2 & 1 \\ \hline -1 & 2 & 0 & -2 & 1 \\ \hline -1 & 2 & 0 & -2 & 1 \\ \hline -1 & 2 & 0 & -2 & 1 \\ \hline \end{array}$$

k_{10}

Haralick's Edge Detector



1. Find $k_1, k_2, k_3, \ldots, k_{10}$ using least square fit, or masks given in Figure 2.8.
2. Compute θ, $\sin\theta$, $\cos\theta$.
3. Compute C_2, C_3.
4. If $C_3 < 0$ and $|\frac{C_2}{3C_3}| < \rho_0$ then that point is an edge point.

Figure 2.9: The steps in Haralick's Edge Detector.

The three Edge Detectors

- Marr-Hildreth
  - Gaussian filter
  - Zerocrossings in Laplacian
- Canny
  - Gaussian filter
  - Maxima in gradient magnitude
- Haralick
  - Smoothing through bi-cubic polynomial
  - Zerocrossings in the second directional derivative, and negative third derivative