

Edge detection

- Gradient-based edge operators
 - Prewitt
 - Sobel
 - Roberts
- Laplacian zero-crossings
- Canny edge detector
- Hough transform for detection of straight lines
- Circle Hough Transform

Gradient-based edge detection

- Idea (continuous-space): local gradient magnitude indicates edge strength

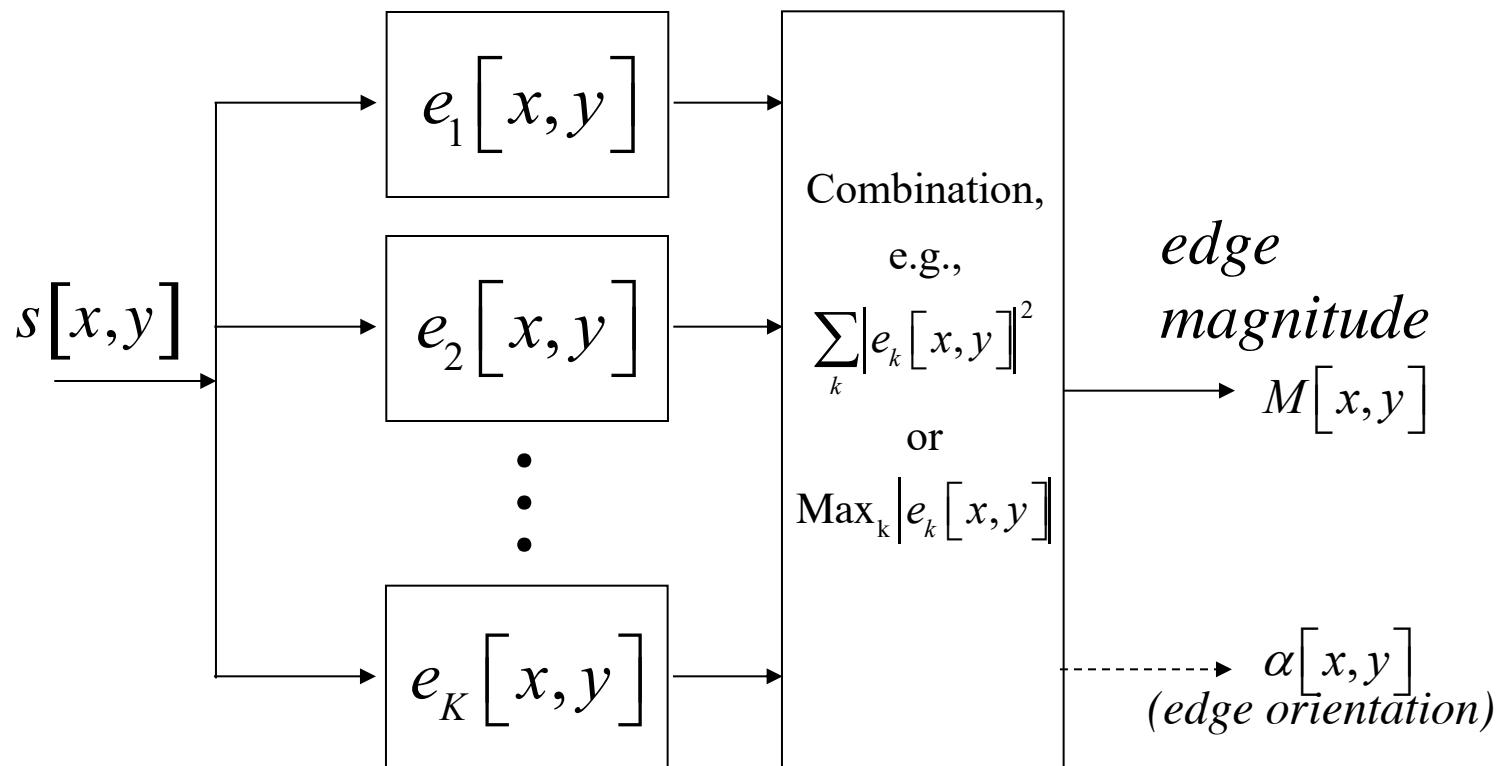
$$\|grad(f(x,y))\| = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2}$$

- Digital image:
use finite differences
to approximate
derivatives

difference	$\begin{pmatrix} -1 & 1 \end{pmatrix}$
central difference	$\begin{pmatrix} -1 & [0] & 1 \end{pmatrix}$
Prewitt	$\begin{pmatrix} -1 & 0 & 1 \\ -1 & [0] & 1 \\ -1 & 0 & 1 \end{pmatrix}$
Sobel	$\begin{pmatrix} -1 & 0 & 1 \\ -2 & [0] & 2 \\ -1 & 0 & 1 \end{pmatrix}$

Practical edge detectors

- Edges can have any orientation
- Typical edge detection scheme uses $K=2$ edge templates
- Some use $K>2$



Gradient filters (K=2)

Central Difference

$$\begin{pmatrix} 0 & 0 & 0 \\ -1 & [0] & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ 0 & [0] & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Roberts

$$\begin{pmatrix} [0] & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} [1] & 0 \\ 0 & -1 \end{pmatrix}$$

Prewitt

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & [0] & 1 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & -1 & -1 \\ 0 & [0] & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Sobel

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & [0] & 2 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & -2 & -1 \\ 0 & [0] & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Kirsch operator (K=8)

$$\text{Kirsch} \quad \begin{pmatrix} +5 & +5 & +5 \\ -3 & [0] & -3 \\ -3 & -3 & -3 \end{pmatrix} \begin{pmatrix} -3 & +5 & +5 \\ -3 & [0] & +5 \\ -3 & -3 & -3 \end{pmatrix} \begin{pmatrix} -3 & -3 & +5 \\ -3 & [0] & +5 \\ -3 & -3 & +5 \end{pmatrix} \begin{pmatrix} -3 & -3 & -3 \\ -3 & [0] & +5 \\ -3 & +5 & +5 \end{pmatrix}$$
$$\begin{pmatrix} -3 & -3 & -3 \\ -3 & [0] & -3 \\ +5 & +5 & +5 \end{pmatrix} \begin{pmatrix} -3 & -3 & -3 \\ +5 & [0] & -3 \\ +5 & +5 & -3 \end{pmatrix} \begin{pmatrix} +5 & -3 & -3 \\ +5 & [0] & -3 \\ +5 & -3 & -3 \end{pmatrix} \begin{pmatrix} +5 & +5 & -3 \\ +5 & [0] & -3 \\ -3 & -3 & -3 \end{pmatrix}$$

Prewitt operator example



Original
 1024×710



Magnitude of
image filtered with
$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & [0] & 1 \\ -1 & 0 & 1 \end{pmatrix}$$
(log display)



Magnitude of
image filtered with
$$\begin{pmatrix} -1 & -1 & -1 \\ 0 & [0] & 0 \\ 1 & 1 & 1 \end{pmatrix}$$
(log display)



Prewitt operator example (cont.)



Sum of squared
horizontal and
vertical gradients
(log display)

threshold = 900

threshold = 4500

threshold = 7200



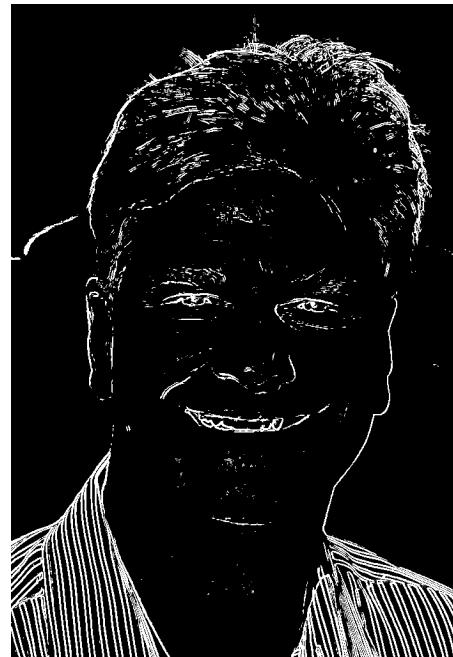
Sobel operator example



Sum of squared
horizontal and
vertical gradients
(log display)



threshold = 1600



threshold = 8000



threshold = 12800



Roberts operator example



Original
 1024×710



Magnitude of
image filtered with
$$\begin{pmatrix} [1] & 0 \\ 0 & -1 \end{pmatrix}$$

(log display)



Magnitude of
image filtered with
$$\begin{pmatrix} [0] & 1 \\ -1 & 0 \end{pmatrix}$$

(log display)



Roberts operator example (cont.)



Sum of squared
diagonal gradients
(log display)

threshold = 100

threshold = 500

threshold = 800

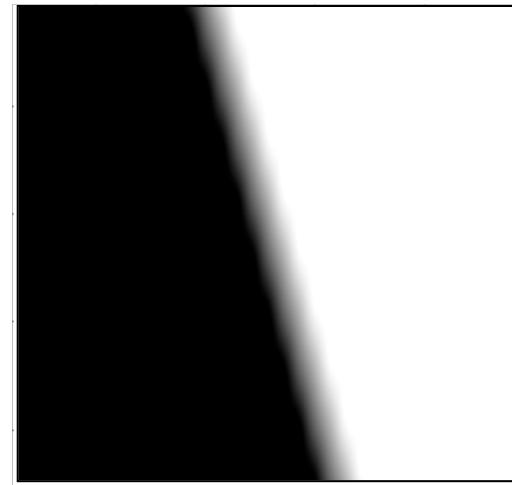


Edge orientation

Central
Difference

$$\begin{pmatrix} 0 & 0 & 0 \\ -1 & [0] & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

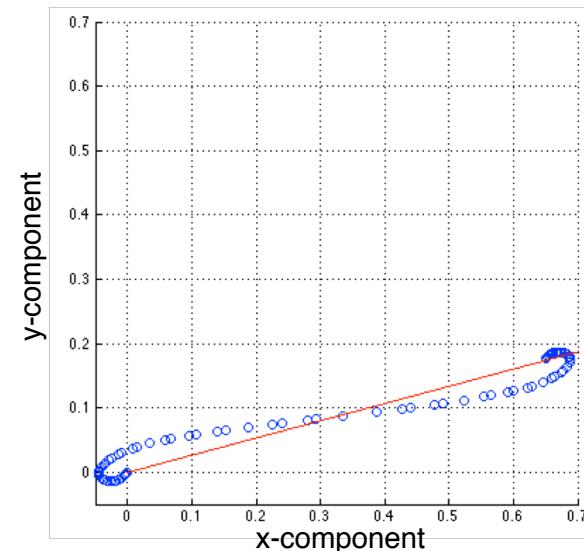
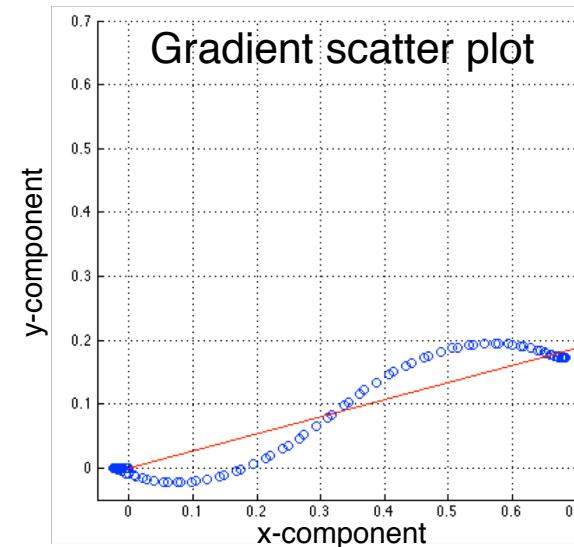
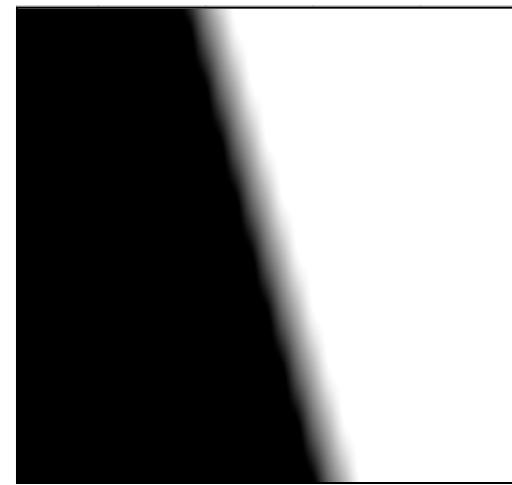
$$\begin{pmatrix} 0 & -1 & 0 \\ 0 & [0] & 0 \\ 0 & 1 & 0 \end{pmatrix}$$



Roberts

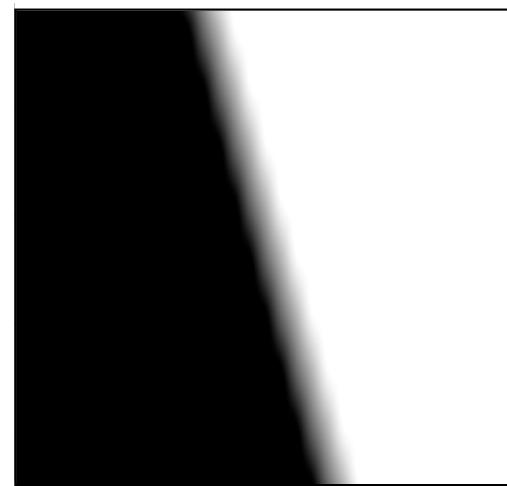
$$\begin{pmatrix} [0] & 1 \\ -1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} [1] & 0 \\ 0 & -1 \end{pmatrix}$$

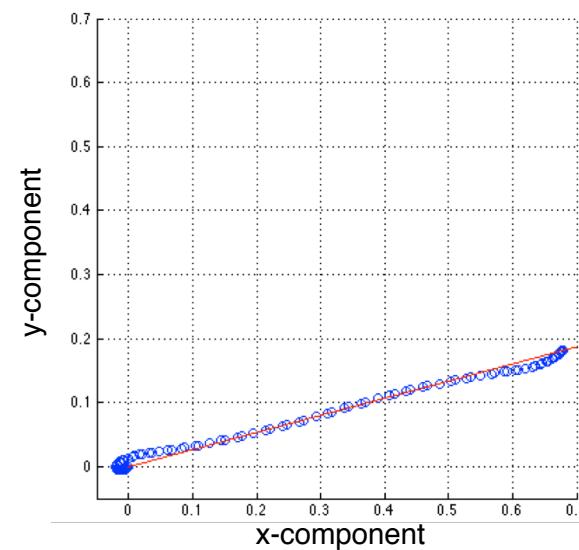
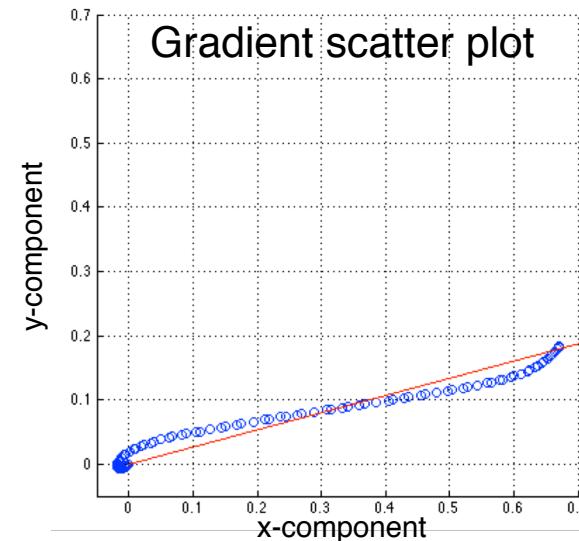
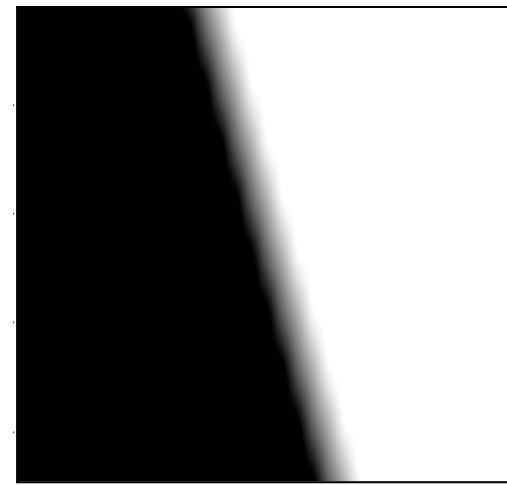


Edge orientation

Prewitt

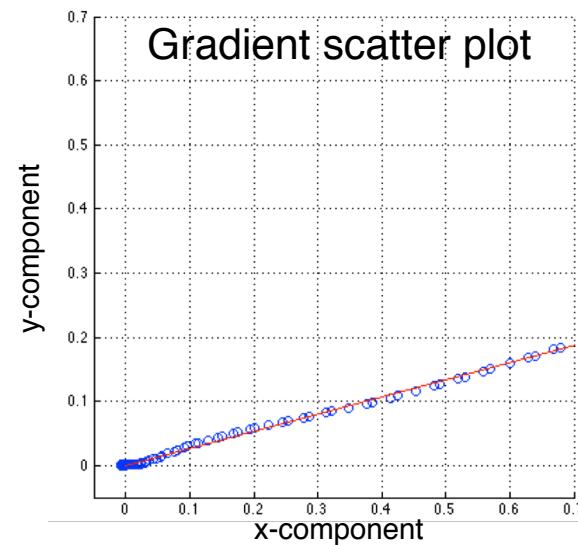
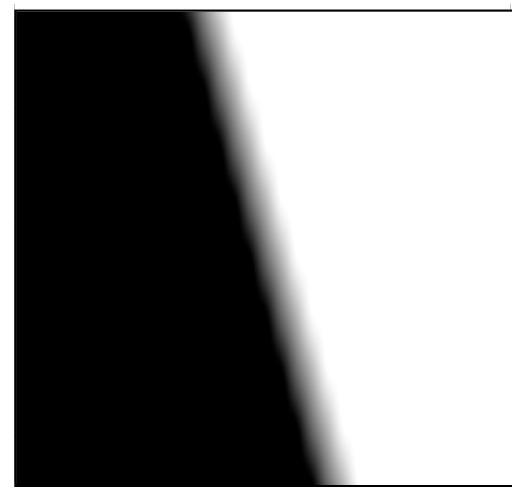
$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & [0] & 1 \\ -1 & 0 & 1 \end{pmatrix}$$
$$\begin{pmatrix} -1 & -1 & -1 \\ 0 & [0] & 0 \\ 1 & 1 & 1 \end{pmatrix}$$


Sobel

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & [0] & 2 \\ -1 & 0 & 1 \end{pmatrix}$$
$$\begin{pmatrix} -1 & -2 & -1 \\ 0 & [0] & 0 \\ 1 & 2 & 1 \end{pmatrix}$$


Edge orientation

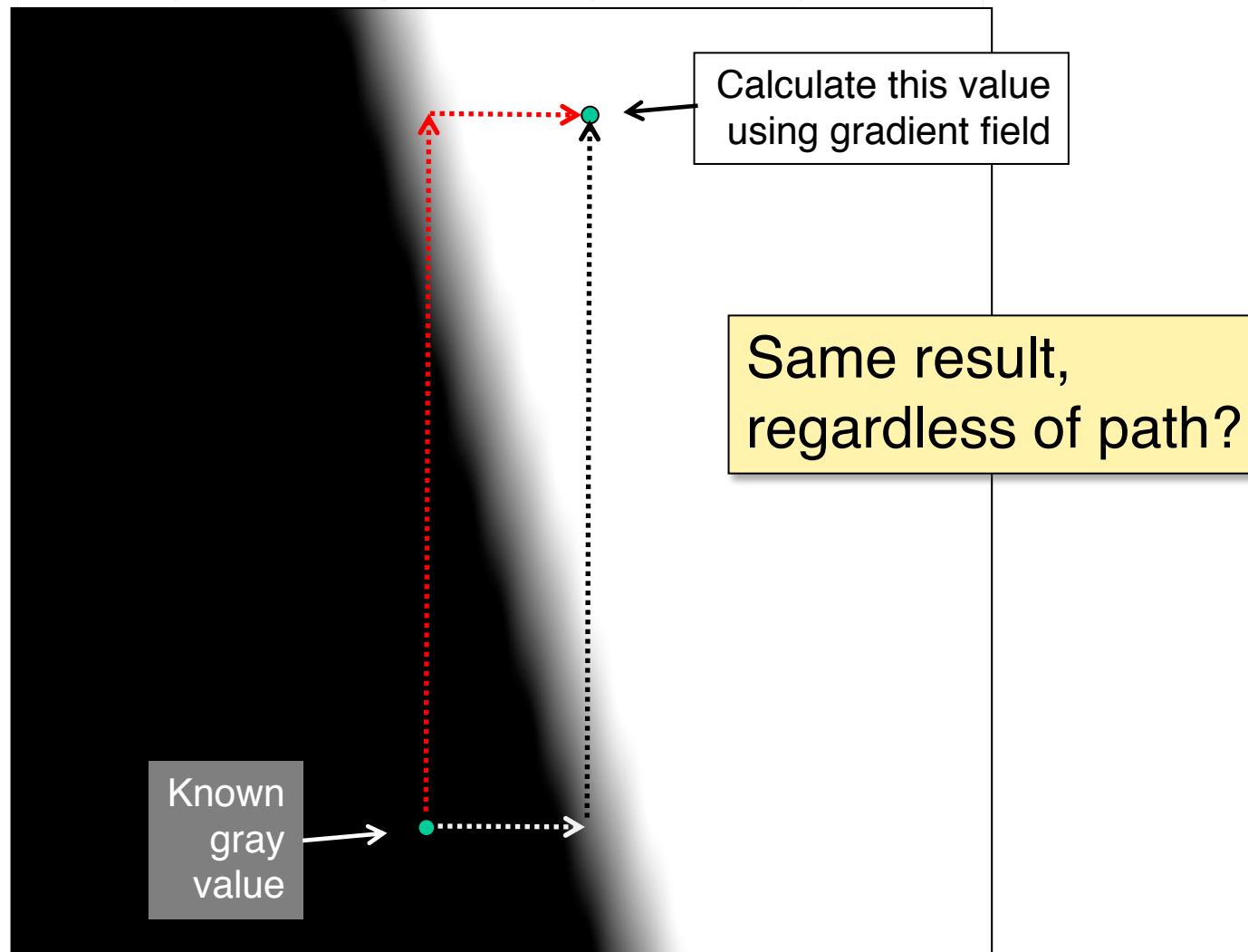
5x5 “consistent”
gradient operator
[\[Ando, 2000\]](#)



$$\begin{pmatrix} -0.0604 & -0.1632 & 0 & 0.1632 & 0.0604 \\ -0.4286 & -1.1335 & 0 & 1.1335 & 0.4286 \\ -0.7448 & -1.9612 & [0] & 1.9612 & 0.7448 \\ -0.4286 & -1.1335 & 0 & 1.1335 & 0.4286 \\ -0.0604 & -0.1632 & 0 & 0.1632 & 0.0604 \end{pmatrix}$$

$$\begin{pmatrix} -0.0604 & -0.4286 & -0.7448 & -0.4286 & -0.0604 \\ -0.1632 & -1.1335 & -1.9612 & -1.1335 & -0.1632 \\ 0 & 0 & [0] & 0 & 0 \\ 0.1632 & 1.1335 & 1.9612 & 1.1335 & 0.1632 \\ 0.0604 & 0.4286 & 0.7448 & 0.4286 & 0.0604 \end{pmatrix}$$

Gradient consistency problem

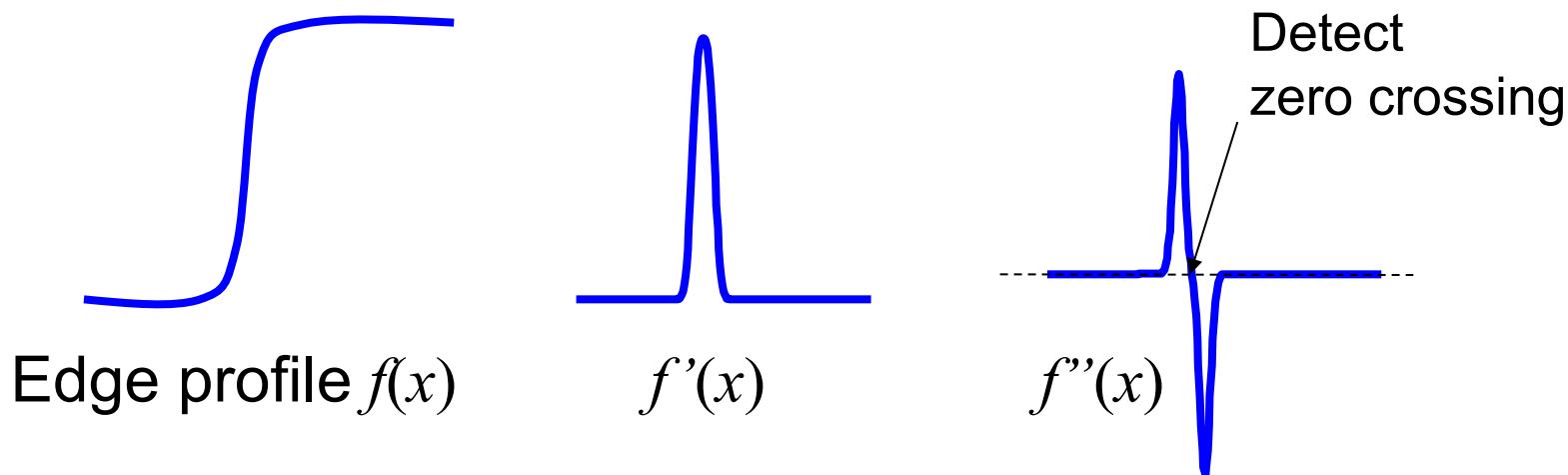


Laplacian operator

- Detect edges by considering second derivative

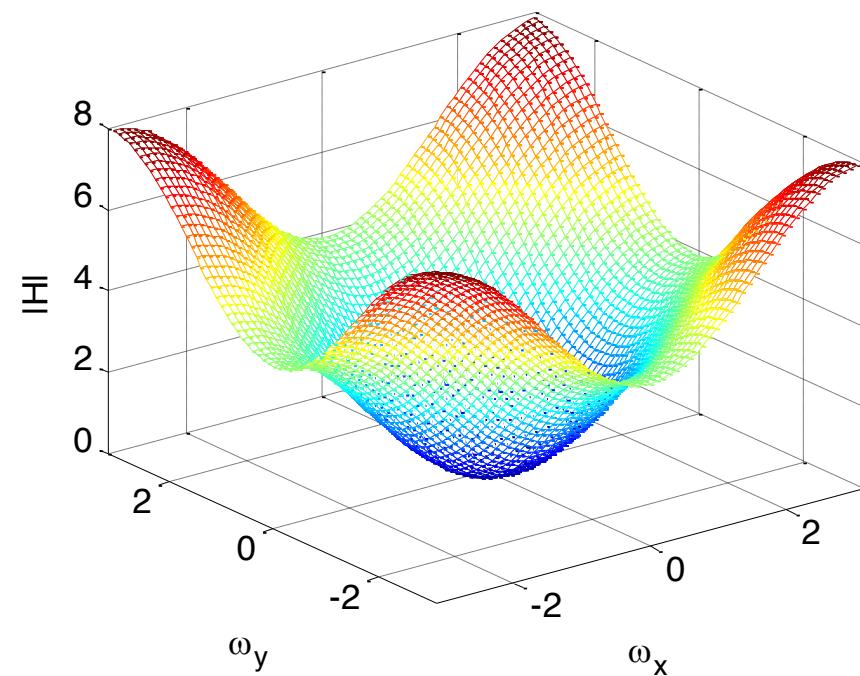
$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}$$

- Isotropic (rotationally invariant) operator
- Zero-crossings mark edge location

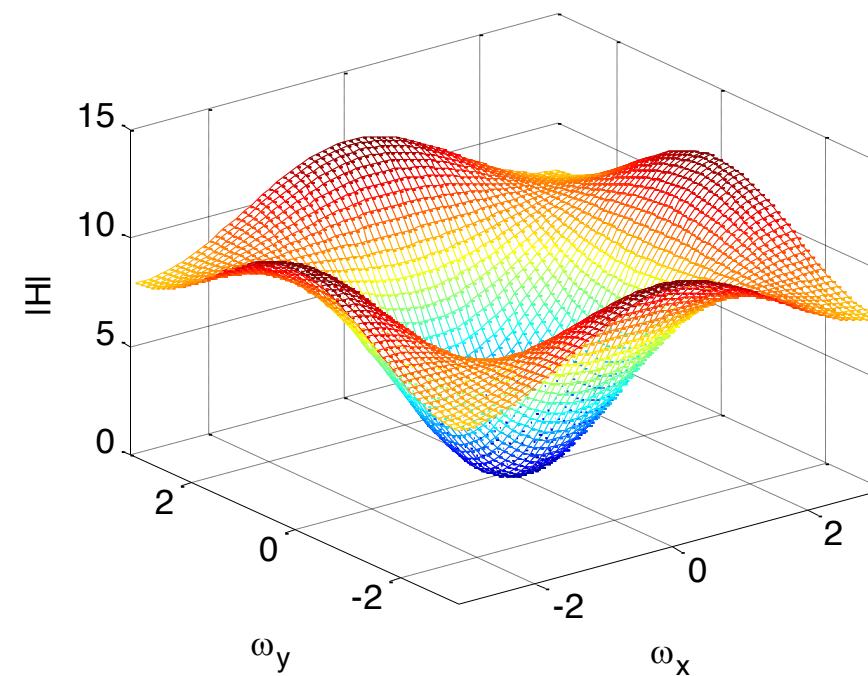


Approximations of Laplacian operator by 3x3 filter

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & [-4] & 1 \\ 0 & 1 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & [-8] & 1 \\ 1 & 1 & 1 \end{pmatrix}$$



Zero crossings of Laplacian



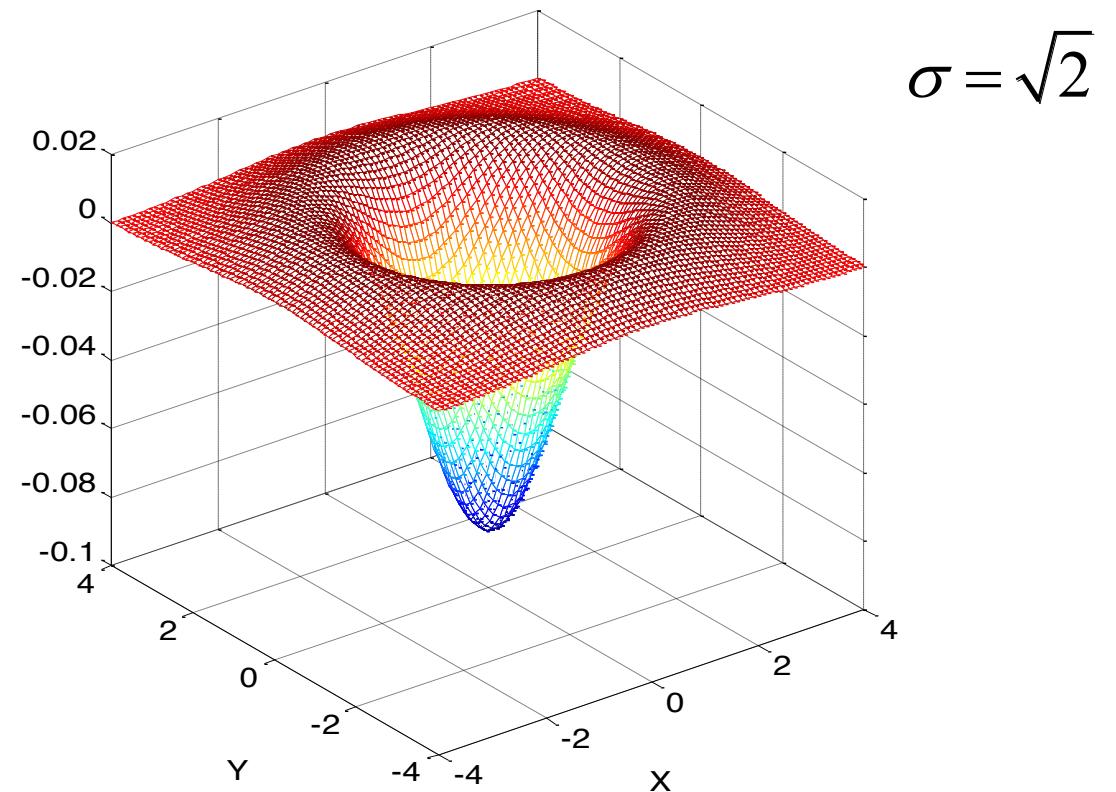
- Sensitive to very fine detail and noise → blur image first
- Responds equally to strong and weak edges
→ suppress zero-crossings with low gradient magnitude



Laplacian of Gaussian

- Filtering of image with Gaussian and Laplacian operators can be combined into convolution with Laplacian of Gaussian (LoG) operator

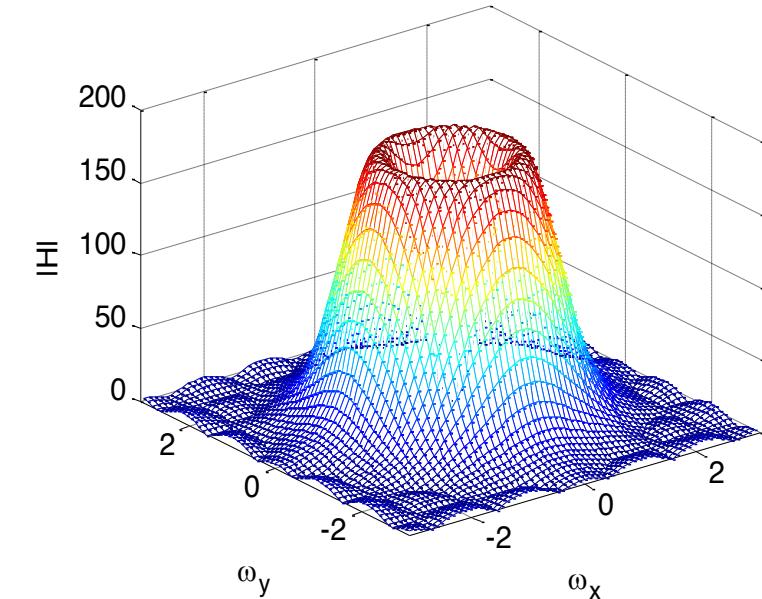
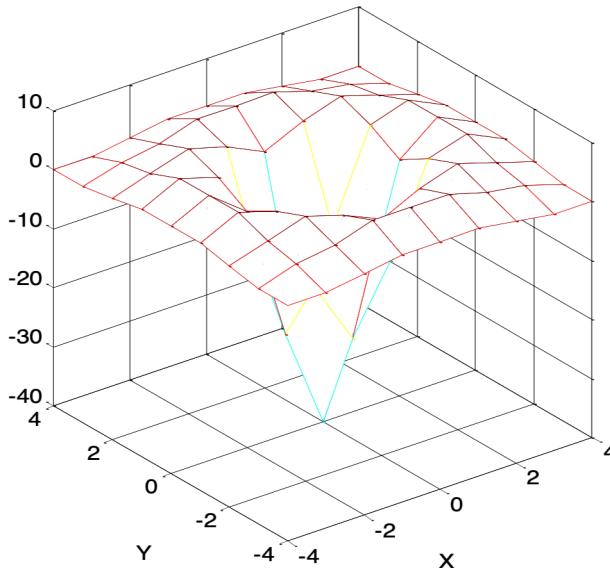
$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$



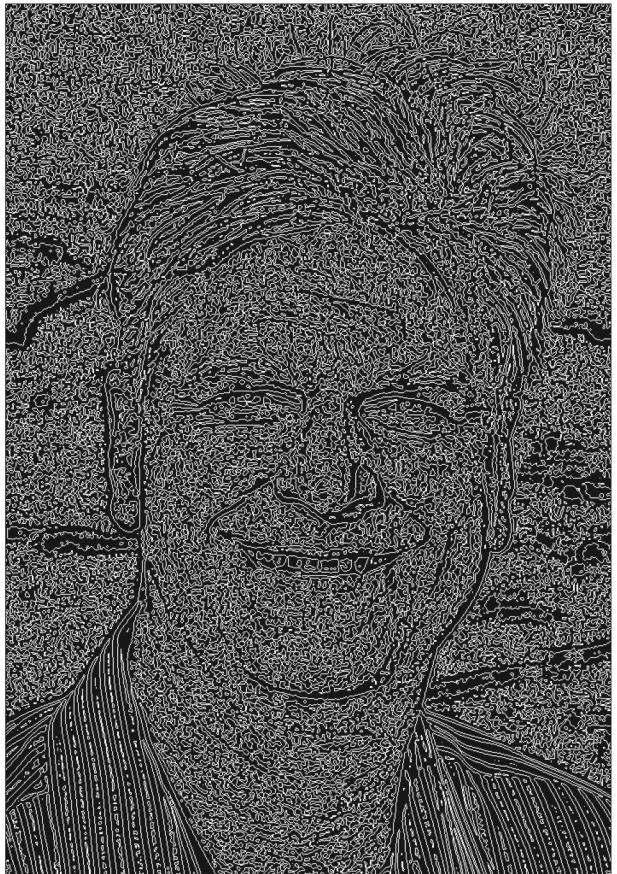
Discrete approximation of Laplacian of Gaussian

$$\sigma = \sqrt{2}$$

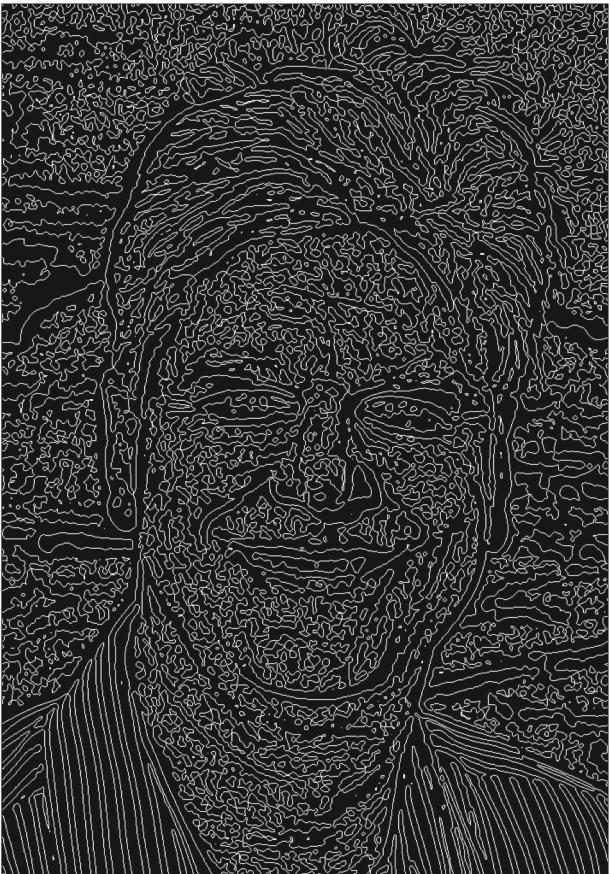
0	0	1	2	2	2	1	0	0
0	2	3	5	5	5	3	2	0
1	3	5	3	0	3	5	3	1
2	5	3	-12	-23	-12	3	5	2
2	5	0	-23	-40	-23	0	5	2
2	5	3	-12	-23	-12	3	5	2
1	3	5	3	0	3	5	3	1
0	2	3	5	5	5	3	2	0
0	0	1	2	2	2	1	0	0



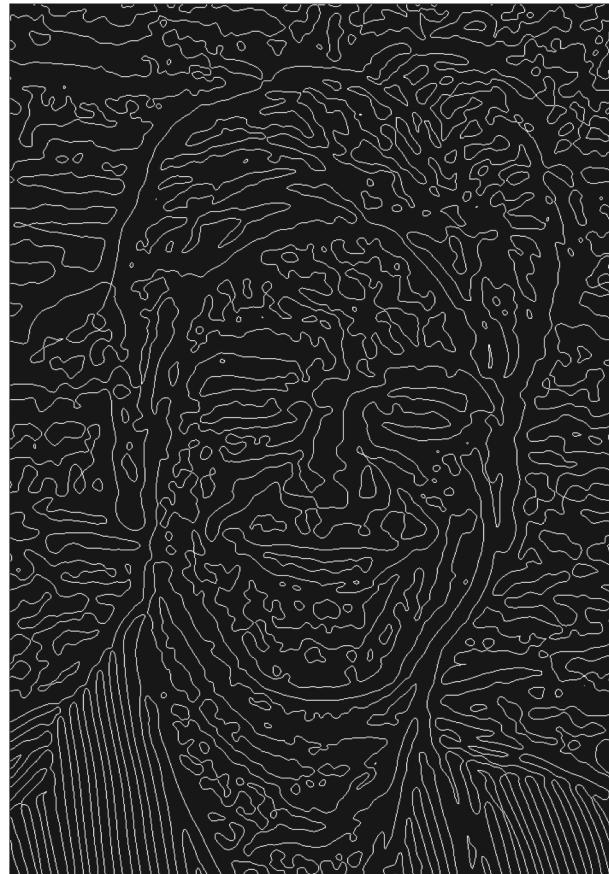
Zero crossings of LoG



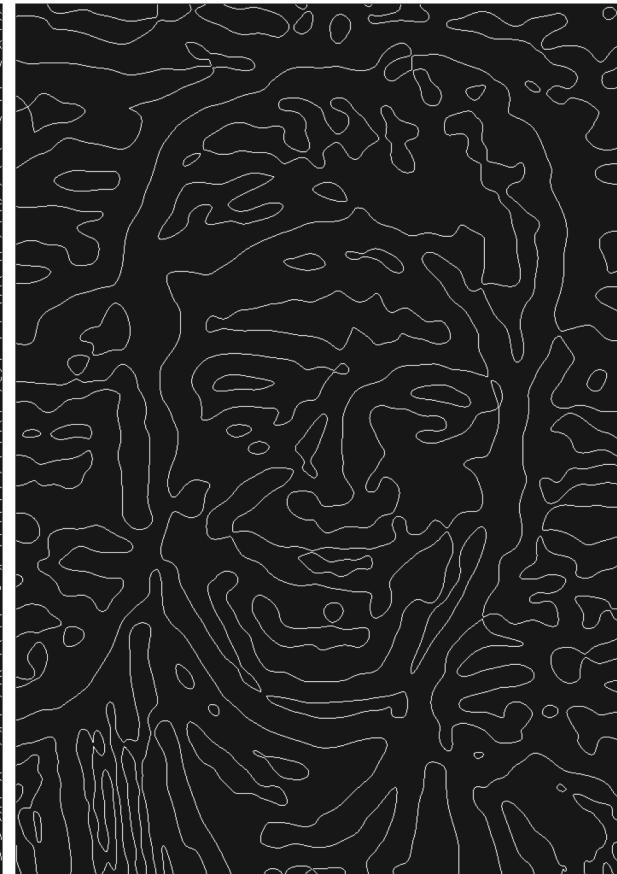
$$\sigma = \sqrt{2}$$



$$\sigma = 2\sqrt{2}$$



$$\sigma = 4\sqrt{2}$$



$$\sigma = 8\sqrt{2}$$



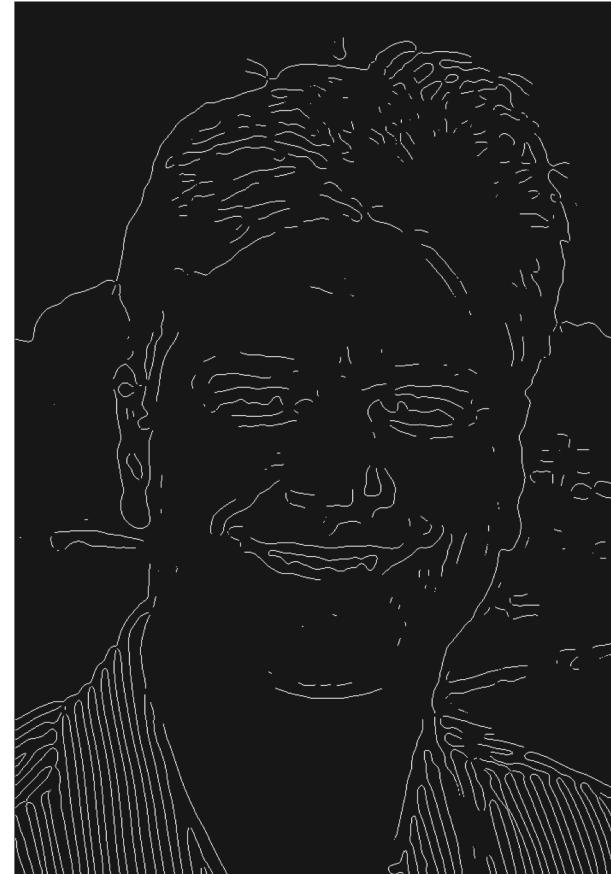
Zero crossings of LoG – gradient-based threshold



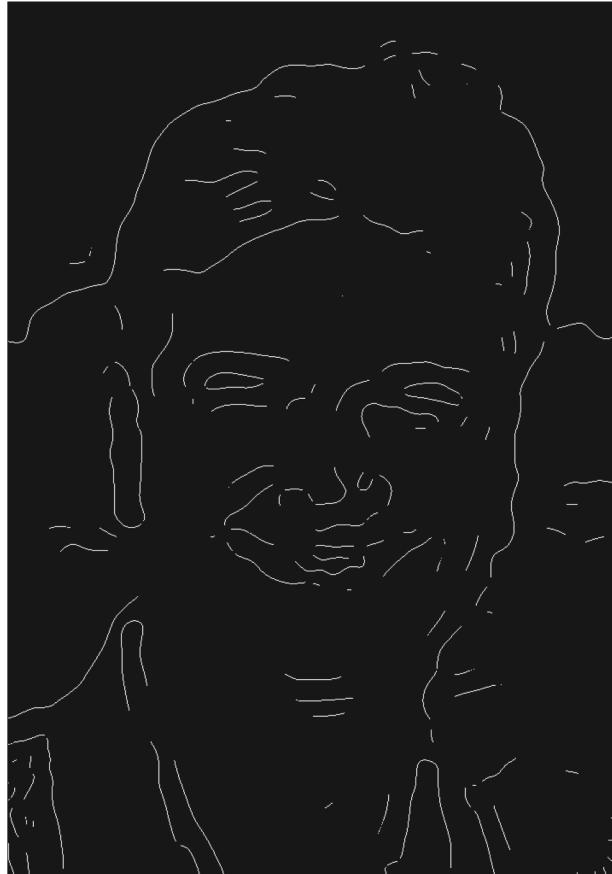
$$\sigma = \sqrt{2}$$



$$\sigma = 2\sqrt{2}$$



$$\sigma = 4\sqrt{2}$$



$$\sigma = 8\sqrt{2}$$



Canny edge detector

1. Smooth image with a Gaussian filter
2. Approximate gradient magnitude and angle (use Sobel, Prewitt . . .)

$$M[x, y] \approx \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

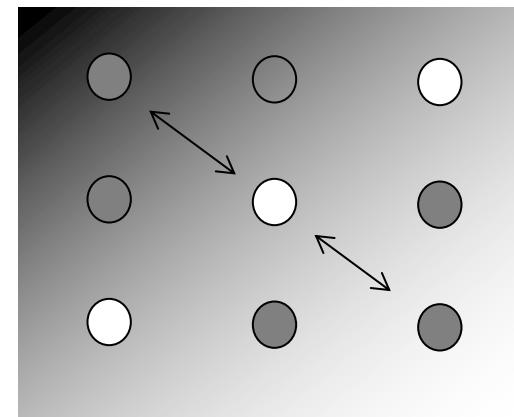
$$\alpha[x, y] \approx \tan^{-1} \left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$$

3. Apply nonmaxima suppression to gradient magnitude
4. Double thresholding to detect strong and weak edge pixels
5. Reject weak edge pixels not connected with strong edge pixels

[Canny, IEEE Trans. PAMI, 1986]

Canny nonmaxima suppression

- Quantize edge normal to one of four directions:
horizontal, -45° , vertical, $+45^\circ$
- If $M[x,y]$ is smaller than either of its neighbors in edge normal direction
→ suppress; else keep.



[Canny, IEEE Trans. PAMI, 1986]

Canny thresholding and suppression of weak edges

- Double-thresholding of gradient magnitude

$$\text{Strong edge: } M[x, y] \geq \theta_{high}$$

$$\text{Weak edge: } \theta_{high} > M[x, y] \geq \theta_{low}$$

- Typical setting: $\theta_{high}/\theta_{low} = 2...3$
- Region labeling of edge pixels
- Reject regions without strong edge pixels

[Canny, IEEE Trans. PAMI, 1986]

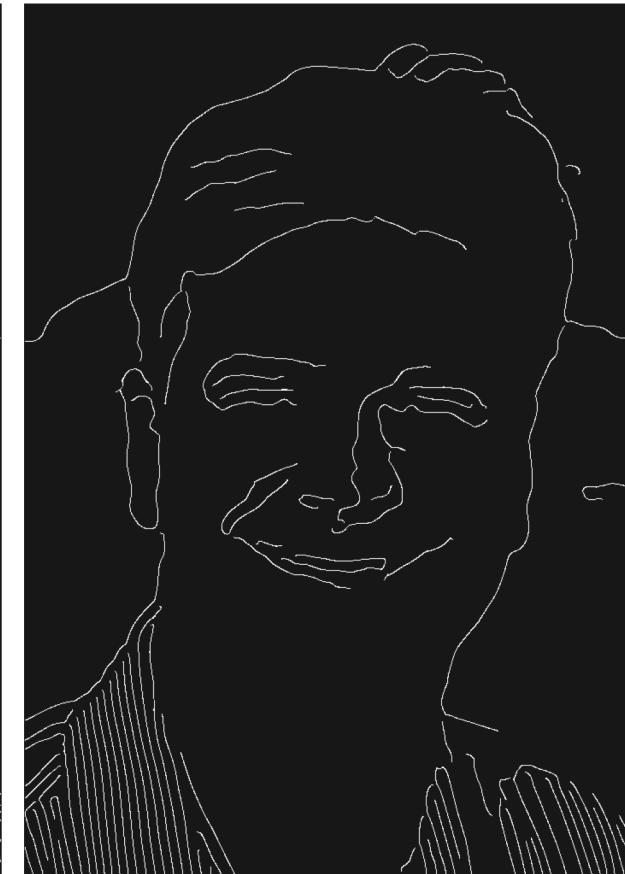
Canny edge detector



$$\sigma = \sqrt{2}$$



$$\sigma = 2\sqrt{2}$$

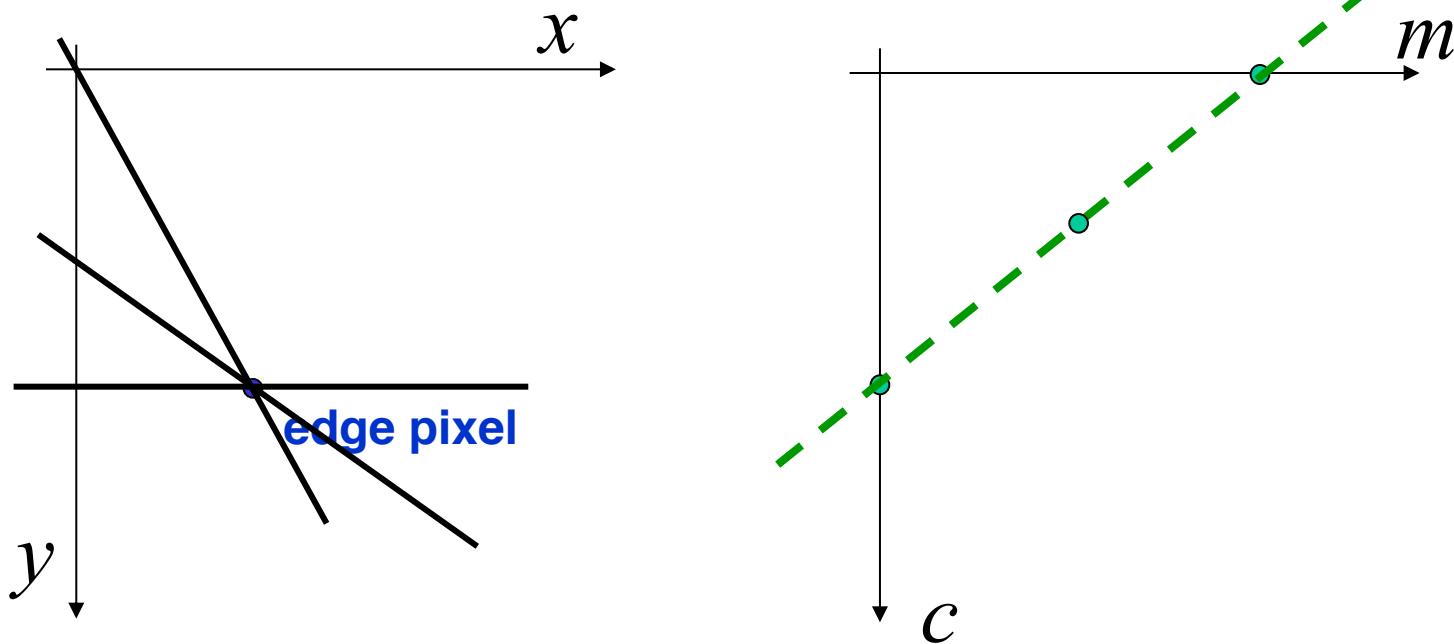


$$\sigma = 4\sqrt{2}$$



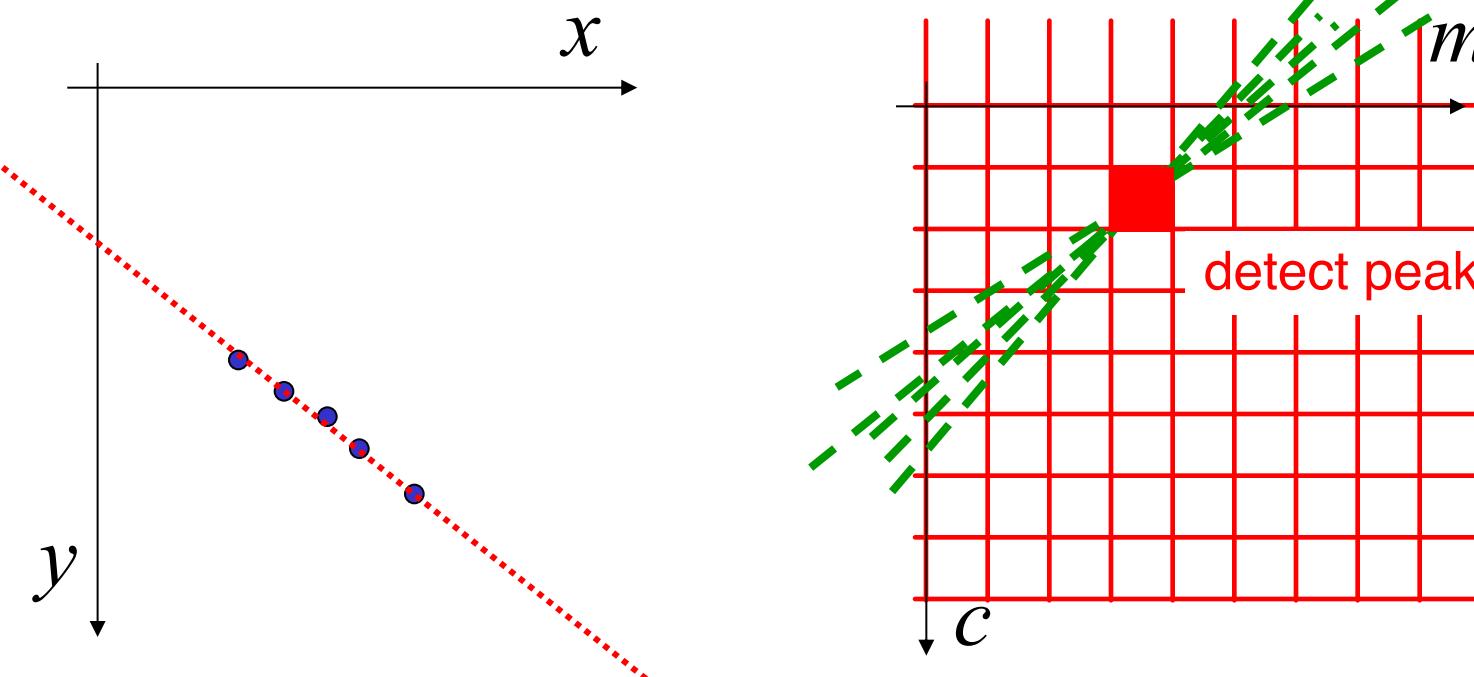
Hough transform

- Problem: fit a straight line (or curve) to a set of edge pixels
- Hough transform (1962): generalized template matching technique
- Consider detection of straight lines $y = mx + c$



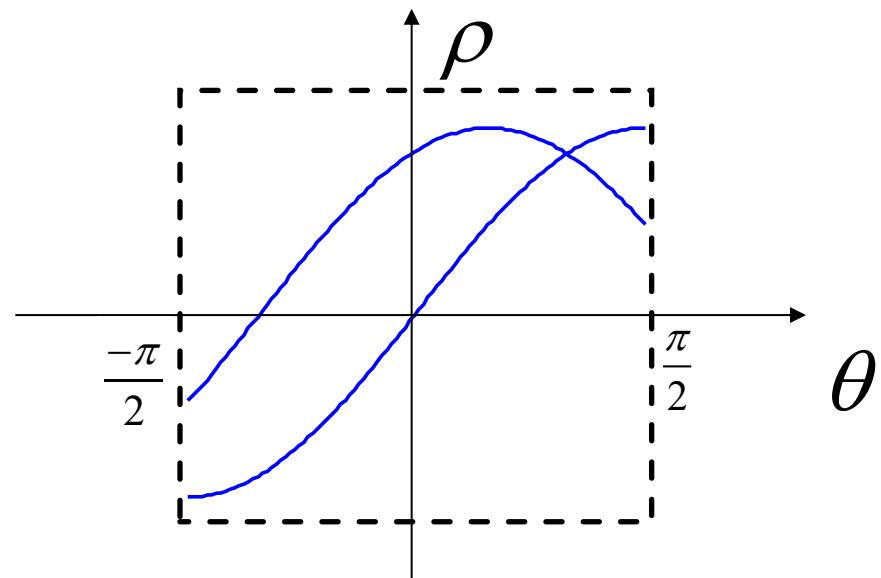
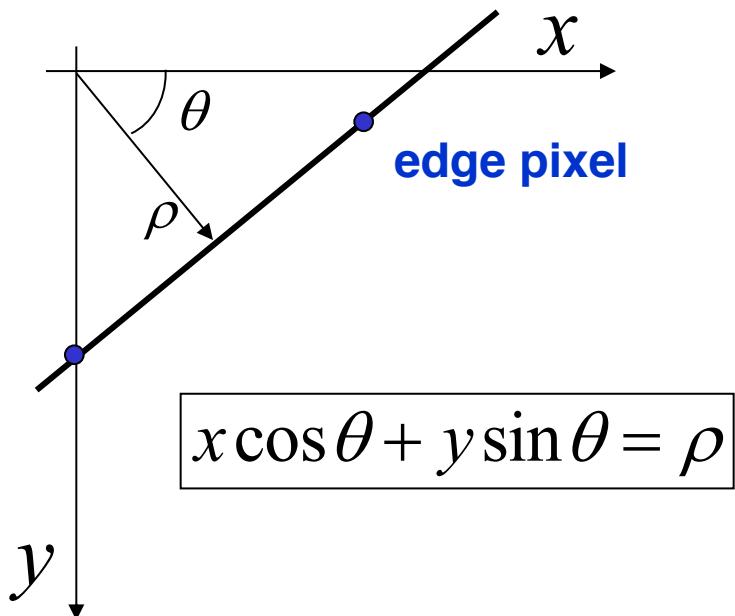
Hough transform (cont.)

- Subdivide (m,c) plane into discrete “bins,” initialize all bin counts by 0
- Draw a line in the parameter space $[m,c]$ for each edge pixel $[x,y]$ and increment bin counts along line.
- Detect peak(s) in $[m,c]$ plane



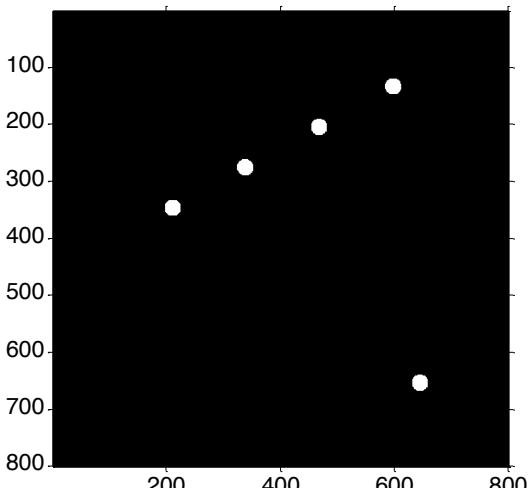
Hough transform (cont.)

- Alternative parameterization avoids infinite-slope problem

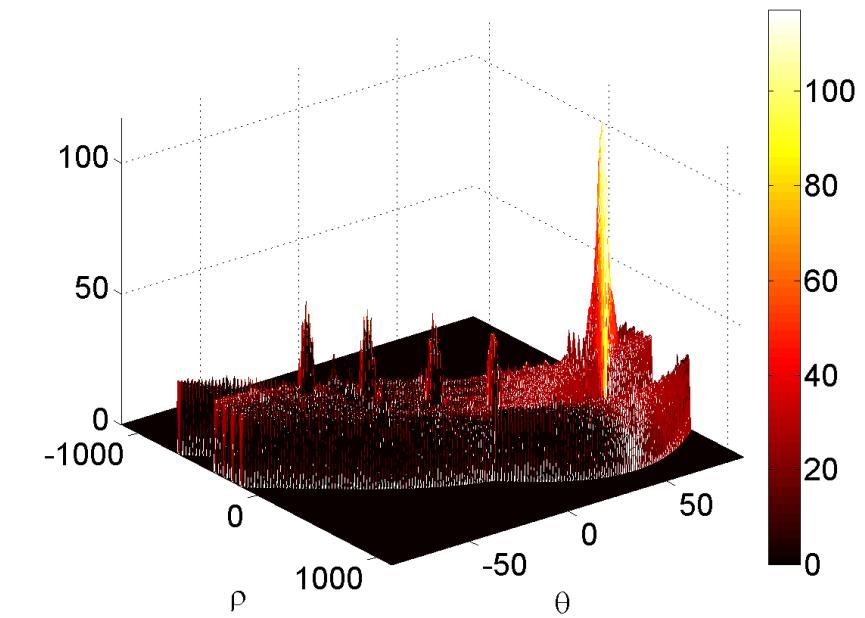
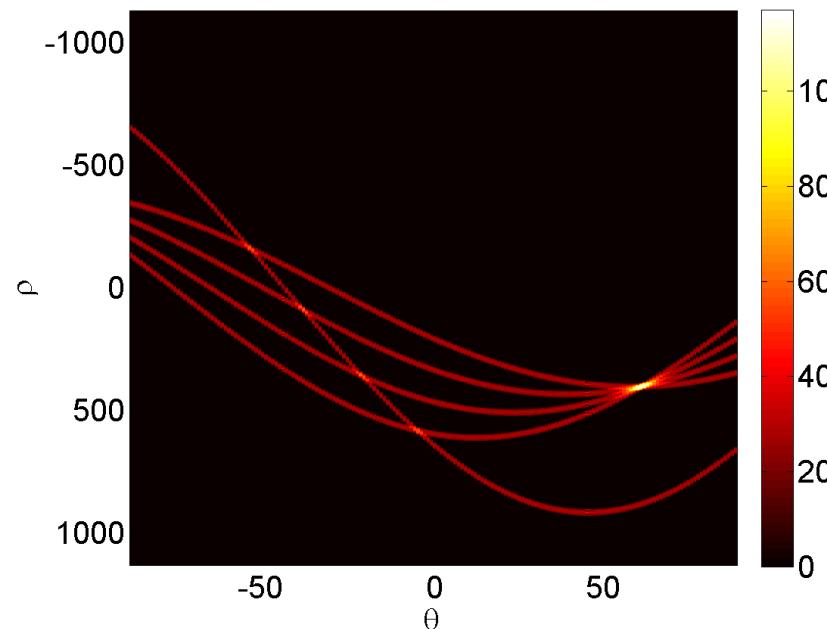


- Similar to Radon transform

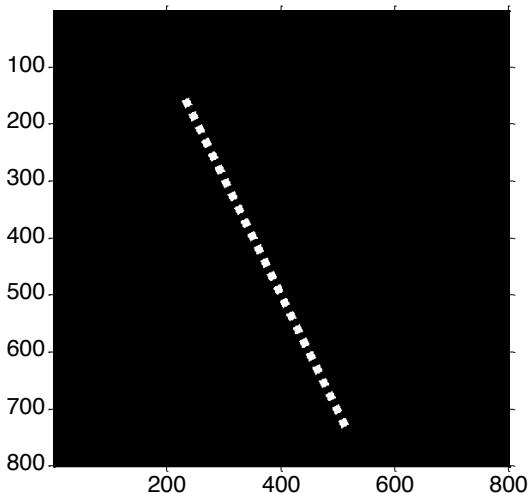
Hough transform example



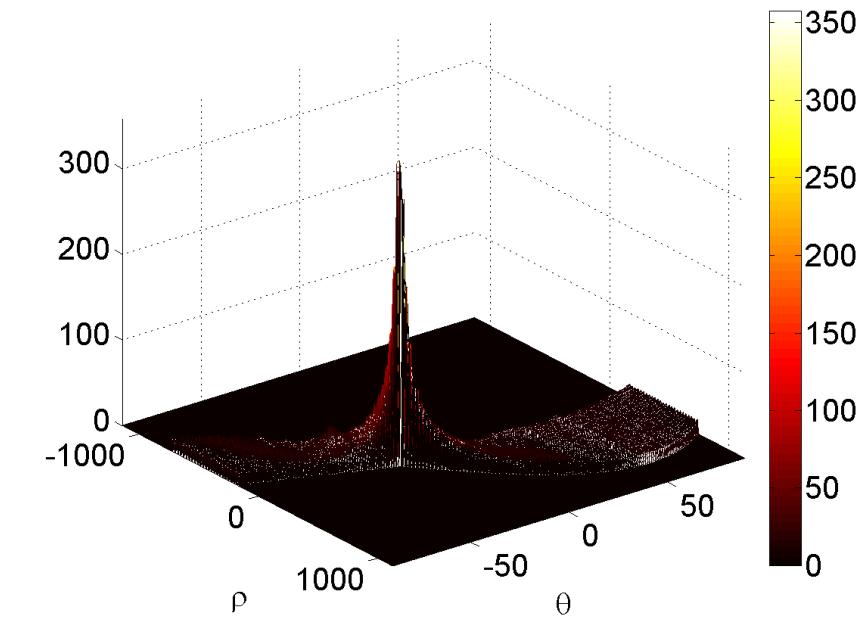
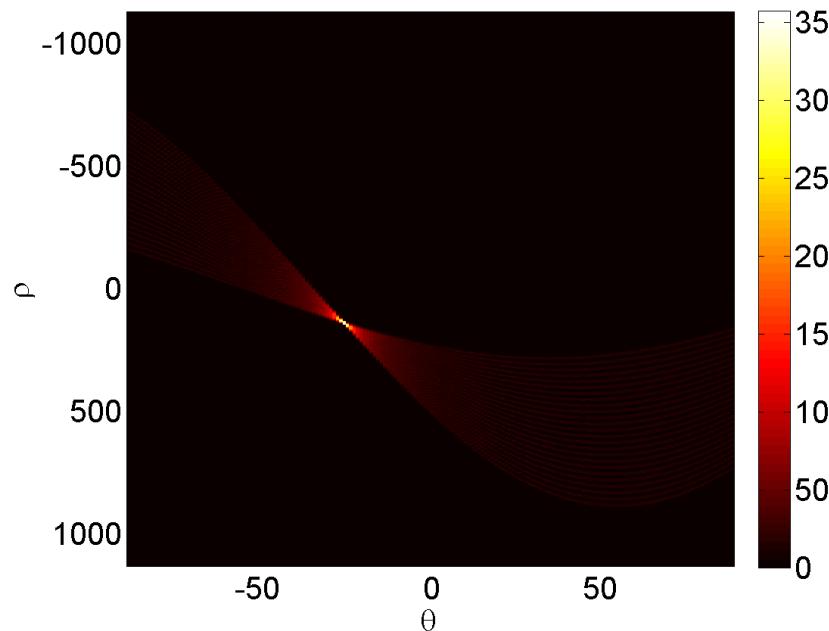
Original image



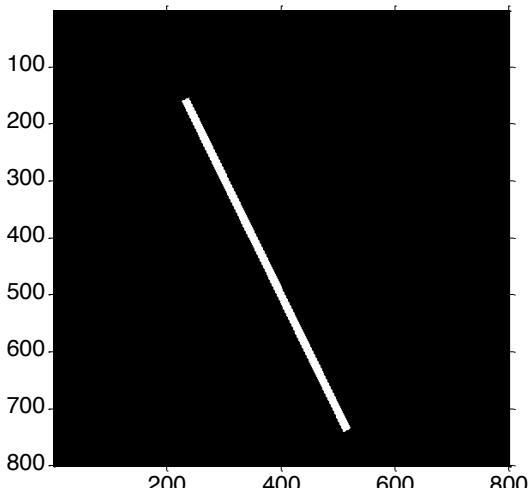
Hough transform example



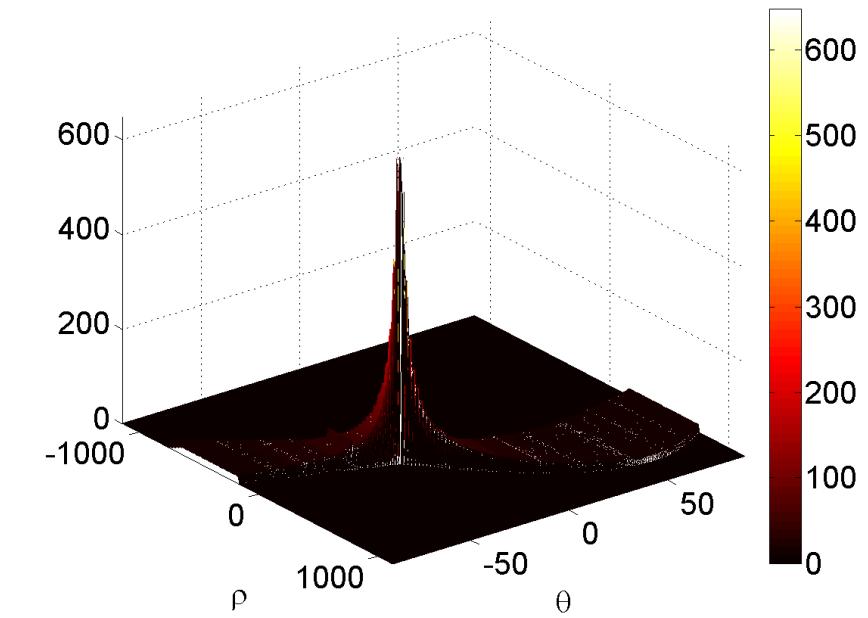
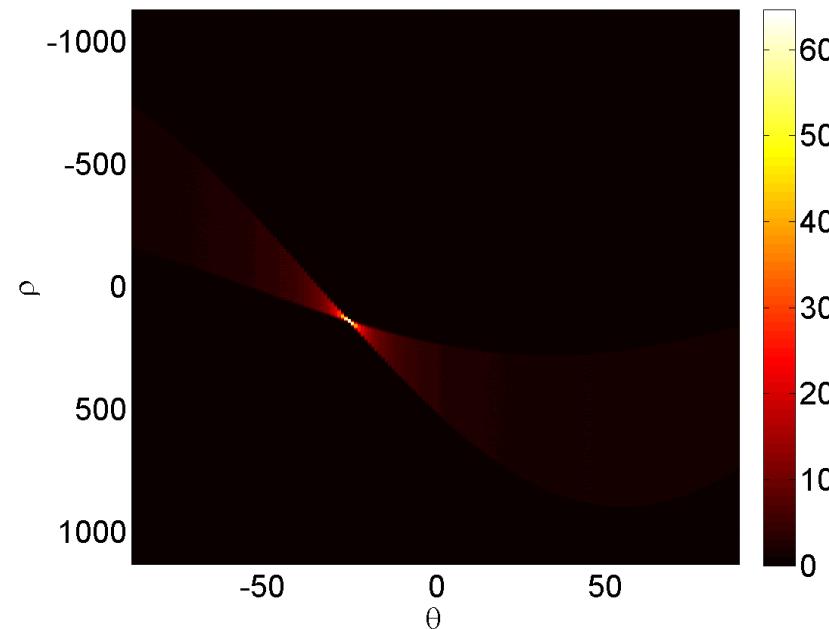
Original image



Hough transform example



Original image



Hough transform example

to decompose an image into layers and compressing them. Usually, layers with large details are untouched or even enhanced. [13] presents a method based on a multistage Retinex theory of color vision. [14] attempts to incorporate traditional photographic techniques to domain for reproducing HDR images. [15] proposes a novel method by adjusting the local histogram through the manipulation of the domain in the logarithmic space. More recent research has been little published research to rendering HDR images on the GPU [17,18] and the...

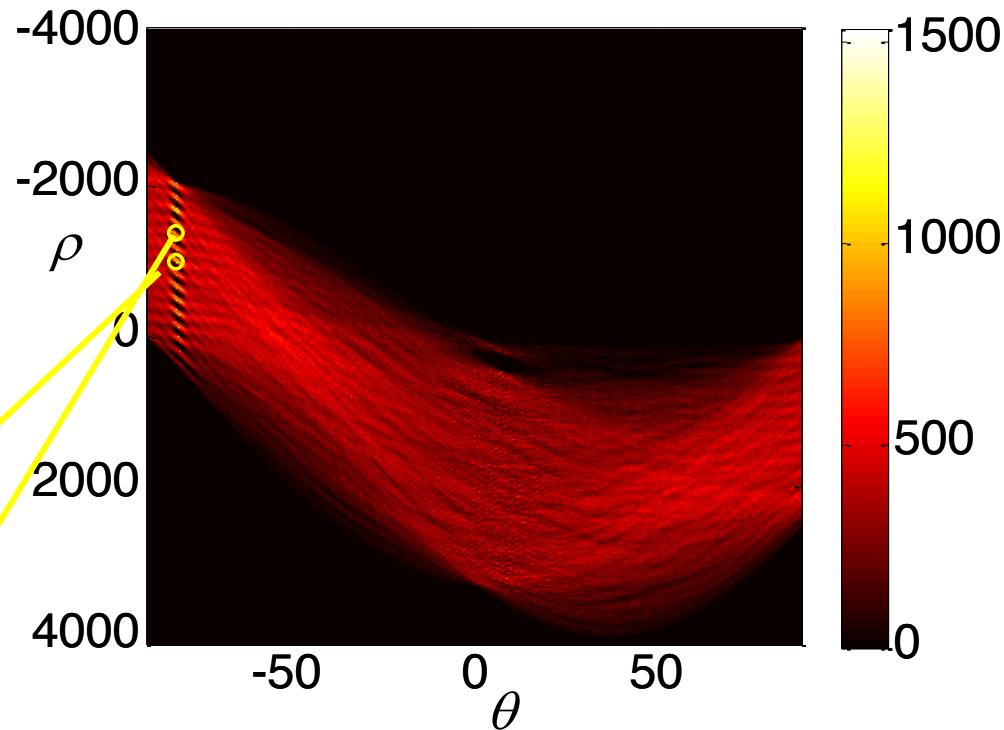
De-skewed Paper

to decompose an image into layers and compressing them. Usually, layers with large details are untouched or even enhanced. [13] presents a method based on a multistage Retinex theory of color vision. [14] attempts to incorporate traditional photographic techniques to domain for reproducing HDR images. [15] proposes a novel method by adjusting the local histogram through the manipulation of the domain in the logarithmic space. More recent research has been little published research to rendering HDR images on the GPU [17,18] and the...

-81.8 deg

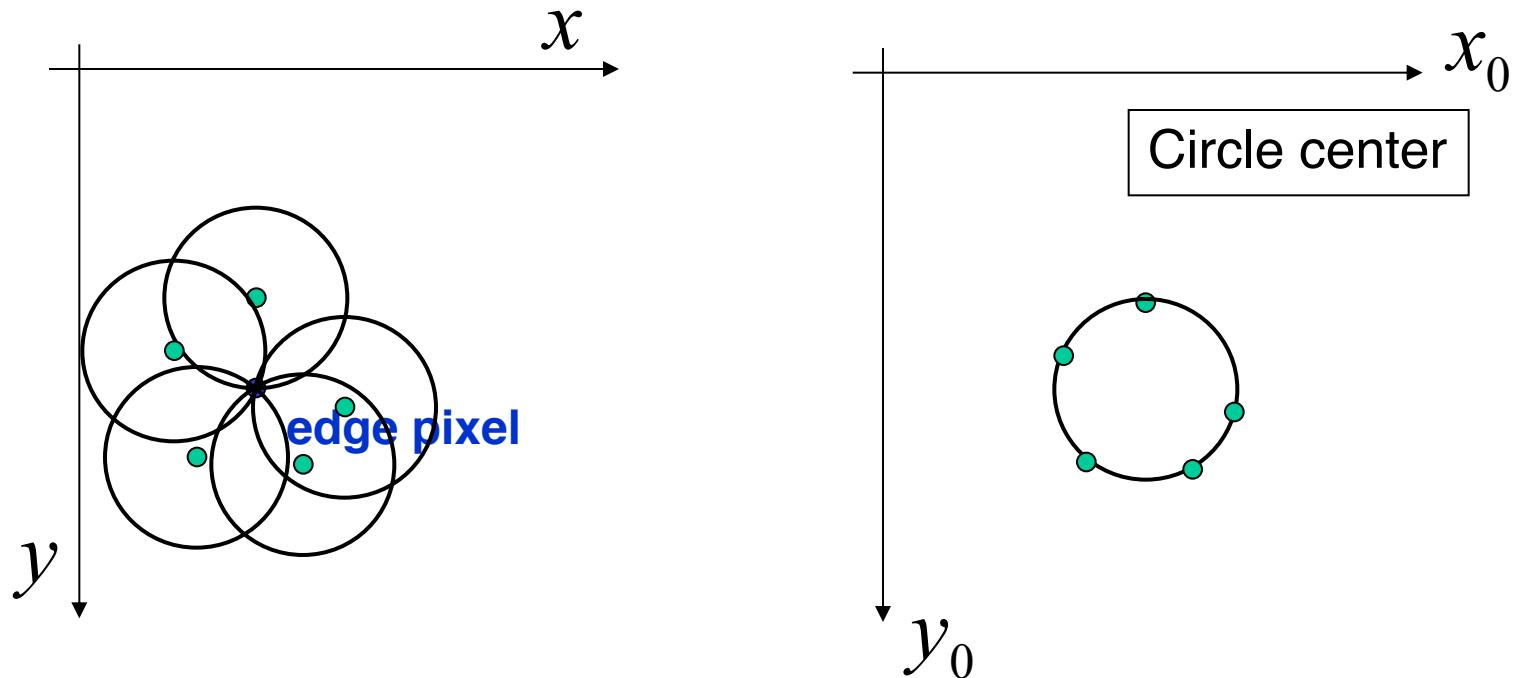
-81.8 deg

Global thresholding



Circle Hough Transform

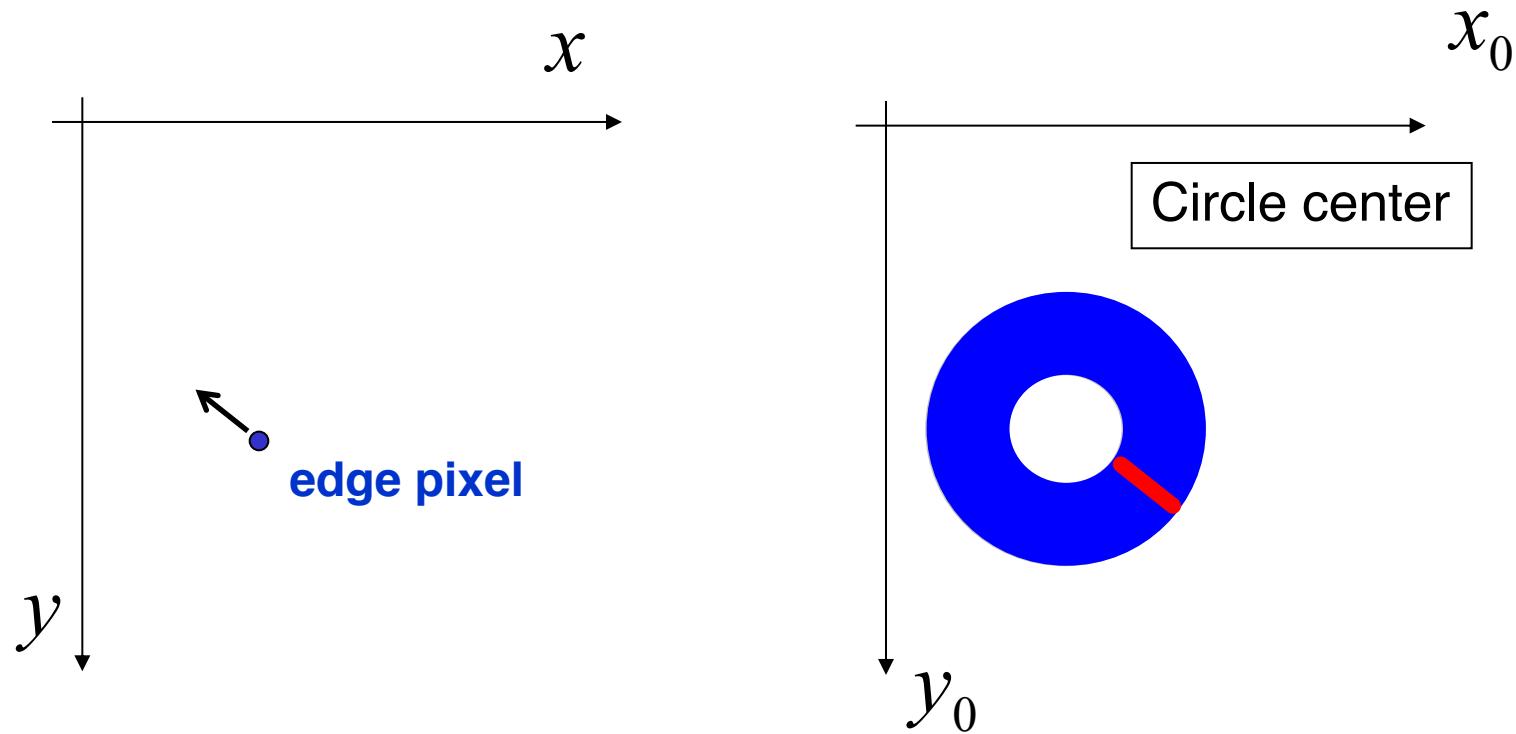
- Find circles of fixed radius r



- Equivalent to convolution (template matching) with a circle

Circle Hough Transform for unknown radius

- 3-d Hough transform for parameters (x_0, y_0, r)
- 2-d Hough transform aided by edge orientation \rightarrow “spokes” in parameter space

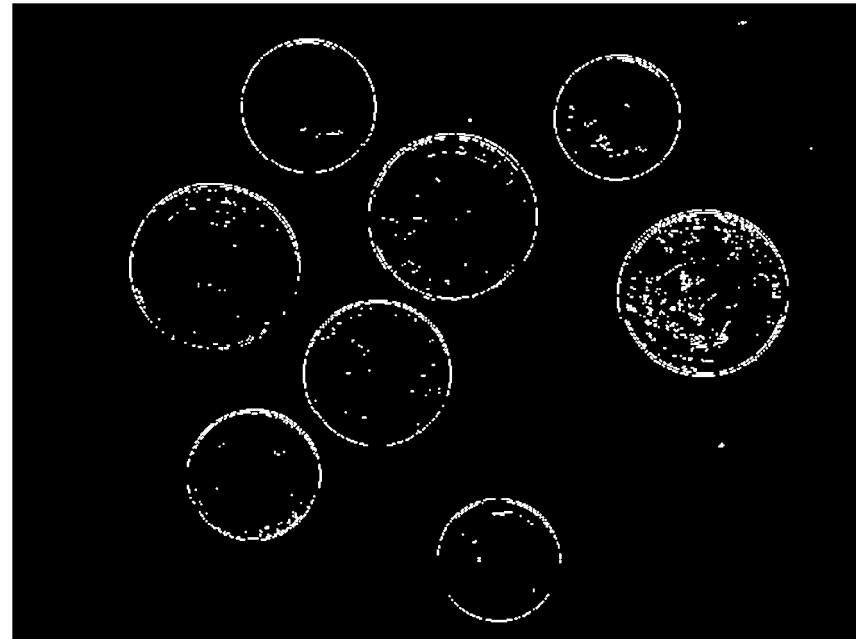


Example: circle detection by Hough transform

Original
coins image



Prewitt edge detection



Detected circles

