Digital Image Processing (CSE 478)
Lecture10: Edges and Hough Transform

Vineet Gandhi

Center for Visual Information Technology (CVIT), IIIT Hyderabad

Today’s Lecture

- Edge models, detection
- Advanced edge detection methods
- HOG (Histogram of Oriented Gradients) features
- Hough Transform

Edge Models

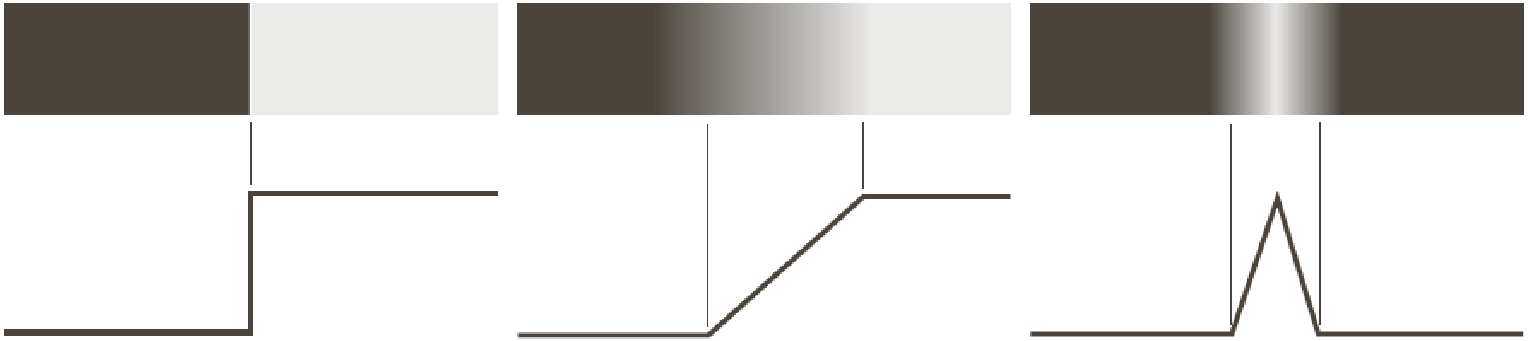
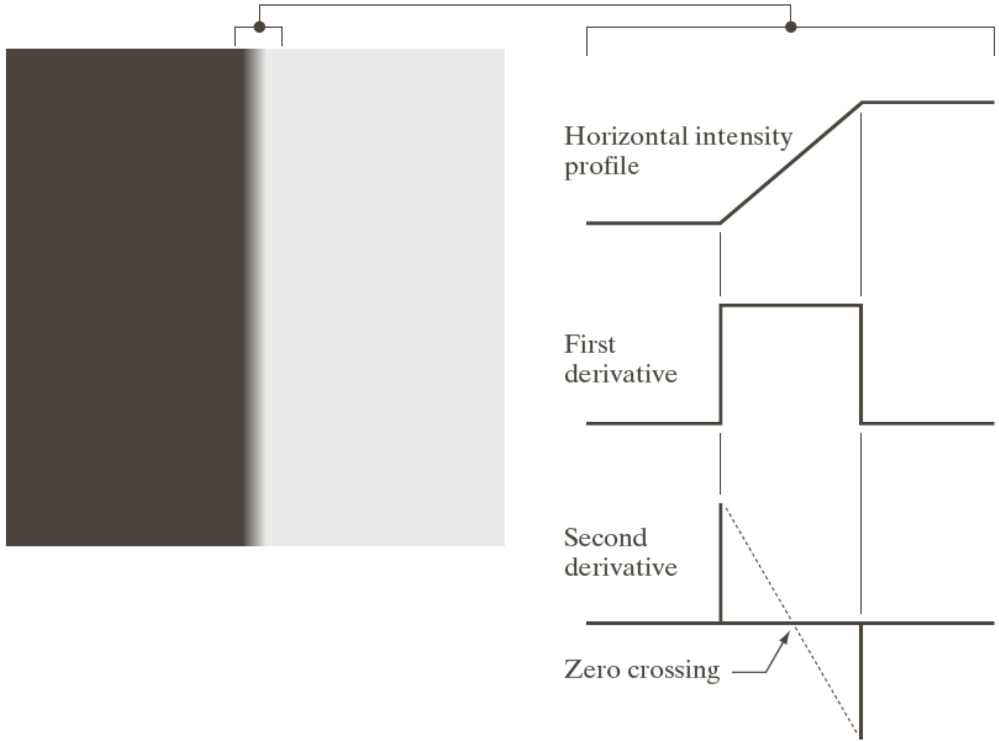


FIGURE 10.8 From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.

Edge Models



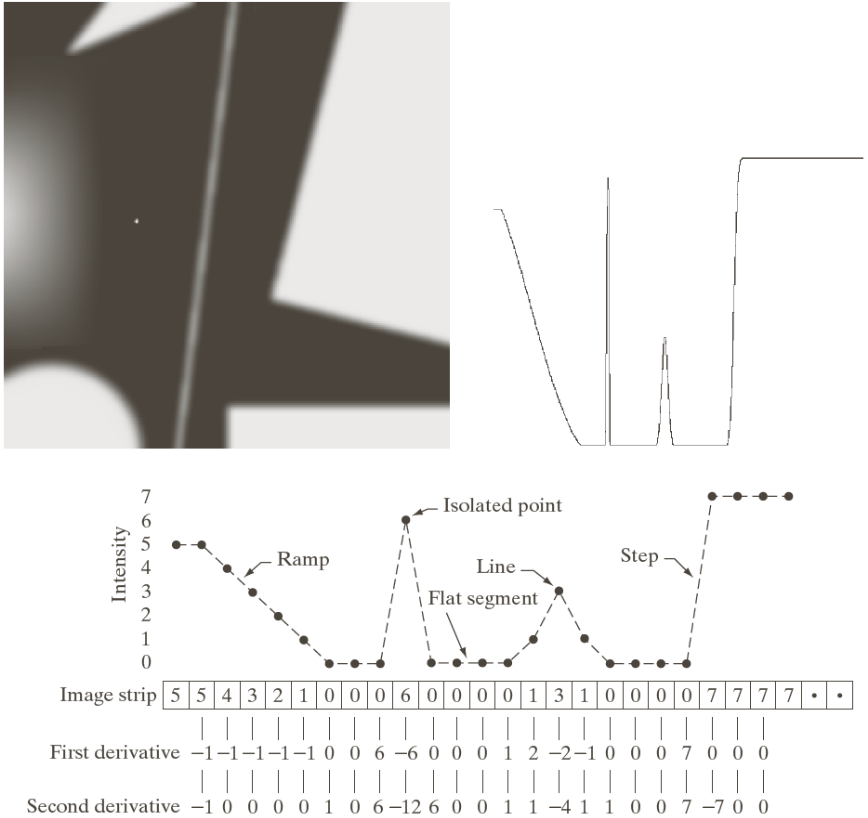
Horizontal intensity profile
First derivative
Second derivative
Zero crossing

Discrete difference as Edge Detection

- Local changes in intensity → derivatives

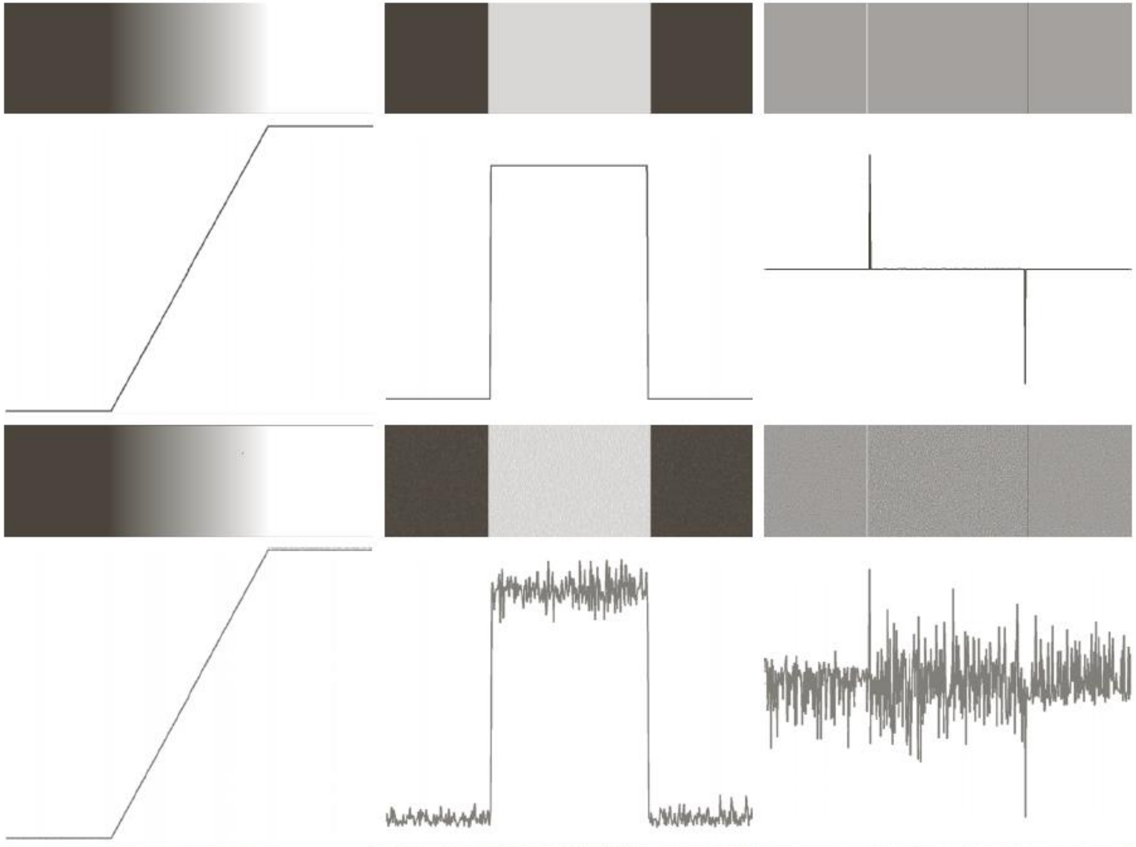
	First order derivative	Second order derivative
Constant intensity	zero	zero
Onset of step or ramp	non zero	non zero
Along ramp	non zero	zero

Edge Detection

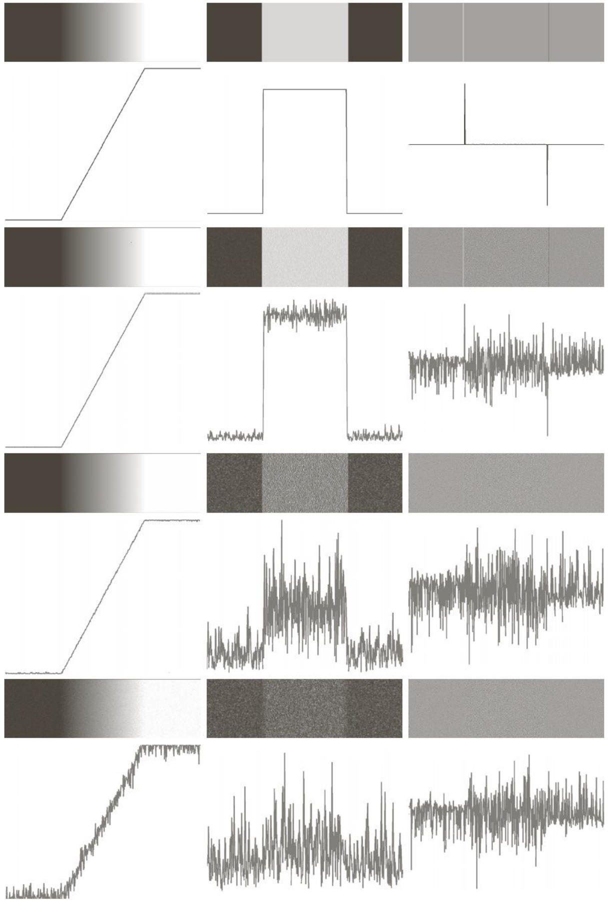


- 1st order derivative generally produces thick edges
- 2nd order derivative has a stronger response to fine details
- 2nd order derivative produce a double edge response
- The sign of 2nd order derivative can be used to determine whether the edge is a transition from light to dark or dark to light

Edge Models (effect of noise)



Edge Models (effect of noise)



Gaussian noise of zero mean and standard deviation 0.0, 0.1, 1.0 and 10.0

Lets explore some more robust edge detectors!

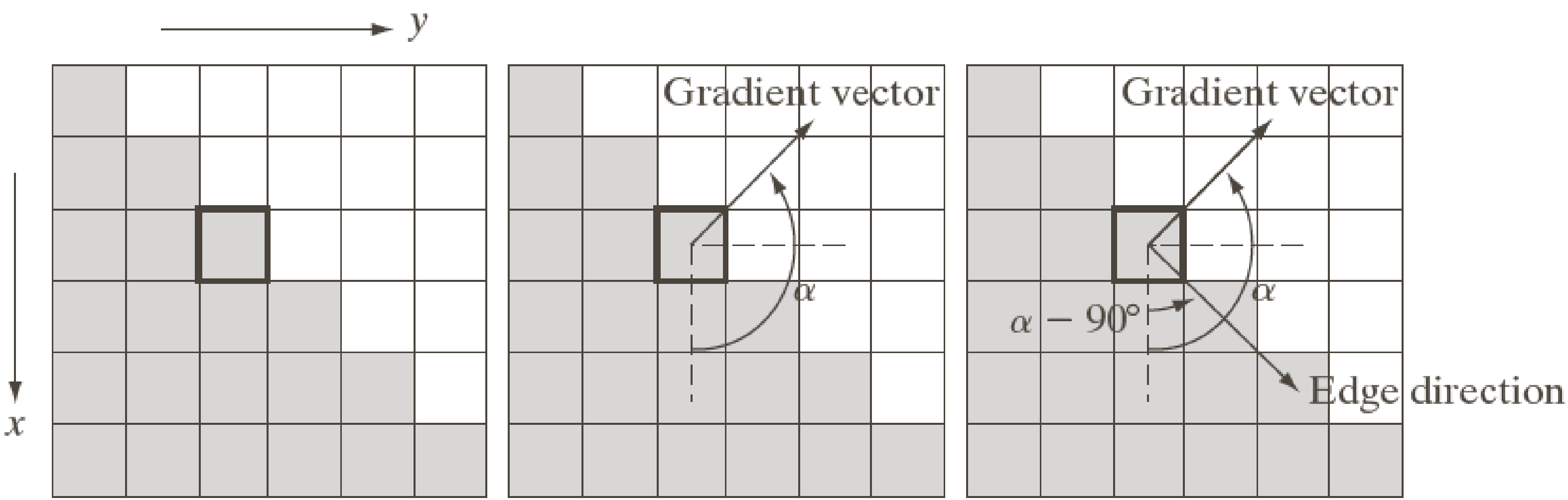
The image gradient and its properties

$$\nabla f = \mathrm{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

$$M(x, y) = \mathrm{mag}(\nabla f) = \sqrt{{g_x}^2 + {g_y}^2}$$

$$\alpha(x, y) = \tan^{-1}\left(\frac{g_x}{g_y}\right)$$

The image gradient and its properties



$$\nabla f = \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

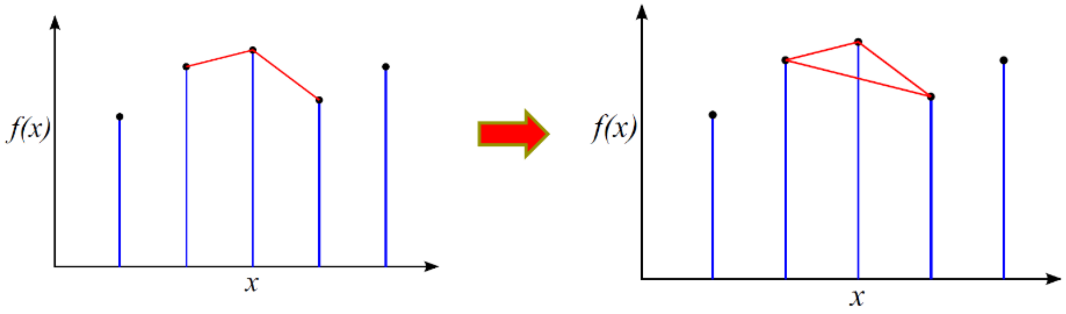
$$M(x, y) = \text{mag}(\nabla f) = \sqrt{{g_x}^2 + {g_y}^2}$$

$$\alpha(x, y) = \tan^{-1}\left(\frac{g_x}{g_y}\right)$$

Gradient operators

−1
1

−1	1

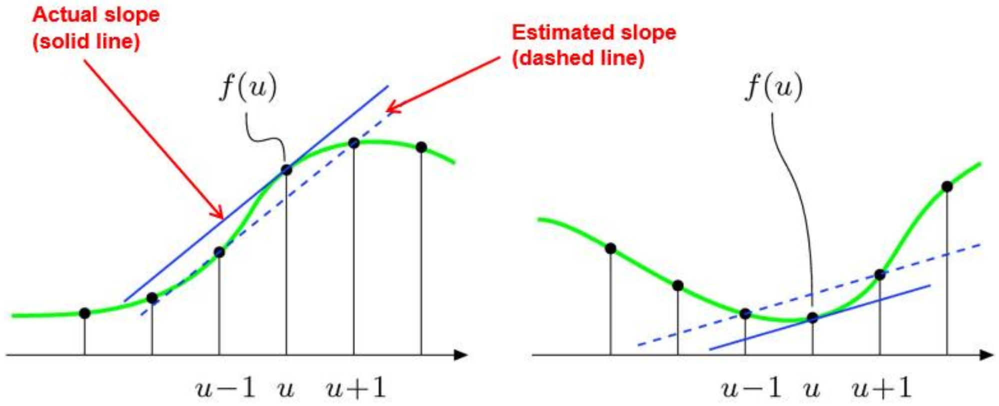


- Left and right slope may not be same

- Solution?

  Take average of left and right slope

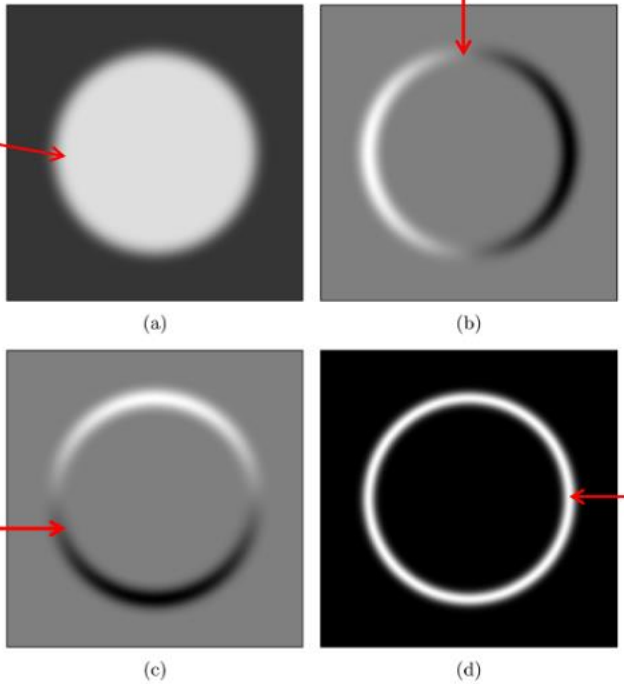
Gradient operators



$$\frac{df}{du}(u) \approx \frac{f(u+1) - f(u-1)}{2} = 0.5 \cdot \big(f(u+1) - f(u-1)\big)$$

Gradient operators

$$H_x^D = \begin{bmatrix} -0.5 & \mathbf{0} & 0.5 \end{bmatrix} = 0.5 \cdot \begin{bmatrix} -1 & \mathbf{0} & 1 \end{bmatrix}$$



$$H_y^D = \begin{bmatrix} -0.5 \\ \mathbf{0} \\ 0.5 \end{bmatrix} = 0.5 \cdot \begin{bmatrix} -1 \\ \mathbf{0} \\ 1 \end{bmatrix}$$

Source: Emmanuel agu

Common Gradient operators

−1	
1	

−1	1

−1	0		0	−1
0	1		1	0

Roberts

−1	−1	−1		−1	0	1
0	0	0		−1	0	1
1	1	1		−1	0	1

Prewitt

−1	−2	−1		−1	0	1
0	0	0		−2	0	2
1	2	1		−1	0	1

Sobel

- Symmetric filters take into account the nature of the data on opposite sides of the center point

Gradient operators (Prewitt)

- Finite differences sensitive to noise
- Derivative more robust if derivative computations are averaged in an neighbourhood

$$H_x^P = \frac{1}{3}\begin{bmatrix}1\\1\\1\end{bmatrix} * [0.5 \quad 0 \quad -0.5] = \frac{1}{6}\begin{bmatrix}1 & 0 & -1\\1 & 0 & -1\\1 & 0 & -1\end{bmatrix}$$

Derivative in *x* direction

Average in *y* direction

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Prewitt

Separability!

Gradient operators (Sobel)

$$H_x^S = \frac{1}{4}\begin{bmatrix}1\\2\\1\end{bmatrix} * [0.5 \quad 0 \quad -0.5] = \frac{1}{8}\begin{bmatrix}1 & 0 & -1\\2 & 0 & -2\\1 & 0 & -1\end{bmatrix}$$

$$H_y^S = \frac{1}{4}[1 \quad 2 \quad 1] * \begin{bmatrix}0.5\\0\\-0.5\end{bmatrix} = \frac{1}{8}\begin{bmatrix}1 & 2 & 1\\0 & 0 & 0\\-1 & -2 & -1\end{bmatrix}$$

Average in *x* direction

Derivative in *y* direction

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

Gradient operators

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

Gradient operators

a b
c d

FIGURE 10.16
(a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1].
(b) $|g_x|$, the component of the gradient in the x-direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image.
(c) $|g_y|$, obtained using the mask in Fig. 10.14(g).
(d) The gradient image, $|g_x| + |g_y|$.

Gradient operators

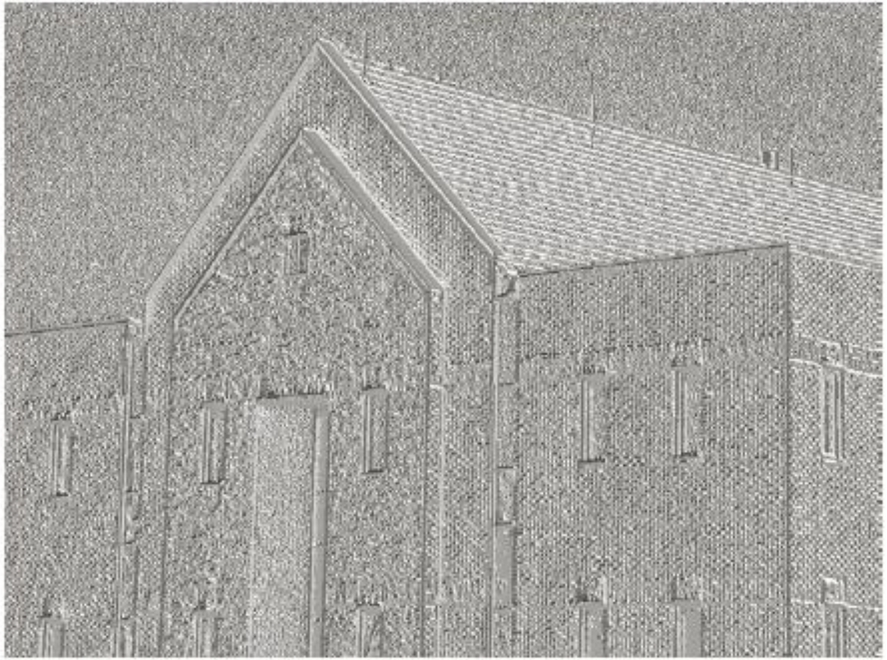
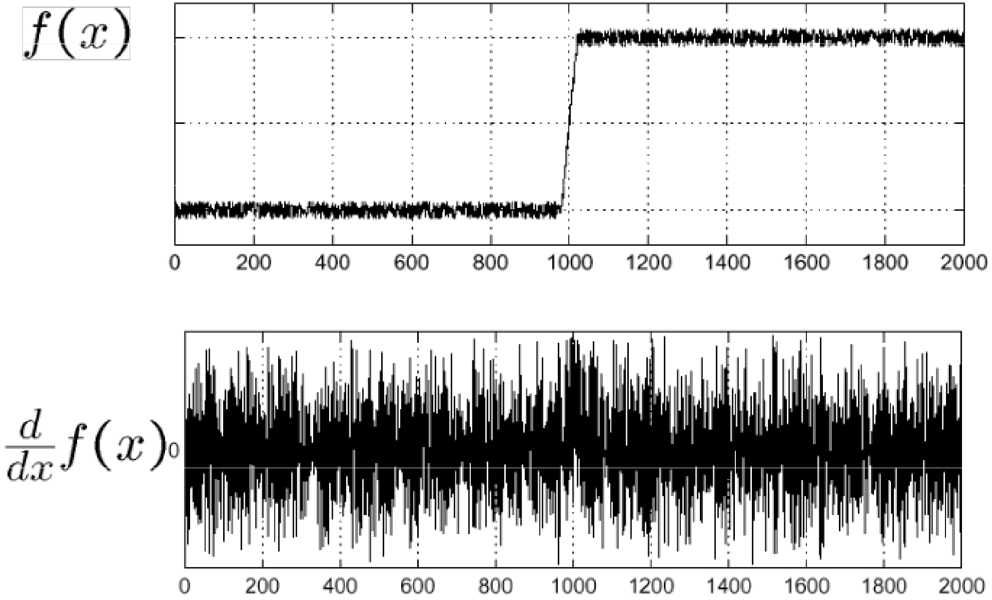
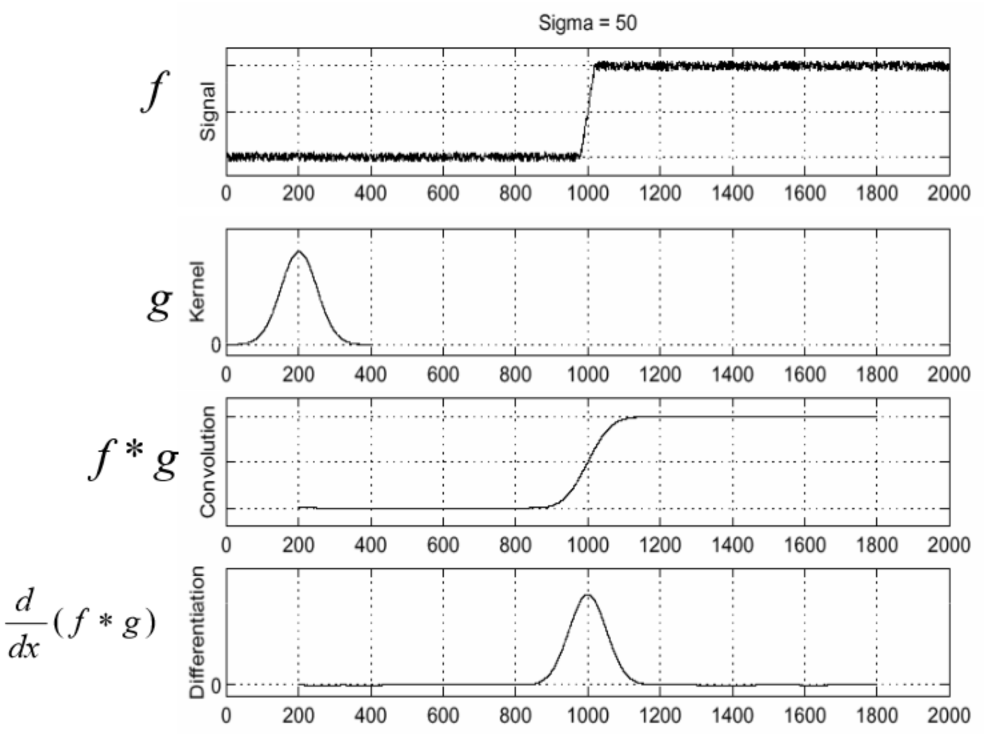


FIGURE 10.17 Gradient angle image computed using Eq. (10.2-11). Areas of constant intensity in this image indicate that the direction of the gradient vector is the same at all the pixel locations in those regions.

Where is the edge?



Solution: smooth first



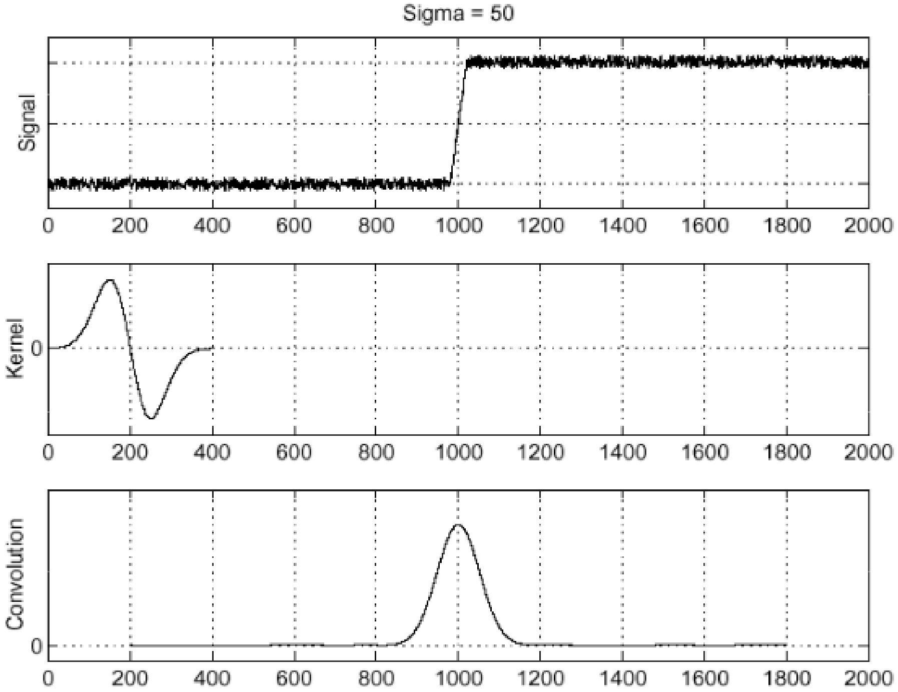
Can do it faster!

- Differentiation in convolution and convolution is associative

f

$\frac{d}{dx} g$

$f * \frac{d}{dx} g$



$$\frac{d}{dx}(f * g) = f * \frac{d}{dx} g$$

Gradient operators (with smoothing)

a b
c d

FIGURE 10.18 Same sequence as in Fig. 10.16, but with the original image smoothed using a 5 × 5 averaging filter prior to edge detection.

Gradient operators

a b

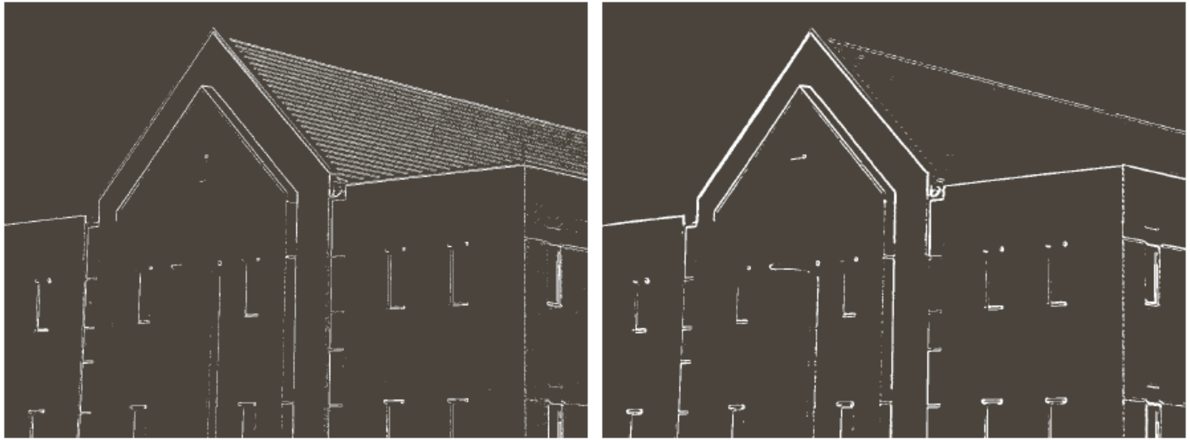
FIGURE 10.19 Diagonal edge detection. (a) Result of using the mask in Fig. 10.15(c). (b) Result of using the mask in Fig. 10.15(d). The input image in both cases was Fig. 10.18(a).

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

Gradient + Thresholding



a b

FIGURE 10.20 (a) Thresholded version of the image in Fig. 10.16(d), with the threshold selected as 33% of the highest value in the image; this threshold was just high enough to eliminate most of the brick edges in the gradient image. (b) Thresholded version of the image in Fig. 10.18(d), obtained using a threshold equal to 33% of the highest value in that image.

Gradient + Thresholding

Original Image

Gradient magnitude, using Sobel operator

```
[fx,fy] = gradient(I);
fmag = (fx.^2 + fy.^2) .^ .5;
```

Image courtesy: William Hoff

Gradient + Thresholding

- We can threshold edge magnitudes to produce binary image

Low threshold

High threshold

Non maximal suppression

Image courtesy: William Hoff

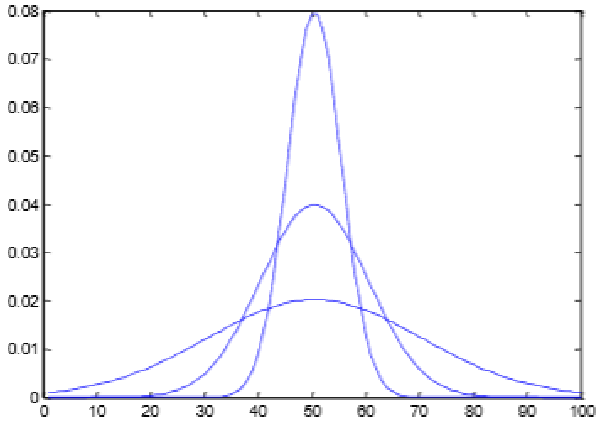
How much to smooth?

Image courtesy: William Hoff

Scale space

- The space of images created by applying a series of operators of different scales
- Different size Gaussians:

$$G_\sigma(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}}$$



Lets explore some advanced edge detection techniques

Advanced edge detectors (Marr Hildreth)

- Marr and Hildreth proposed the filter $\nabla^2 G$
  - ∇^2 is the Laplacian operator
  - G is 2D gaussian

$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

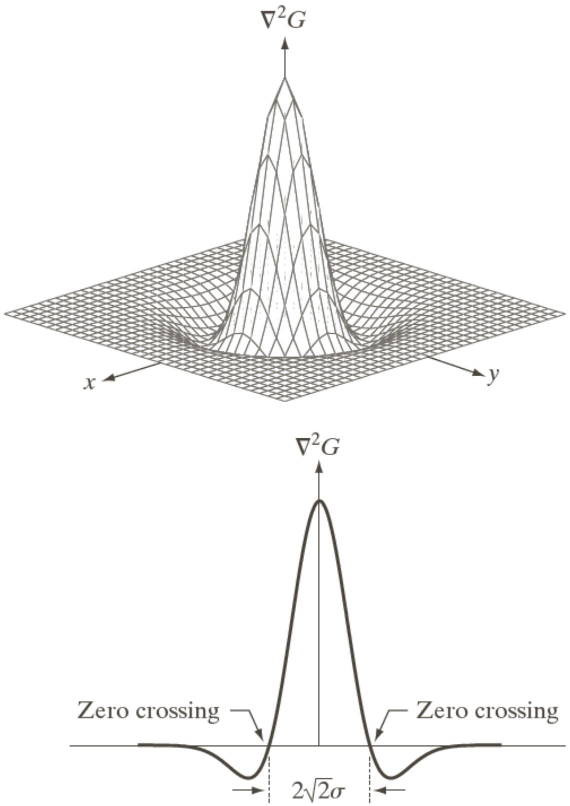
$$\nabla^2 G(x,y) = \frac{\partial^2 G(x,y)}{\partial x^2} + \frac{\partial^2 G(x,y)}{\partial y^2} = \frac{\partial}{\partial x}\left[\frac{-x}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\right] + \frac{\partial}{\partial y}\left[\frac{-y}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\right]$$

$$= \left[\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} + \left[\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$= \left[\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Laplacian of Gaussian!

Advanced edge detectors (Marr Hildreth)



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

Mexican hat!

a	b
c	d

FIGURE 10.21 (a) Three-dimensional plot of the *negative* of the LoG. (b) Negative of the LoG displayed as an image. (c) Cross section of (a) showing zero crossings. (d) 5×5 mask approximation to the shape in (a). The negative of this mask would be used in practice.

$$\nabla^2 G(x,y) = \left[\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Advanced edge detectors (Marr Hildreth)

- Marr-Hildreth edge detection

$$g(x, y) = \ [\nabla^2 G(x, y)] * f(x, y)$$

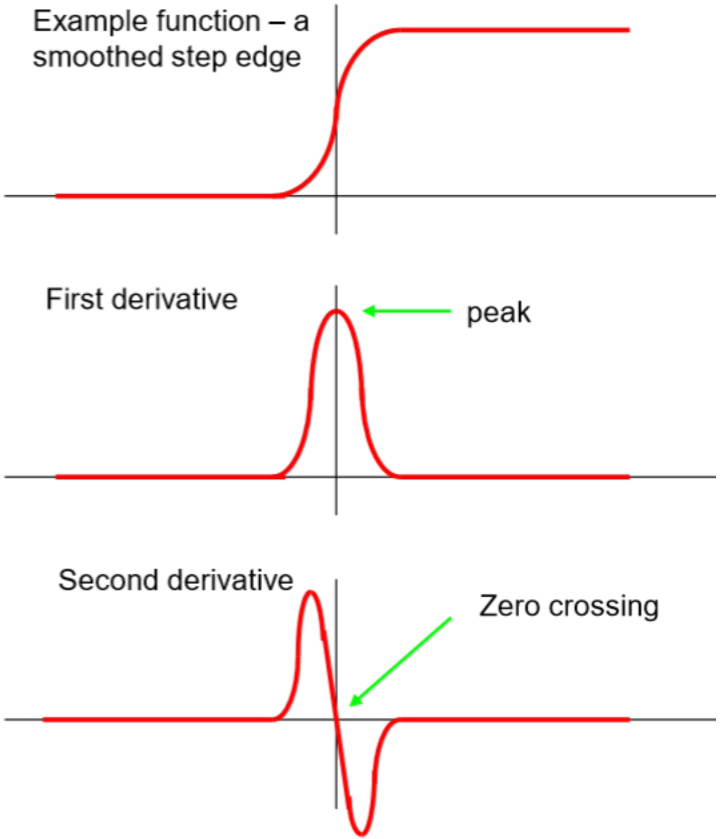
$$g(x, y) = \nabla^2 [G(x, y) * f(x, y)]$$

1. Filter the image with a gaussian low pass filter
2. Compute the Laplacian of the resulting image
3. Find zero crossings

Zero crossings

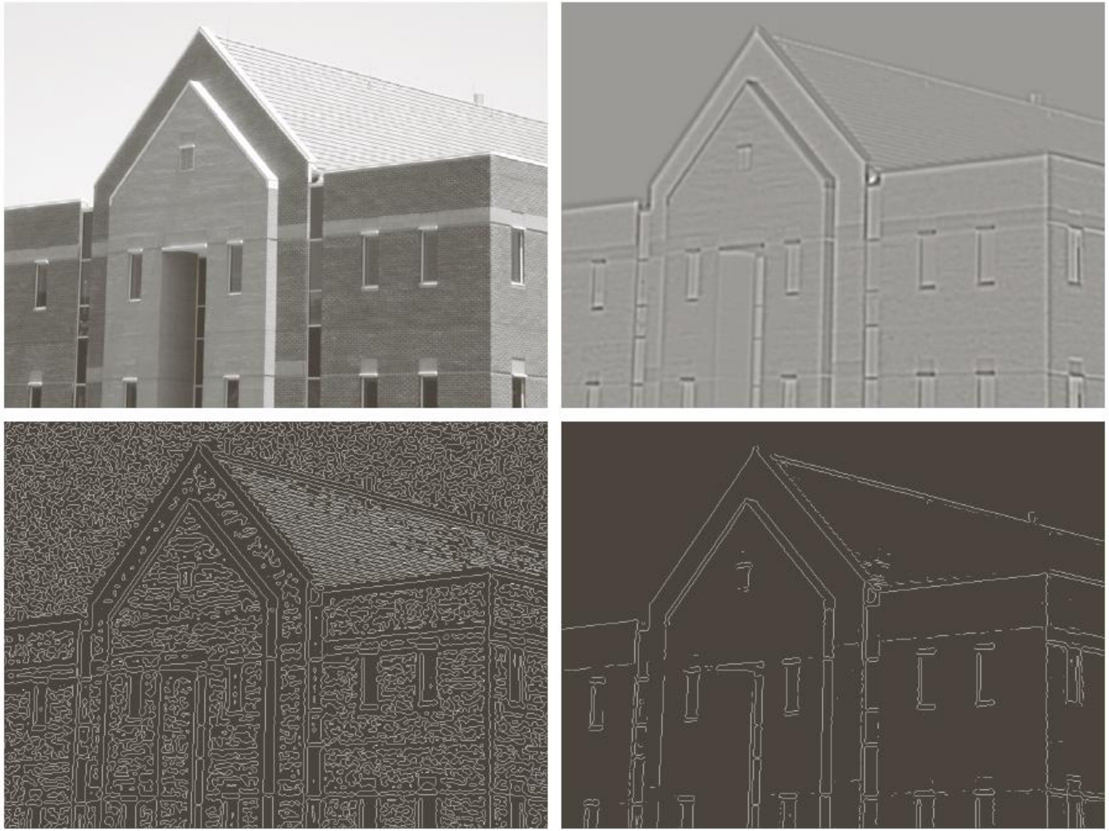
- Zero crossing indicates an edge
- Peak in first derivative → zeros in second derivative

  - How to detect a zero crossing?



Source: William Hoff

Zero crossings



a b
c d

FIGURE 10.22 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Results of Steps 1 and 2 of the Marr-Hildreth algorithm using $\sigma = 4$ and $n = 25$. (c) Zero crossings of (b) using a threshold of 0 (note the closed-loop edges). (d) Zero crossings found using a threshold equal to 4% of the maximum value of the image in (b). Note the thin edges.

Advanced edge detectors (Canny)

- Most common and widely used
  - Good detection (minimize false edges and missing real edges)
  - Good localization (detected edges close to true edges)
  - Single Response (return only one point for each true edge point)

- Essence of Canny’s work: optimal solution to these three formulation by expressing the three criteria mathematically

- He found that a good approximation to an optimal operator is the first derivative of the Gaussian in the direction of the gradient
  - Then do nonmaxima suppression

Advanced edge detectors (Canny)

- Smooth the input image with a Gaussian filter
- Compute the gradient magnitude and angle images
- Apply non maxima suppression to the gradient magnitude image
  - Quantize the angle image into four directions
  - If magnitude is less than one of the two neighbours along its direction, suppress
- Use double thresholding and connectivity analysis to detect and link edges

Advanced edge detectors (Canny)

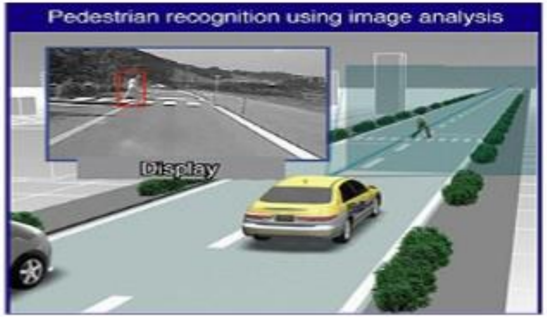
a b
c d

FIGURE 10.25
(a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1].
(b) Thresholded gradient of smoothed image.
(c) Image obtained using the Marr-Hildreth algorithm.
(d) Image obtained using the Canny algorithm. Note the significant improvement of the Canny image compared to the other two.

Lets explore an application of the stuff we learnt!

HOG for person detection

- Applications
  - Multimedia analysis
  - Pedestrian detection for smart cars
  - Visual surveillance, behaviour analysis



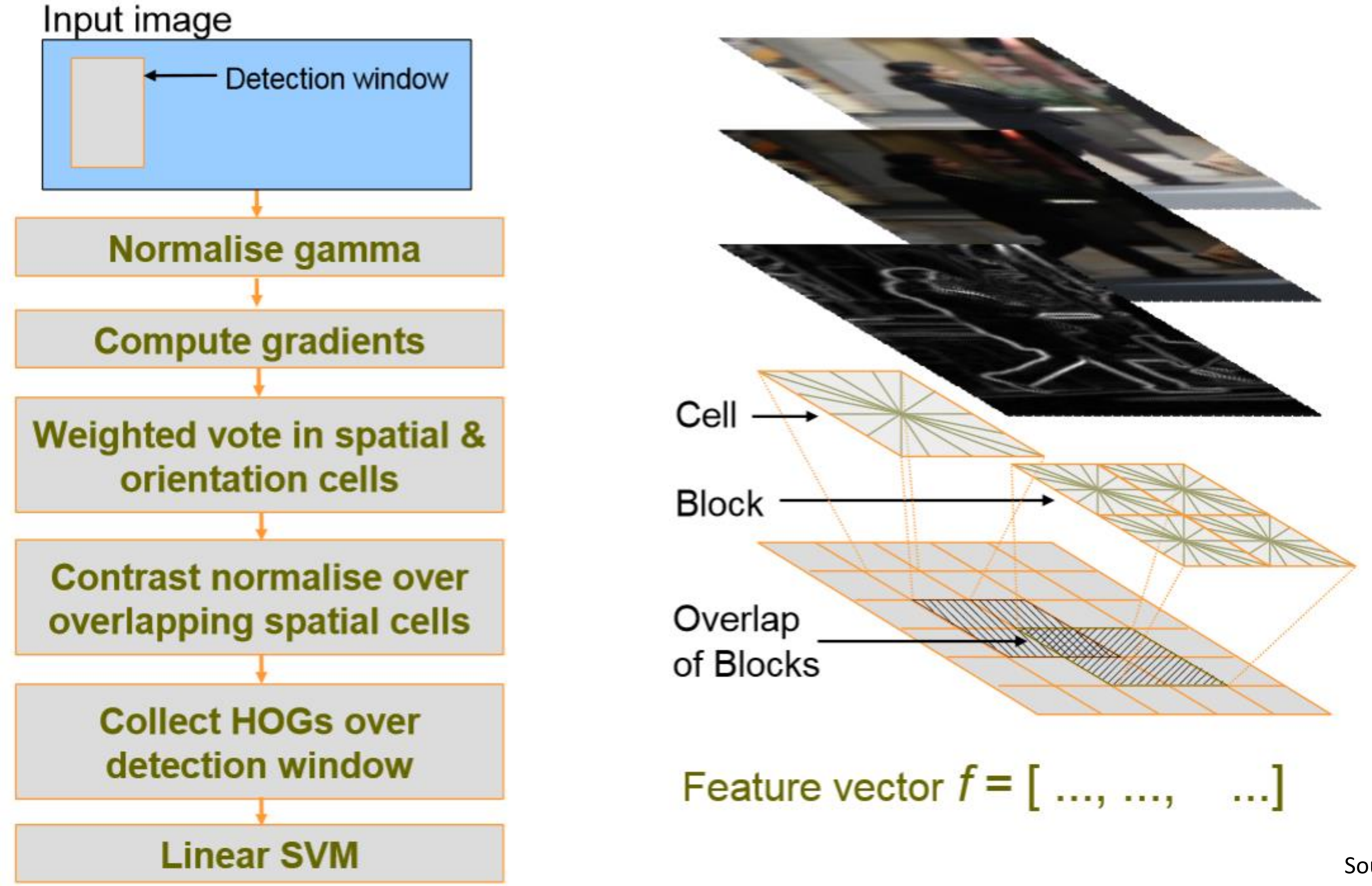
Source: Navneet Dalal

HOG for person detection

- Histogram of oriented gradients
- Proposed by Dalal and Triggs (INRIA Grenoble)
- Extremely popular paper: 12375 citations (todays figure from google scholar)
- Lets understand it in 2 minutes

Source: Navneet Dalal

HOG for person detection

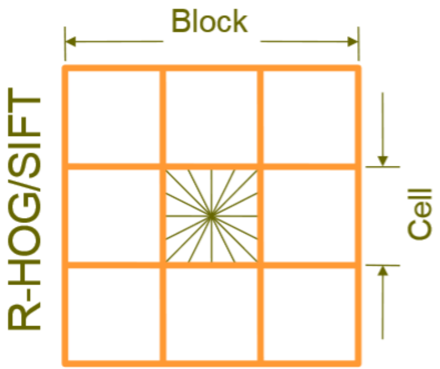


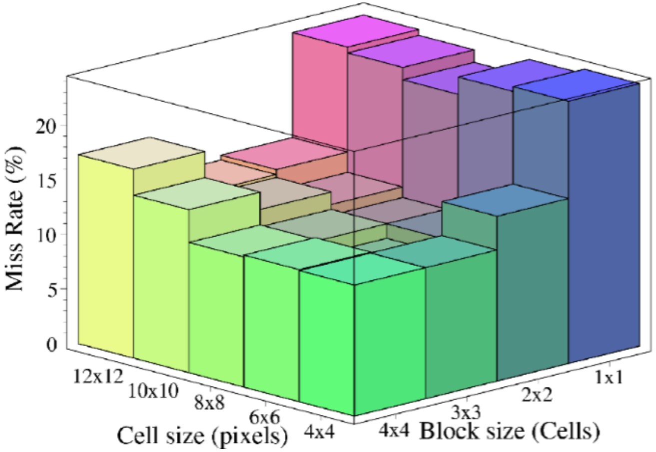
Source: Navneet Dalal

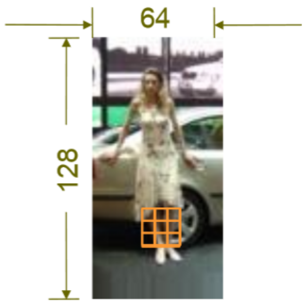
HOG for person detection

Rest is tuning!

- What size of cells? What size of blocks?
- How to normalize?
- How many orientations?
- Which color space?

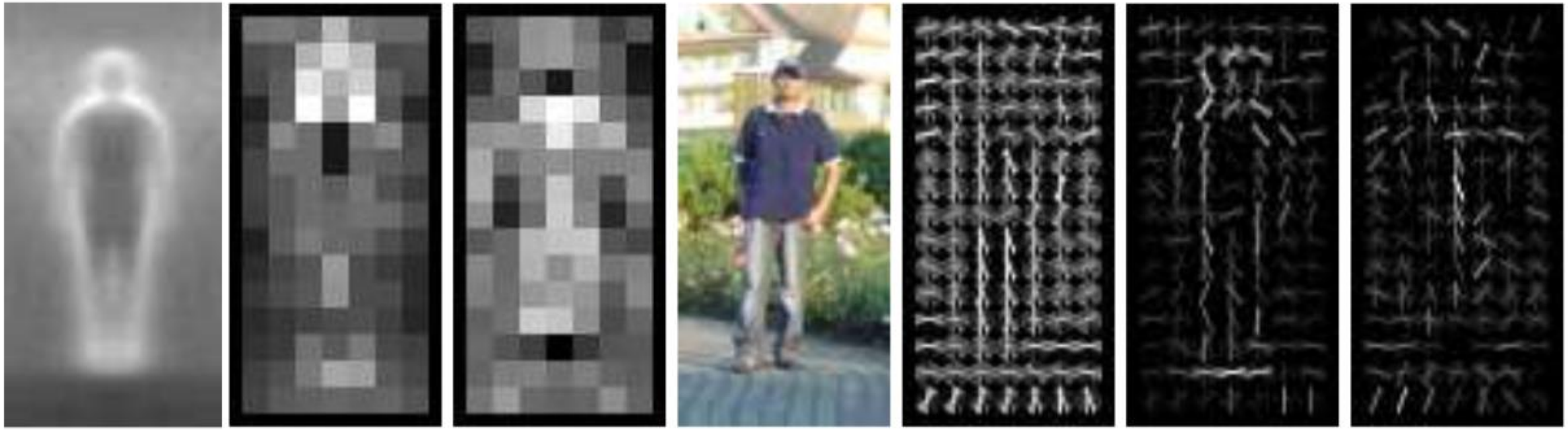






Source: Navneet Dalal

HOG for person detection



(a) (b) (c) (d) (e) (f) (g)

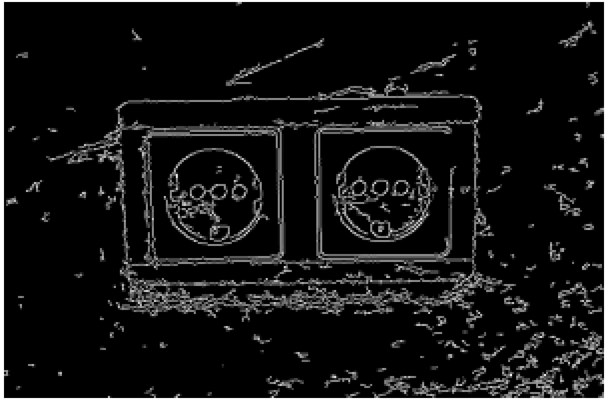
Source: Navneet Dalal

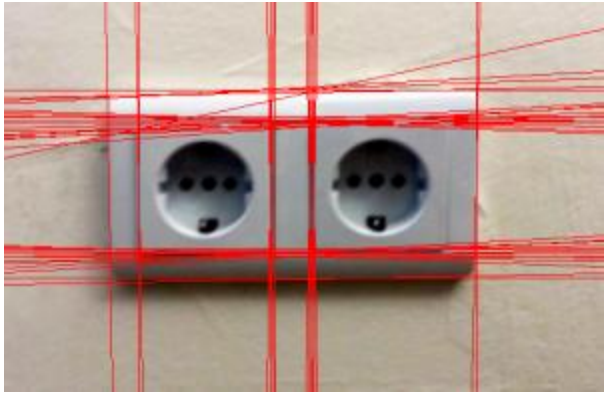
Can we find principle lines or curves from edge images!

Group gradients into Edges or Curves

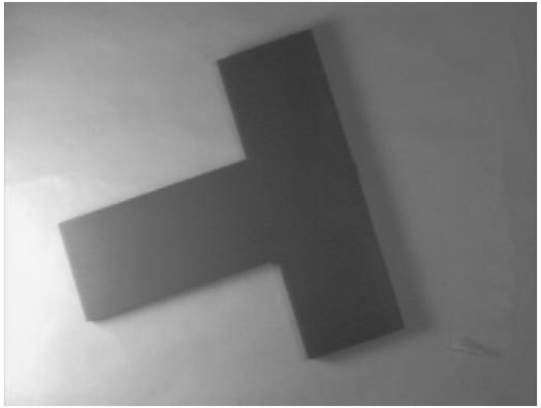
- Local approach
- Global approach

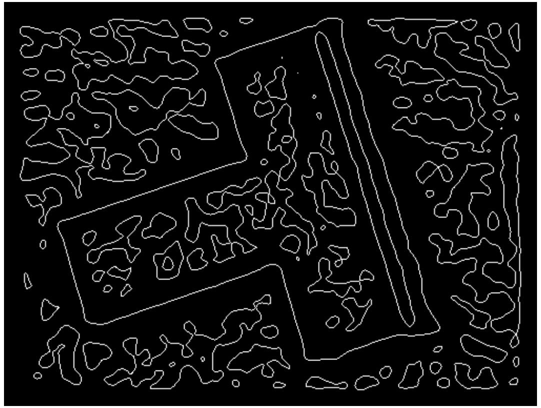
Hough Transform



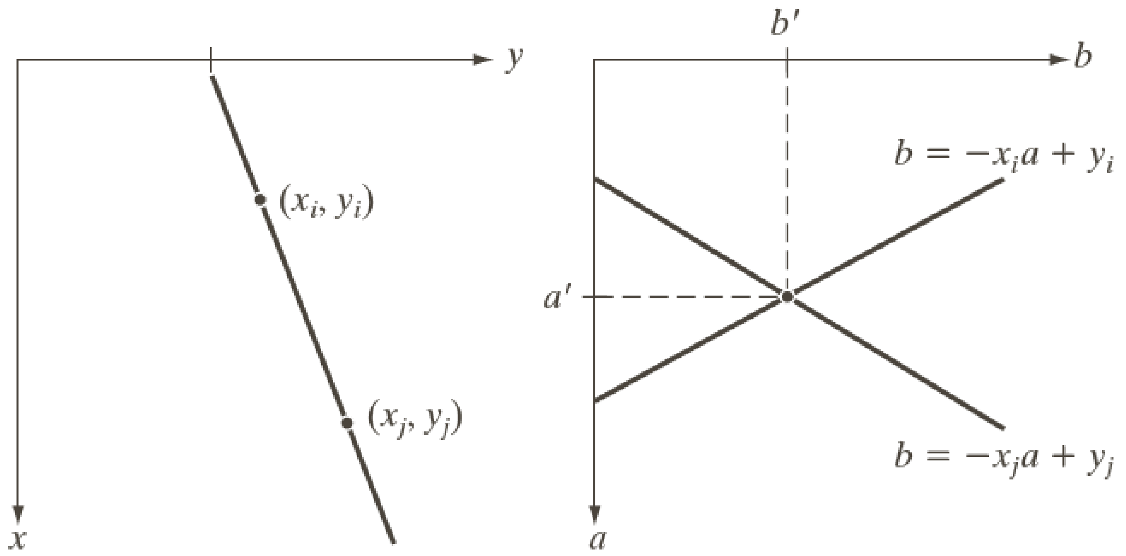


Hough Transform





Hough Transform



a b

FIGURE 10.31
(a) xy-plane.
(b) Parameter space.

Hough Transform

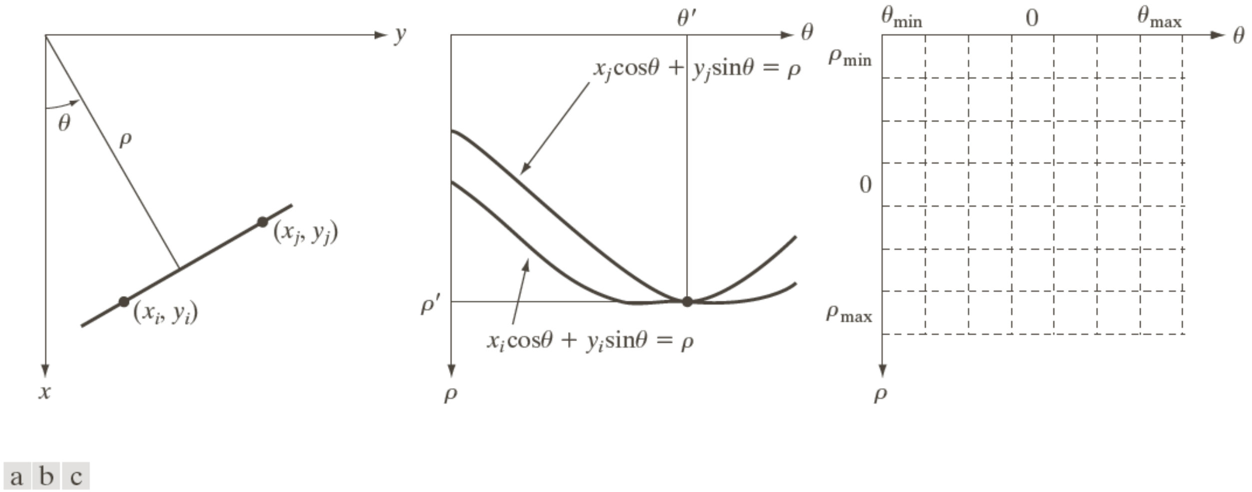


FIGURE 10.32 (a) (ρ, θ) parameterization of line in the xy-plane. (b) Sinusoidal curves in the $\rho\theta$-plane; the point of intersection (ρ', θ') corresponds to the line passing through points (x_i, y_i) and (x_j, y_j) in the xy-plane. (c) Division of the $\rho\theta$-plane into accumulator cells.