Outline

- Introduction
- Noise removal using low-pass filters
- Sharpening by edge enhancement
  - Edge detection using high-pass filters
  - Edge enhancement by high emphasis filters
- Edge detection
  - First order gradient
  - Second order gradient
- Summary

Problems

- Smoothing
  - Noise removal
  - Detail preserving image smoothing
- Sharpening
  - Edge enhancement
  - Detail focusing
- Edge detection

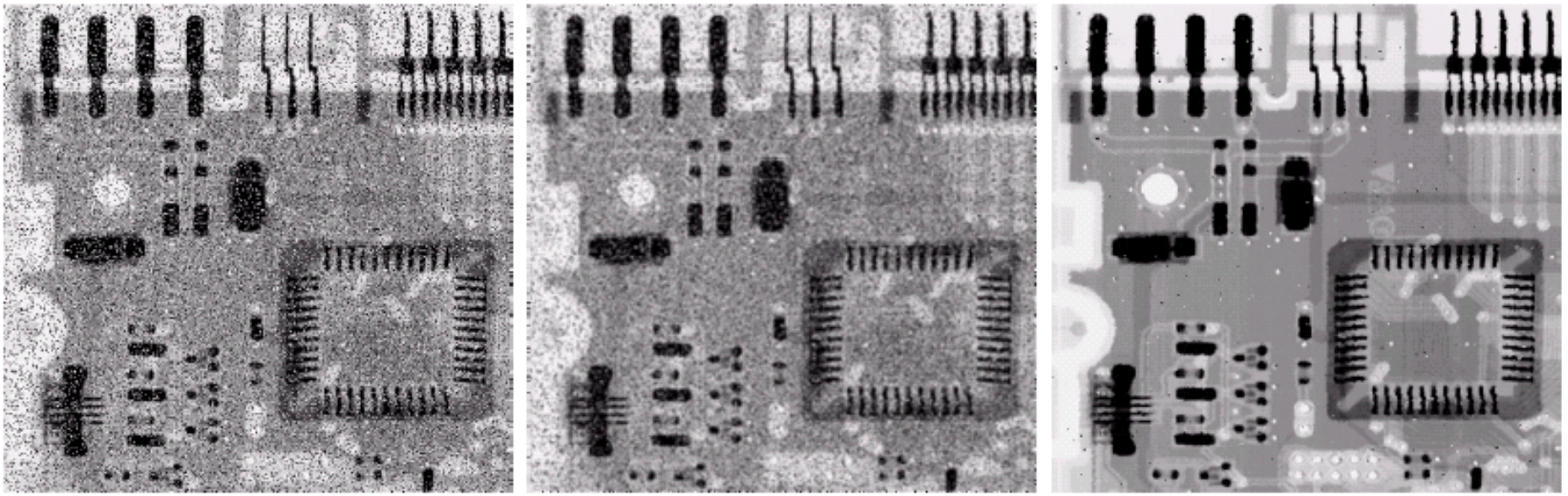
Approaches

- Spatial domain operation or filtering (the processed value for the current pixel depends on both itself and surrounding pixels)
  - Linear filtering
  - Non-linear filtering
    - Rank order filtering including median
    - Morphological filtering
  - Adaptive filtering

Noise Removal (Image Smoothing)

- An image may be “dirty” (with dots, speckles,stains)
- Noise removal:
  - To remove speckles/dots on an image
  - Dots can be modeled as impulses (salt-and-pepper or speckle) or continuously varying (Gaussian noise)
  - Can be removed by taking mean or median values of neighboring pixels (e.g. 3x3 window)
  - Equivalent to low-pass filtering
- Problem with low-pass filtering
  - May blur edges
  - More advanced techniques: adaptive, edge preserving

Example



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Averaging Filter: An Intuitive Approach

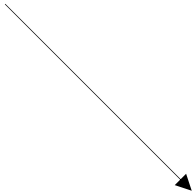
- Replace each pixel by the average of pixels in a square window surrounding this pixel

$$g(m,n)=\frac{1}{9}(f(m-1,n-1)+f(m-1,n)+f(m-1,n+1)+f(m,n-1)+f(m,n)+f(m,n+1)+f(m+1,n-1)+f(m+1,n)+f(m+1,n+1))$$

- Trade-off between noise removal and detail preserving:
  - Larger window -> can remove noise more effectively, but also blur the details/edges

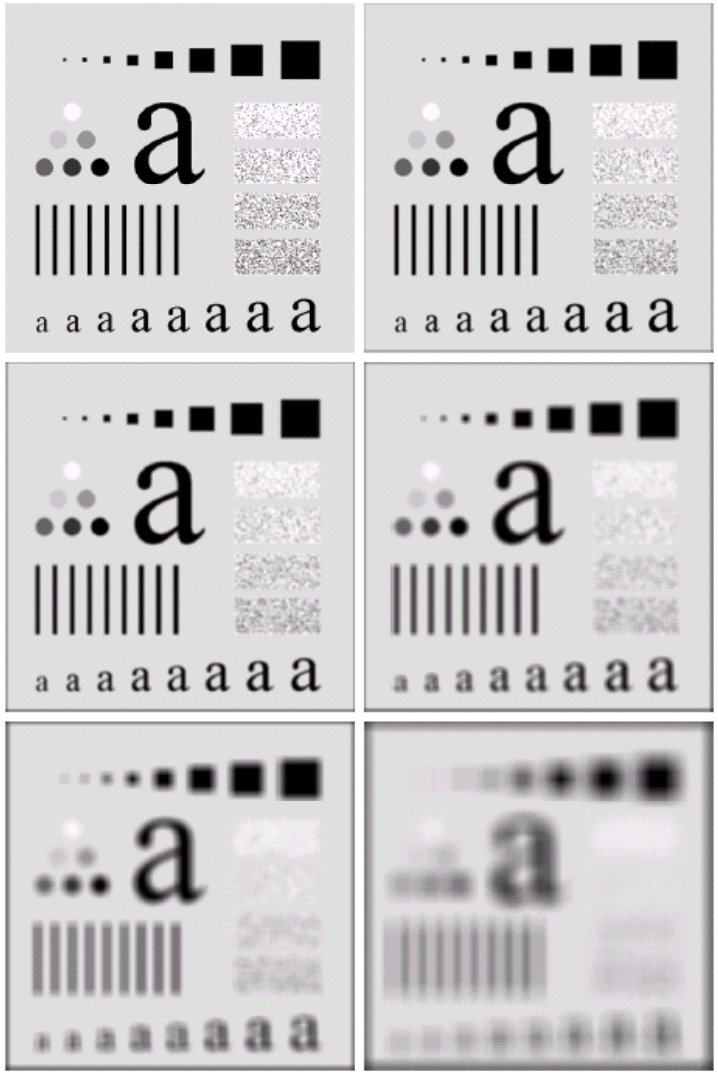
Example: 3x3 average

100	100	100	100	100
100	200	205	203	100
100	195	200	200	100
100	200	205	195	100
100	100	100	100	100



100	100	100	100	100
100	144	167	145	100
100	167	200	168	100
100	144	166	144	100
100	100	100	100	100

Example



a b
c d
e f

FIGURE 3.35 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $n = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their gray levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.

Weighted Averaging Filter

- Instead of averaging all the pixel values in the window, give the closer-by pixels higher weighting, and far-away pixels lower weighting.

$$g(m,n) = \sum_{k=k_0}^{k_1} \sum_{l=l_0}^{l_1} h(k,l)\, f(m-k, n-l) = f(m,n) * h(m,n)$$

- This type of operation is in fact 2-D linear convolution of f(m,n) by a filter h(m,n).
- Weighted average filter retains low frequency and suppresses high frequency = low-pass filter

Example Weighting Mask

$\frac{1}{9}\times$

1	1	1
1	1	1
1	1	1

$\frac{1}{16}\times$

1	2	1
2	4	2
1	2	1

All weights must sum to one

Example: Weighted Average

100	100	100	100	100
100	200	205	203	100
100	195	200	200	100
100	200	205	195	100
100	100	100	100	100

$\frac{1}{9}\times$

1	1	1
1	1	1
1	1	1

$\frac{1}{16}\times$

1	2	1
2	4	2
1	2	1

100	100	100	100	100
100	144	167	145	100
100	167	200	168	100
100	144	166	144	100
100	100	100	100	100

100	100	100	100	100
100	156	176	158	100
100	174	201	175	100
100	156	175	156	100
100	100	100	100	100

Example

Original image

Average filtered image

Weighted Average filtered image

Common Smoothing Filters

$$\frac{1}{10}\begin{bmatrix}1 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 1\end{bmatrix};\quad \frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix};\quad \frac{1}{(b+2)^2}\begin{bmatrix}1 & b & 1\\ b & b^2 & b\\ 1 & b & 1\end{bmatrix},\ with\ b \geq 1.$$

- Are any of these separable?
- Criteria for designing a smoothing filter
  - h(k,l)≥0 (function as averaging)
  - $\sum_{k=k_0}^{k_1}\sum_{l=l_0}^{l_1} h(k,l) = 1$ (the mean value is preserved)

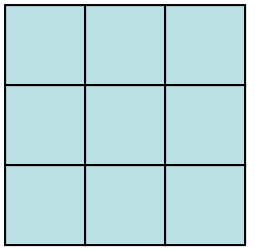
Median Filter

- Problem with Averaging Filter
  - Blur edges and details in an image
  - Not effective for impulse noise (Salt-and-pepper)
- Median filter:
  - Taking the median value instead of the average or weighted average of pixels in the window
    - Sort all the pixels in an increasing order, take the middle one
  - The window shape does not need to be a square
  - Special shapes can preserve line structures

Median Filter: 3x3 Square Window

100	100	100	100	100
100	200	205	203	100
100	195	200	200	100
100	200	205	195	100
100	100	100	100	100

Window shape



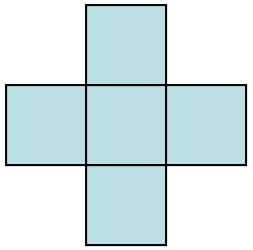
100	100	100	100	100
100	100	200	100	100
100	200	200	200	100
100	100	195	100	100
100	100	100	100	100

Matlab command: medfilt2(A,[3 3])

Median Filter: 3x3 Cross Window

100	100	100	100	100
100	200	205	203	100
100	195	200	200	100
100	200	205	195	100
100	100	100	100	100

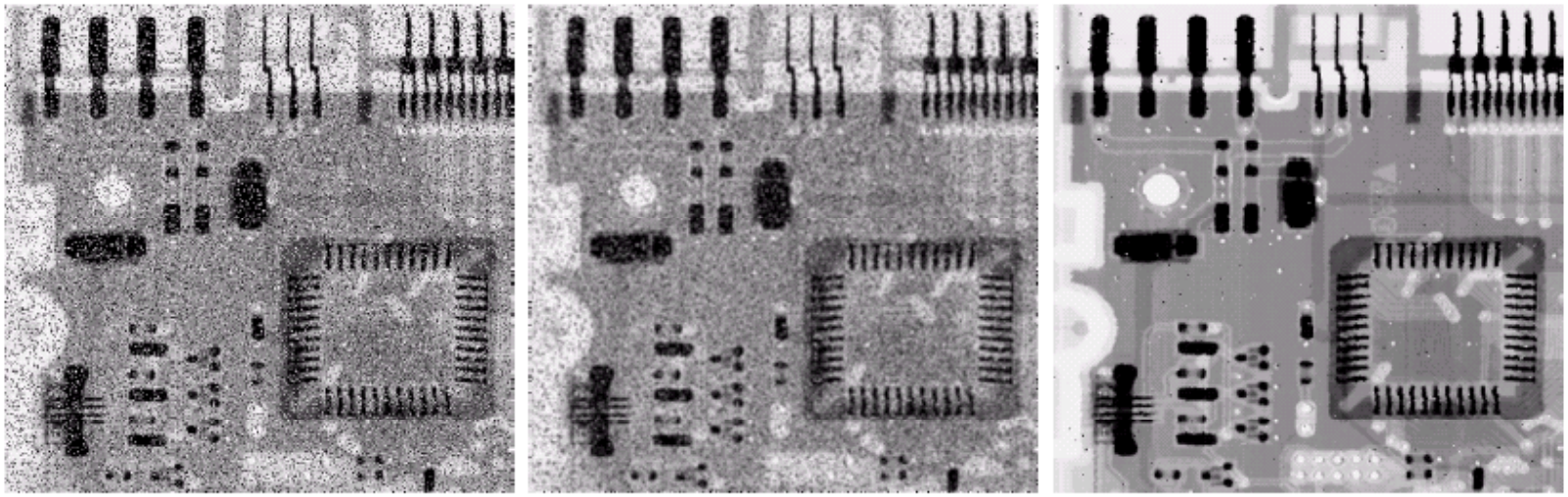
Window shape



100	100	100	100	100
100	195	200	200	100
100	200	200	200	100
100	195	200	195	100
100	100	100	100	100

Note that the edges of the center square are better reserved

Example



a b c

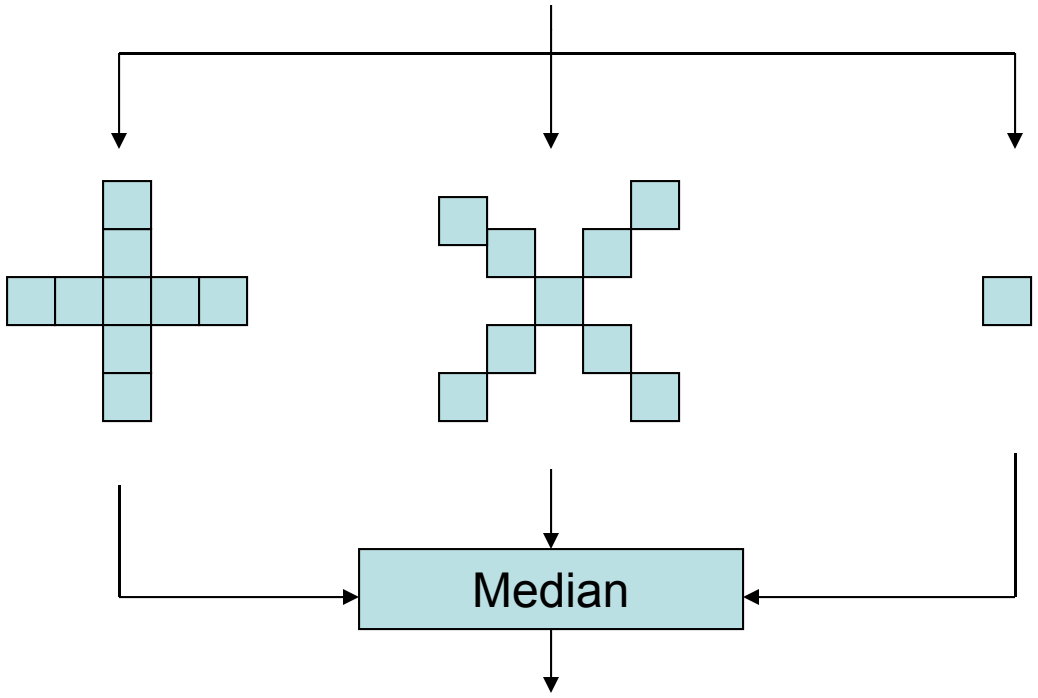
FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Rank order filters

- Rank order filters
  - Instead of taking the mean, rank all pixel values in the window, take the n-th order value.
  - E.g. max or min or median
- Properties
  - Non-linear $T(f_1 + f_2) \neq T(f_1) + T(f_2)$

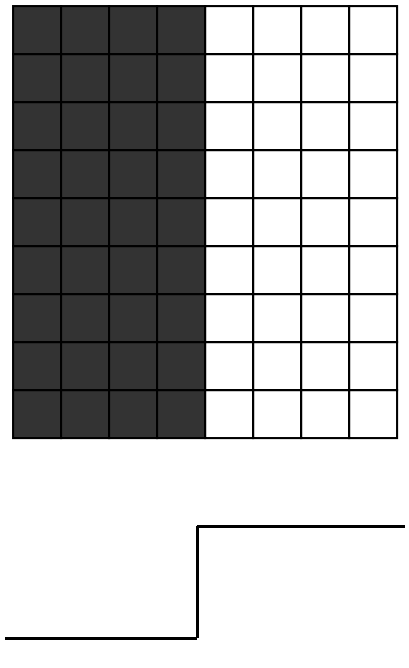
Multi-level Median Filtering

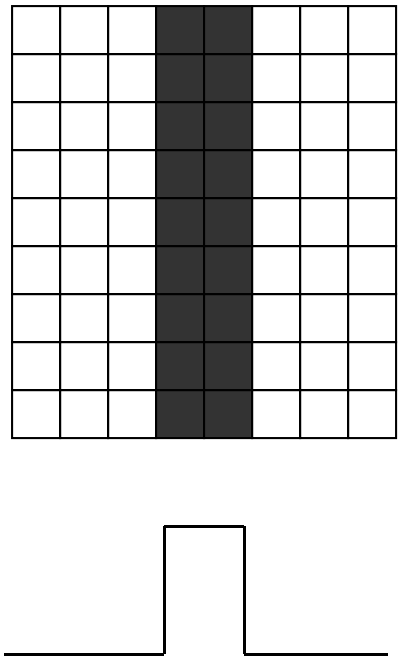
- To reduce the computation, one can concatenate several small median filters to realize a large window operation.
- When the small windows are designed properly, this approach can also help reserve edges better.

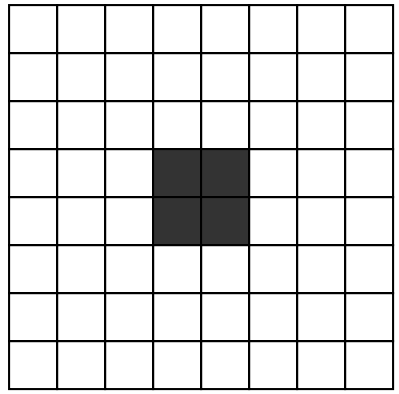


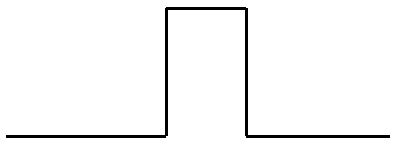
Edge Detection

- What is an edge?
- Difference between edge and line and point









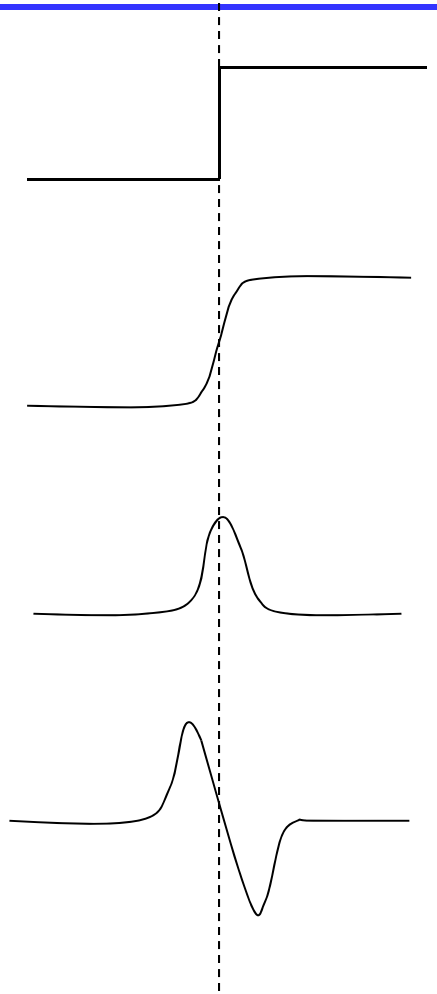
Characterization of Edges

Ideal step edge

Real edge has a slope

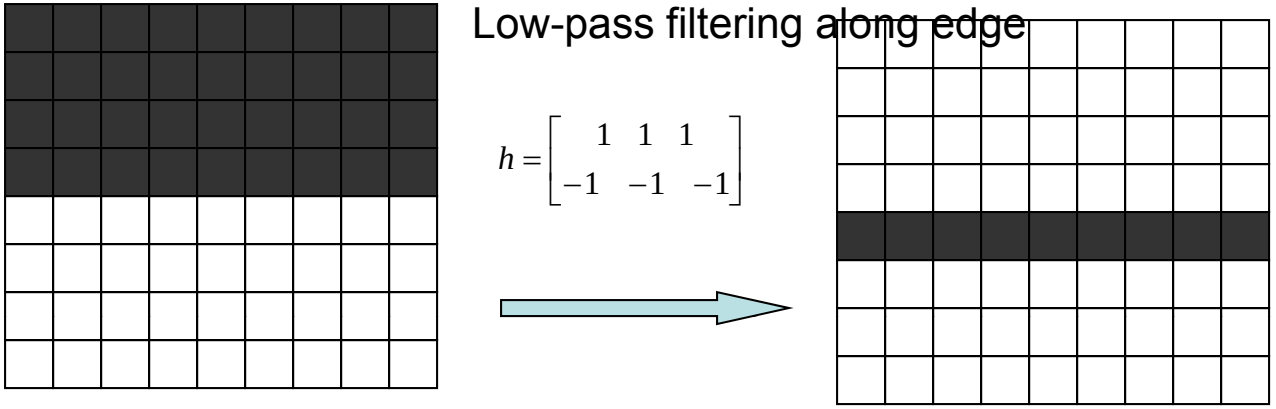
First order derivative:
Maximum at edge location

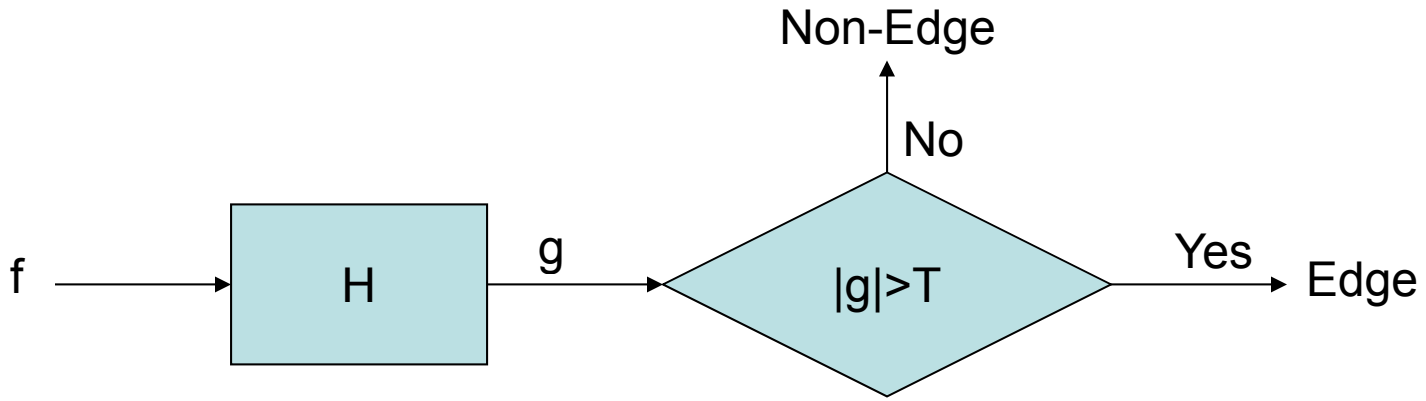
Second order derivative:
Zero crossing at edge location



Edge Detection Based on First Order Derivatives

- Edge





What if we don't know edge direction?

Directional Edge Detector

- High-pass (or band-pass) in one direction (simulating first order derivative)
- Low pass in the orthogonal direction (smooth noise)
- Prewitt edge detector

LP BP

$$H_x = \frac{1}{3}\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}; \quad H_y = \frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

- Sobel edge detector

$$H_x = \frac{1}{4}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}; \quad H_y = \frac{1}{4}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

The sobel filter provides better smoothing along the edge

Example of Sobel Edge Detector

Original image

Filtered image by H_x

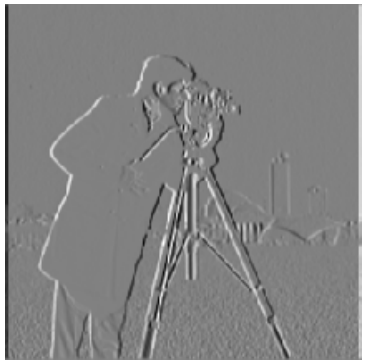
Filtered image by H_y

How to set threshold?

- Trial and error
- According to edge magnitude distribution
  - E.g assuming only 5% pixels should be edge pixels, then the threshold should be the 95% percentile of the edge magnitude
  - Illustrate on board

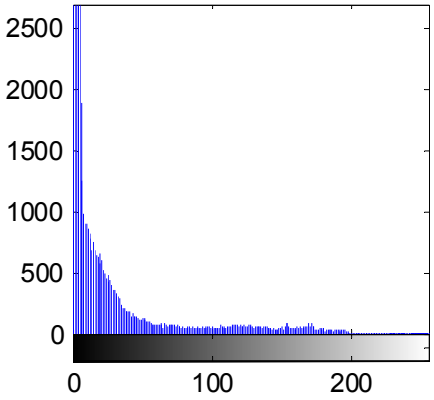
gx

gy



gm

Histogram of gm



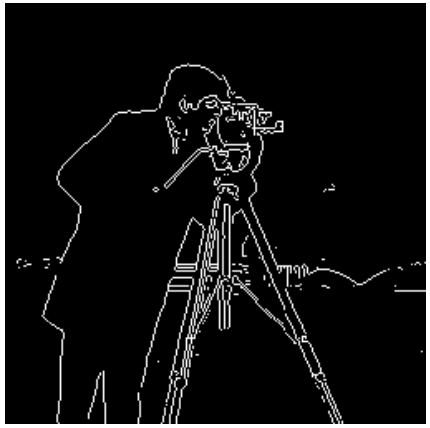
T=100

T=50

T=20

Results using MATLAB “edge( )” function

Sobel,T=0.14

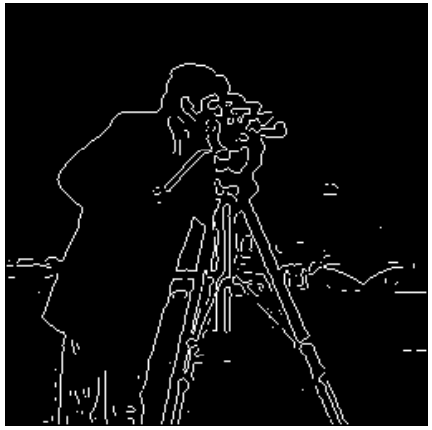


LOG, T=0.0051

canny,T=[0.0313,0.0781]

Sobel,T=0.1

LOG, T=0.01



canny,T=[0.1,0.15]