

# Image Filtering

## Chetan Arora

Disclaimer: The contents of these slides are taken from various publicly available resources such as research papers, talks and lectures. To be used for the purpose of classroom teaching, and academic dissemination only.

# **Convolution**



# Discrete Convolution

$$h(i) = (f * g)(i) = \sum_{k=-\infty}^{k=\infty} f(k)g(i - k)$$

$f(k)$

1	2	2	2	1
---	---	---	---	---

$g(k)$

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

$g(-k)$

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

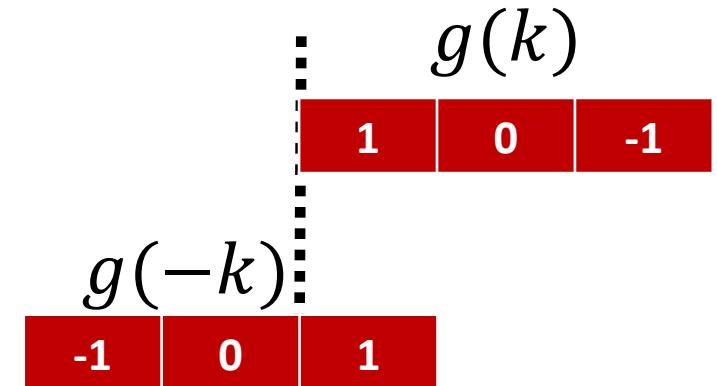


# Discrete Convolution

$$h(i) = (f * g)(i) = \sum_{k=-\infty}^{k=\infty} f(k)g(i - k)$$



$h(0)$





# Discrete Convolution

$$h(i) = (f * g)(i) = \sum_{k=-\infty}^{k=\infty} f(k)g(i - k)$$



$g(1 - k)$

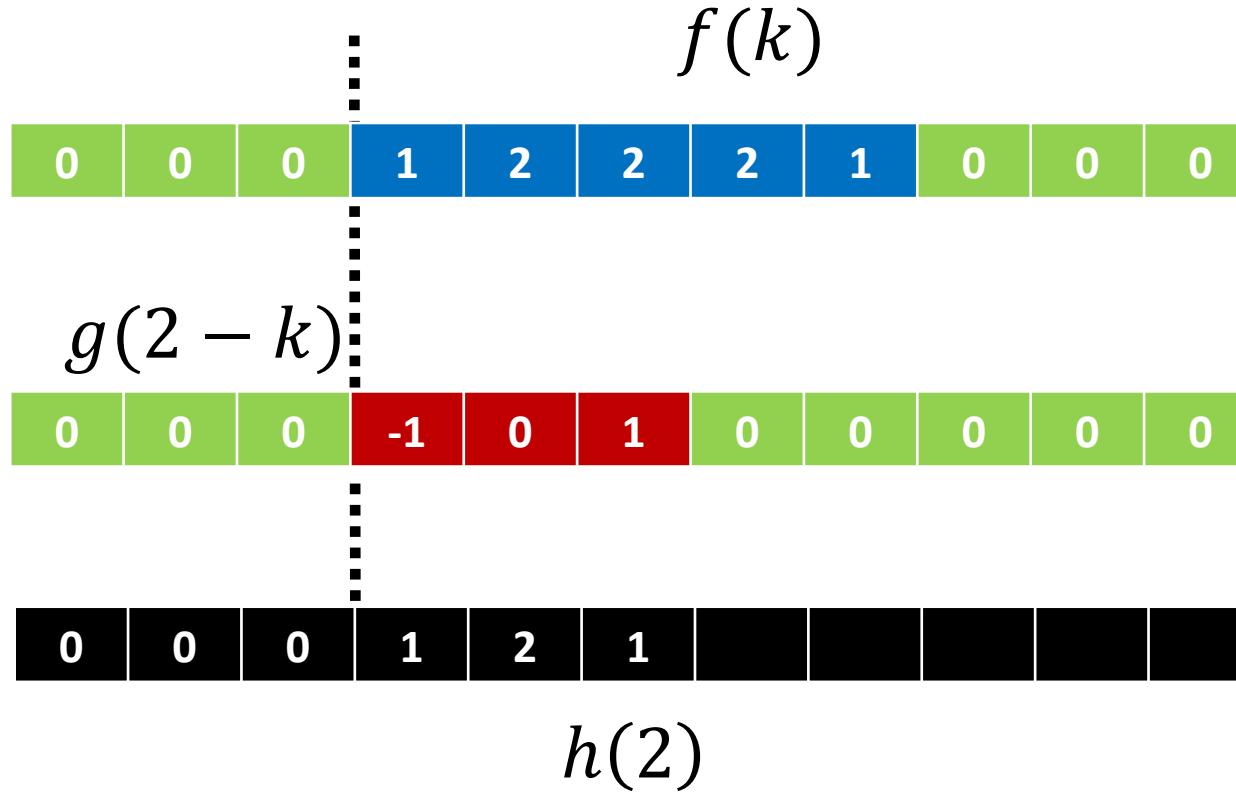


$h(1)$



# Discrete Convolution

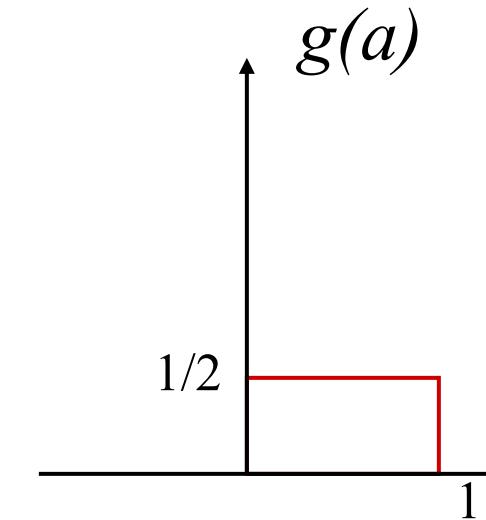
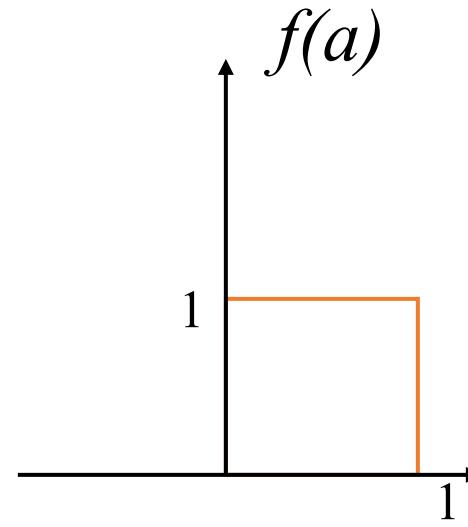
$$h(i) = (f * g)(i) = \sum_{k=1}^n f(k)g(i - k)$$





# Continuous Convolution

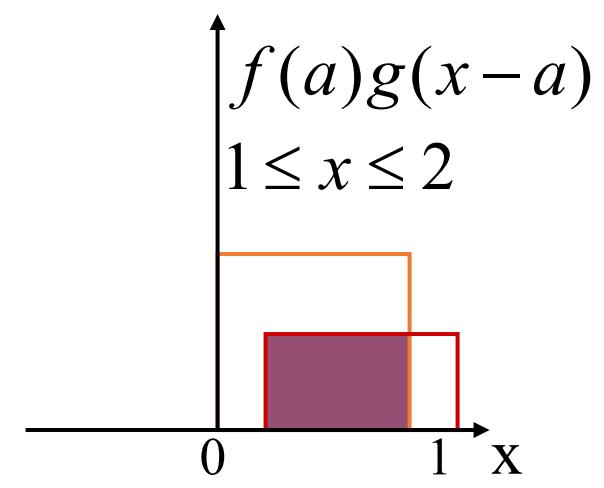
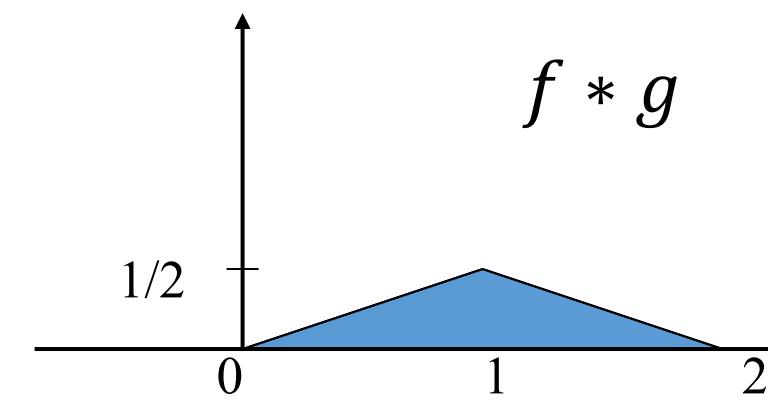
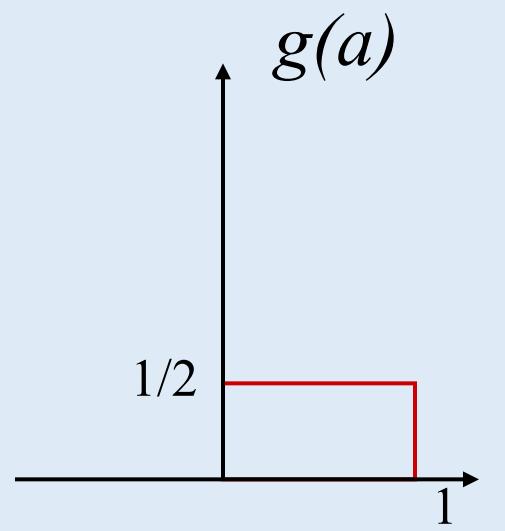
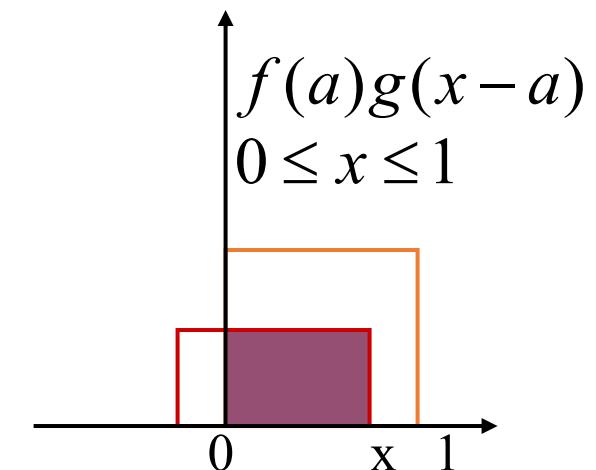
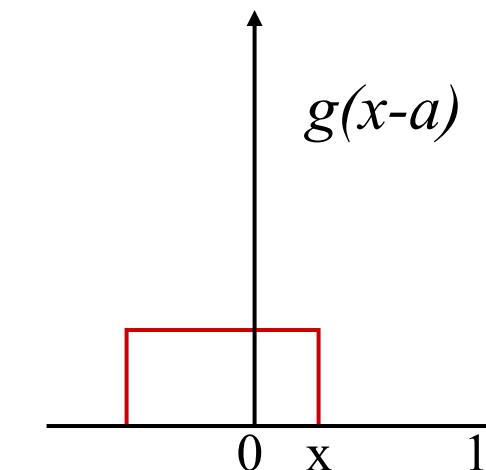
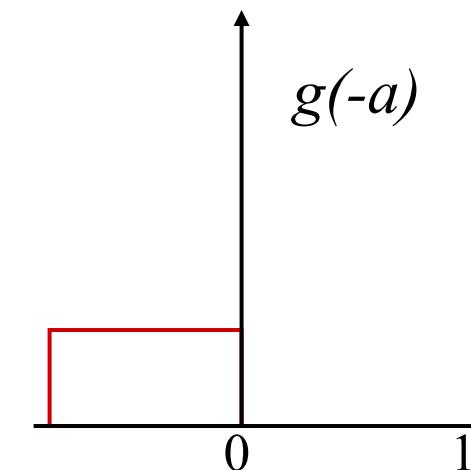
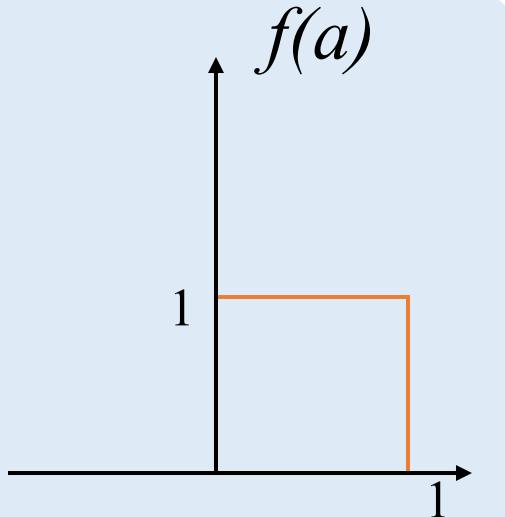
$$(f * g)(x) = \int_{-\infty}^{\infty} f(a)g(x - a)da$$





# Continuous Convolution

$$(f * g)(x) = \int_{-\infty}^{\infty} f(a)g(x-a)da$$





# Convolution

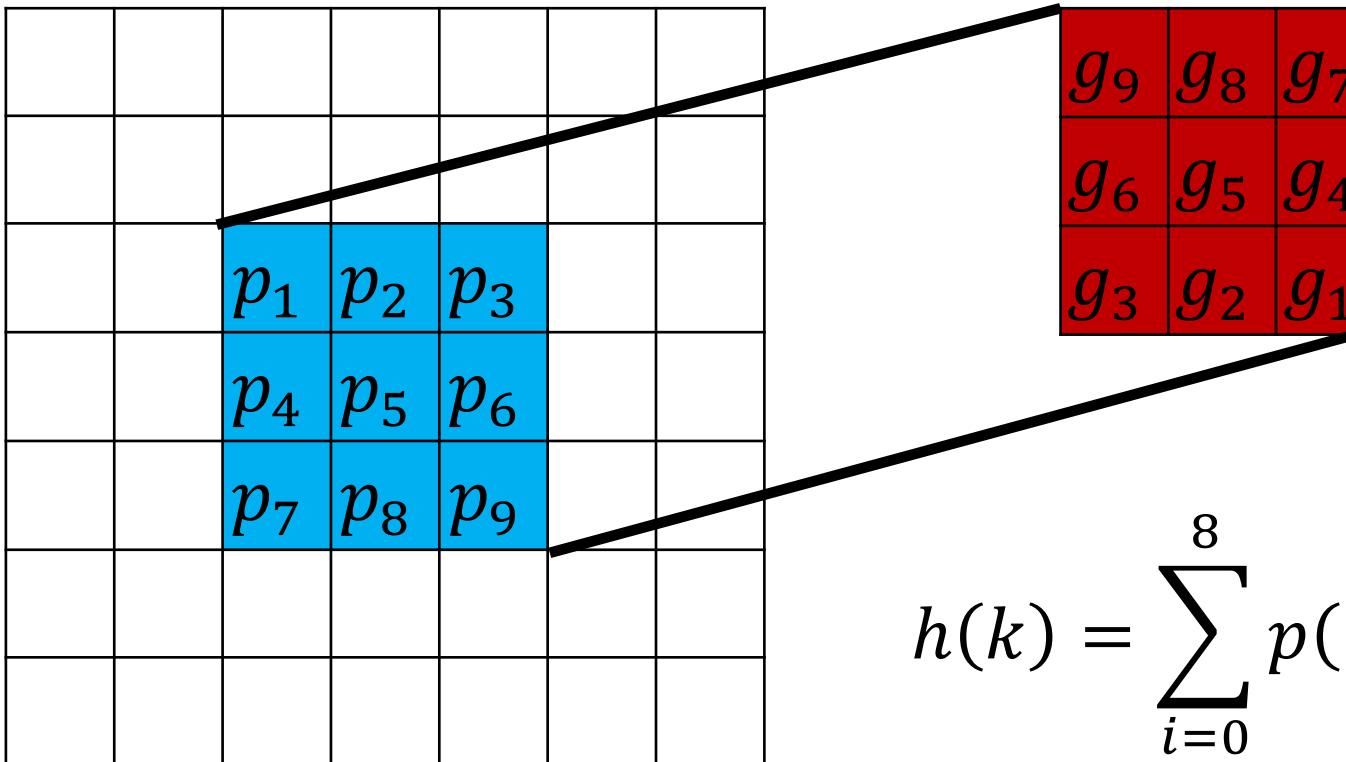
- **Why convolution?**
  - Response to Physics (sensors): Frequency response to color by sensors (color space - wavelength). Area response of sensors (image space)
- **Why Reflect?**
  - Commutative:  $f * g = g * f$
  - Associative:  $f * (g * h) = (f * g) * h$
  - Distributive:  $f * (g + h) = f * g + f * h$



# 2D Discrete Convolution

$$h = f * g$$

$$h(i, j) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f(k, l)g(i - k, j - l)$$





# Simple Convolutions

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \end{bmatrix}^n$$

$$\begin{bmatrix} 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 & 3 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

# **Image Filtering**



# Smoothing, Noise Cleaning

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



Original Image



Noisy Image



Filtered Image

Noise Assumption: Additive, Independent



# Smoothing, Noise Cleaning



Original



Smoothed

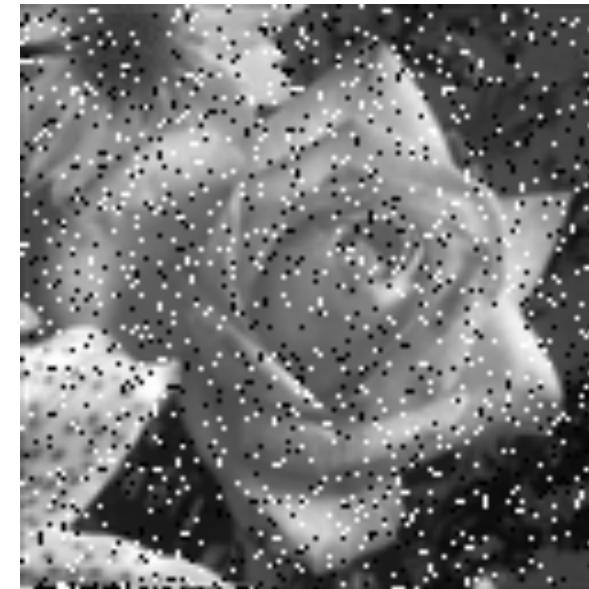
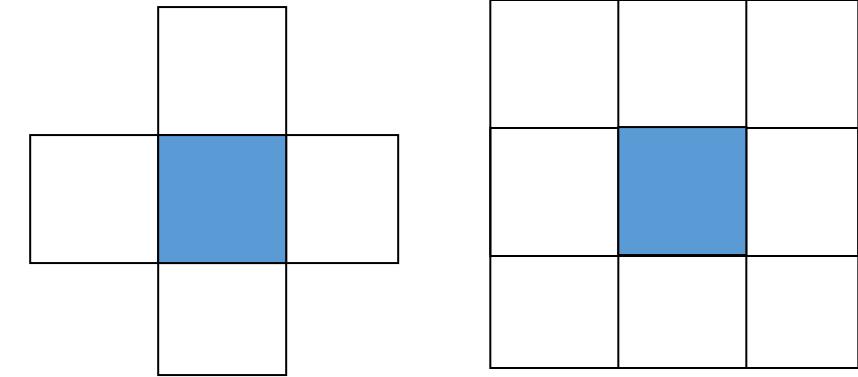


Smoothing by a  
larger kernel



# Noise Cleaning by Median Filtering

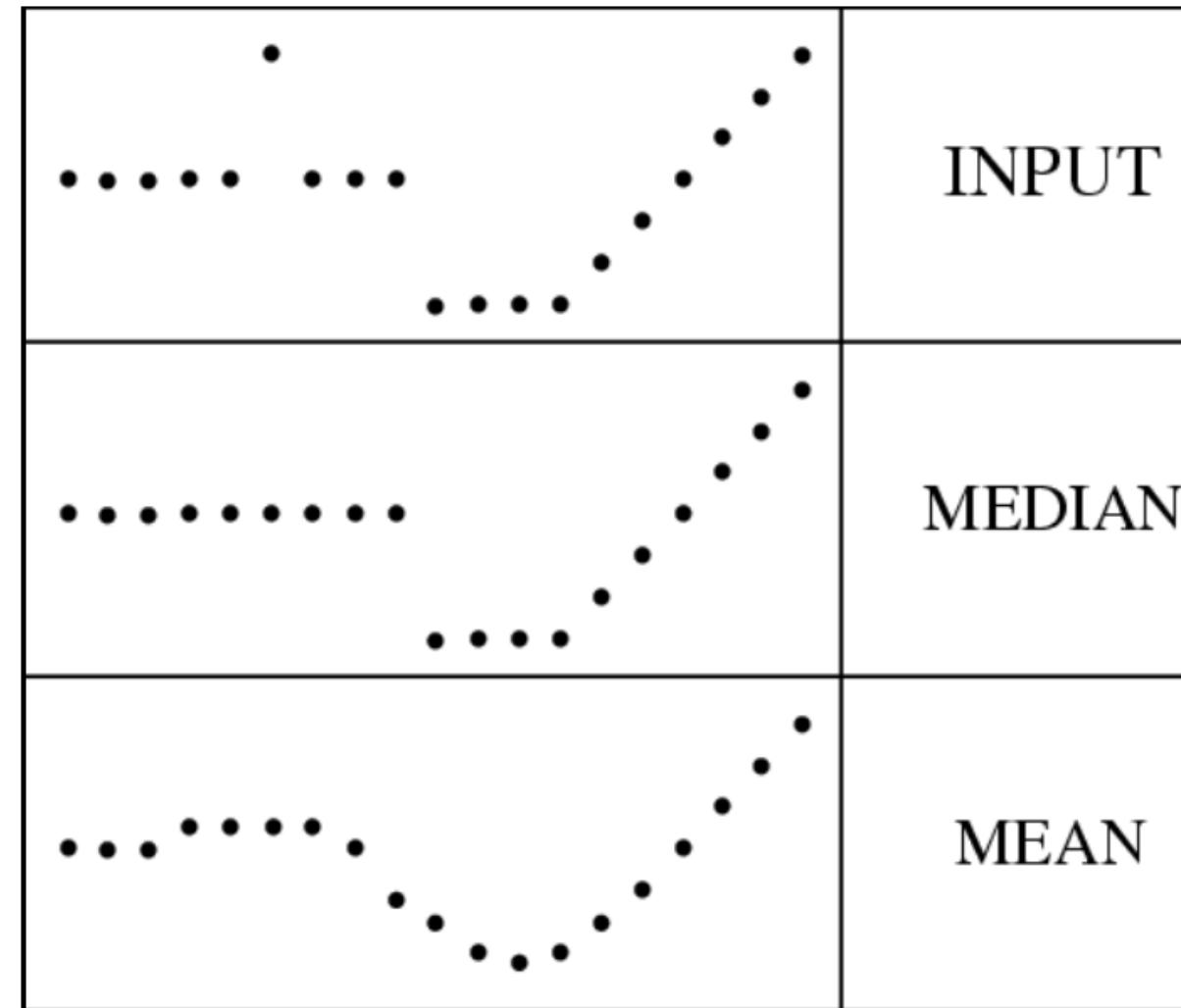
- Replace the value of a pixel with the MEDIAN of its neighborhood
- Depends on the definition of “neighborhood”





# Average Vs Median Smoothing

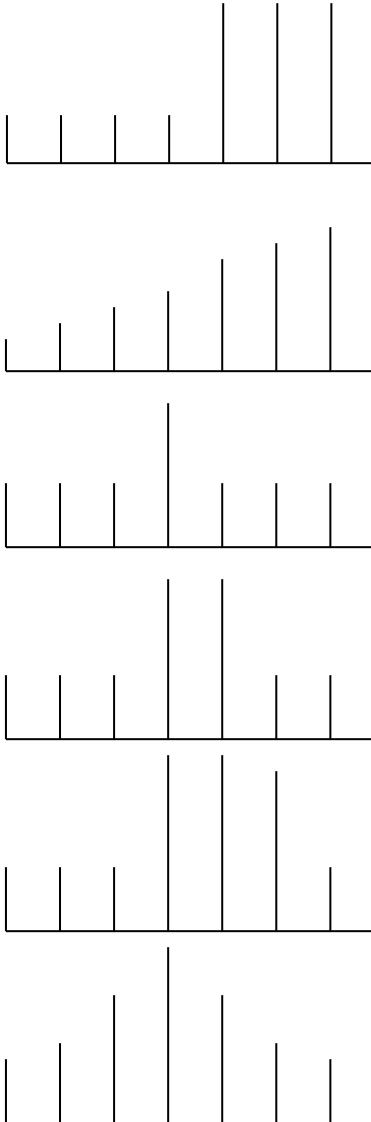
Filter width = 5



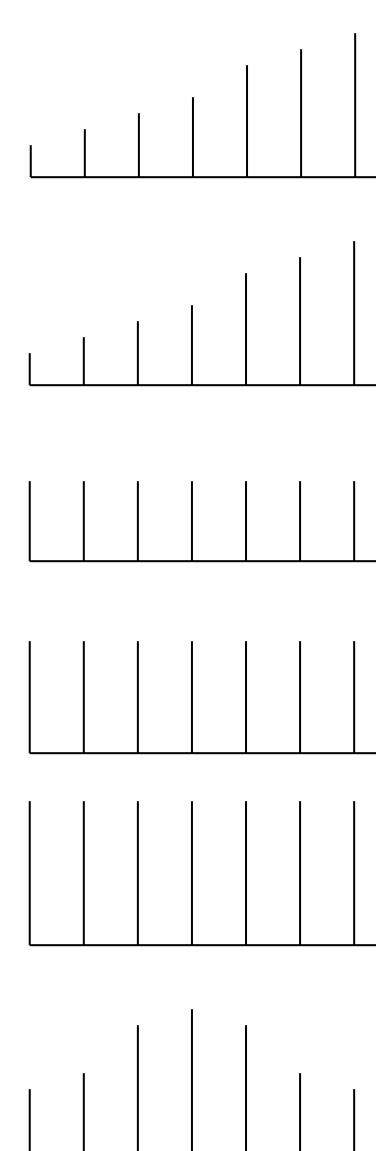


# Average Vs Median Smoothing

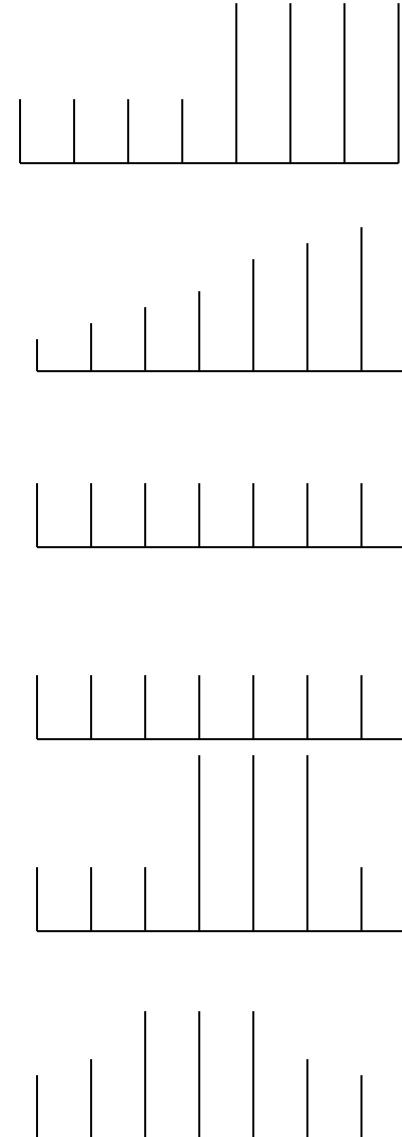
Input



Averaging



Median





# Average Vs Median Smoothing

- Averaging / Smoothing - Loss of Detail
- Median - Blockiness
- Min, Max etc.



Noisy Image  
Salt & Pepper Noise



Averaging



Median



# Image Derivatives

$$\frac{\partial}{\partial x} f(i, j) = \lim_{\epsilon \rightarrow 0} \frac{f(i, j) - f(i - \epsilon, j)}{\epsilon}$$
$$\cong f(i, j) - f(i - 1, j)$$



**Implement**  
Convolution with  $[1 \quad -1]$



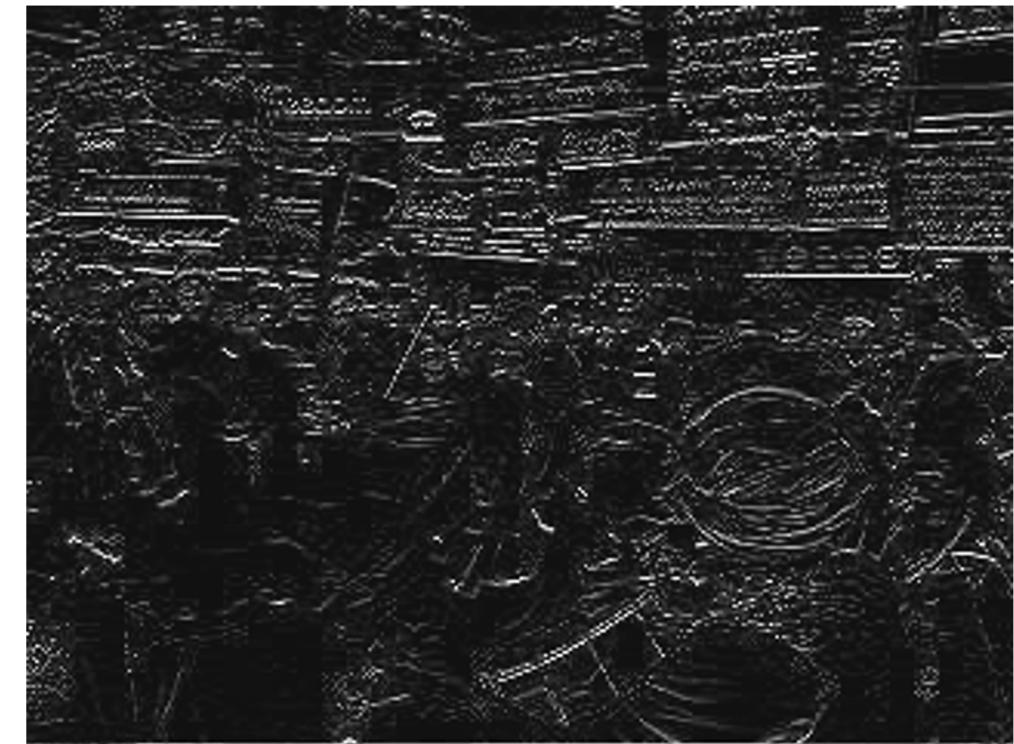


# Image Derivatives

$$\frac{\partial}{\partial x} f(i, j) = \lim_{\epsilon \rightarrow 0} \frac{f(i, j) - f(i - \epsilon, j)}{\epsilon}$$
$$\cong f(i, j) - f(i - 1, j)$$



Implement  
Convolution with  $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$



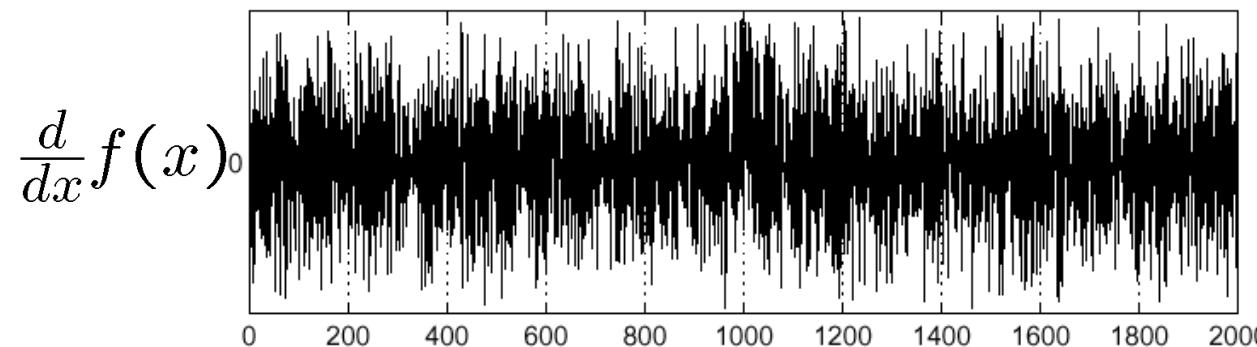
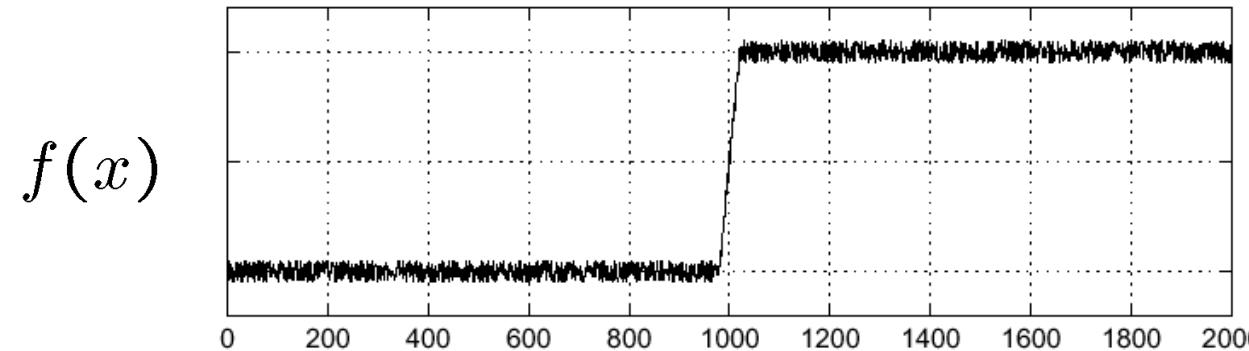


# Types of Discrete derivative in 1D

- Backward:  $\frac{\partial f}{\partial x} = f(x) - f(x - 1) \Rightarrow [-1 \quad 1 \quad 0]$
- Forward:  $\frac{\partial f}{\partial x} = f(x) - f(x + 1) \Rightarrow [0 \quad 1 \quad -1]$
- Central:  $\frac{\partial f}{\partial x} = f(x + 1) - f(x - 1) \Rightarrow [-1 \quad 0 \quad +1]$



# Effect of Noise on Image Gradients



**Where is the edge?**

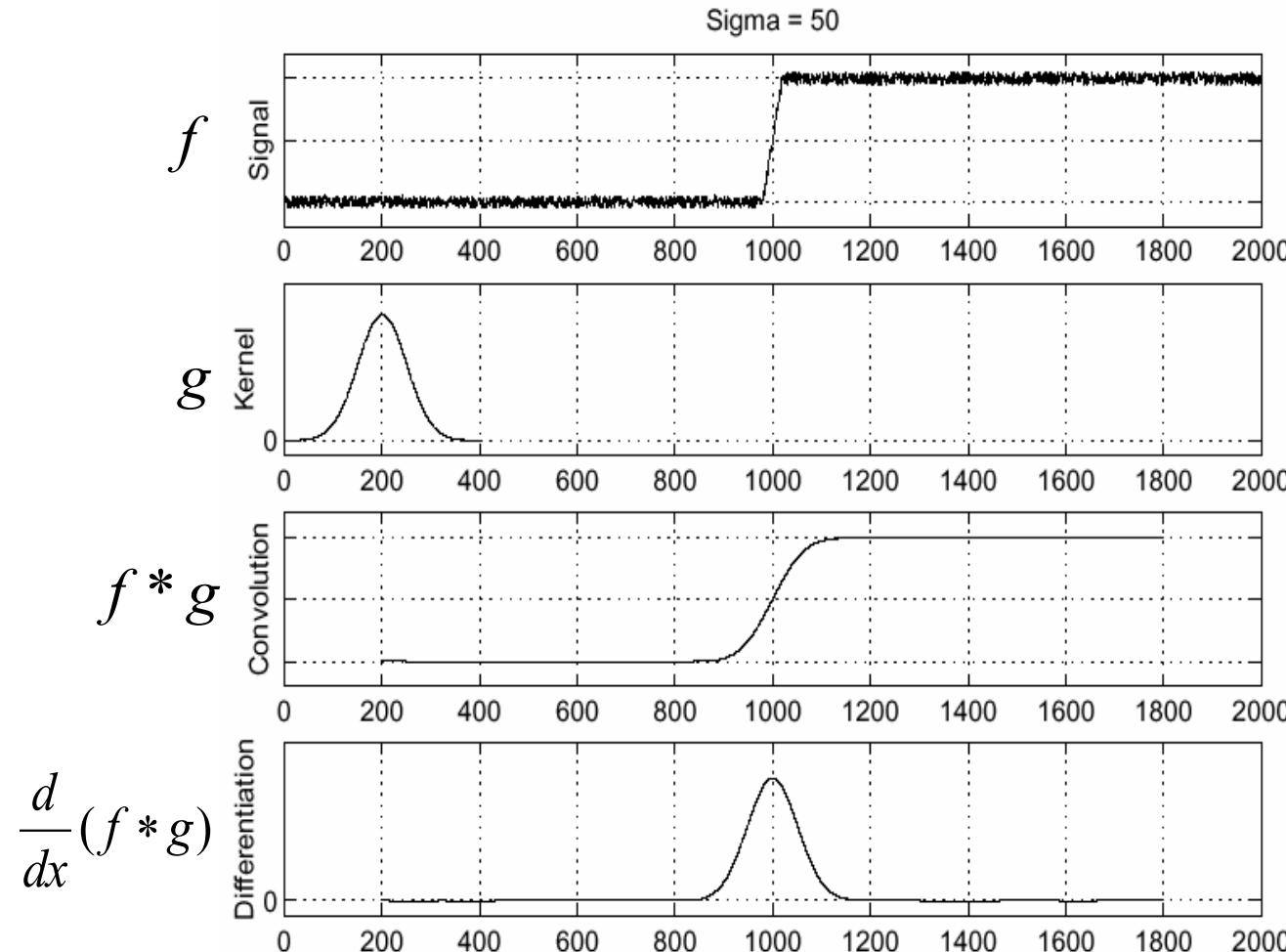


# Effect of Noise on Image Gradients

- Finite difference filters respond strongly to noise
  - Image noise results in pixels that look very different from their neighbors
  - Generally, the larger the noise the stronger the response
- What is to be done?
  - Smoothing the image should help, by forcing pixels different to their neighbors (=noise pixels?) to look more like neighbors



# Effect of Noise on Image Gradients





# Image Derivatives

- Good derivative filters are high-pass (edge) in one direction and low-pass (blur) in the orthogonal direction.

Sobel ( $X$ )

$$\frac{1}{8} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Sobel ( $Y$ )

$$\frac{1}{8} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

- Derivative/Difference of Gaussians