Welcome to

Computer Vision
Week #3 - Lecture

AJ R&D

Agenda

Week #3 - Lecture
Linear Image processing

1. Linearity and convolution
2. Filters as template
3. Edge detection: Gradients
4. Edge detection: 2D operators

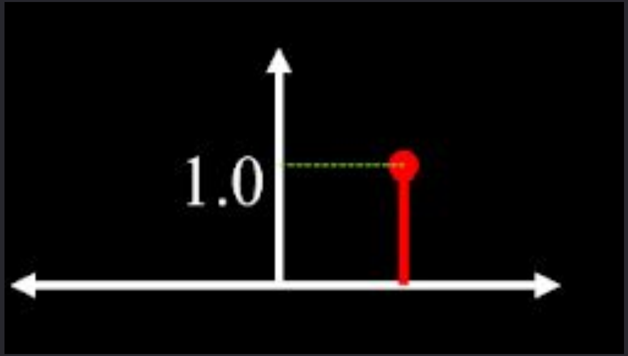
Agenda

Week #3 - Lecture
Linear Image processing

1. Linearity and convolution
2. Filters as template
3. Edge detection: Gradients
4. Edge detection: 2D operators

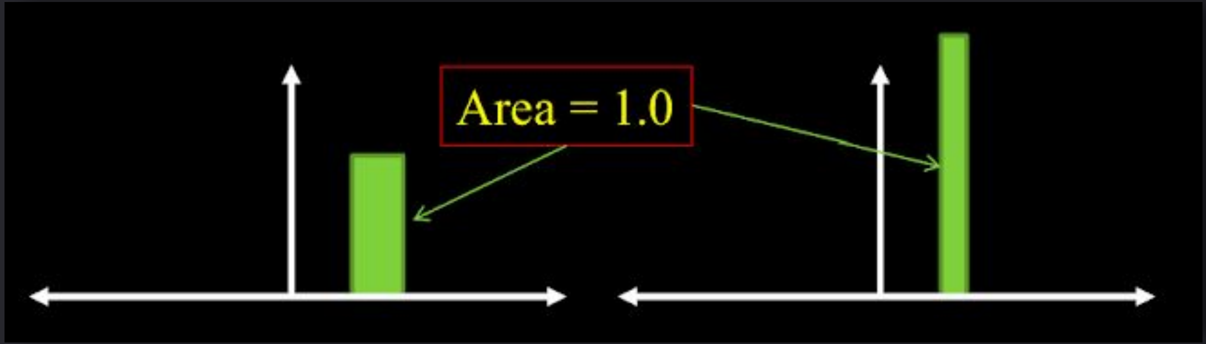
An impulse function

In the discrete world, an impulse is a very easy signal to understand: it is just a value of 1 at a single location



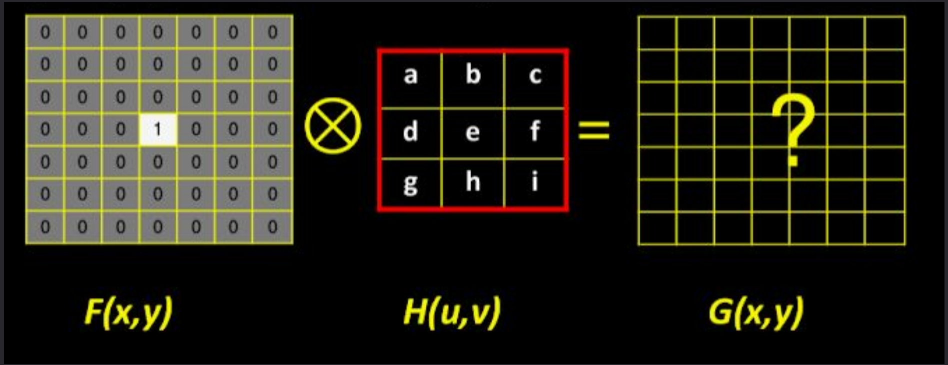
An impulse function

In the continuous world, an impulse is an idealized function that is very narrow and very tall so that it has a unit area



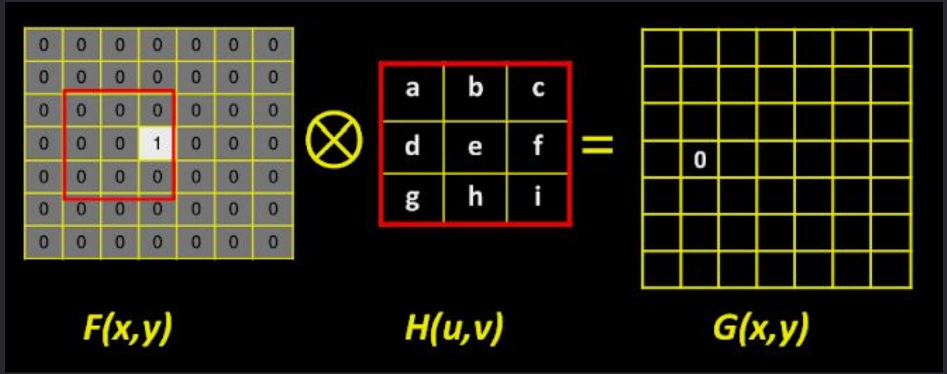
Filtering an impulse signal

What is the result of filtering the impulse signal F with the arbitrary kernel H?



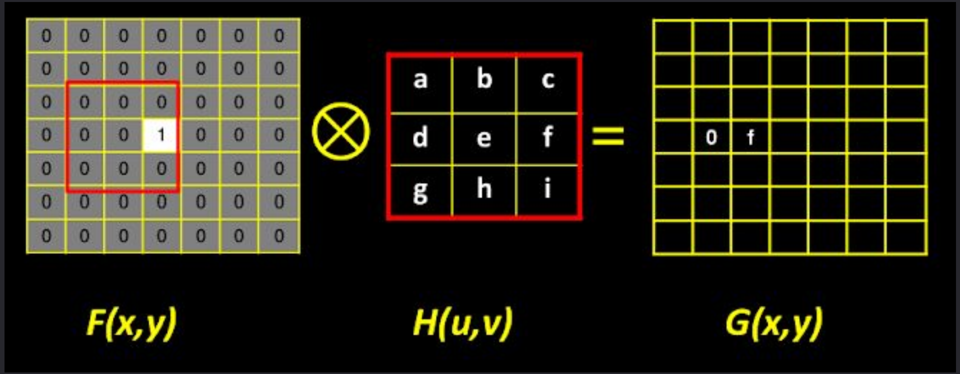
Filtering an impulse signal

What is the result of filtering the impulse signal F with the arbitrary kernel H?



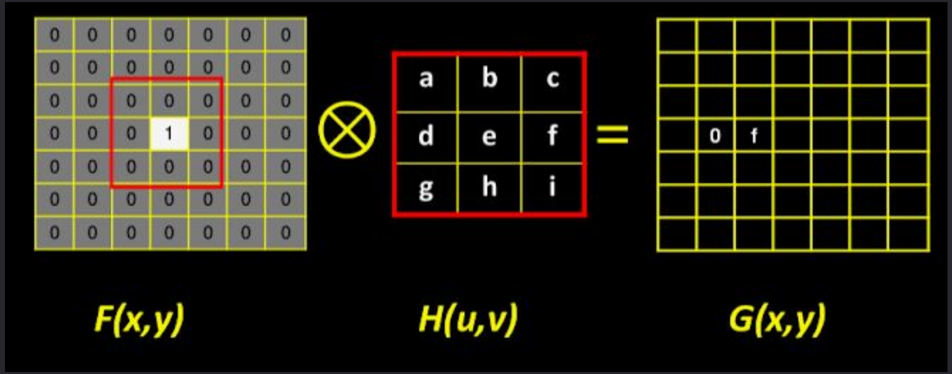
Filtering an impulse signal

What is the result of filtering the impulse signal F with the arbitrary kernel H?



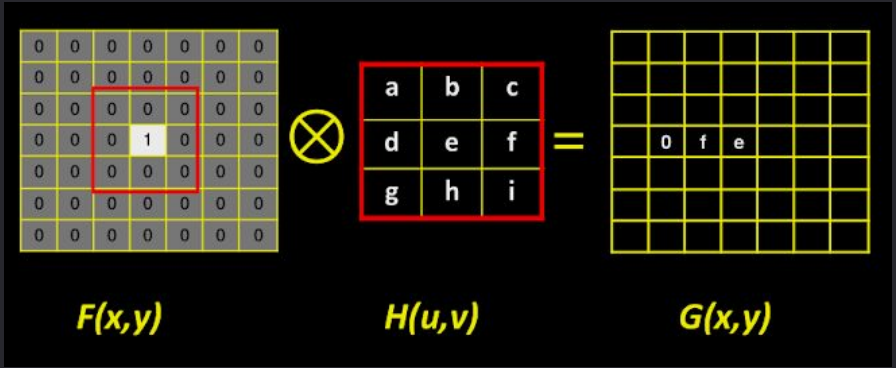
Filtering an impulse signal

What is the result of filtering the impulse signal F with the arbitrary kernel H?



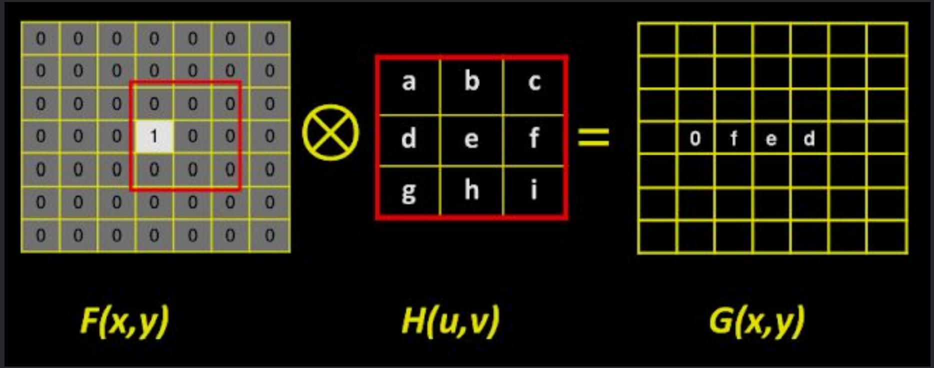
Filtering an impulse signal

What is the result of filtering the impulse signal F with the arbitrary kernel H?



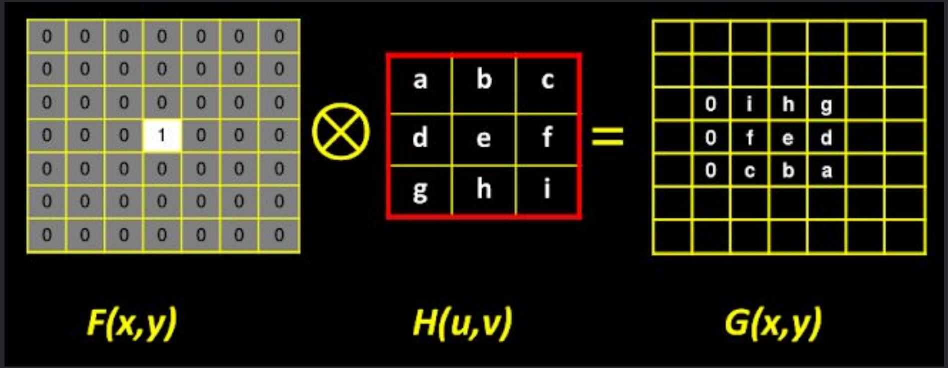
Filtering an impulse signal

What is the result of filtering the impulse signal F with the arbitrary kernel H?



Filtering an impulse signal

What is the result of filtering the impulse signal F with the arbitrary kernel H?



Correlation vs Convolution

Cross-correlation

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i+u, j+v]$$

$$G = H \otimes F$$

Convolution

Flip in both dimensions
(bottom to top, right to left)

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i-u, j-v]$$

$$G = H \star F$$

Correlation vs Convolution

Cross-correlation

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i+u, j+v]$$

$$G = H \otimes F$$

Convolution

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i-u, j-v]$$

$$G = H \star F$$

Flip in both dimensions
(bottom to top, right to left)

Convolution

Centered at zero!

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]\,F[i-u,\,j-v]$$

$$G = H * F$$

Notation for convolution operator

Properties of Convolution

- Linear & shift invariant
- Commutative:

  $f * g = g * f$
- Associative

  $(f * g) * h = f * (g * h)$
- Identity:

  unit impulse $e = [\ldots, 0, 0, 1, 0, 0, \ldots]$. $f * e = f$
- Differentiation: $\frac{\partial}{\partial x}(f * g) = \frac{\partial f}{\partial x} * g$

Separability

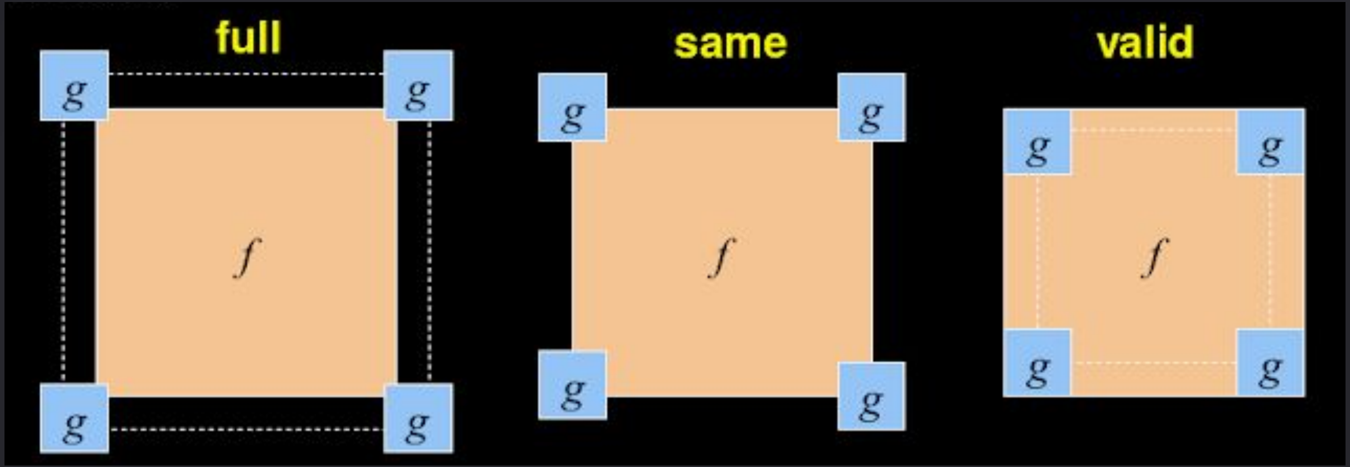
$$\mathbf{c} \; \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \overset{\mathbf{r}}{\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \; \mathbf{H}$$

$$G = H * F = (C * R) * F = C * (R * F)$$

- So we do two convolutions but each is W*N*N. So this is useful if W is big enough such that $2 \cdot W \cdot N^2 << W^2 \cdot N^2$
- Used to be ***very*** important. Still, if W=31, save a factor of 15.

Boundary issue

What is the size of the output?

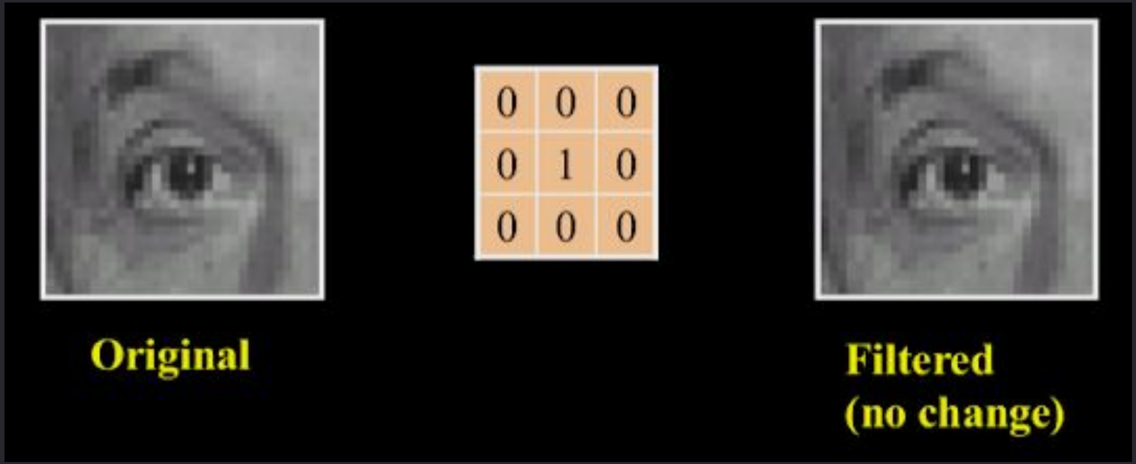


Boundary issue

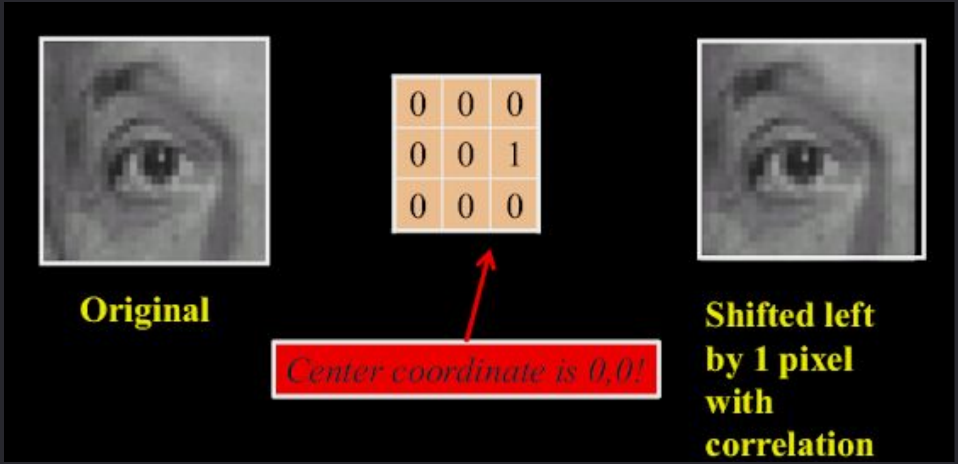
What about near the edge?

- methods:
  - clip filter (black)
  - wrap around
  - copy edge
  - reflect across edge

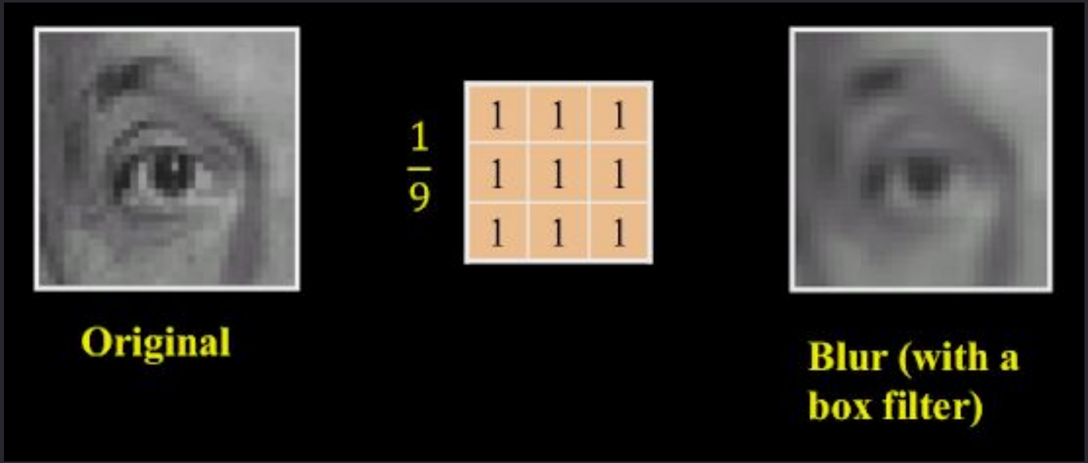
Practice with linear filters



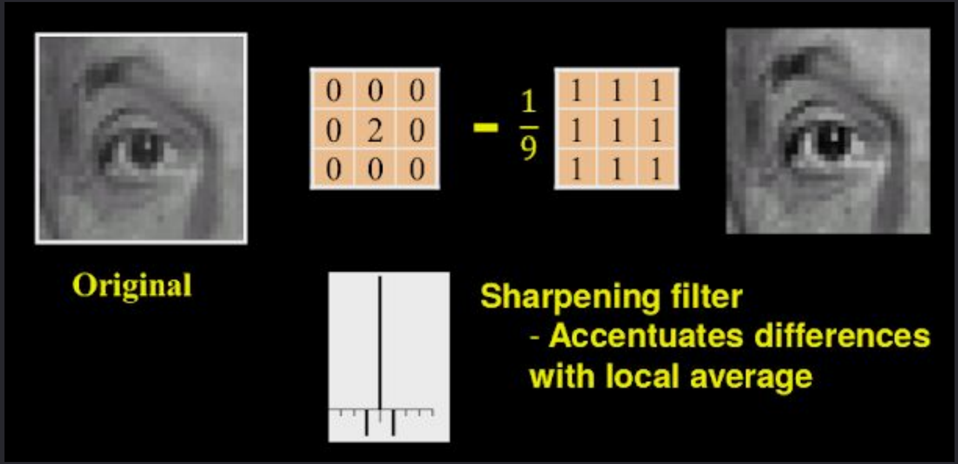
Practice with linear filters



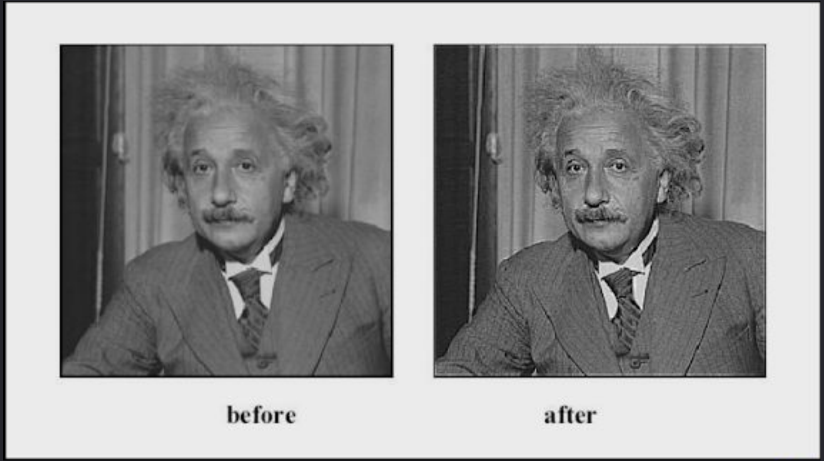
Practice with linear filters



Practice with linear filters

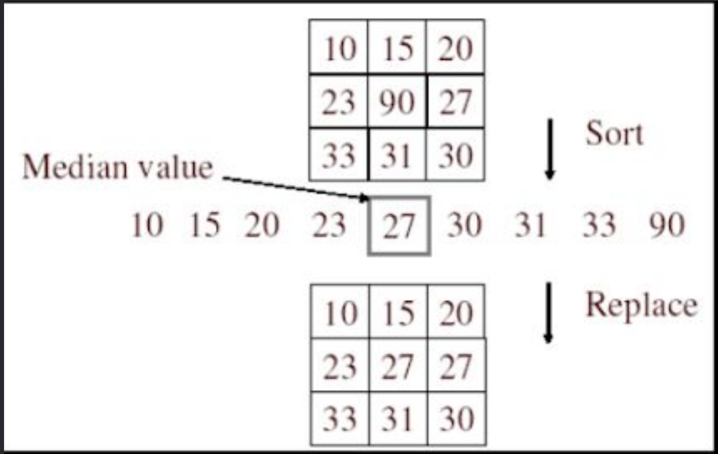


Practice with linear filters



Practice with linear filters

Median filter



- No new pixel values introduced
- Removes spikes: good for impulse, salt & pepper noise
- Linear?

Practice with linear filters

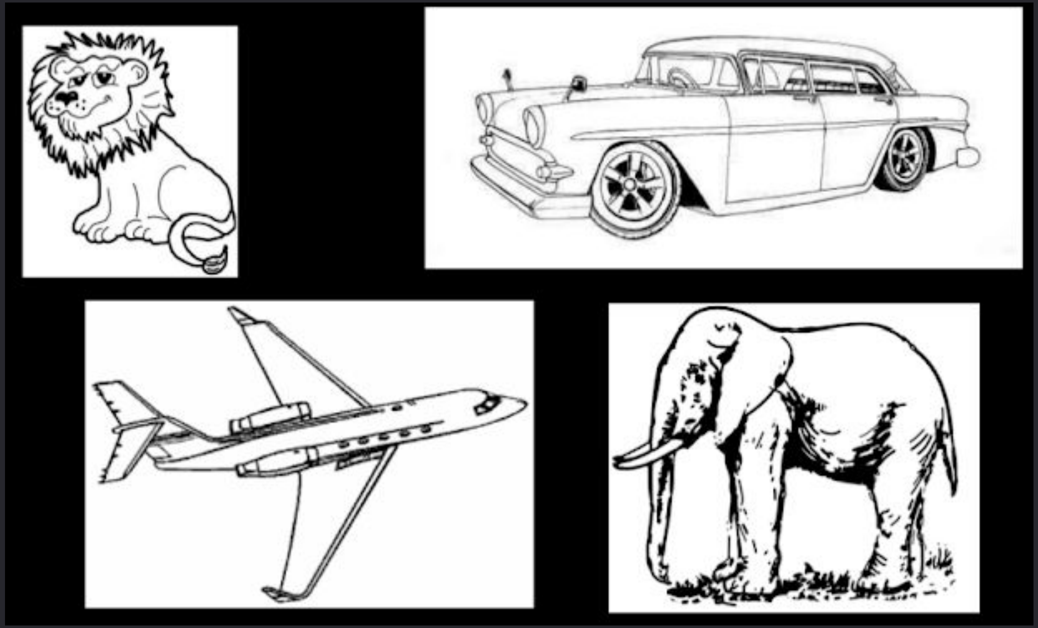
Median filter

Agenda

Week #3 - Lecture
Linear Image processing

1. Linearity and convolution
2. Filters as template
3. Edge detection: Gradients
4. Edge detection: 2D operators

Reduced Images



How do we identify edges in image?

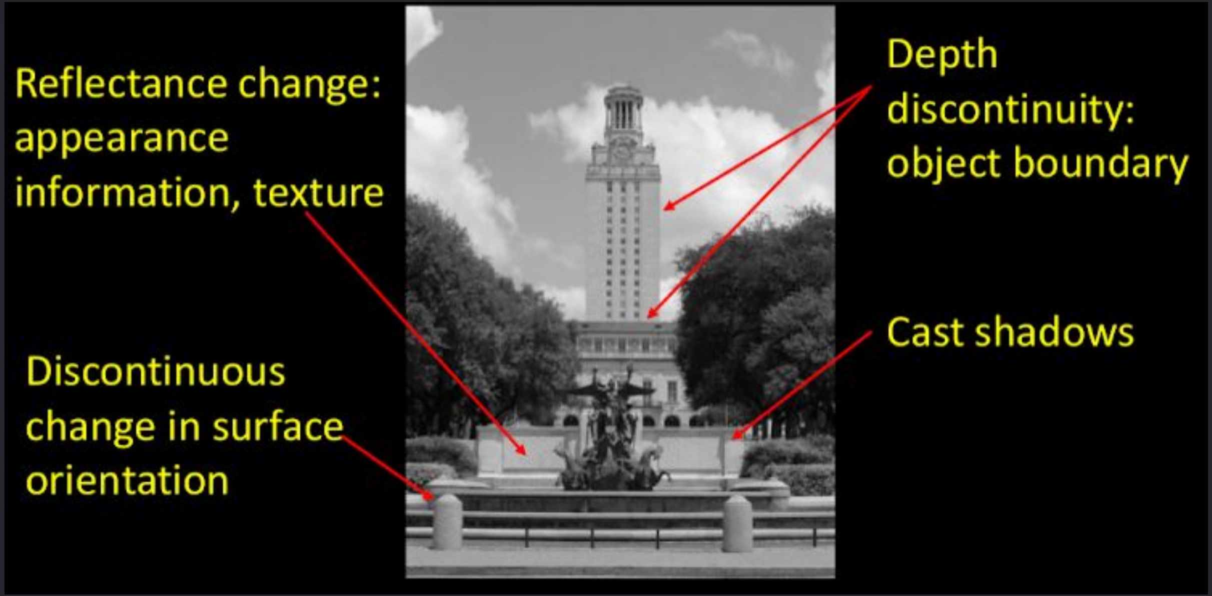
Sample Images

How do we identify edges in image?

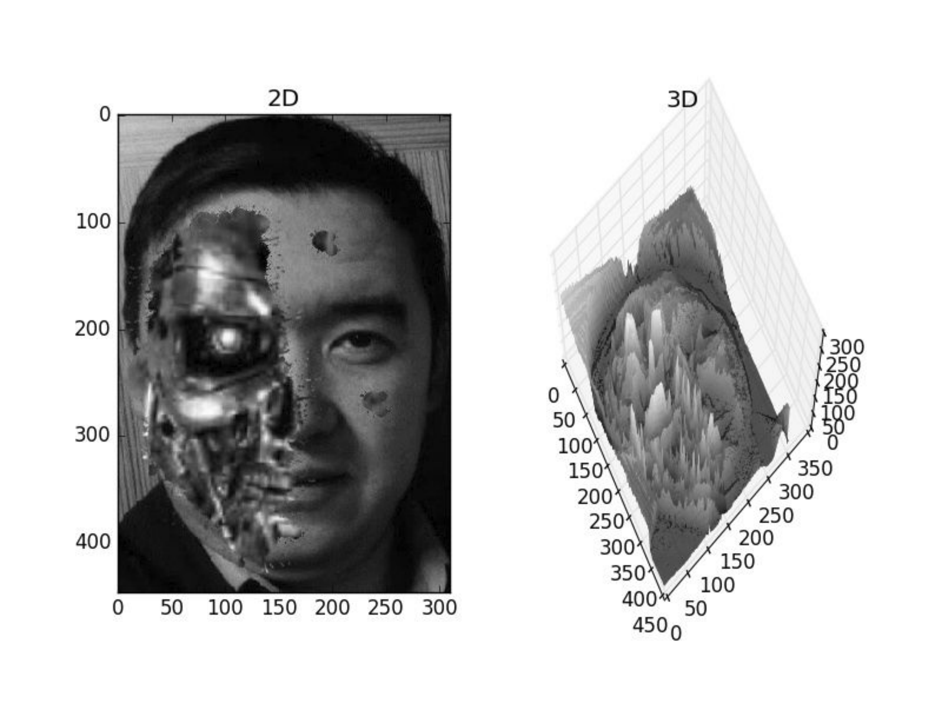
Goal of edge detection:

Identify sudden changes (discontinuities) in image

In a real image



In a real image



Edge detection

Basic idea: look for a neighborhood with strong signs of change.

Problems:

- neighborhood size
- how to detect change

81	82	26	24
82	33	25	25
81	82	26	24

Edge detection

An edge is a place of rapid change in the image intensity function

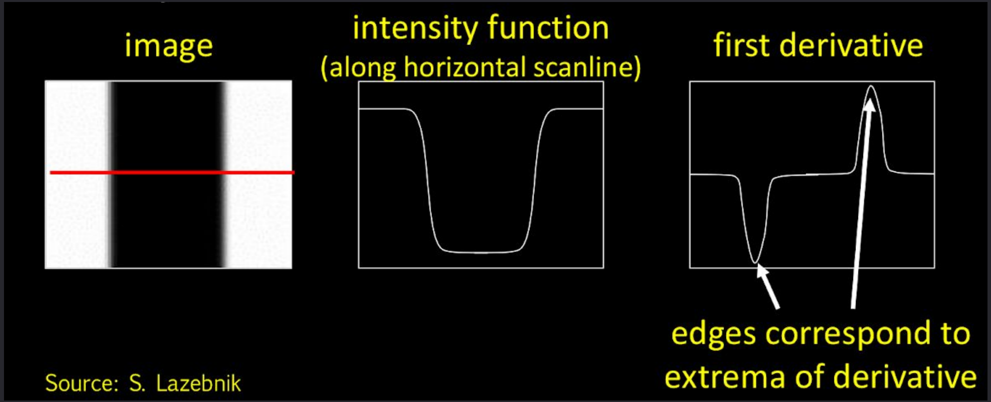


Image gradient

The gradient of an image:

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

Edge detection: Gradient |

Image gradient

The gradient of an image:

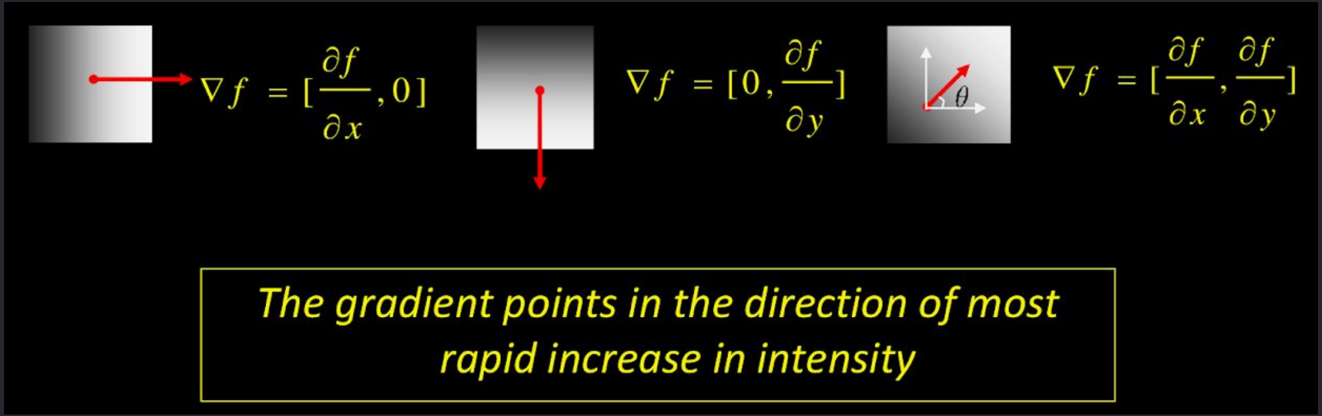


Image gradient

The gradient of an image:

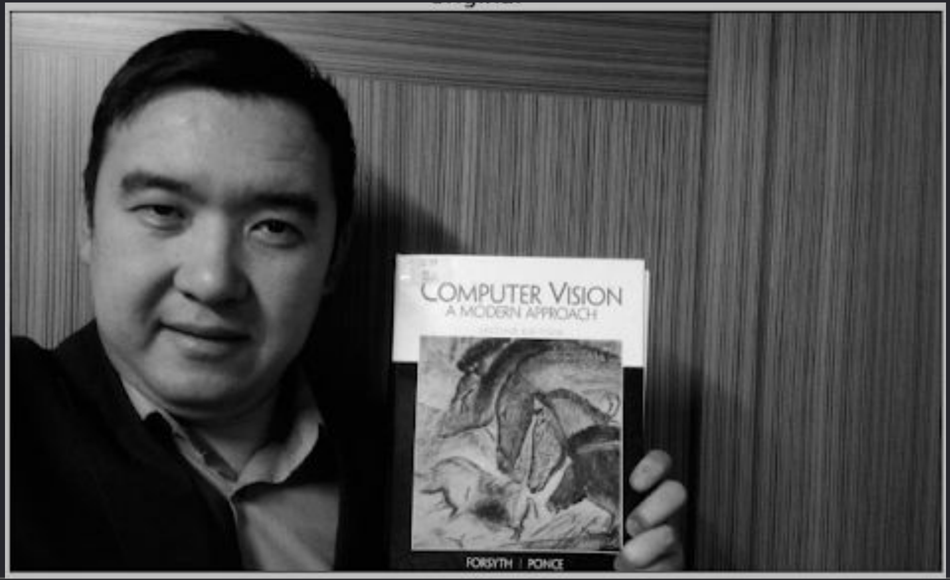
The gradient of an image: $\nabla f = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}]$

The gradient direction is given by: $\theta = \tan^{-1}(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x})$

The *edge strength* is given by the gradient magnitude: $\|\nabla f\| = \sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2}$

Finite differences

Original image



Finite differences

x or y direction?

Edge detection: Gradient |

Finite differences

x or y direction?

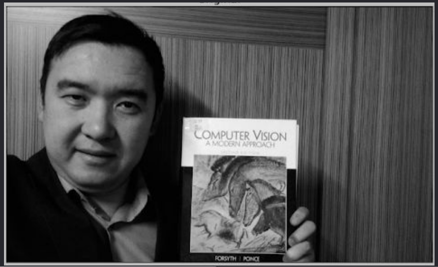
Partial Derivatives of an Image

x or y direction?

Partial Derivatives of an Image

x or y direction?

$$\frac{\partial f(x, y)}{\partial x}$$

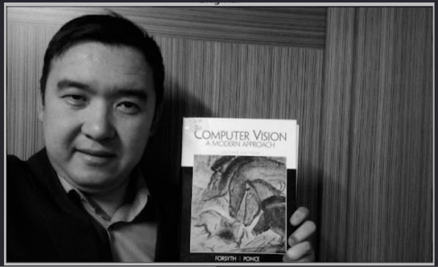


$$\frac{\partial f(x, y)}{\partial y}$$

Partial Derivatives of an Image

x or y direction?

$$\frac{\partial f(x, y)}{\partial x}$$



$$\frac{\partial f(x, y)}{\partial y}$$

Sobel Operator

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

$$s_x \qquad\qquad s_y$$

(Sobel) Gradient is $\nabla \mathbf{I} = [\mathbf{g_x} \;\; \mathbf{g_y}]^{\mathbf{T}}$

$g = (g_x{}^2 + g_y{}^2)^{1/2}$ is the gradient magnitude.

$\theta = \mathrm{atan2}(g_y, g_x)$ is the gradient direction.

Some well known Gradient Masks

	Sx	Sy

- **Sobel:**

Sx:

-1	0	1
-2	0	2
-1	0	1

Sy:

1	2	1
0	0	0
-1	-2	-1

- **Prewitt:**

Sx:

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

Sy:

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

- **Roberts:**

Sx:

0	1
-1	0

Sy:

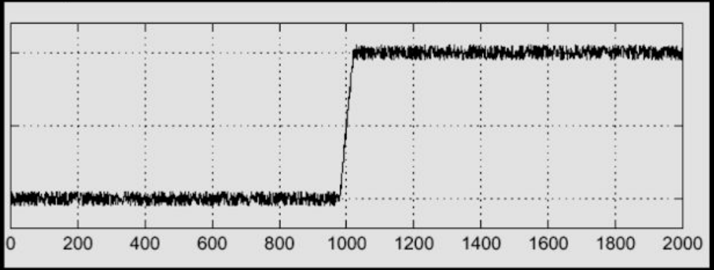
1	0
0	-1

In real world -

Consider a single row or column of the image

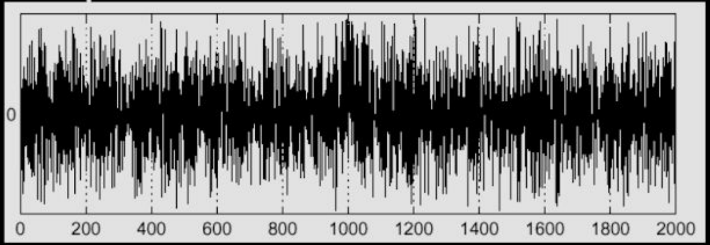
(plotting intensity as a function of *x)*

$f(x)$

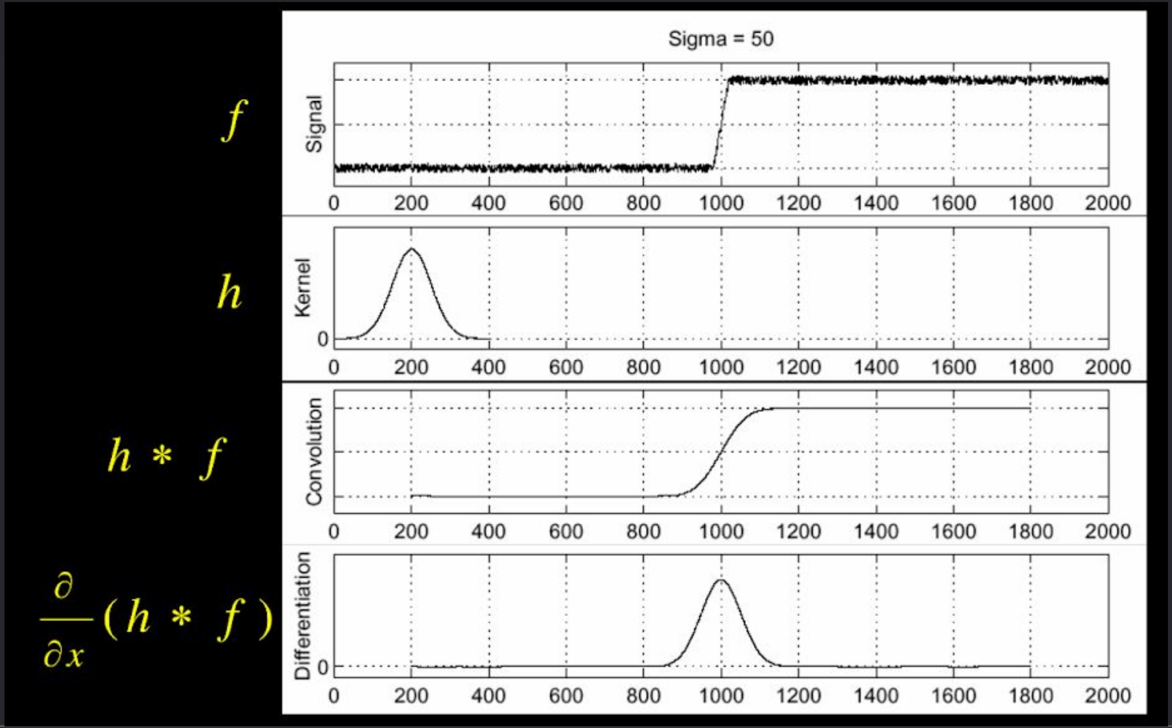


Apply derivative operator....

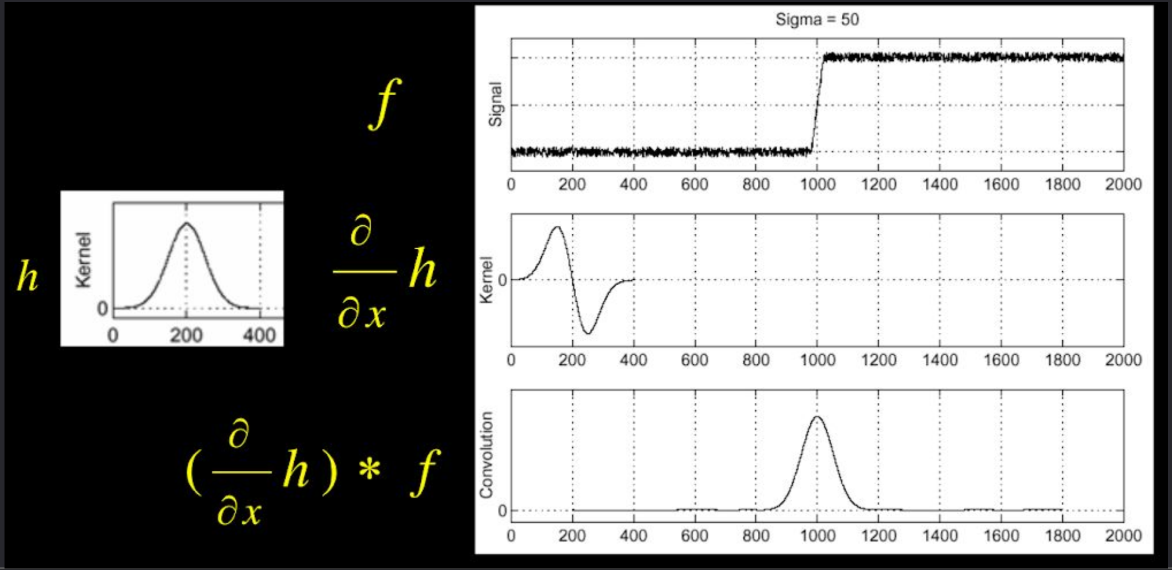
$\frac{d}{dx} f(x)$



Smooth first -



Smooth first -

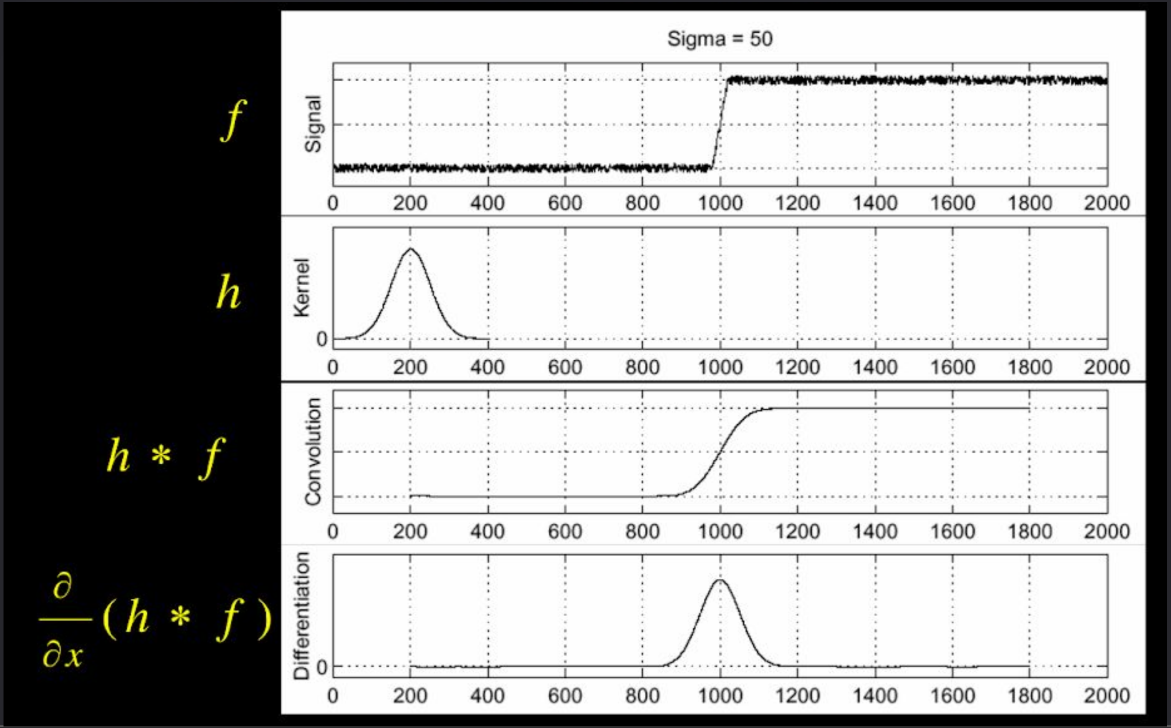


Agenda

Week #3 - Lecture
Linear Image processing

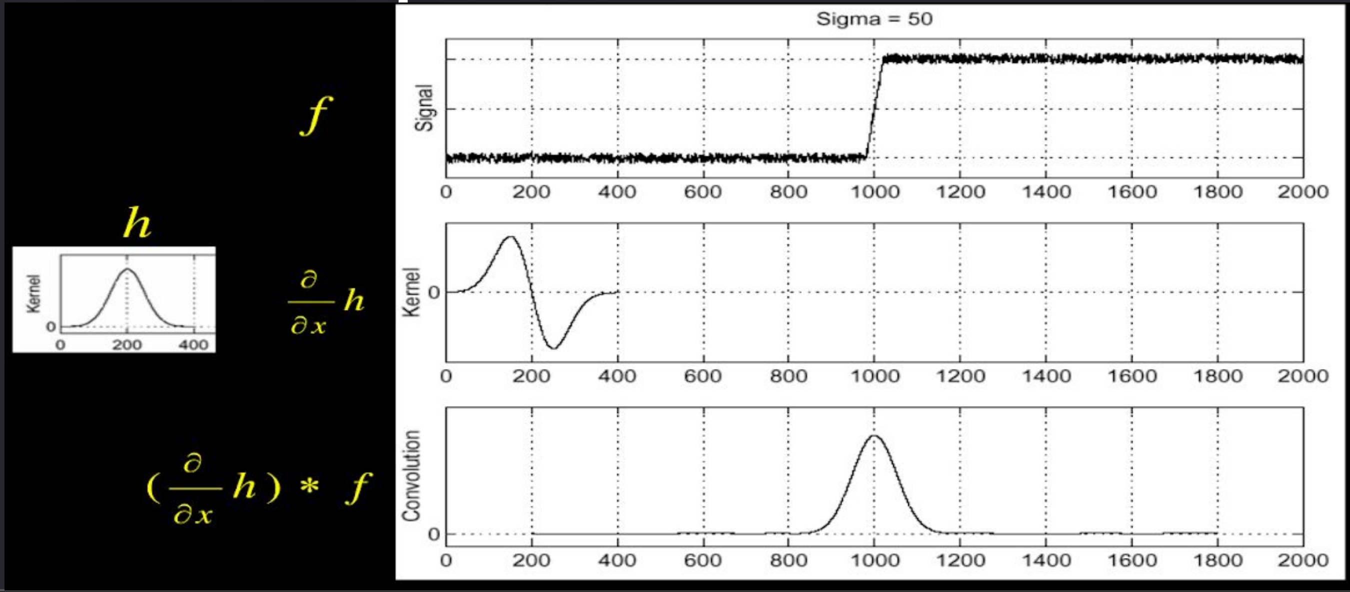
1. Linearity and convolution
2. Filters as template
3. Edge detection: Gradients
4. **Edge detection: 2D operators**

Derivative theorem of convolution - 1D



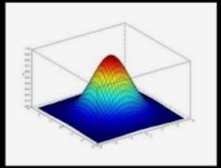
Derivative theorem of convolution - 1D

Saves one operation:

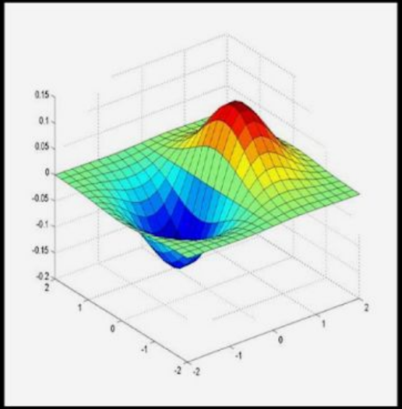


Derivative of Gaussian filter - 2D

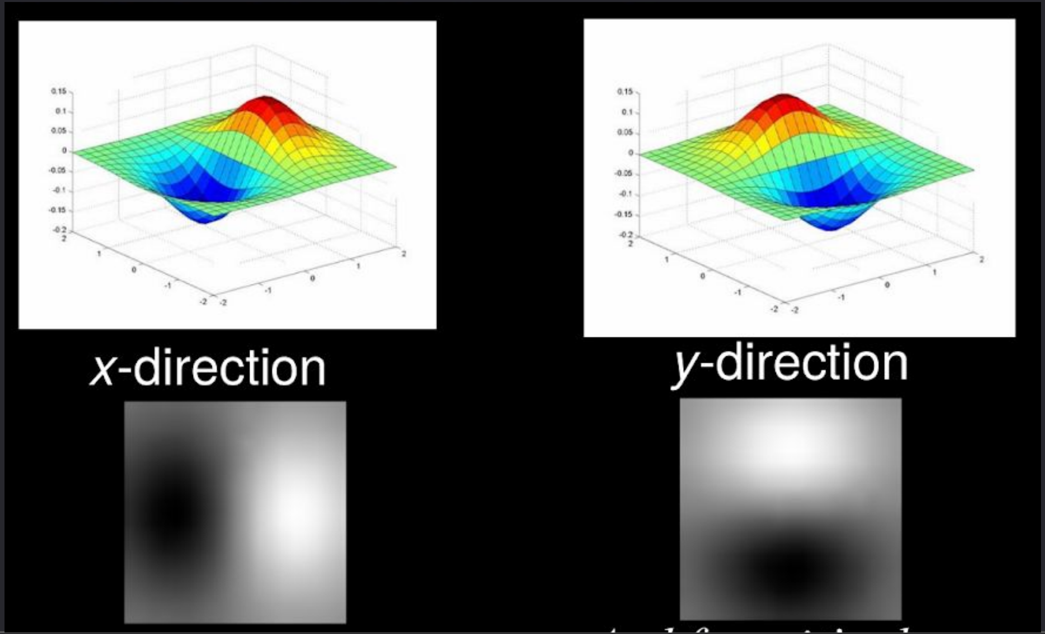
$$(I \otimes g) \otimes h = I \otimes (g \otimes h)$$



$$\begin{bmatrix} 0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \\ 0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\ 0.0219 & 0.0983 & 0.1621 & 0.0983 & 0.0219 \\ 0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\ 0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \end{bmatrix} \otimes \begin{bmatrix} -1 & 1 \end{bmatrix} =$$



Derivative of Gaussian filter



Smoothing with Gaussian

Sigma 1

Sigma 3

Sigma 9

Effects of sigma on derivatives

Sigma 1

Larger values: larger scale edges detected

Sigma 3

Sigma 9

Smaller values: finer features detected

Canny edge detector

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. “Thin” to get localized edge pixels
4. Link or connect edge pixels

Primary edge detection steps

1. Smoothing derivatives to suppress noise and compute gradient
2. Threshold to find regions of “significant” gradient

Primary edge detection steps

1. **Non-maximum suppression:**

   Thin multi-pixel wide “ridges” down to single pixel width

2. **Linking and thresholding (hysteresis):**

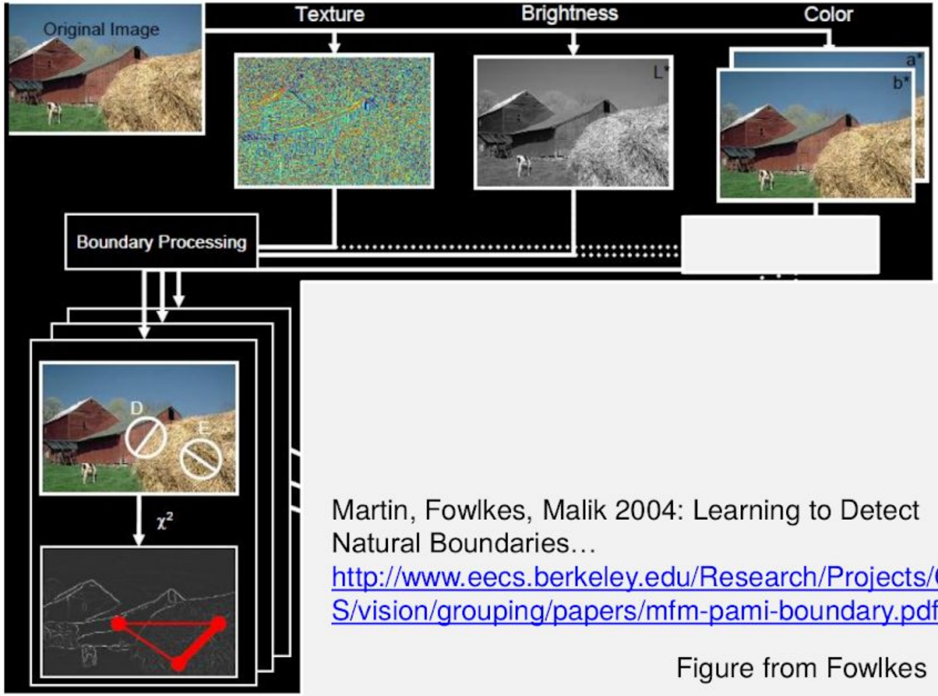
   Define two thresholds: low and high
   Use the high threshold to start edge curves and the low threshold to continue them

Canny edge


Original Image
Edge Image
COMPUTER VISION
A MODERN APPROACH
FORSYTH | PONCE

Canny edge

pB boundary detector

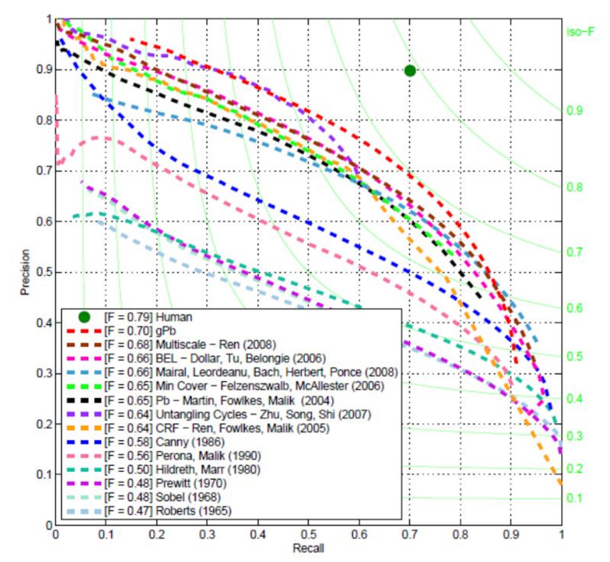


Martin, Fowlkes, Malik 2004: Learning to Detect Natural Boundaries…
http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/papers/mfm-pami-boundary.pdf

Figure from Fowlkes

pB boundary detector

45-years of boundary detection



Source: Arbelaez, Maire, Fowlkes, and Malik. TPAMI 2011 (pdf)

References & Sources |

1. https://www.udacity.com/course/introduction-to-computer-vision--ud810

Thank you for your time!