ĐẠI HỌC TÔN ĐỨC THẮNG
Ton Duc Thang University (TDTU)

Digital Image Processing

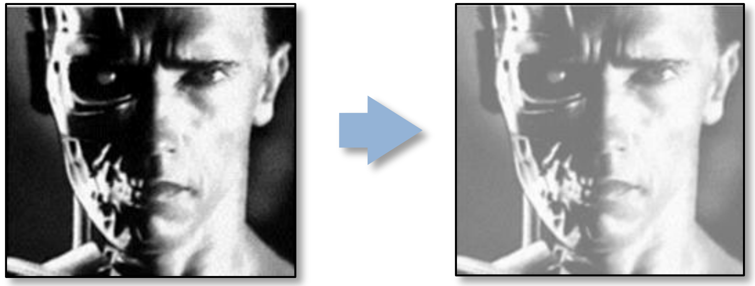
PHAM VAN HUY (PhD.) – INFORMATION TECHNOLOGY FACULTY
Email: phamvanhuy@tdtu.edu.vn

Image filtering

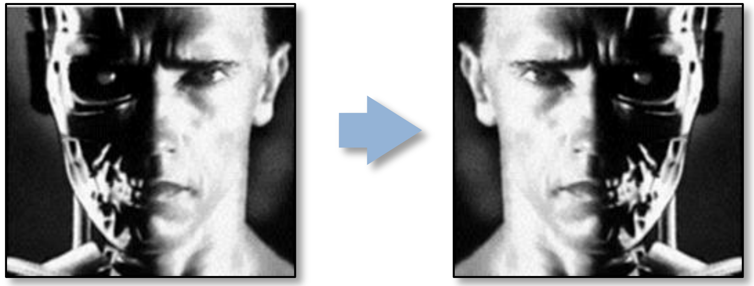
In Digital Image Processing

Image transformations

- As with any function, we can apply operators to an image



$g(x,y) = f(x,y) + 20$



$g(x,y) = f(-x,y)$

- We'll talk about a special kind of operator, *convolution* (linear filtering)

Question: Noise reduction

ĐẠI HỌC TÔN ĐỨC THẮNG
Ton Duc Thang University (TDTU)

- Given a camera and a still scene, how can you reduce noise?

Take lots of images and average them!

What's the next best thing?

Source: S. Seitz

Image filtering

- Modify the pixels in an image based on some function of a local neighborhood of each pixel

10	5	3
4	5	1
1	1	7

Local image data

Some function →

	7	

Modified image data

Source: L. Zhang

Image filtering

- **Filtering:**
  – Form a new image whose pixels are a combination original pixel values

Goals:

-Extract useful information from the images
- Features (edges, corners, blobs...)

- Modify or enhance image properties:
- super-resolution; in-painting; de-noising

De-noising

Salt and pepper noise

Super-resolution

In-painting

Bertamio et al

Fei-Fei Li Lecture 4- 11 6-Oct-16

Linear filtering

- One simple version: linear filtering (cross-correlation, convolution)
  - Replace each pixel by a linear combination of its neighbors
- The prescription for the linear combination is called the “kernel” (or “mask”, “filter”)

10	5	3
4	6	1
1	1	8

Local image data

0	0	0
0	0.5	0
0	1	0.5

kernel

	8	

Modified image data

Source: L. Zhang

Convolution

- Same as cross-correlation, except that the kernel is “flipped” (horizontally and vertically)

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

This is called a **convolution** operation:

$$G = H * F$$

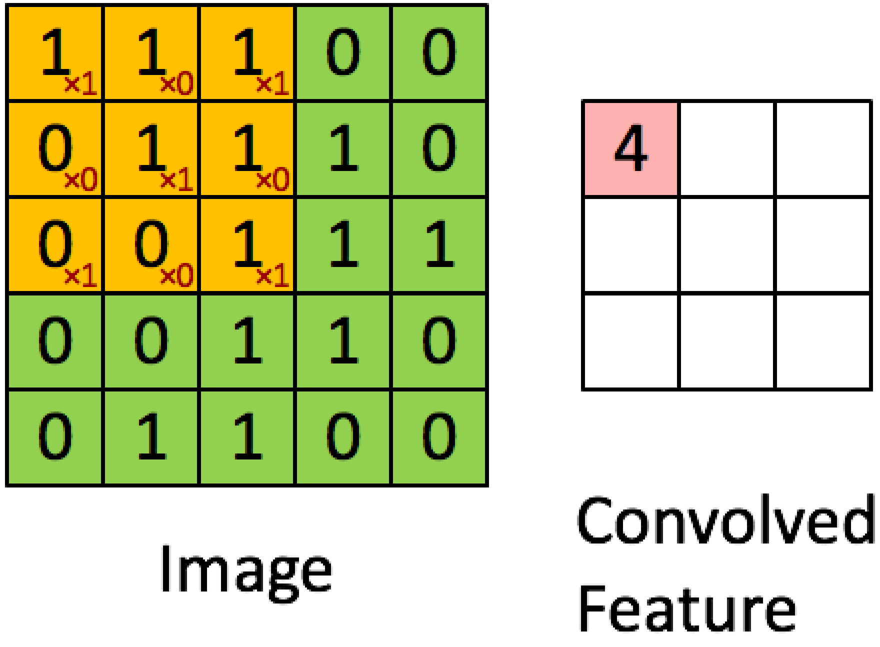
- Convolution is **commutative** and **associative**

2D Convolution

$$g(x,y) = h(x,y) * f(x,y)$$

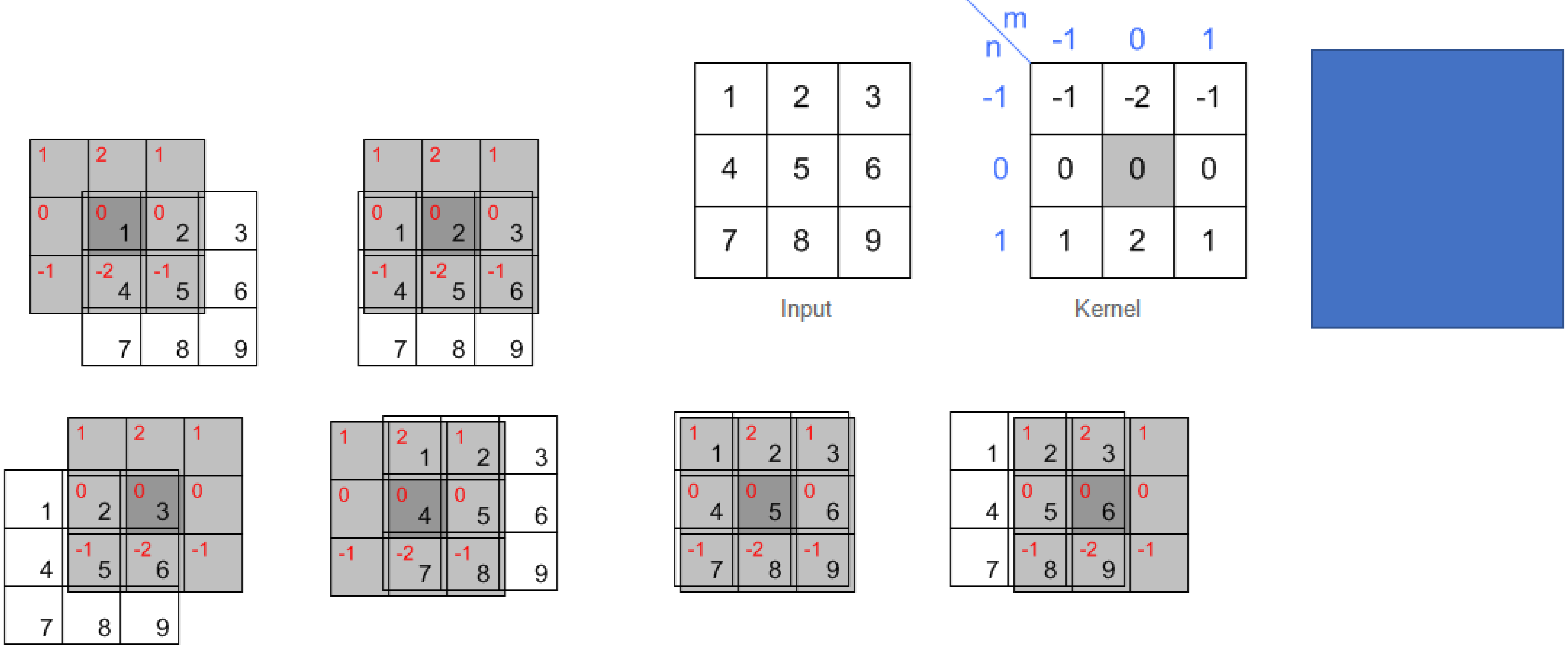
- f, g: input/output
- h: mask/filter/kernel

1. Flip the mask (horizontally and vertically) only once
2. Slide the mask onto the image.
3. Multiply the corresponding elements and then add them
4. Repeat this procedure until all values of the image has been calculated.

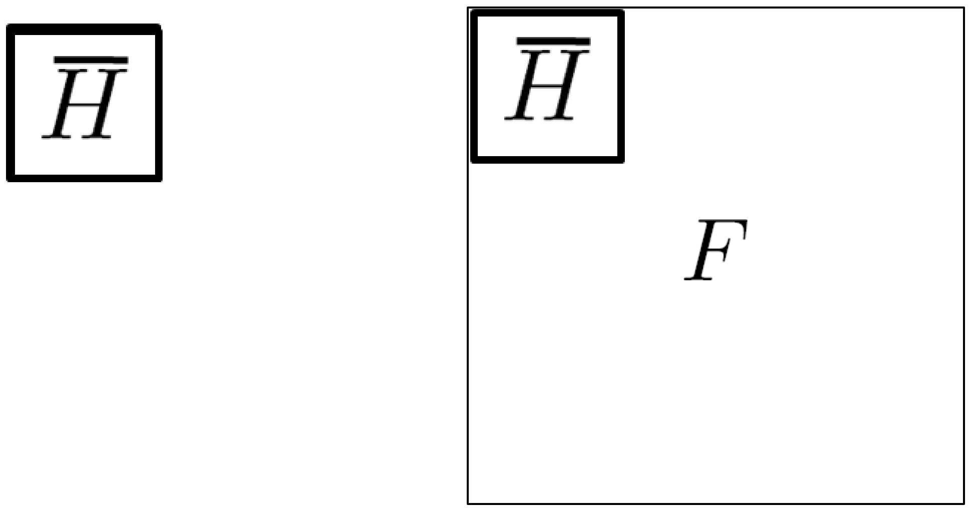


Example

- http://www.songho.ca/dsp/convolution/convolution2d example.html



Convolution



Adapted from F. Durand

Cross-correlation

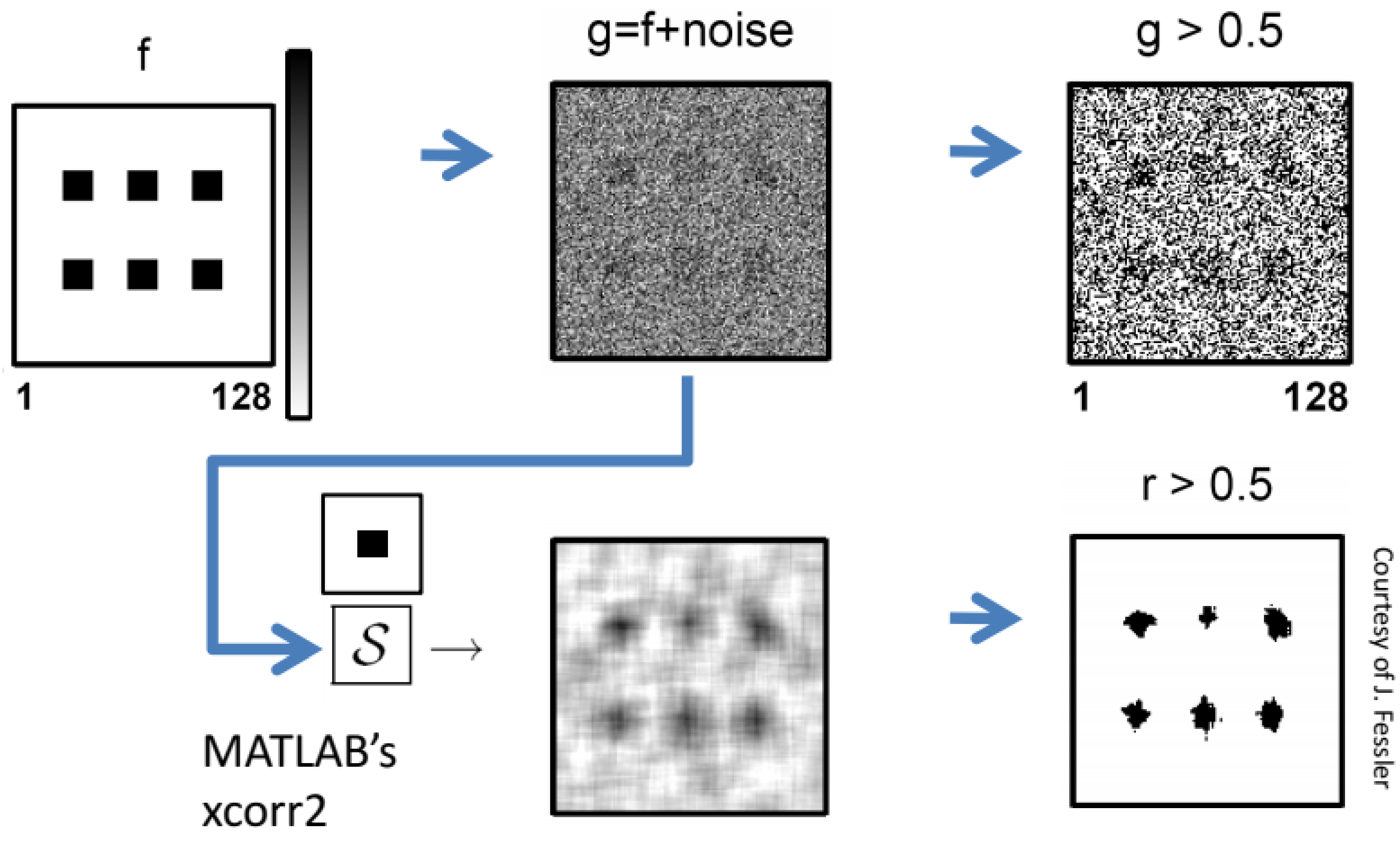
Let F be the image, H be the kernel (of size 2k+1 x 2k+1), and G be the output image

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

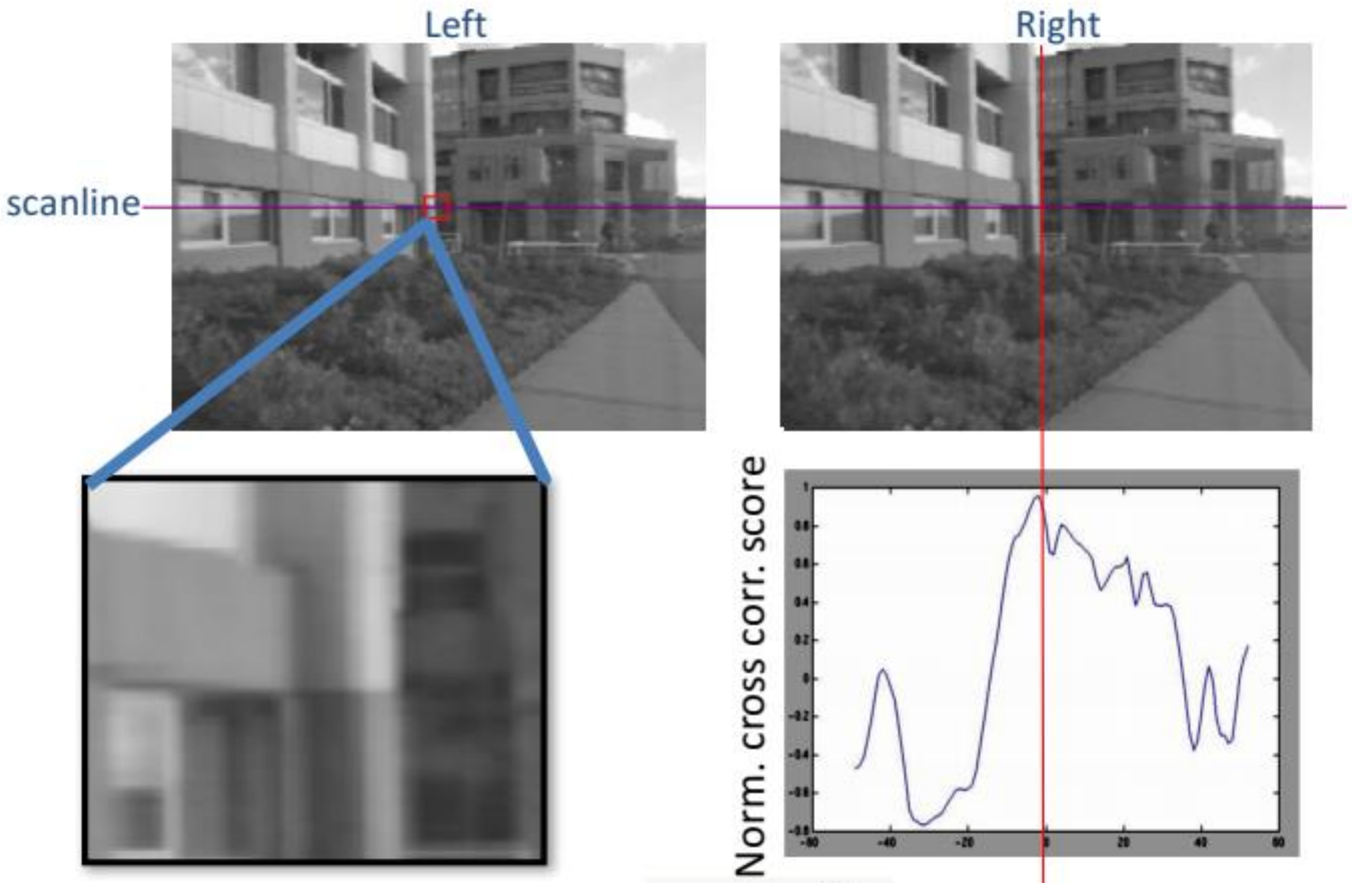
This is called a **cross-correlation** operation:

$$G = H \otimes F$$

(Cross) correlation – example



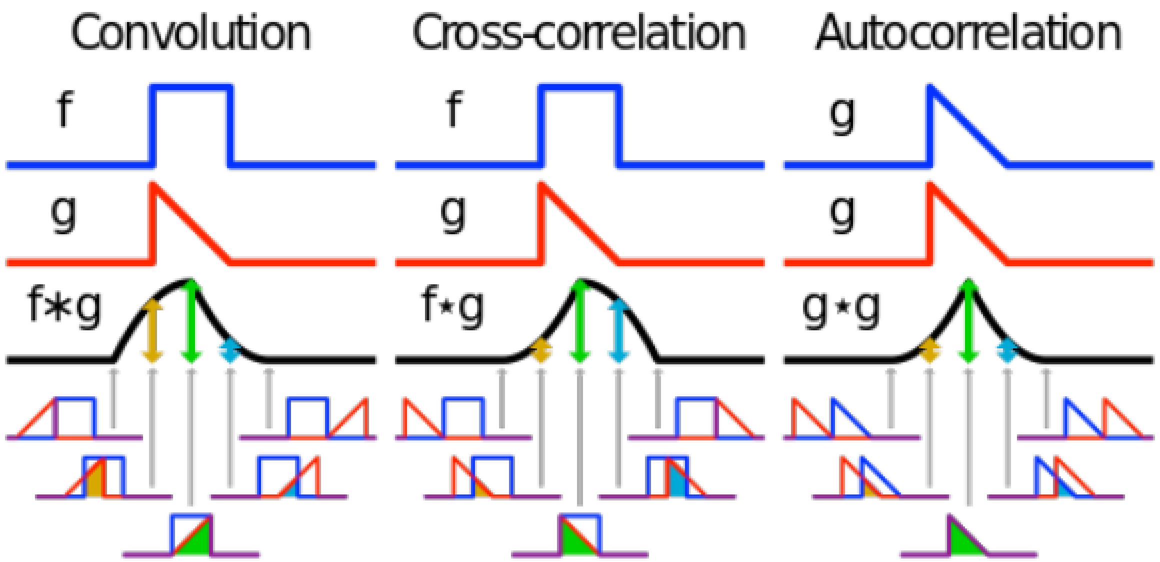
(Cross) correlation – example



$$dc(y_1, y_2) = \frac{y_1^T y_2}{|y_1||y_2|}$$

Fei-Fei Li Lecture 4- 50 6-Oct-16

Convolution vs. (Cross) Correlation



Fei-Fei Li Lecture 4- 51 6-Oct-16

Convolution vs. (Cross) Correlation

- A **<u>convolution</u>** is an integral that expresses the amount of overlap of one function as it is shifted over another function.
  - convolution is a filtering operation

- **<u>Correlation</u>** compares the ***similarity*** *of* ***two*** *sets of* ***data.*** Correlation computes a measure of similarity of two input signals as they are shifted by one another. The correlation result reaches a maximum at the time when the two signals match best .
  - correlation is a measure of relatedness of two signals

Fei-Fei Li Lecture 4- 52 6-Oct-16

Cross Correlation Application: Vision system for TV remote control

- uses template matching

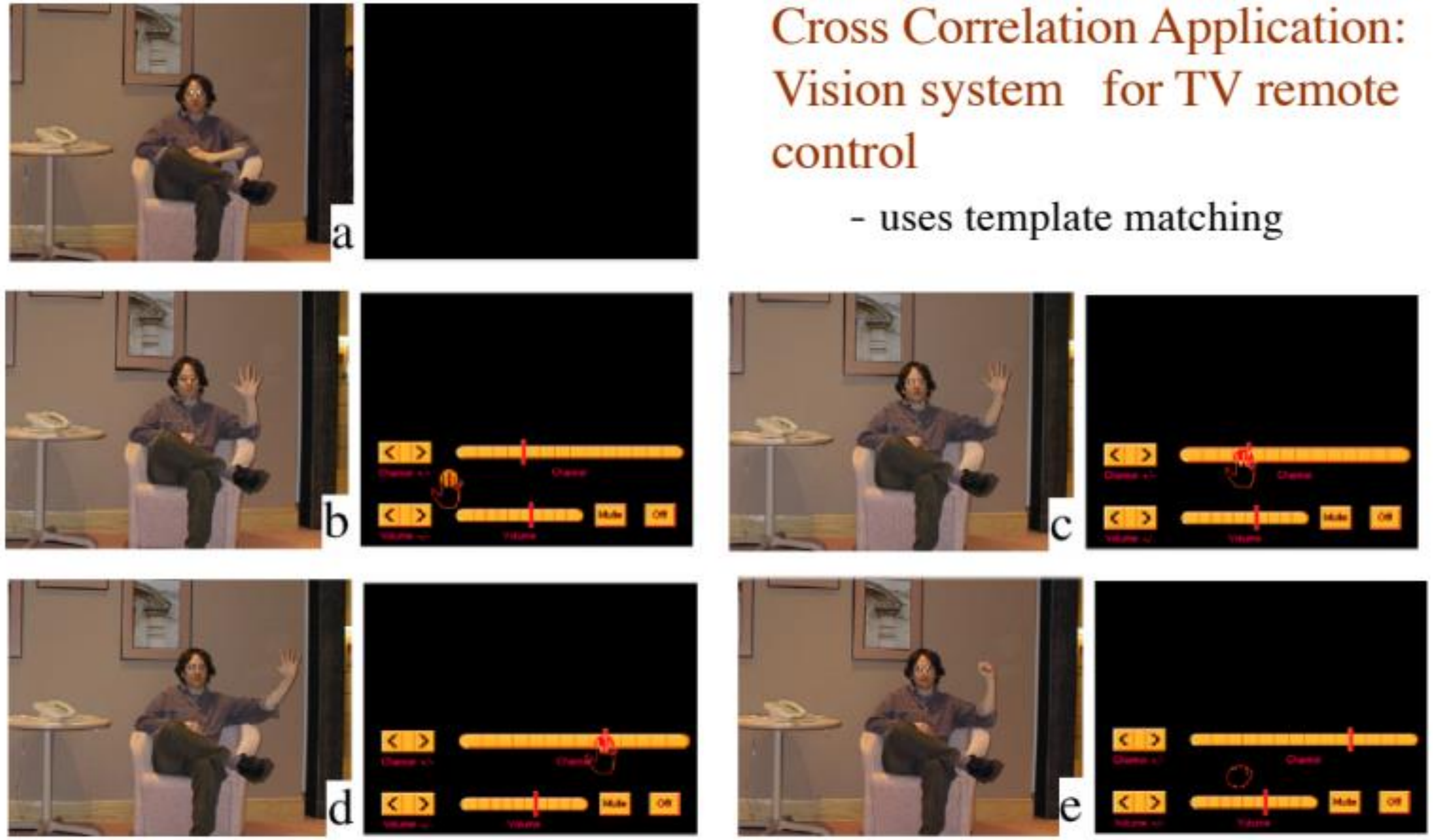


Figure from "Computer Vision for Interactive Computer Graphics," W.Freeman et al, IEEE Computer Graphics and Applications, 1998 copyright 1998, IEEE

Mean filtering

H

$*$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

F

$=$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	$\boxed{20}$	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

G

Linear filters: examples

Original

$$* \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} =$$

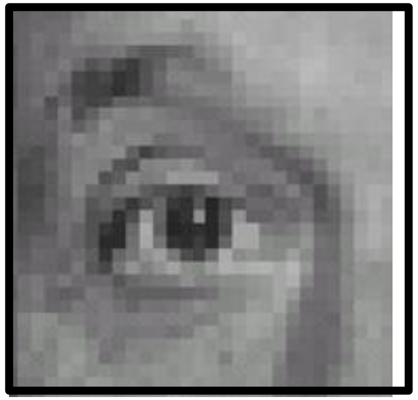
Identical image

Source: D. Lowe

Linear filters: examples

Original

$$* \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} =$$



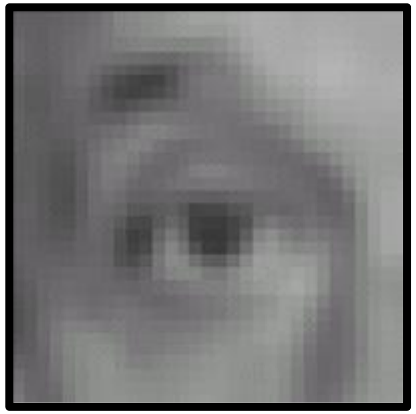
Shifted left
By 1 pixel

Source: D. Lowe

Linear filters: examples

Original

$$* \; \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} =$$



Blur (with a mean filter)

Source: D. Lowe

Linear filters: examples

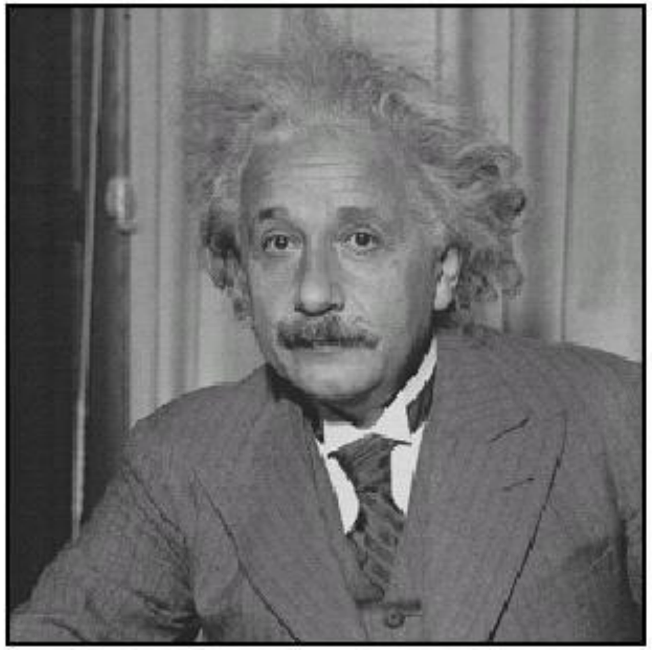
Original

$$* \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right) =$$

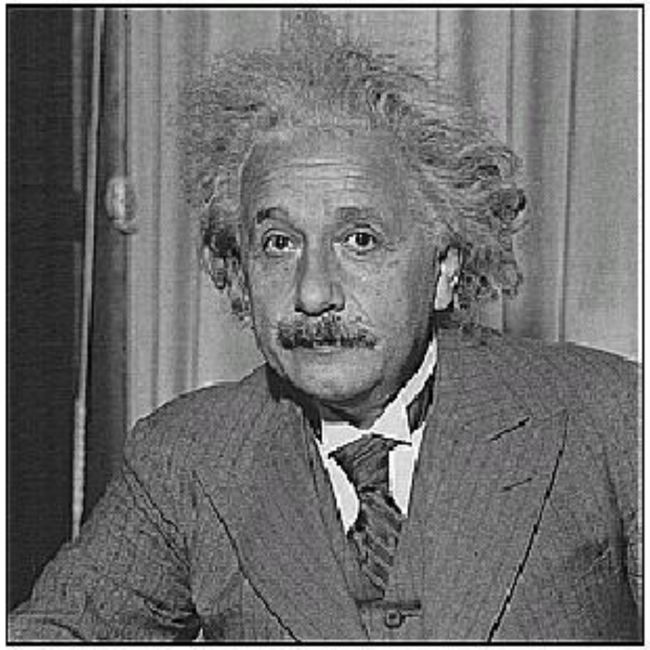
Sharpening filter
(accentuates edges)

Source: D. Lowe

Sharpening



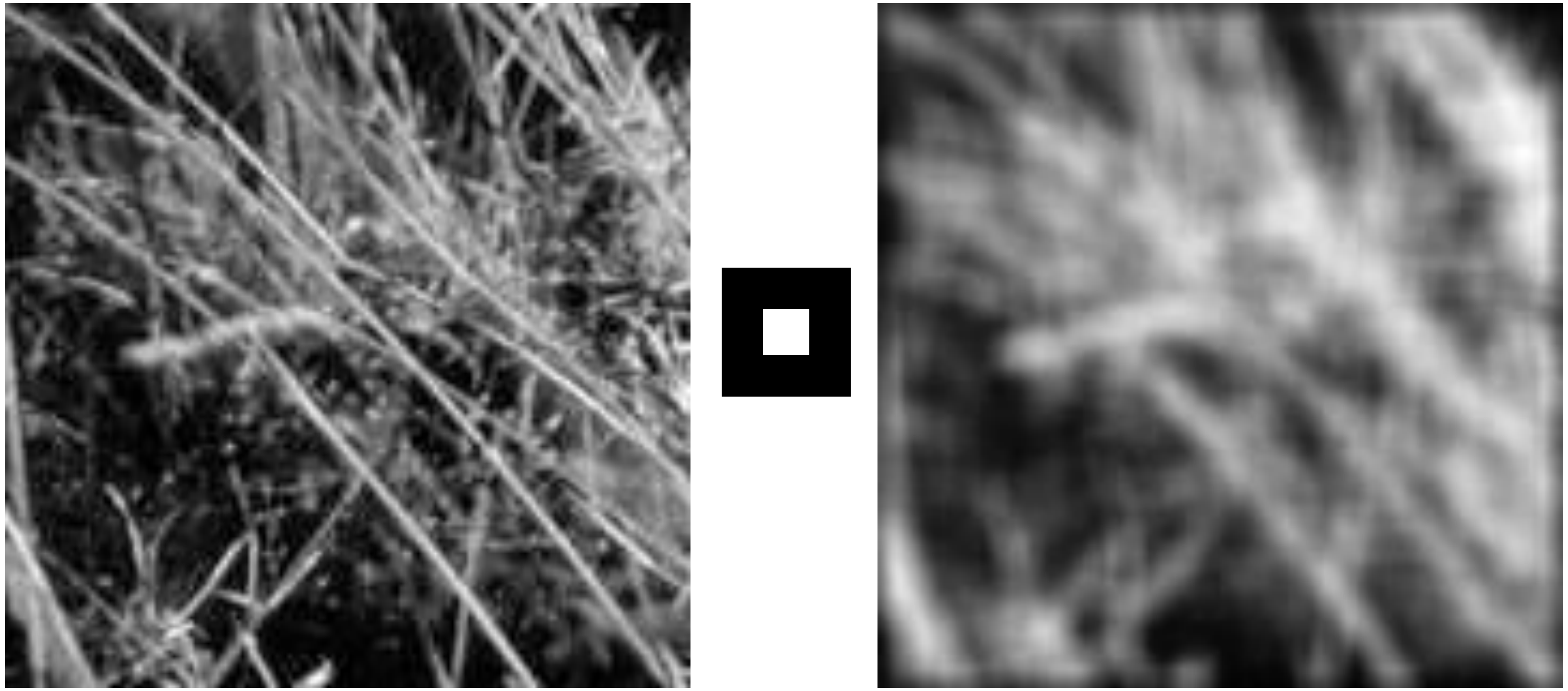
before



after

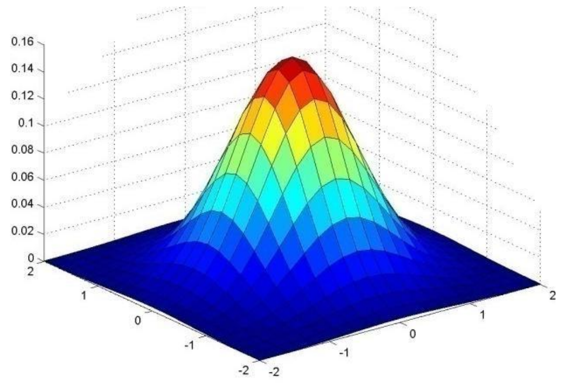
Source: D. Lowe

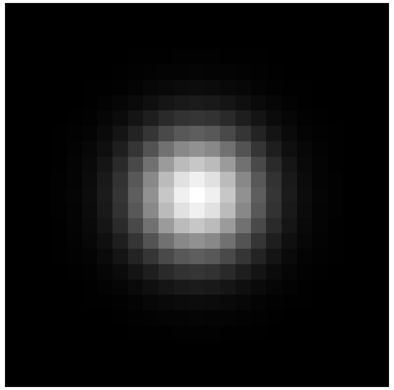
Smoothing with box filter revisited



Source: D. Forsyth

Gaussian Kernel

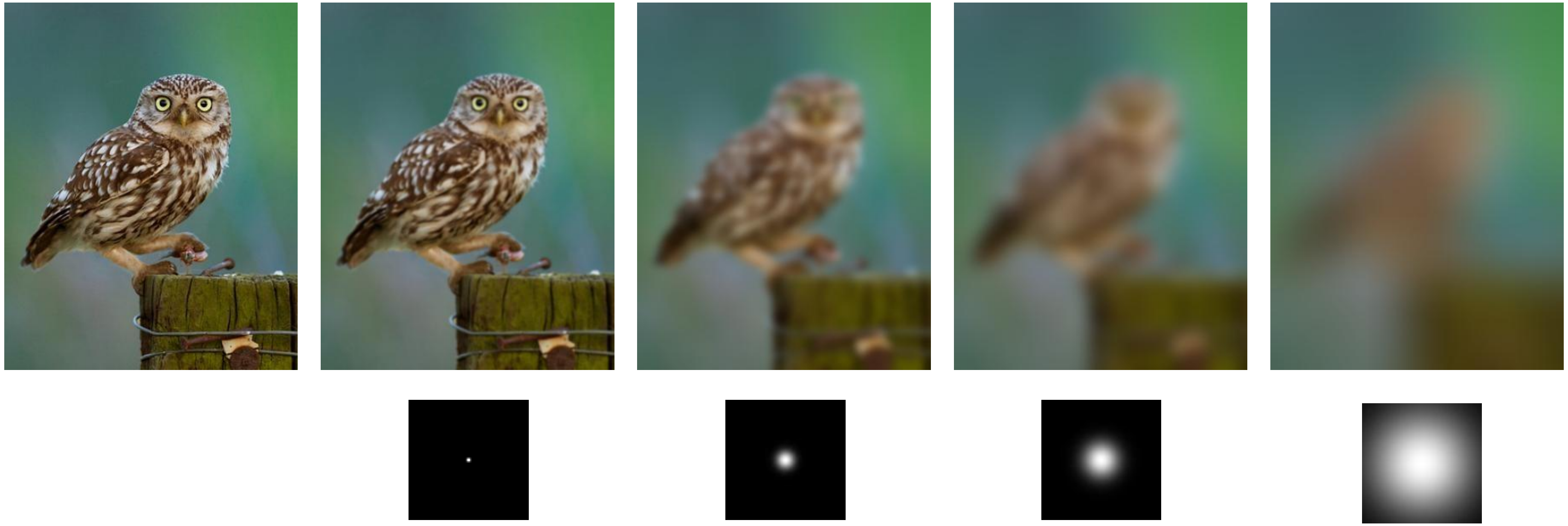




$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Source: C. Rasmussen

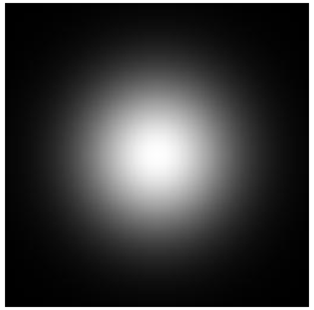
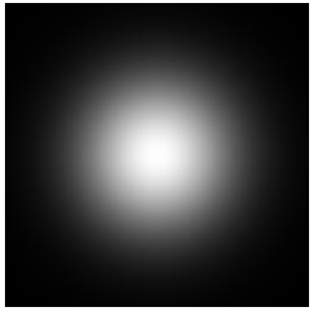
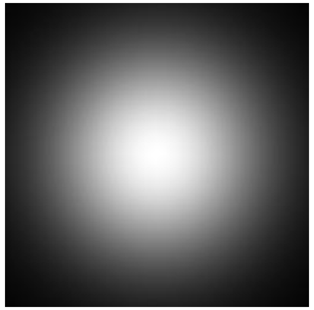
Gaussian filters



$\sigma = 1$ pixel $\sigma = 5$ pixels $\sigma = 10$ pixels $\sigma = 30$ pixels

Gaussian filter

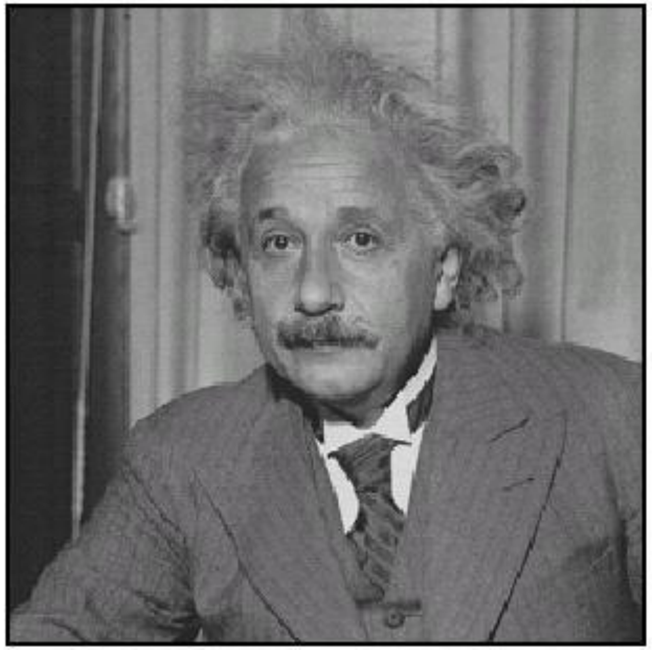
- Removes “high-frequency” components from the image (low-pass filter)
- Convolution with self is another Gaussian

 $*$  $=$ 

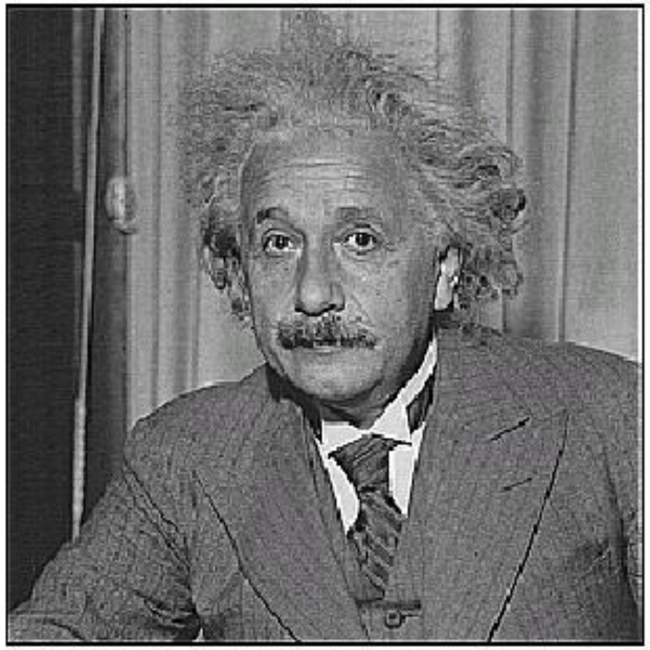
  - Convolving two times with Gaussian kernel of width σ = convolving once with kernel of width $\sigma\sqrt{2}$

Source: K. Grauman

Sharpening



before



after

Source: D. Lowe

Sharpening revisited

- What does blurring take away?

–

=

Let’s add it back:

+ α

=

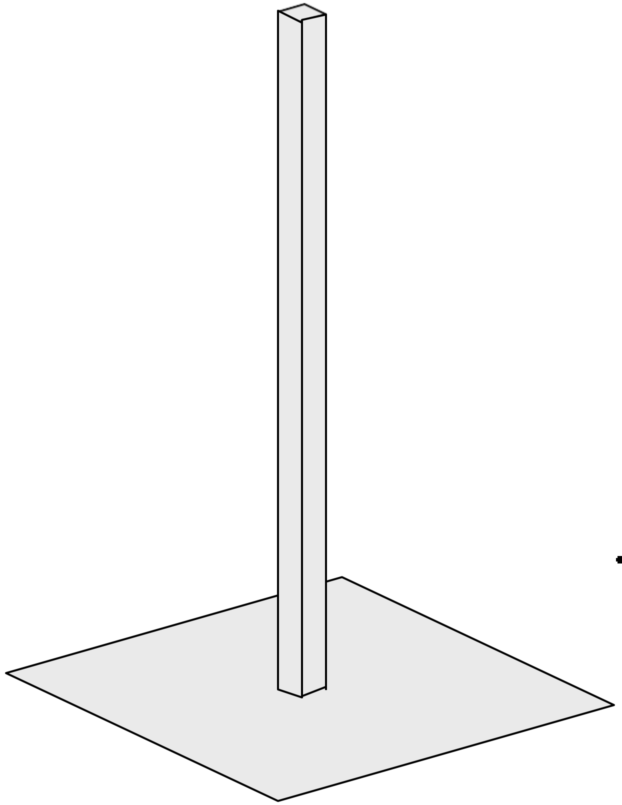
Source: S. Lazebnik

Sharpen filter

$$F + \alpha\left(F - \underbrace{F * H}_{\text{blurred image}}\right)$$

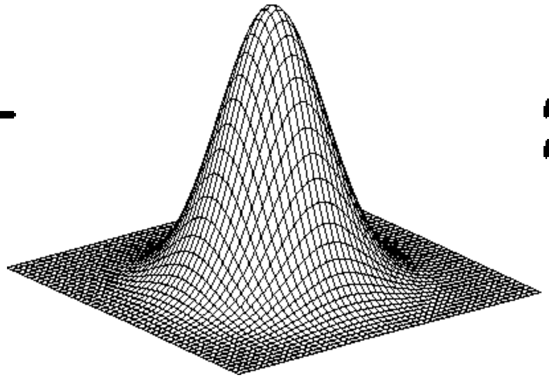
image

unit impulse
(identity)



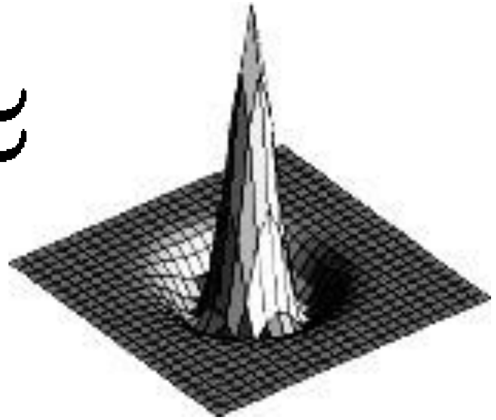
scaled impulse

$-$



Gaussian

$\approx$

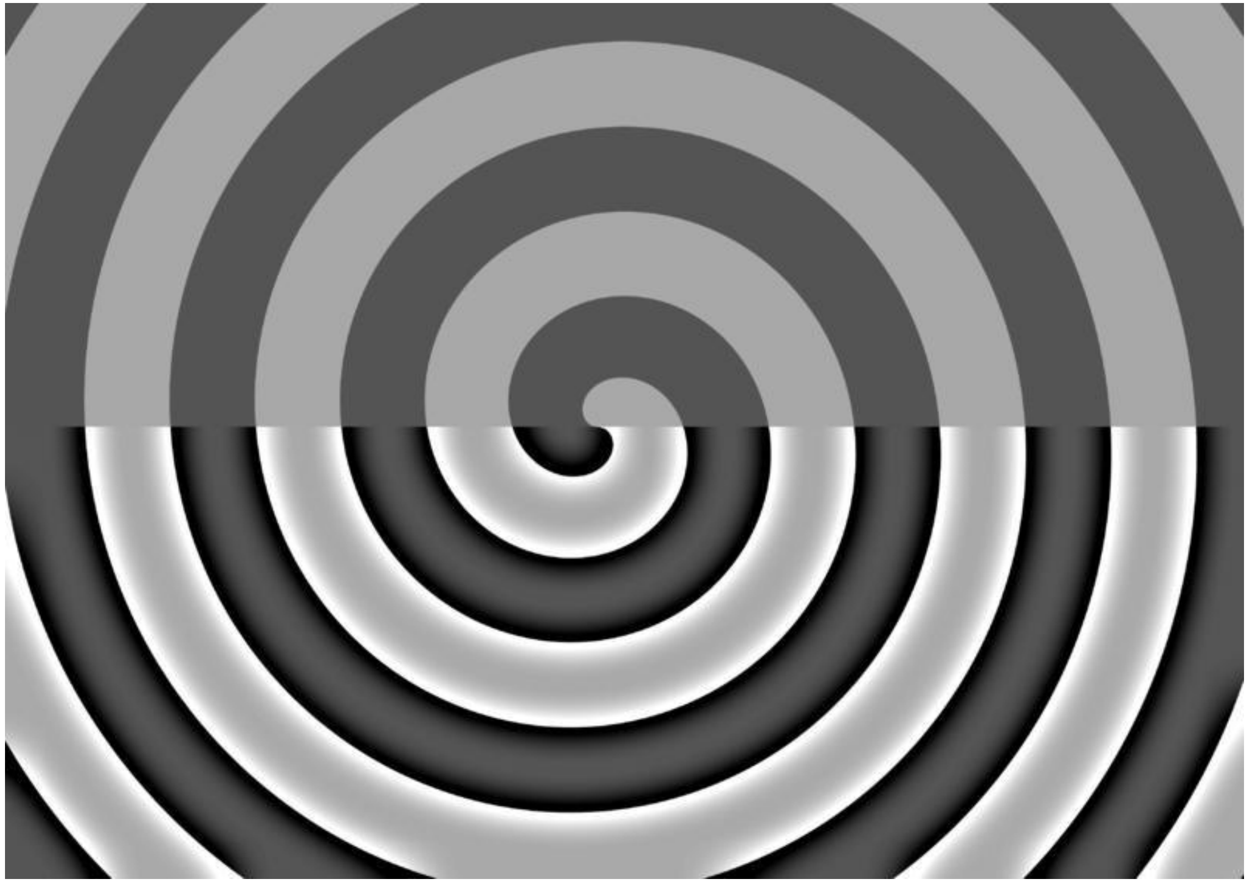


Laplacian of Gaussian

Sharpen filter

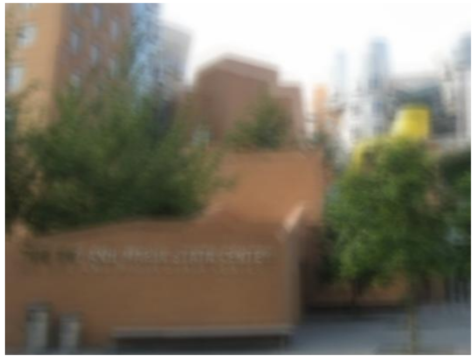
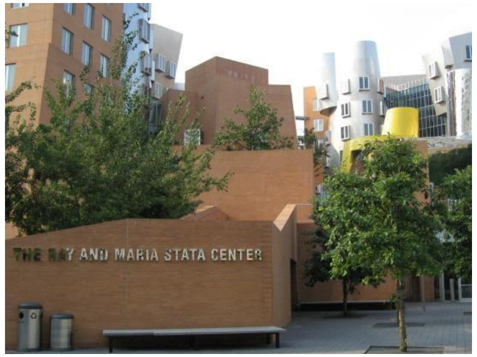
unfiltered

filtered



Convolution in the real world

Camera shake

 =  $*$ 

Source: Fergus, *et al.* "Removing Camera Shake from a Single Photograph", SIGGRAPH 2006

Bokeh: Blur in out-of-focus regions of an image.




Source: http://lullaby.homepage.dk/diy-camera/bokeh.html

Questions?