UNIVERSIDADE FEDERAL DO PARANÁ – UFPR

Departamento de Informática

Computer Vision and Perception

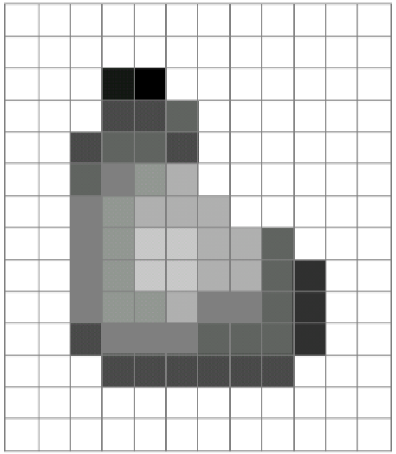
Image filtering

Prof. Eduardo Todt
2023

Sumary

- Text book
- Introduction
  - recognition problem
  - image basics

Remember image structure



255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	20	0	255	255	255	255	255	255	255
255	255	255	75	75	75	255	255	255	255	255	255
255	255	75	95	95	75	255	255	255	255	255	255
255	255	96	127	145	175	255	255	255	255	255	255
255	255	127	145	175	175	175	255	255	255	255	255
255	255	127	145	200	200	175	175	95	255	255	255
255	255	127	145	200	200	175	175	95	47	255	255
255	255	127	145	145	175	127	127	95	47	255	255
255	255	74	127	127	127	95	95	95	47	255	255
255	255	255	74	74	74	74	74	74	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Filters

Filtering

Form a new image whose pixels are a combination of the original pixels

Why?

To get useful information from images

E.g., extract edges or contours (to understand shape)

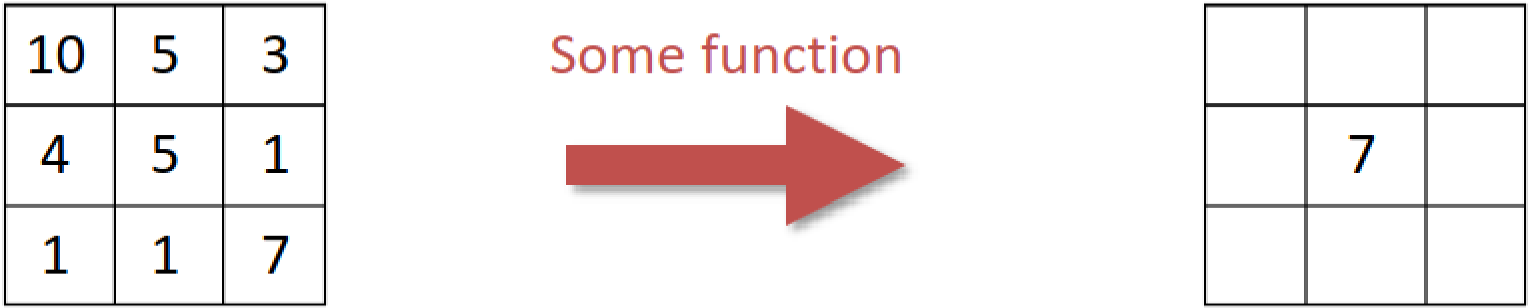
To enhance the image

E.g., to remove noise

E.g., to sharpen to enhance image

Image filtering

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



Local image data

Modified image data

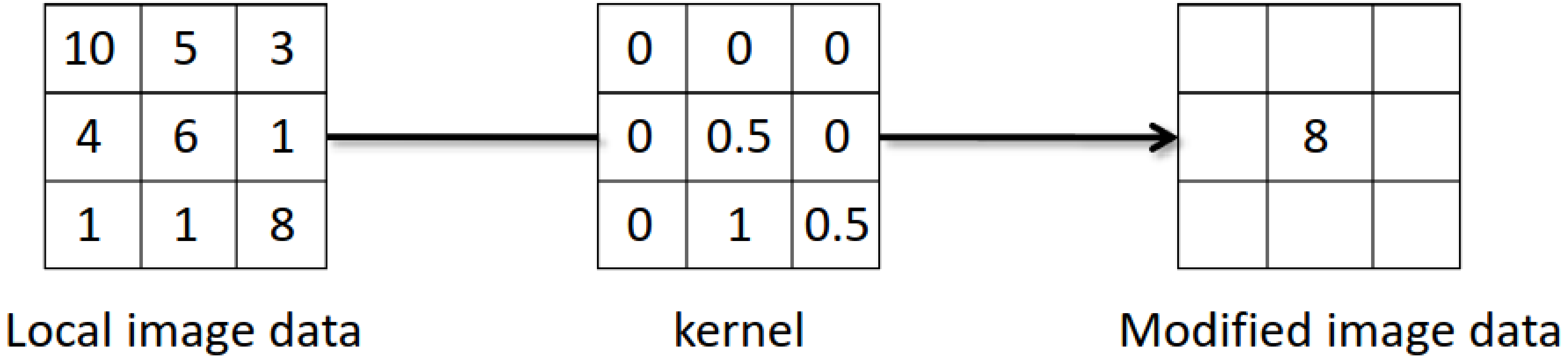
https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Lienar Filtering

One simple version of filtering: linear filtering (cross-correlation, convolution)

Replace each pixel by a linear combination (a weighted sum) of its neighbors

The prescription for the linear combination is called “kernel” (or “mask”, “filter”)



https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Cross-correlation

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

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

$$G = H \otimes F$$

Like a “dot product” between local neighborhood and kernel for each pixel

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Convolution

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

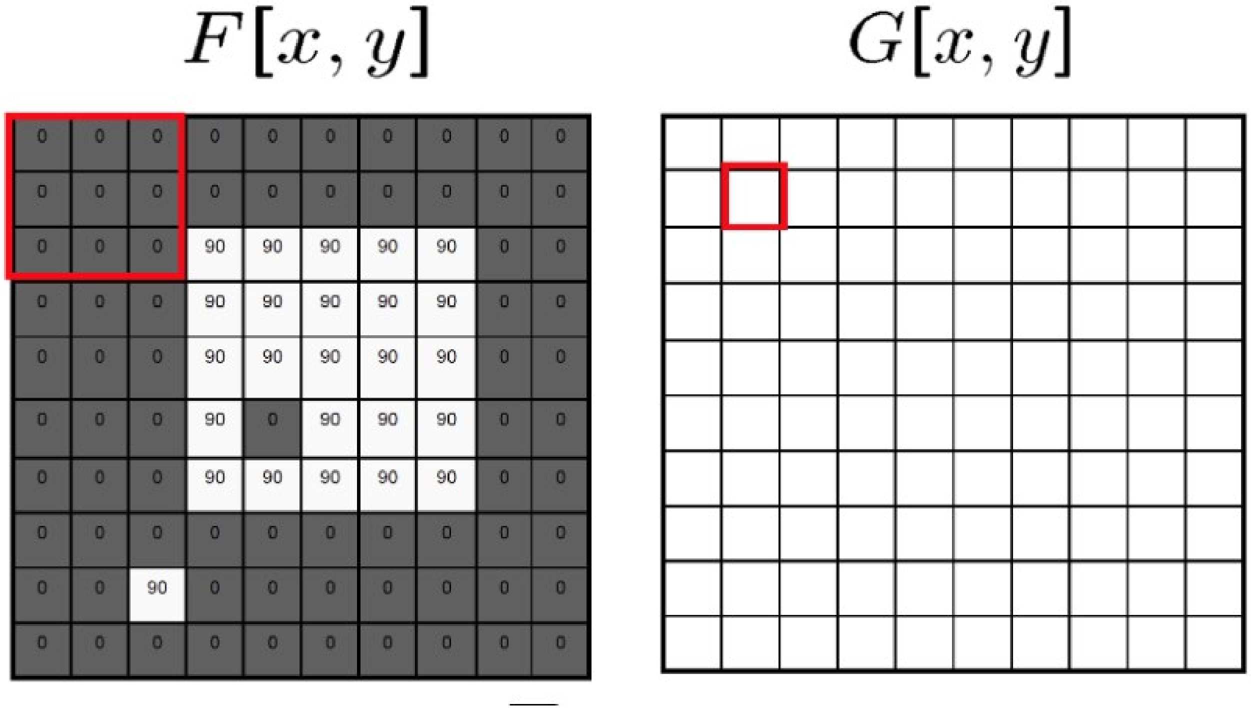
$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i-u,j-v]$$

$$G = H * F$$

Convolution is commutative and associative

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Mean filtering



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

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Mean filtering

$F[x, y]$

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

$G[x, y]$

	0	10							

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Mean filtering

$F[x, y]$

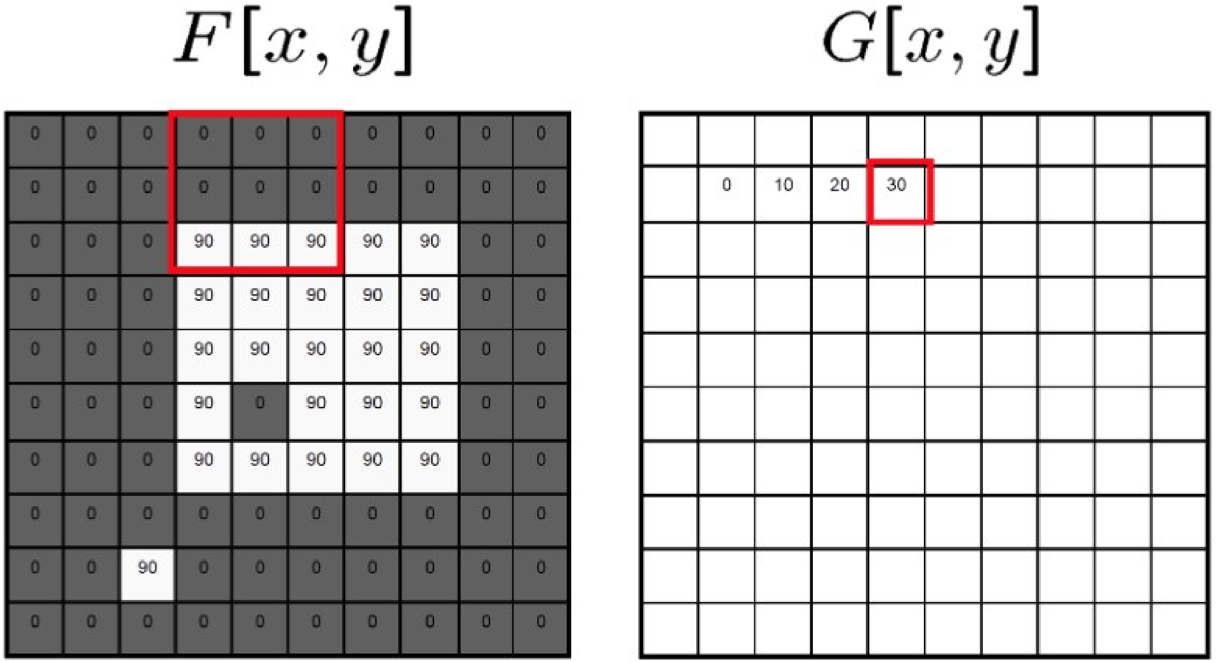
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

	0	10	20						

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Mean filtering



https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Mean filtering

$F[x,y]$

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

$G[x,y]$

	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	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

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	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

G

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Linear filters

Original

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

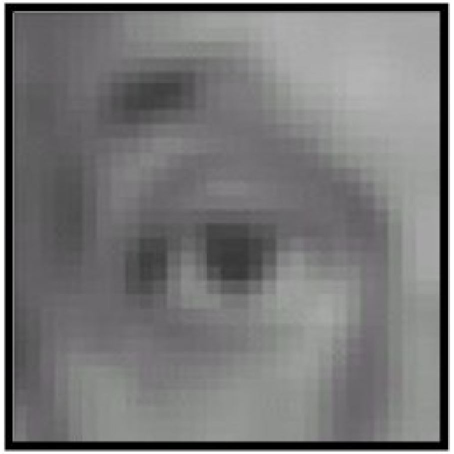
Identical image

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Linear filters

Original

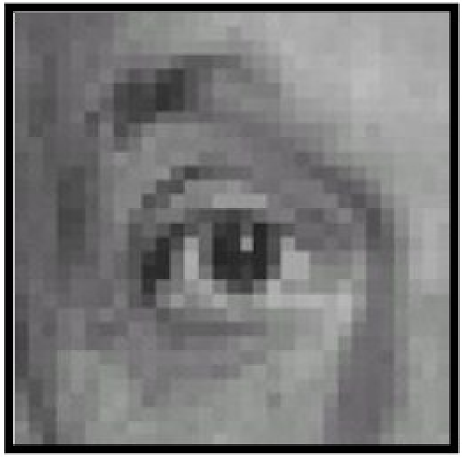
$$* \ \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} =$$



Blur (with a mean filter)

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Linear filters



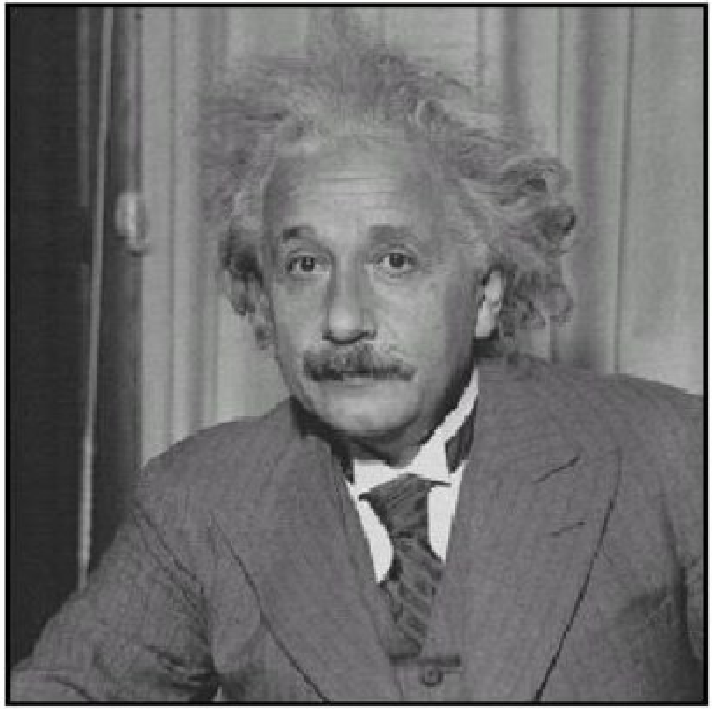
Original

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

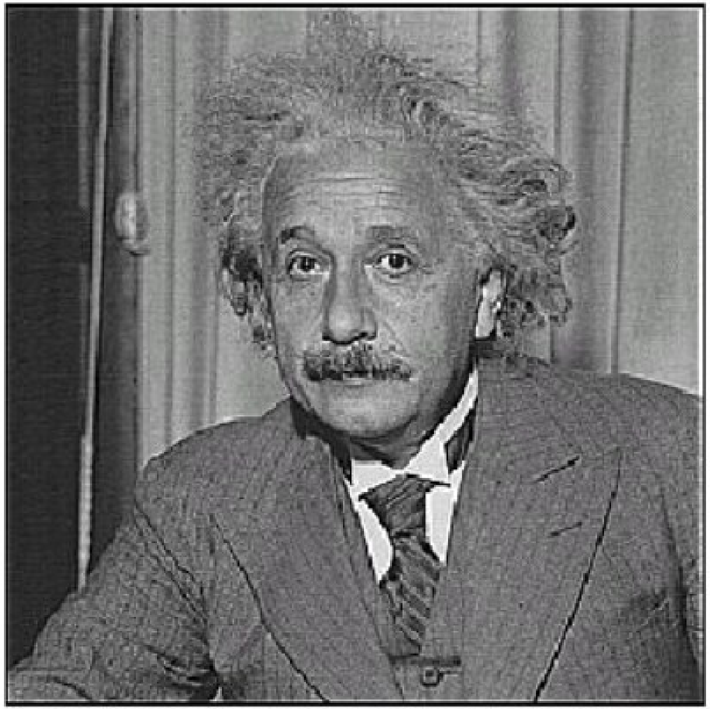
Sharpening filter
(accentuates edges)

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Sharpening



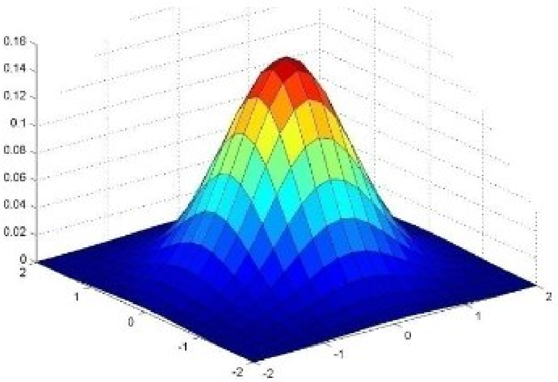
before

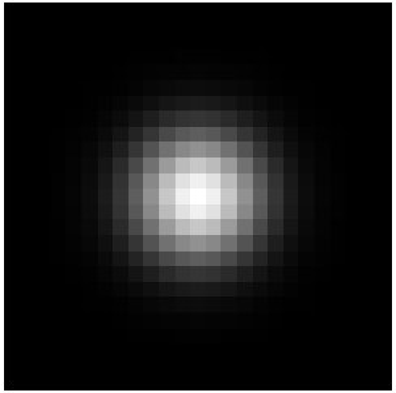


after

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Gaussian Kernel

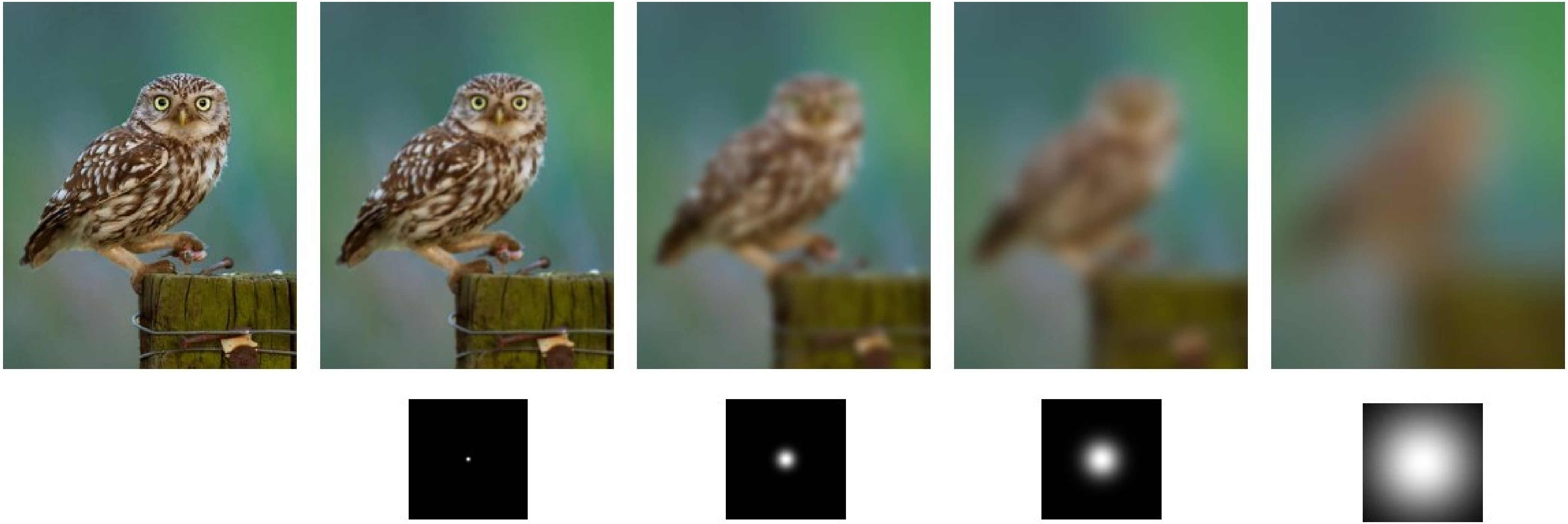




$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Gaussian Filters



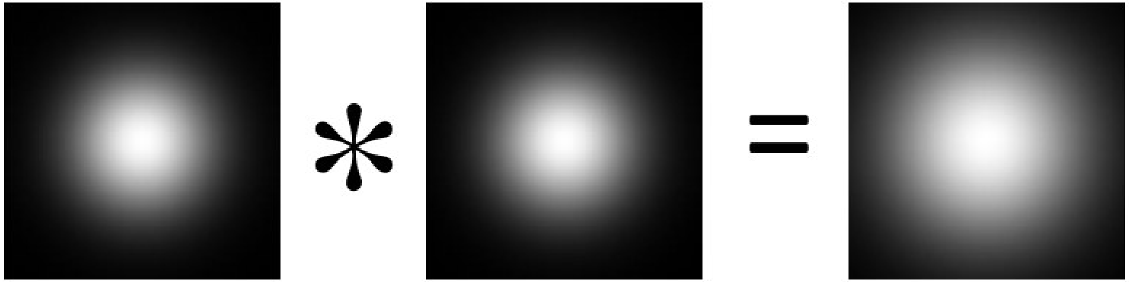
σ = 1 pixel σ = 5 pixels σ = 10 pixels σ = 30 pixels

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Gaussian Filter

Removes “high-frequency” components from the image (low-pass filter)

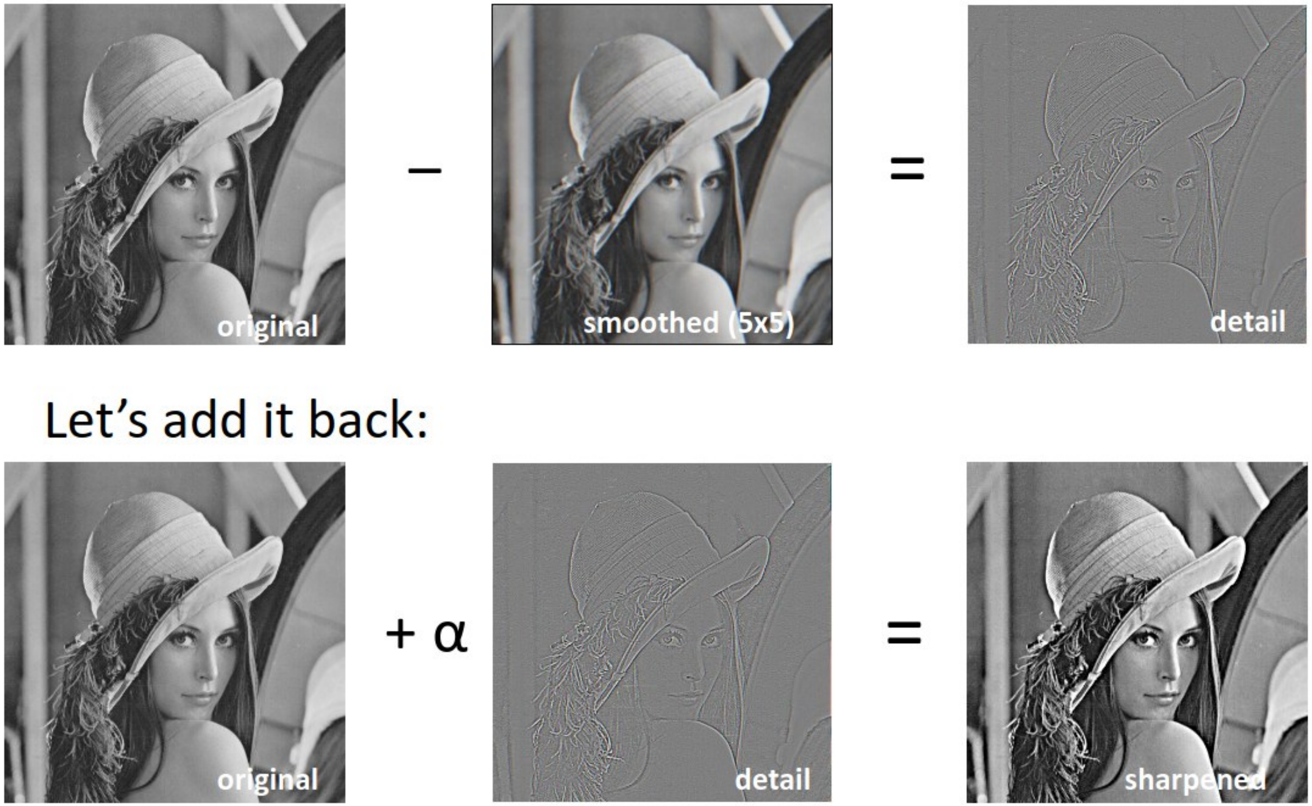
Convolution with self is another Gaussian: original width σ results width σ sqrt(2)



https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Sharpening revisited

What does blurring take away?



Source: S. Lazebnik

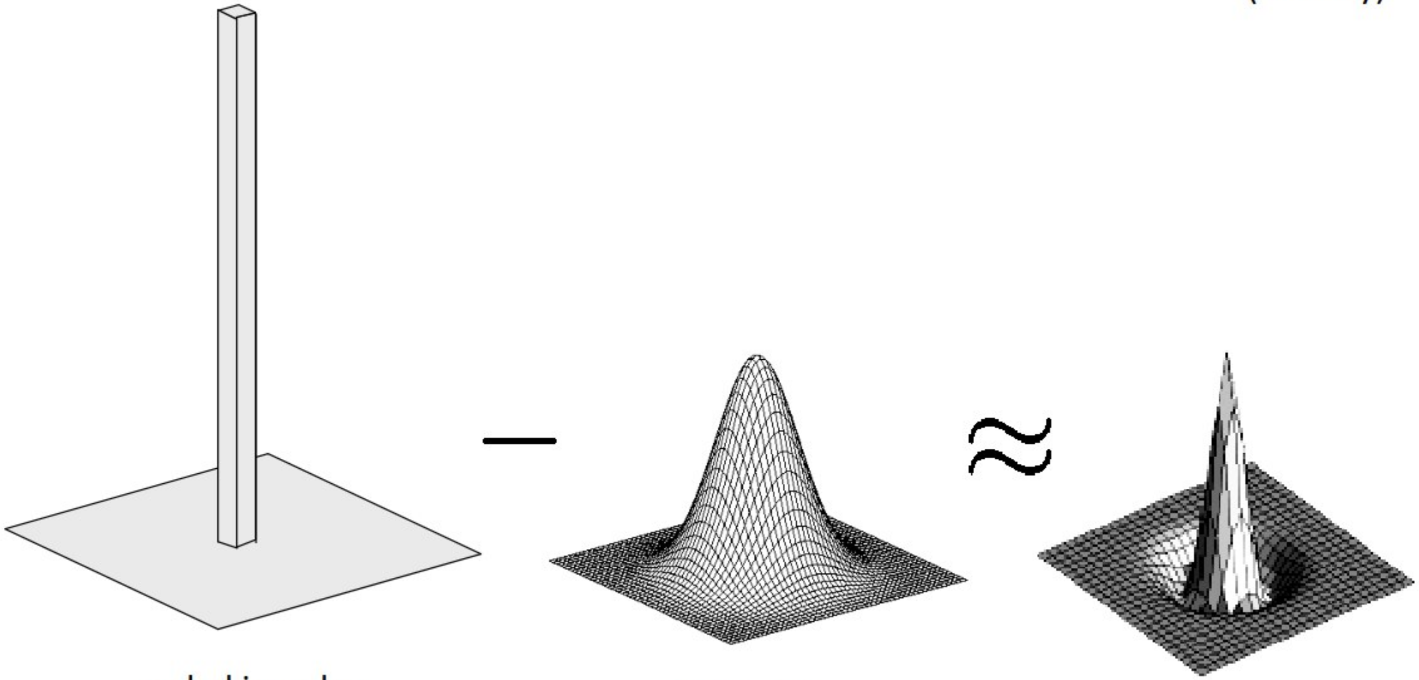
https://www.cs.cornell.edu/cou

Sharpen filter

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

image

unit impulse (identity)



scaled impulse — Gaussian ≈ Laplacian of Gaussian

Sharpen filter

https://www.

Optical convolution

Camera shake

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

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Gaussian Kernel

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

Computational Vision and Perception