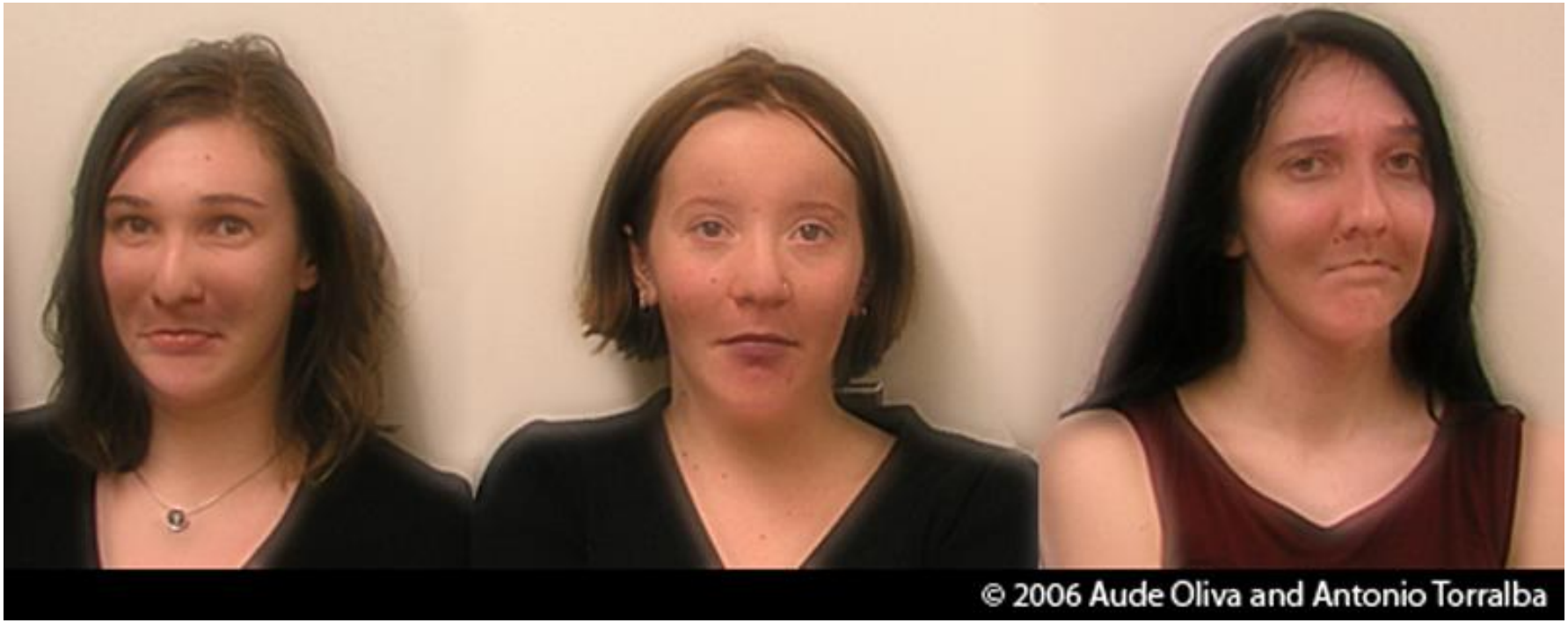
CS4670/5670: Intro to Computer Vision

Noah Snavely

Lecture 1: Images and image filtering

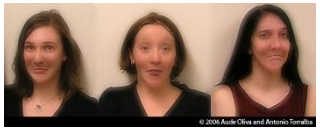


Hybrid Images, Oliva et al., http://cvcl.mit.edu/hybridimage.htm

CS4670: Computer Vision

Noah Snavely

Lecture 1: Images and image filtering

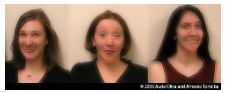


Hybrid Images, Oliva et al., http://cvcl.mit.edu/hybridimage.htm

CS4670: Computer Vision

Noah Snavely

Lecture 1: Images and image filtering

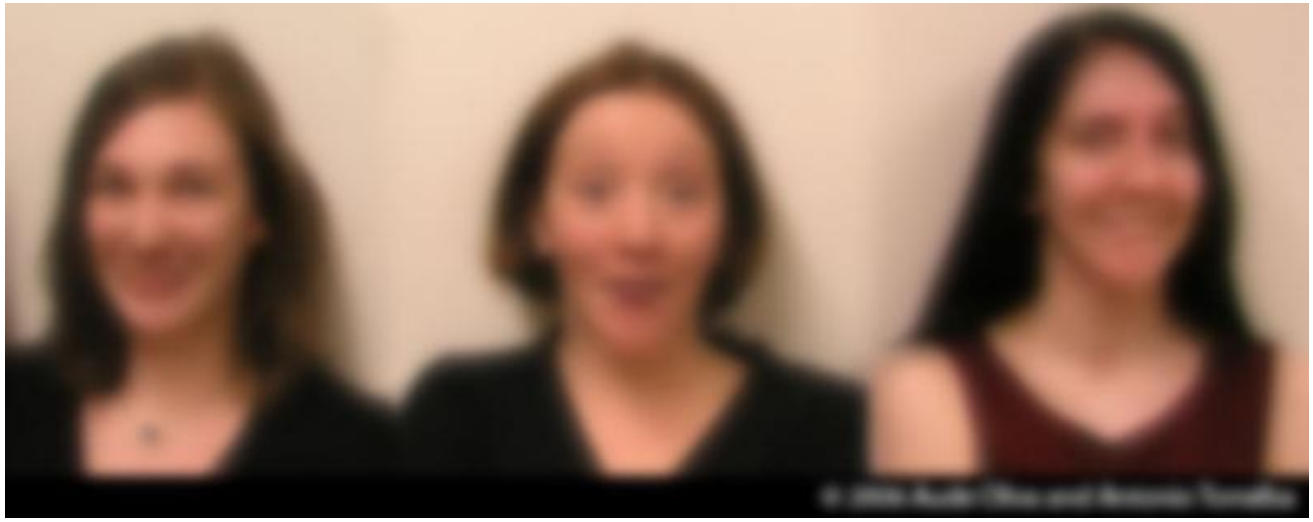


Hybrid Images, Oliva et al., http://cvcl.mit.edu/hybridimage.htm

CS4670: Computer Vision

Noah Snavely

Lecture 1: Images and image filtering



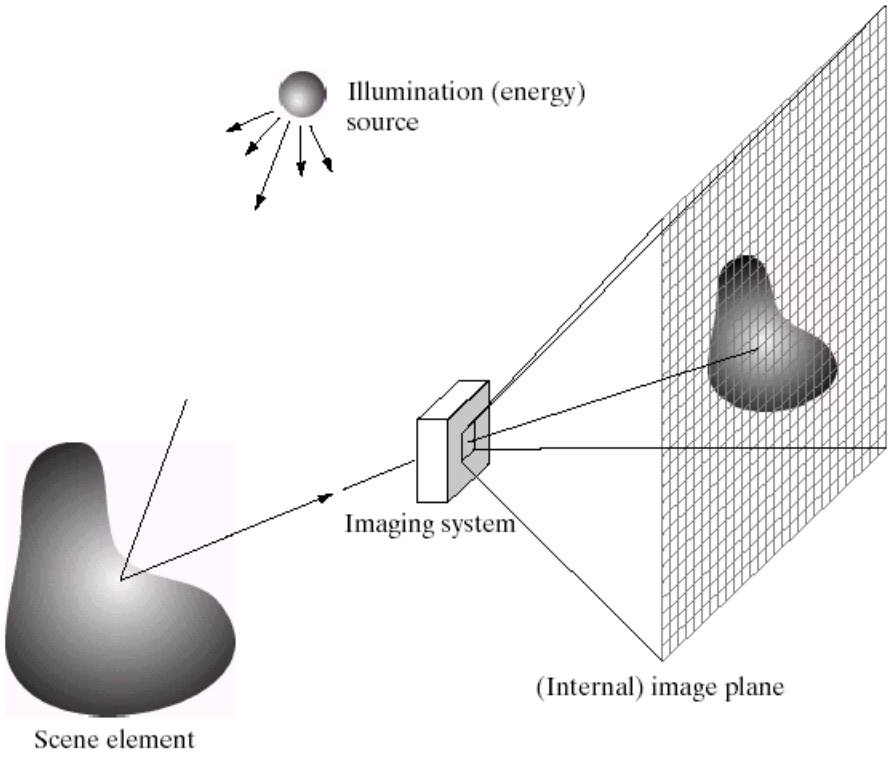
Hybrid Images, Oliva et al., http://cvcl.mit.edu/hybridimage.htm

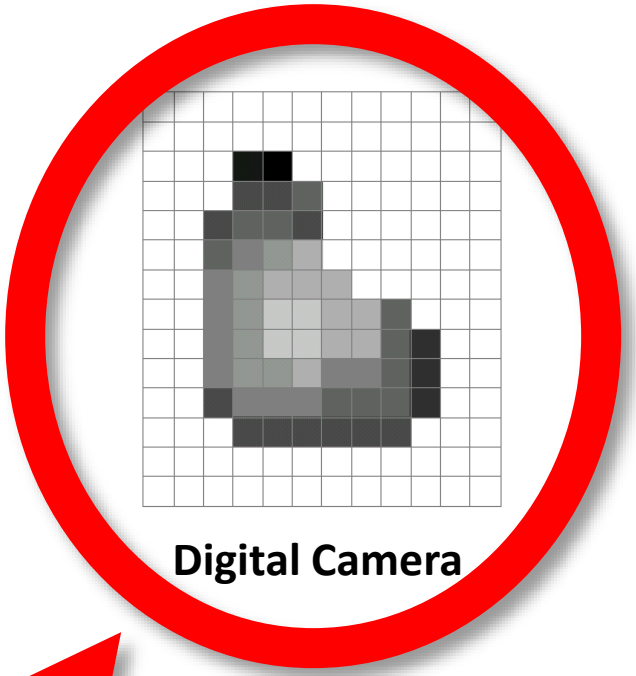
Reading

- Szeliski, Chapter 3.1-3.2

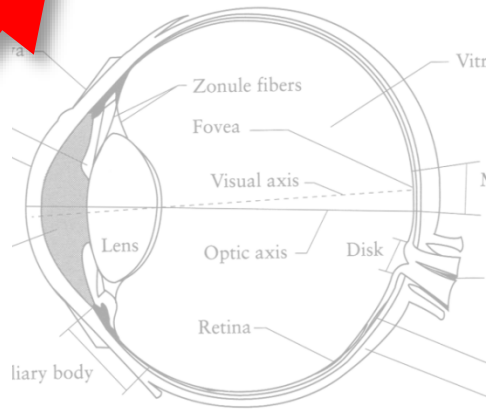
What is an image?

What is an image?





We'll focus on these in this class

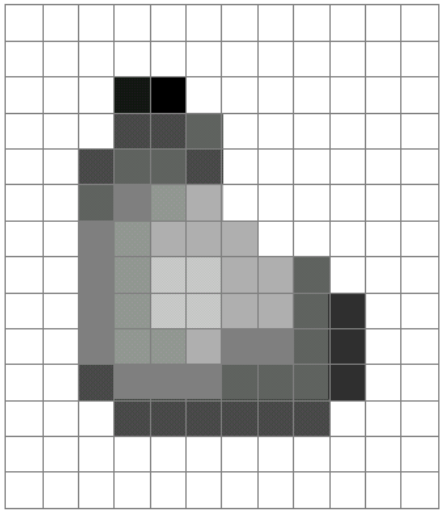


(More on this process later)

Source: A. Efros

What is an image?

- A grid (matrix) of intensity values



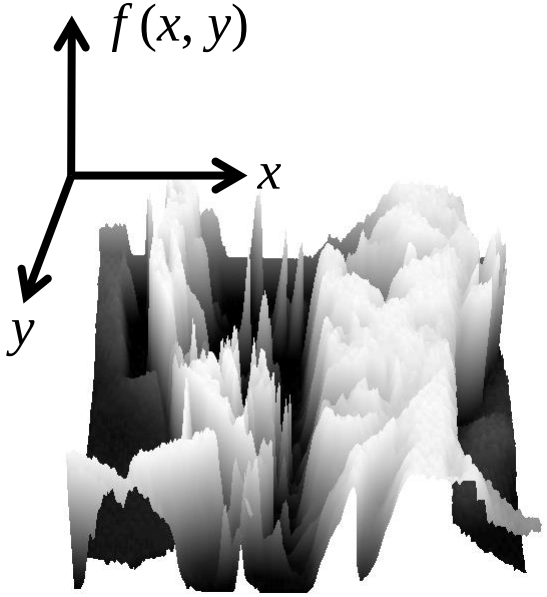
=

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

(common to use one byte per value: 0 = black, 255 = white)

What is an image?

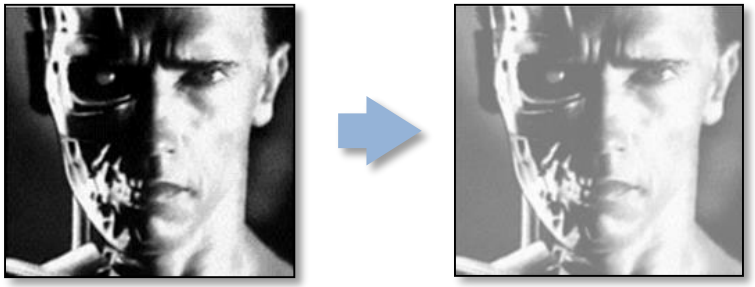
- We can think of a (grayscale) image as a **function**, f, from R^2 to R:
  - $f(x,y)$ gives the **intensity** at position (x,y)



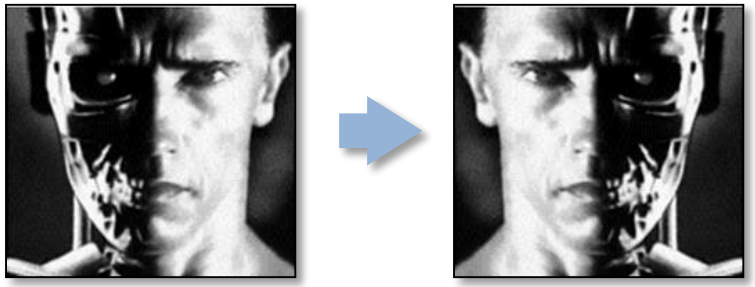
  - A **digital** image is a discrete (**sampled**, **quantized**) version of this function

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

- 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

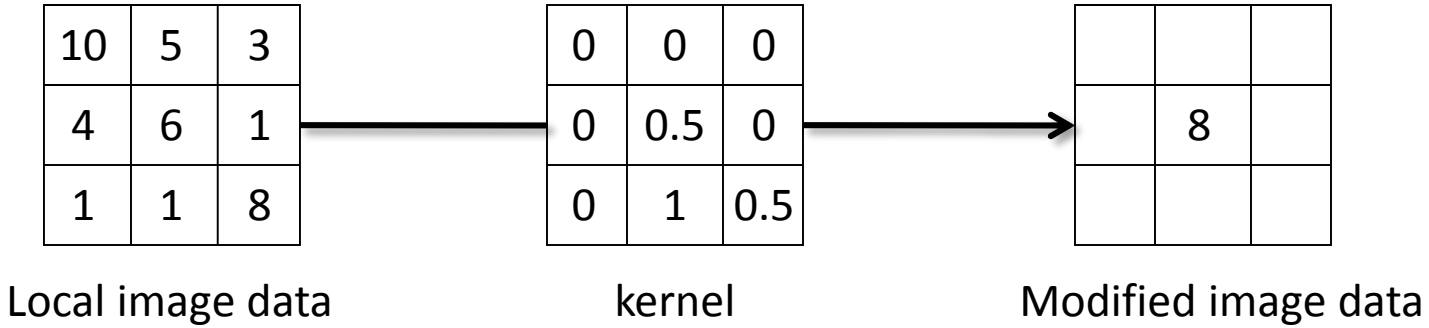
Local image data

Modified image data

Source: L. Zhang

Linear filtering

- One simple version: 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 the "kernel" (or "mask", "filter")



Source: L. Zhang

Cross-correlation

- Let F be the image, H be the kernel (of size 2*k*+1 x 2*k*+1), and G be the output image. The **cross-correlation** operation is defined as:

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

- Can think of as a "dot product" between local neighborhood and kernel for each pixel
- Short notation: $G = H \otimes F$

Convolution

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

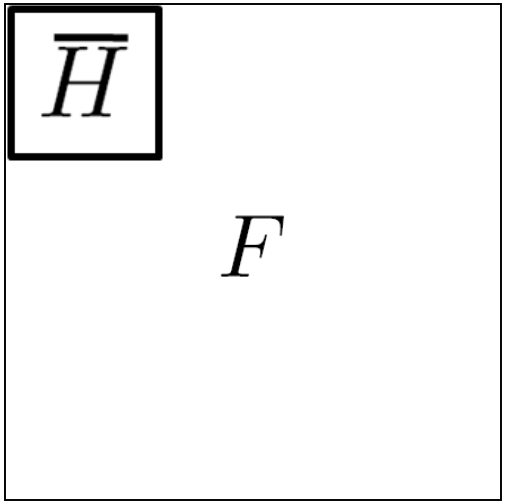
$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i-u,j-v]$$

This is called a **convolution** operation:

$$G = H * F$$

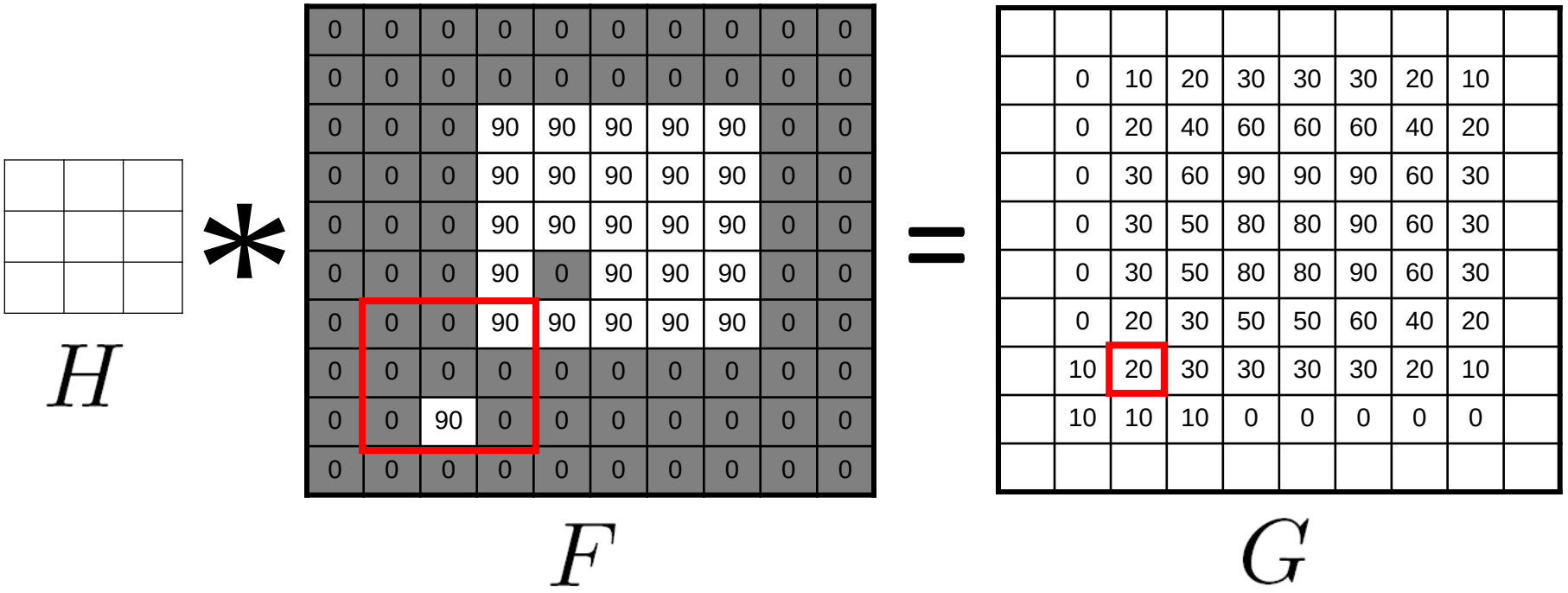
- Convolution is **commutative** and **associative**

Convolution



Adapted from F. Durand

Mean filtering



Linear filters: examples

Original

$*$

0	0	0
0	1	0
0	0	0

$=$

Identical image

Source: D. Lowe

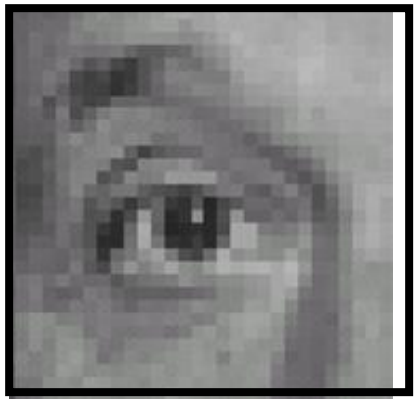
Linear filters: examples

Original

$*$

0	0	0
0	0	1
0	0	0

$=$



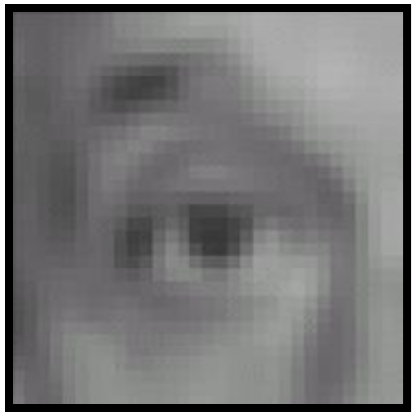
Shifted left
By 1 pixel

Source: D. Lowe

Linear filters: examples

Original

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



Blur (with a mean filter)

Source: D. Lowe

Linear filters: examples

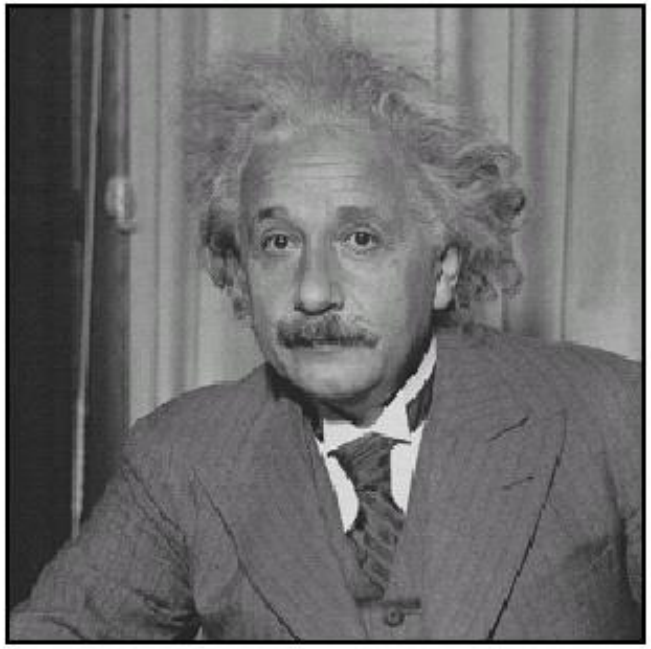
Original

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

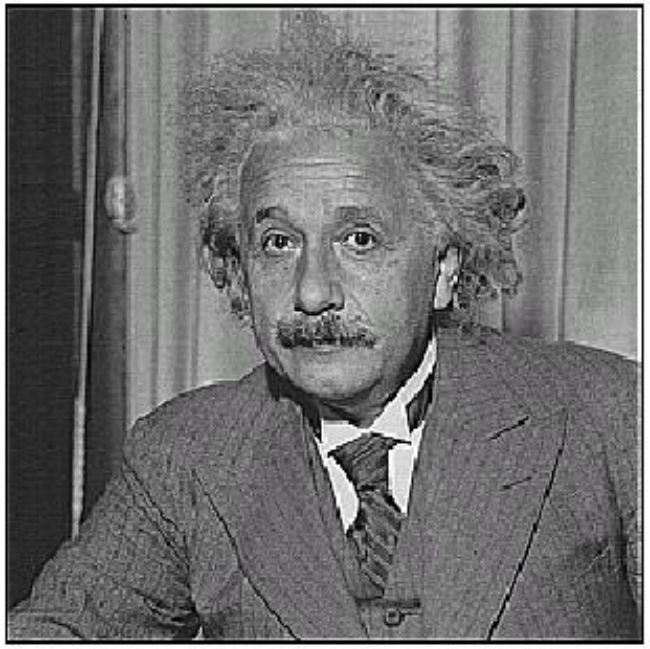
Sharpening filter
(accentuates edges)

Source: D. Lowe

Sharpening



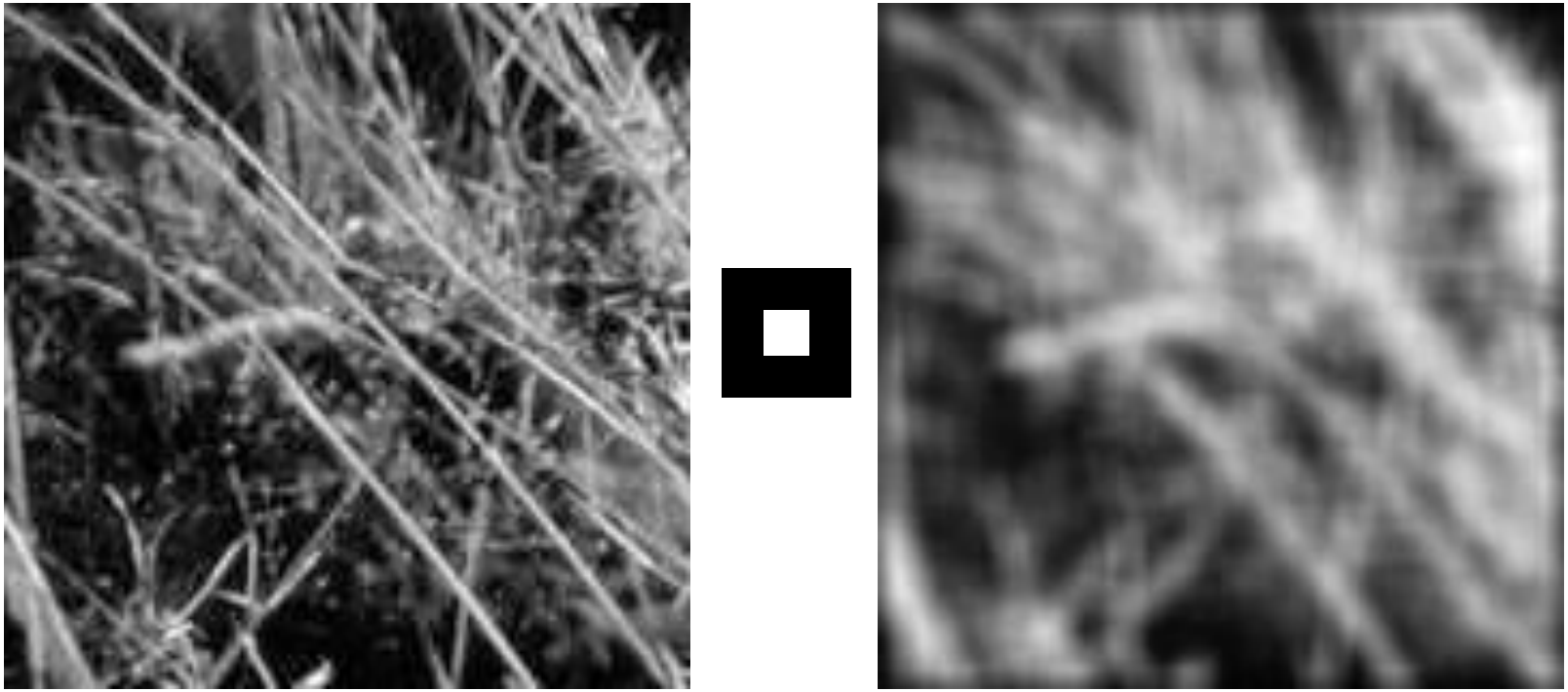
before



after

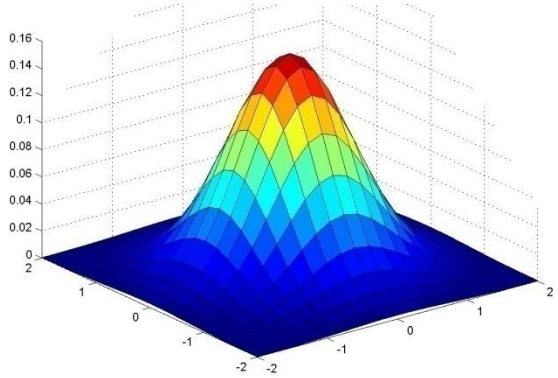
Source: D. Lowe

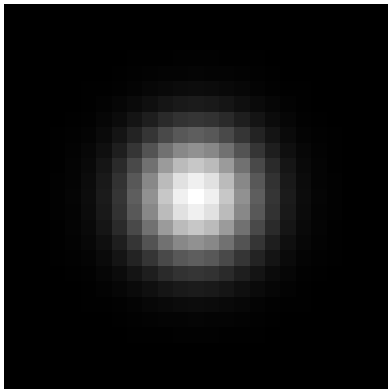
Smoothing with box filter revisited



Source: D. Forsyth

Gaussian Kernel

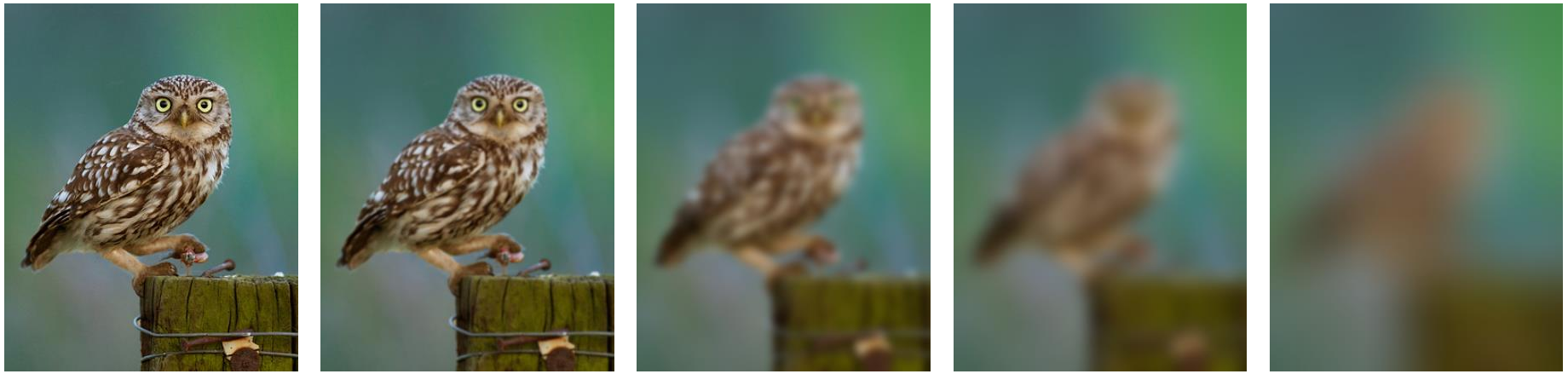




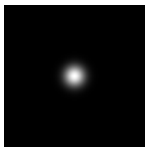
$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Source: C. Rasmussen

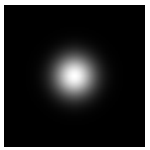
Gaussian filters



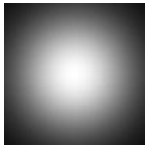
σ = 1 pixel



σ = 5 pixels

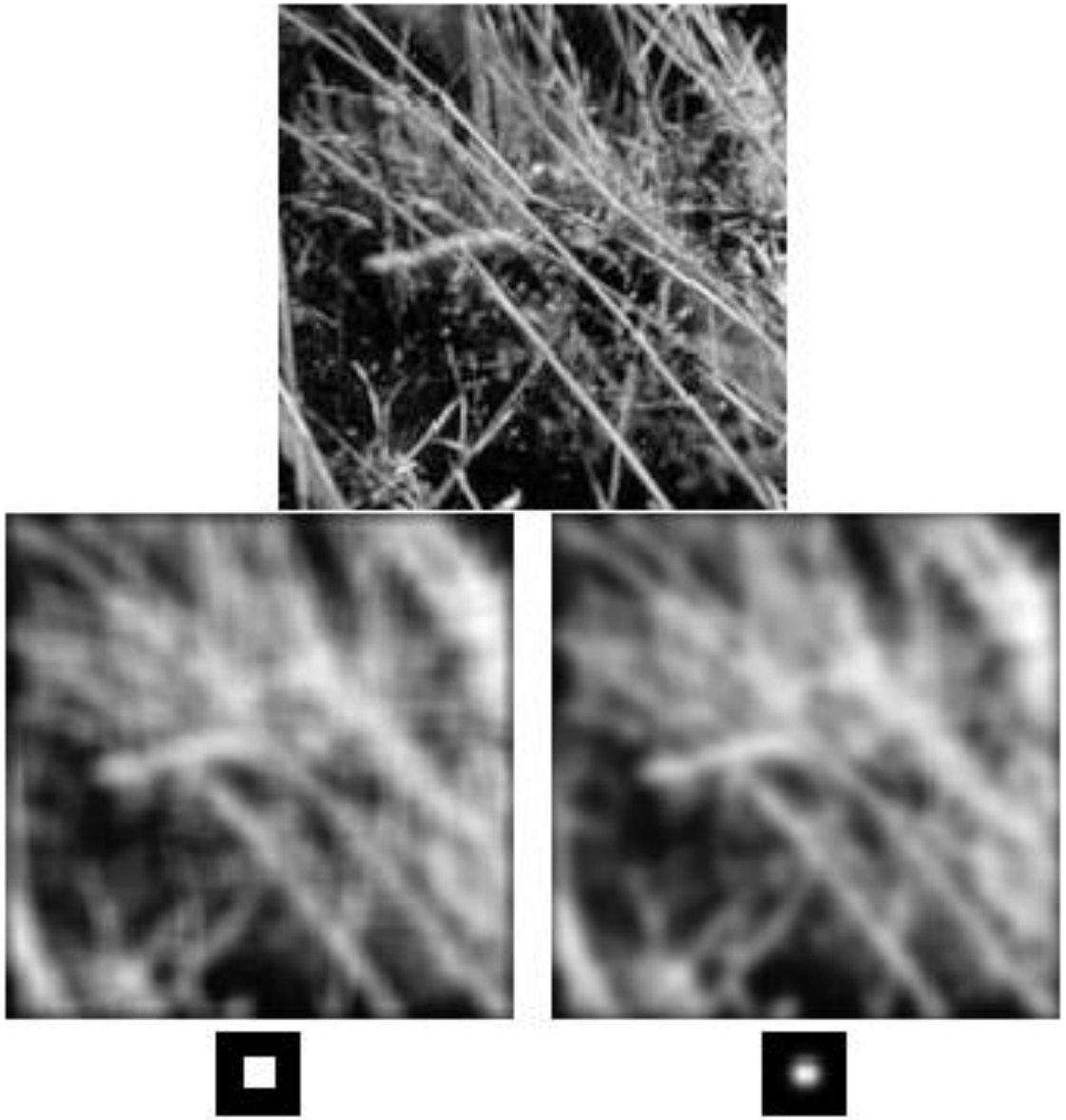


σ = 10 pixels



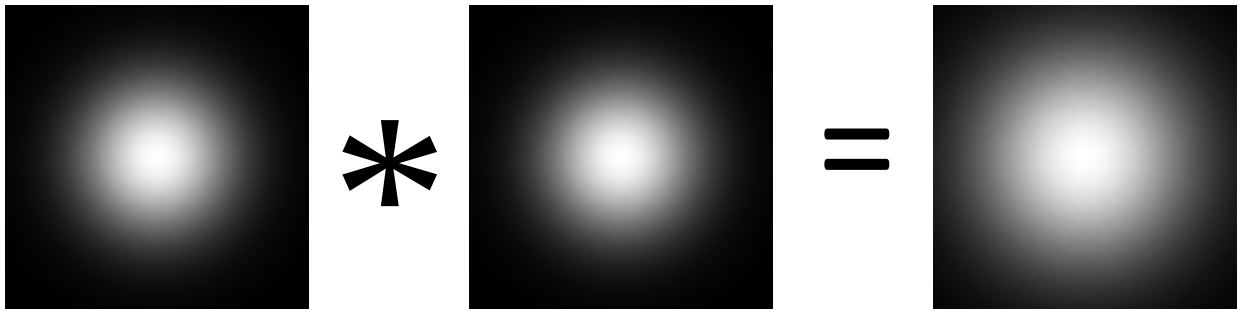
σ = 30 pixels

Mean vs. Gaussian filtering



Gaussian filter

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



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

Source: K. Grauman

Sharpening revisited

- What does blurring take away?

–

=

Let's add it back:

+ α

=

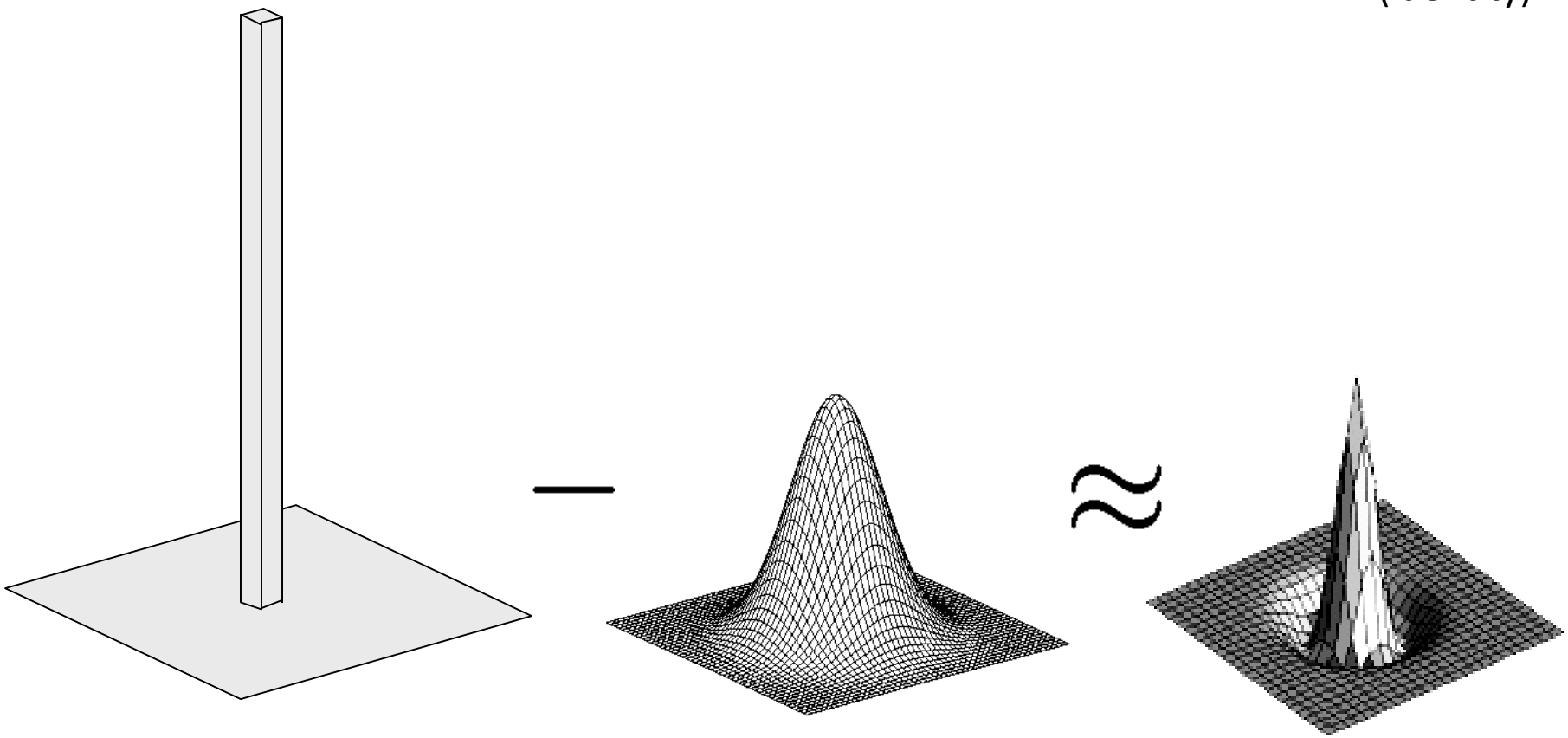
Source: S. Lazebnik

Sharpen filter

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

image

unit impulse (identity)



scaled impulse

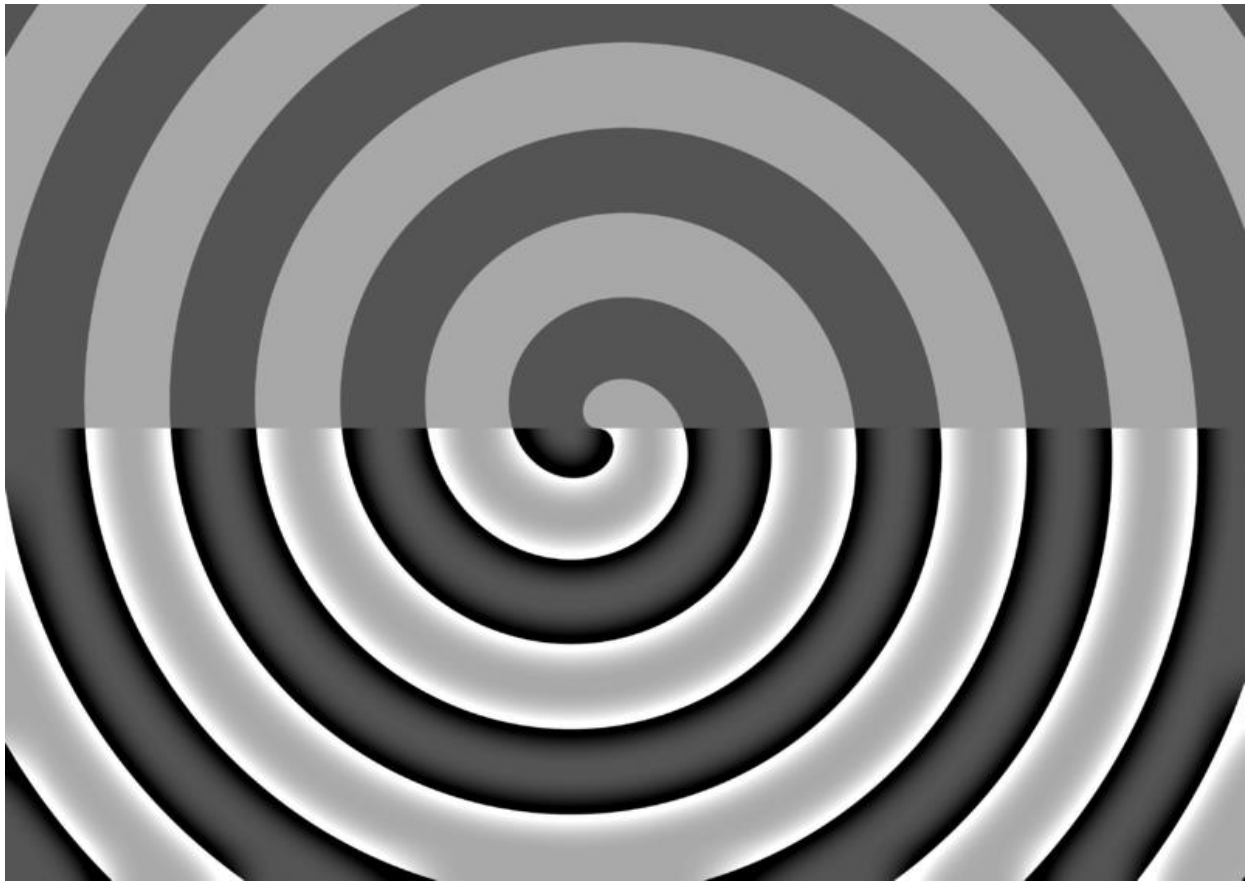
Gaussian

Laplacian of Gaussian (LoG)

Sharpen filter

unfiltered

filtered



“Optical” Convolution

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://www.diyphotography.net/diy_create_your_own_bokeh/

Questions?

- For next time:
  - Read Szeliski, Chapter 3.1-3.2