Image Filtering

Module 2

What is a Feature?

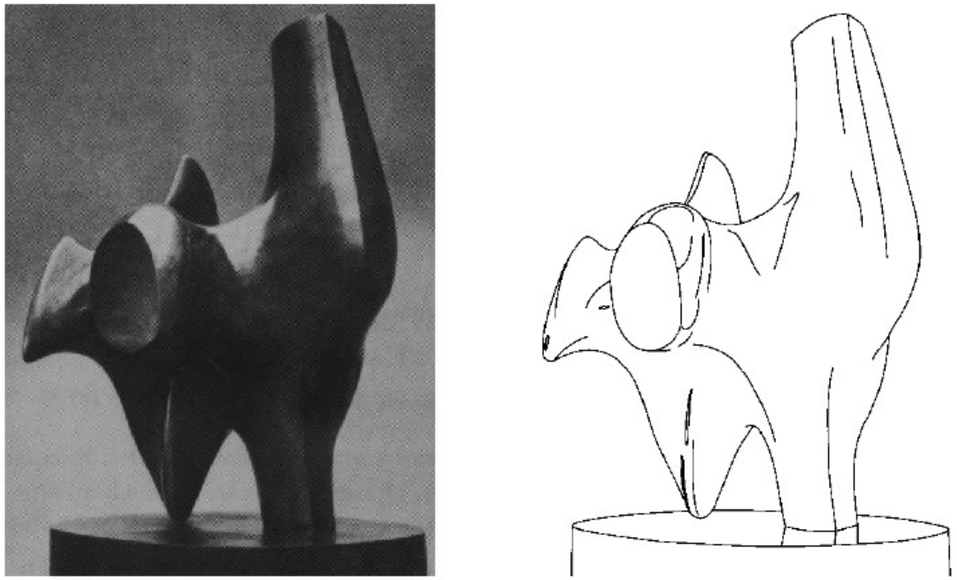
How do we understand the different parts of this image?

In computer vision, a "feature" refers to a distinctive piece of information in an image that is useful for specific computational tasks such as recognition, detection, or tracking. Features can range from simple elements like corners and edges to more complex structures like color regions, or texture patterns.


sky
building
flag
face
banner
wall
street lamp
bus
bus
cars
slide credit: Fei-Fei, Fergus & Torralba

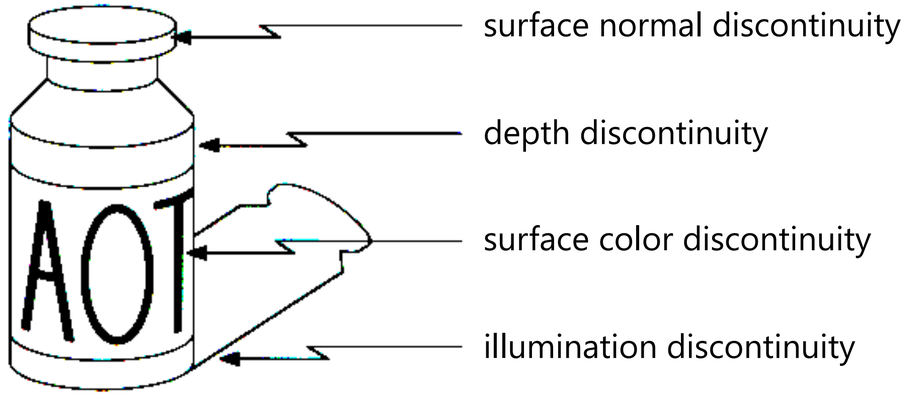
We already discussed color but what about edges?
How do we define an edge?

Edge detection



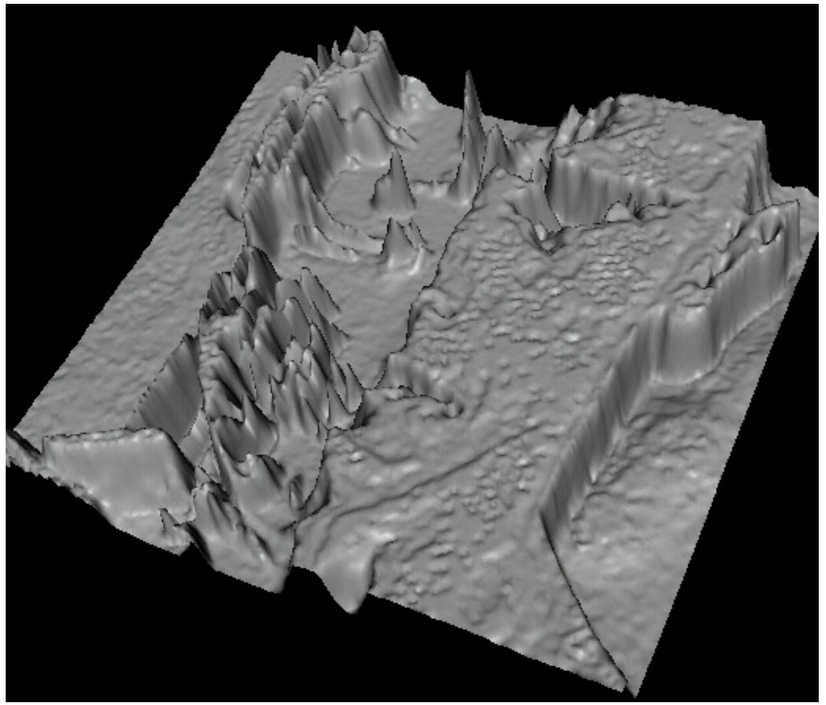
- Convert a 2D image into a set of curves
  - Extracts salient features of the scene
  - More compact than pixels

Origin of edges



- Edges are caused by a variety of factors

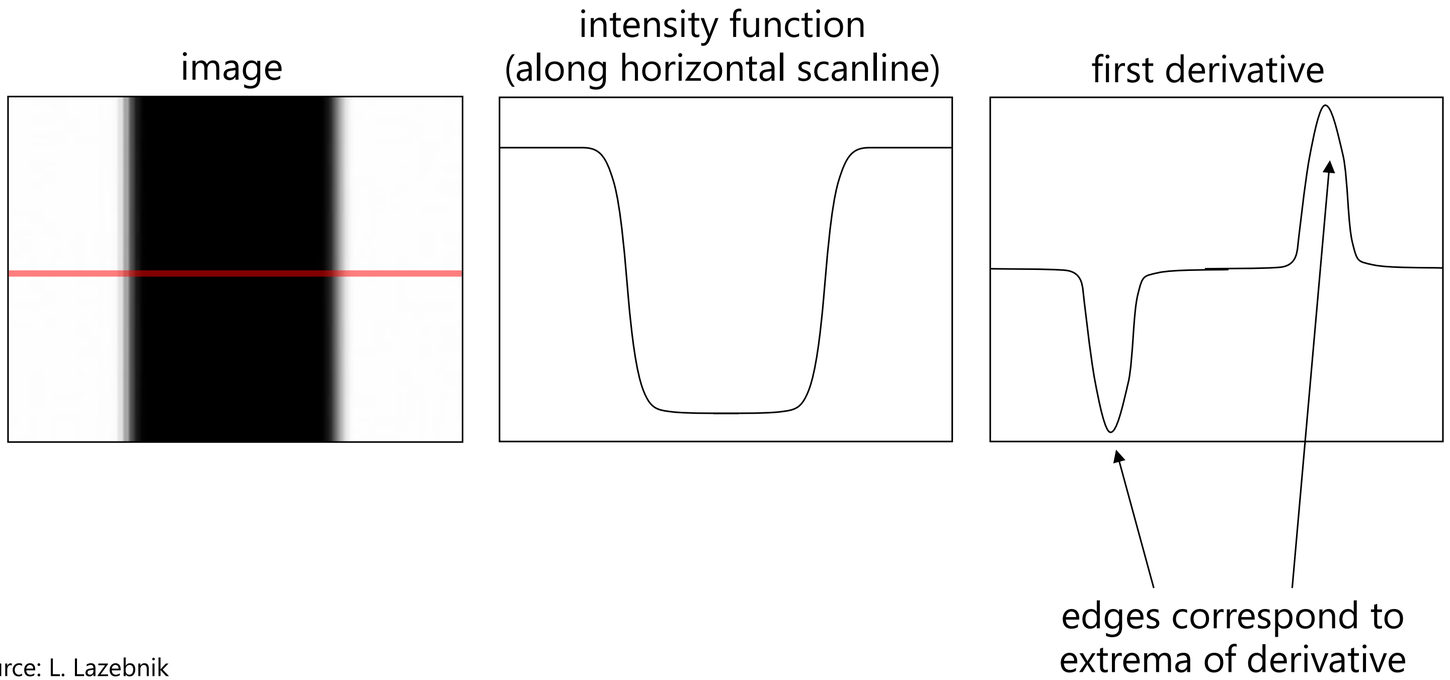
Images as functions…



- Edges look like steep cliffs

Characterizing edges

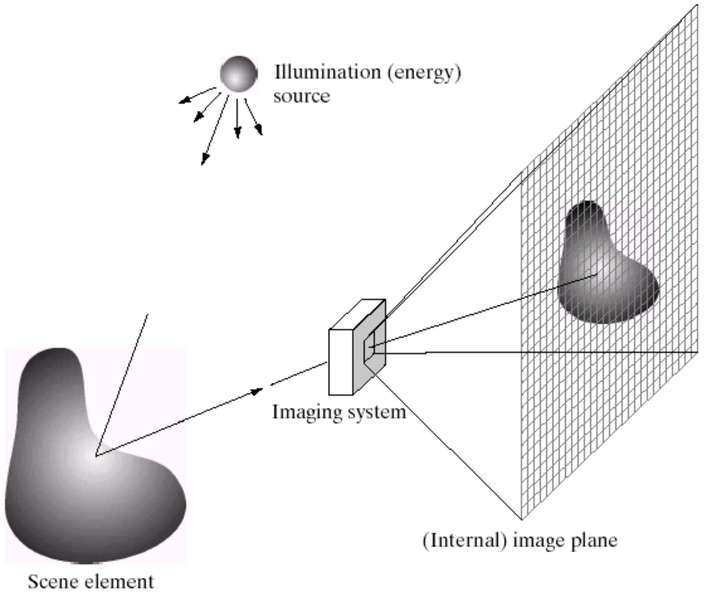
- An edge is a place of *rapid change* in the image intensity function

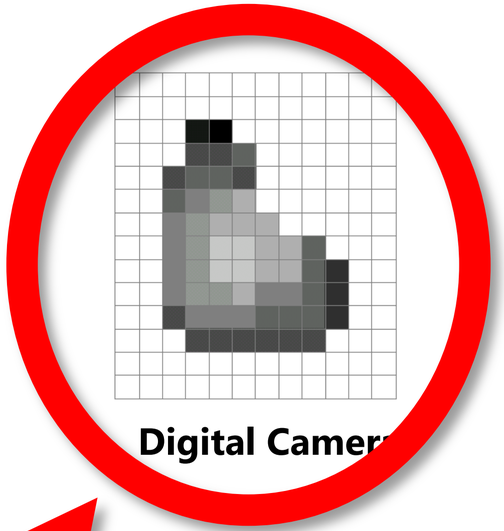


Source: L. Lazebnik

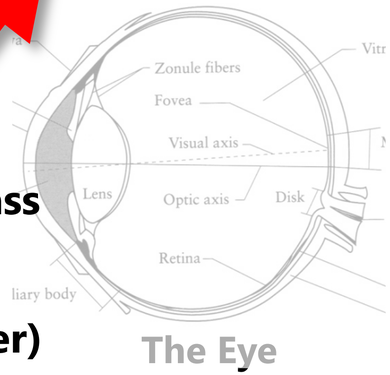
How do we discover edge regions within our image?

What is an image?





We'll focus on these in this class

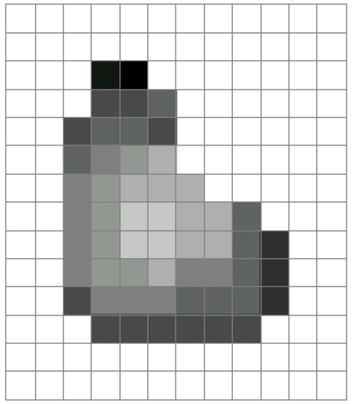


(More on this process later)

Source: A. Efros

What is an image?

- A grid (matrix) of intensity values



=

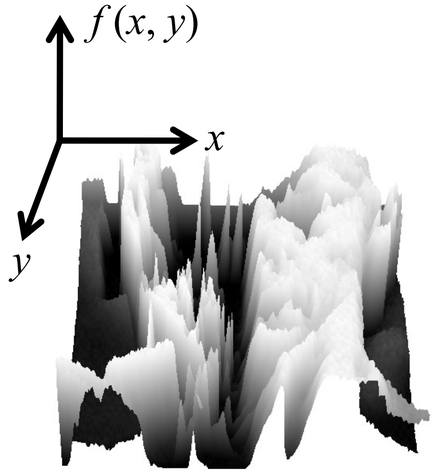
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	20	0	255	255	255	255	255	255	255
255	255	255	75	75	75	255	255	255	255	255	255
255	255	75	95	95	75	255	255	255	255	255	255
255	255	96	127	145	175	255	255	255	255	255	255
255	255	127	145	175	175	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?

- Can think of a (grayscale) image as a **function** *f* from $\mathbb{R}^2$ to $\mathbb{R}$:
  - $f(x,y)$ gives the **intensity** at position (x,y)

snoop

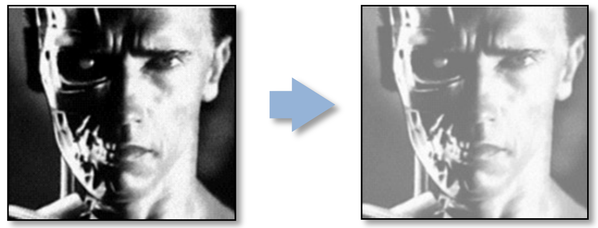


3D view

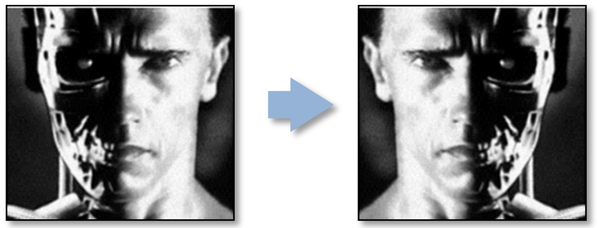
  - A **digital** image is a discrete (**sampled**, **quantized**) version of this function

Image transformations

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



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



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

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

Filters

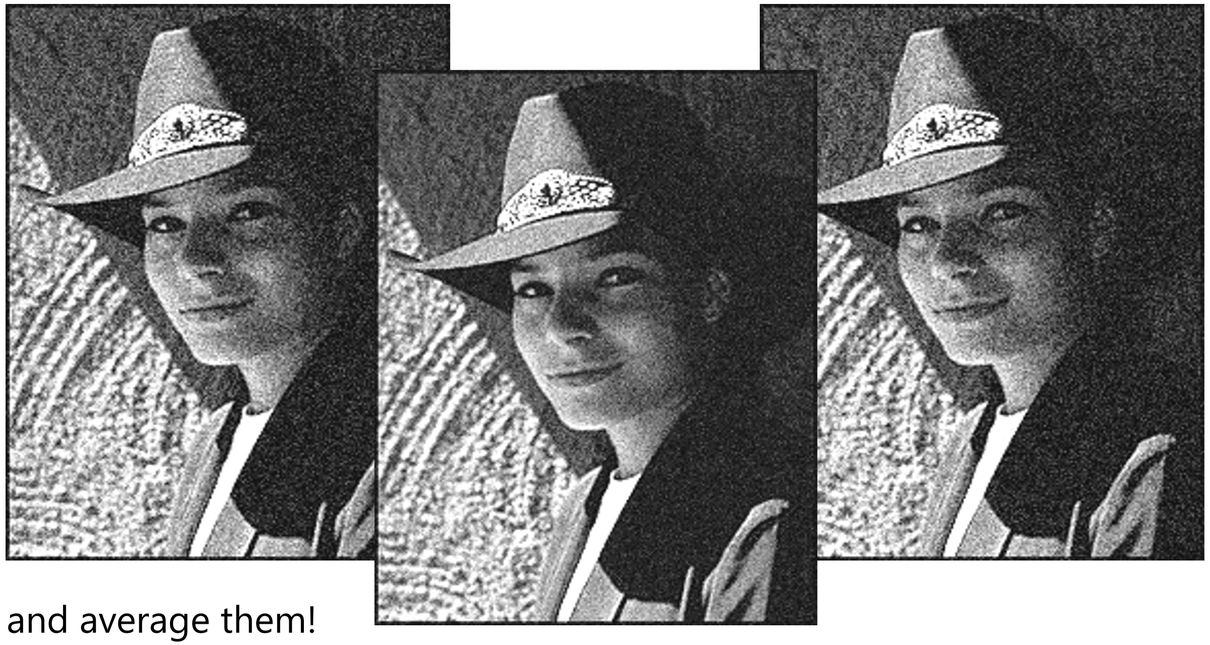
- Filtering
  - Form a new image whose pixel values are a combination of the original pixel values
- Why?
  - To get useful information from images
    - E.g., extract edges or contours (to understand shape)
  - To enhance the image
    - E.g., to remove noise
    - E.g., to sharpen and “enhance image” a la CSI
  - A key operator in Convolutional Neural Networks

Image Processing Problems

- Image Restoration
  - denoising
  - deblurring
- Image Compression
  - JPEG, HEIF, MPEG, …
- Locating Structural Features
  - corners
  - edges

Question: Noise reduction

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



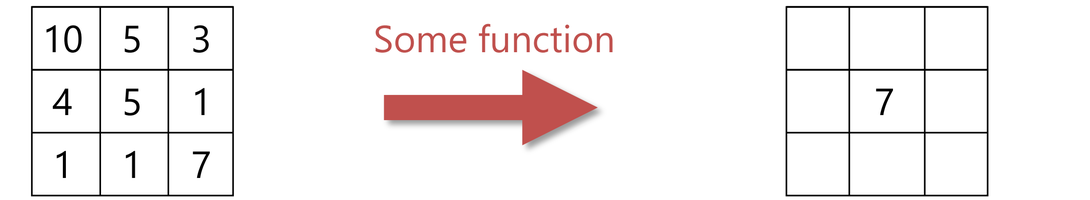
Take lots of images and average them!

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



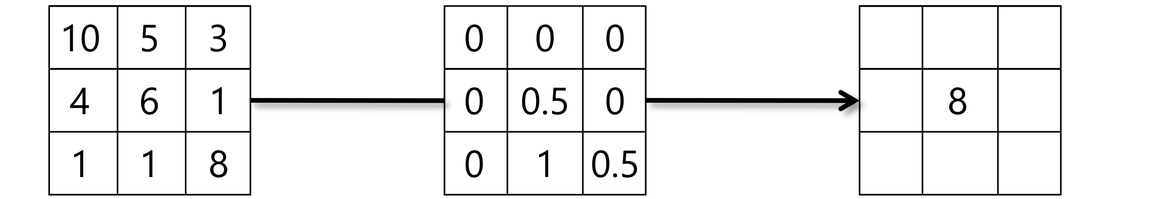
Local image data

Modified image data

Source: L. Zhang

Linear filtering

- One simple version 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 the “kernel” (or “mask”, “filter”)



Local image data kernel Modified image data

Source: L. Zhang

Convolution

- Let F be the image and H be the Kernel (2k +1 x 2k +1) and G will be the output image

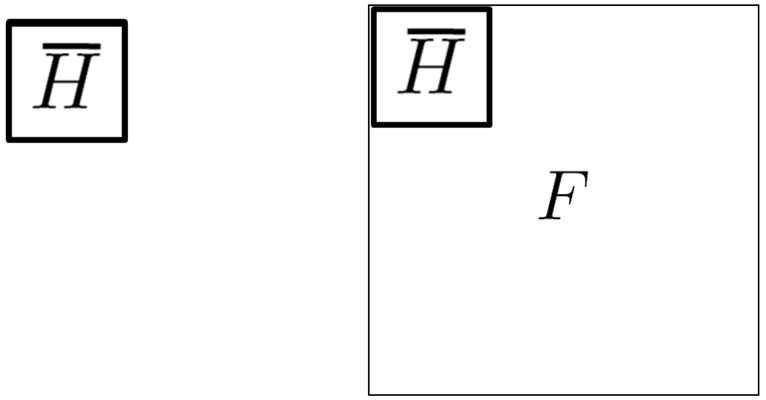
$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i-u,j-v]$$

This is called a **convolution** operation:

$$G = H * F$$

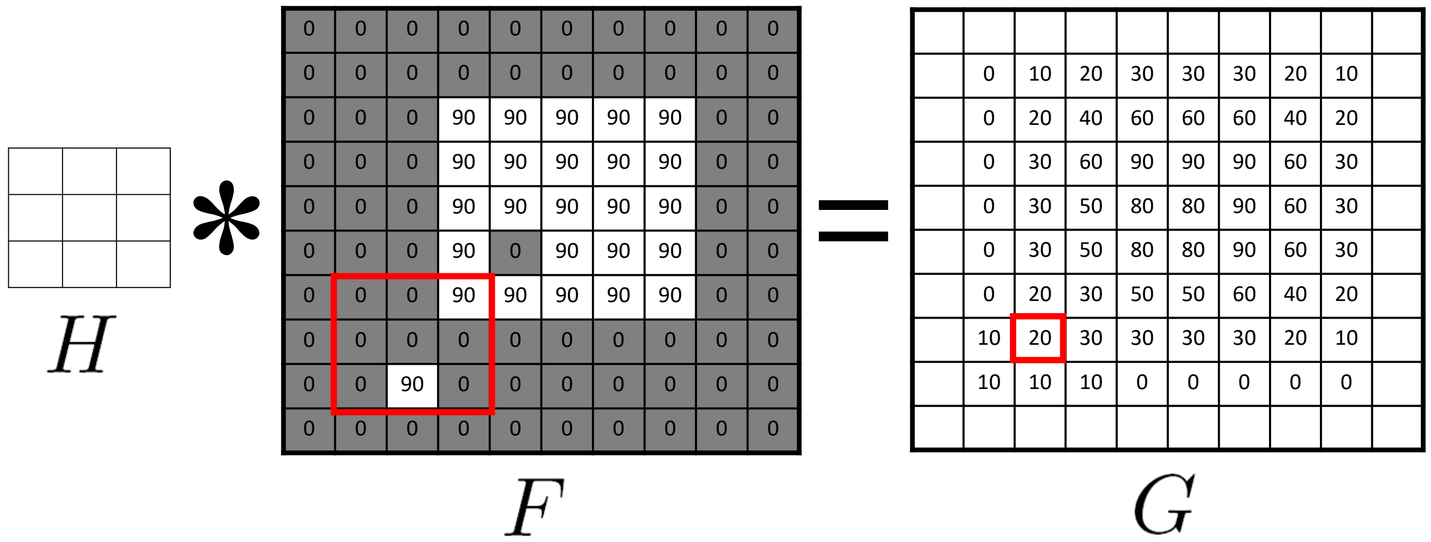
- Convolution is **commutative** and **associative**

Convolution



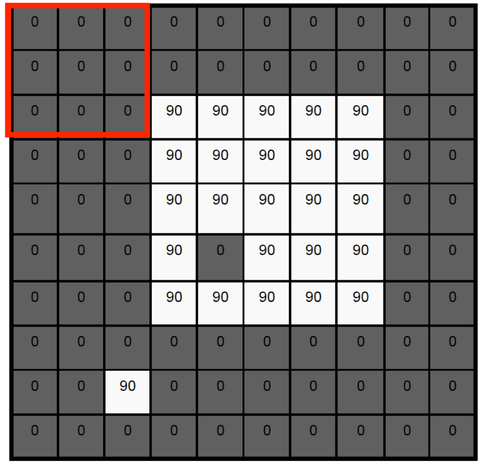
Adapted from F. Durand

Mean filtering

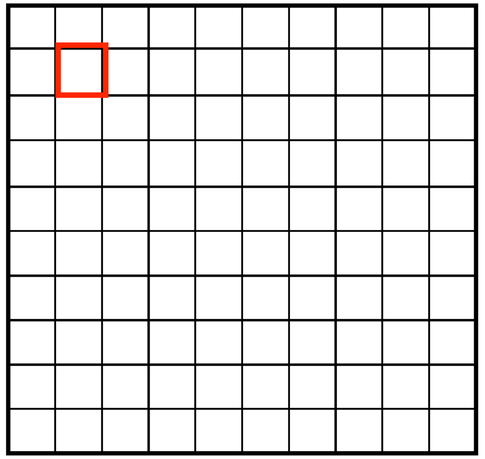


Mean filtering/Moving average

$F[x, y]$

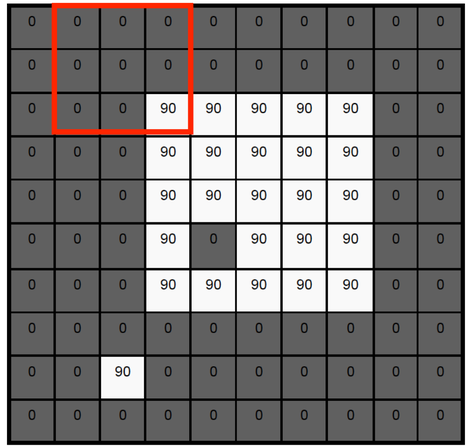


$G[x, y]$

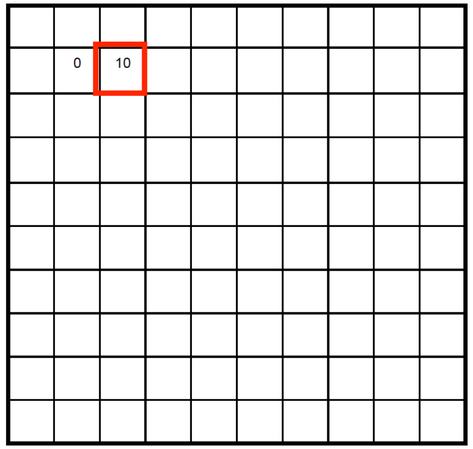


Mean filtering/Moving average

$F[x, y]$

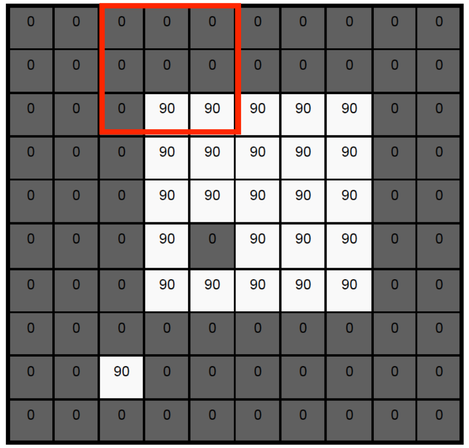


$G[x, y]$

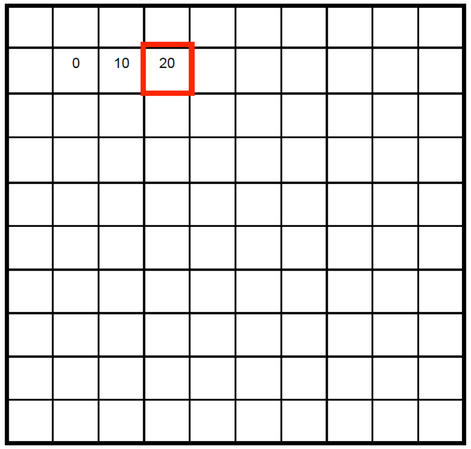


Mean filtering/Moving average

$F[x, y]$

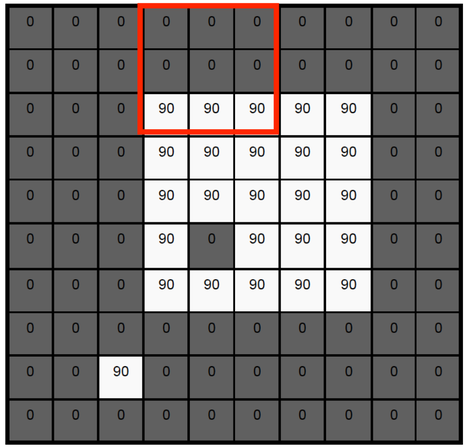


$G[x, y]$

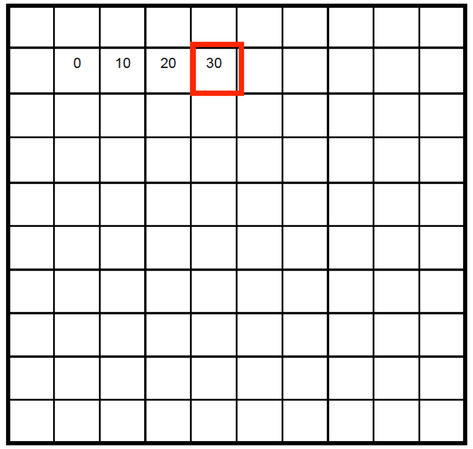


Mean filtering/Moving average

$F[x, y]$

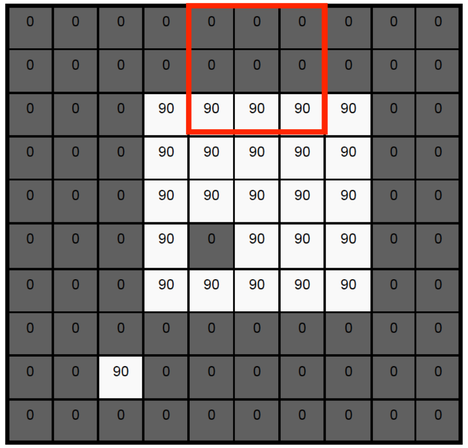


$G[x, y]$

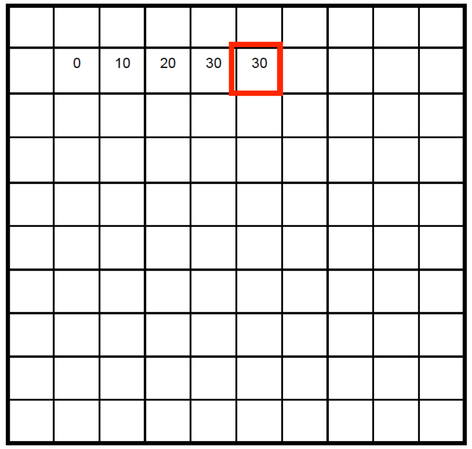


Mean filtering/Moving average

$F[x, y]$

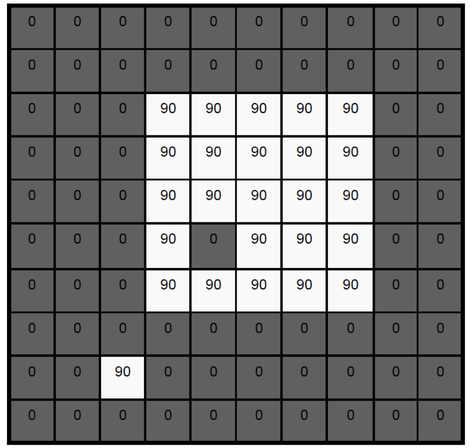


$G[x, y]$

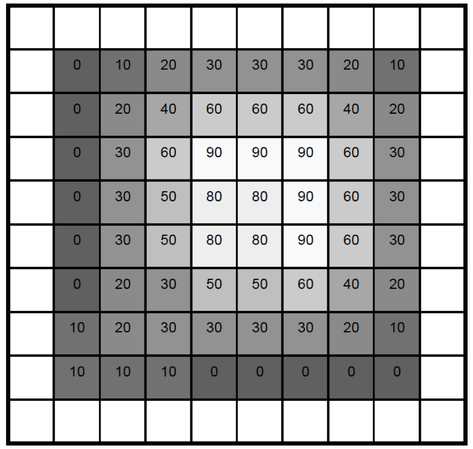


Mean filtering/Moving average

$F[x, y]$



$G[x, y]$



Linear filters: examples

Original

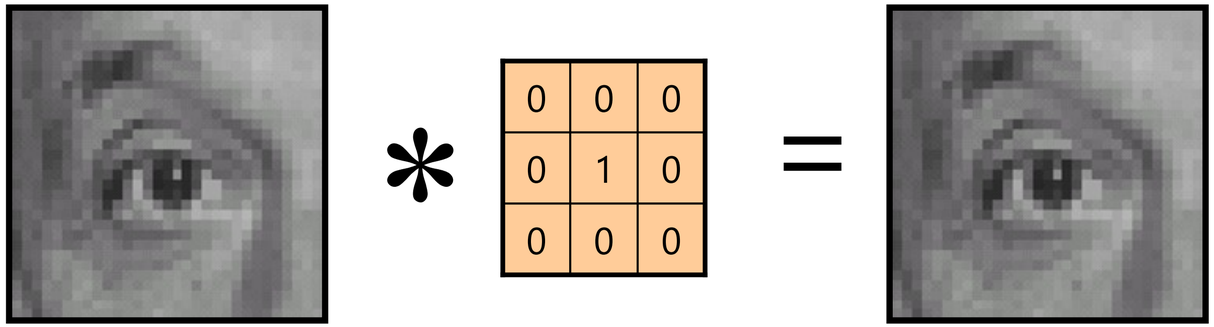
$$*$$

0	0	0
0	1	0
0	0	0

Source: D. Lowe

What image operation does filtering with this kernel perform?
([0 0 0; 0 1 0; 0 0 0])

Linear filters: examples

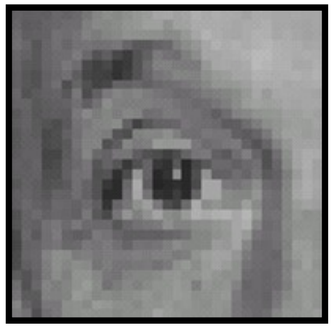


Original

Identical image

Source: D. Lowe

Linear filters: examples



Original

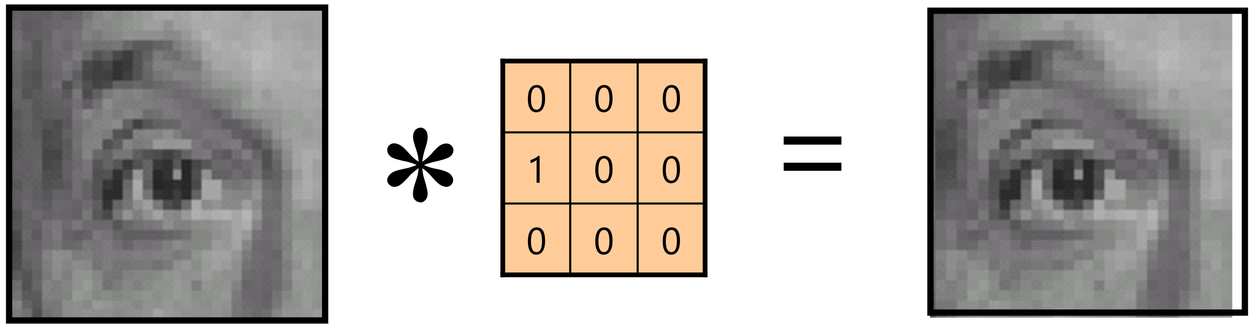
$$*$$

0	0	0
1	0	0
0	0	0

Source: D. Lowe

What image operation does filtering with this kernel perform?
([0 0 0; 1 0 0; 0 0 0])

Linear filters: examples

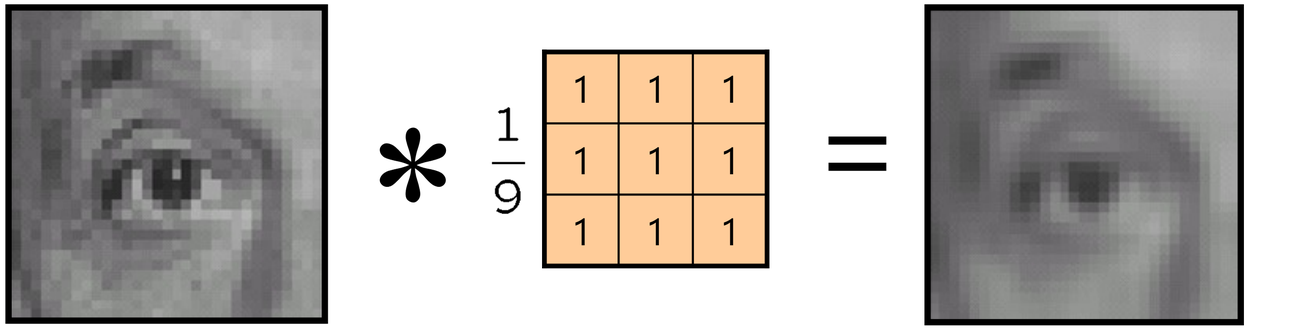


Original

Shifted left by 1 pixel

Source: D. Lowe

Linear filters: examples

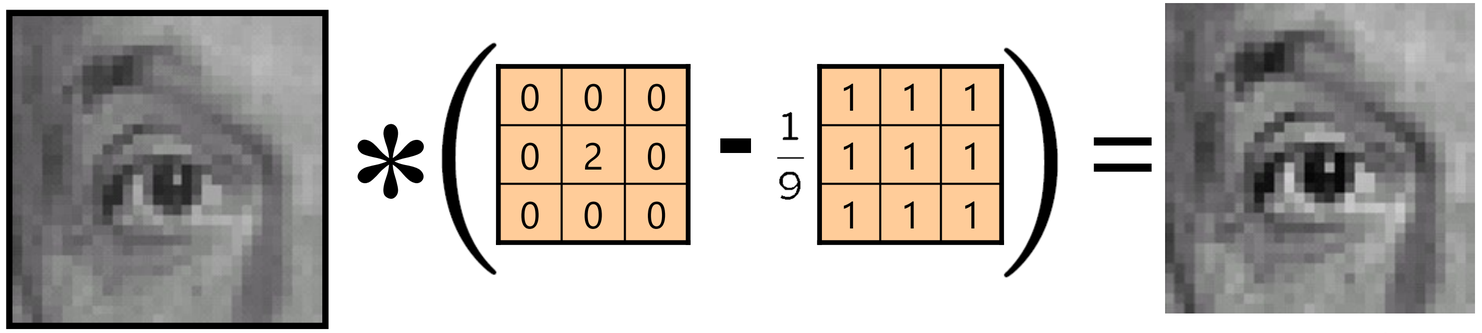


Original

Blur (with a mean filter)

Source: D. Lowe

Linear filters: examples

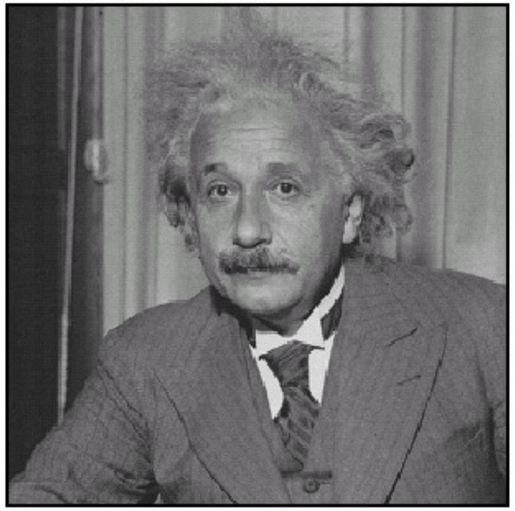


Original

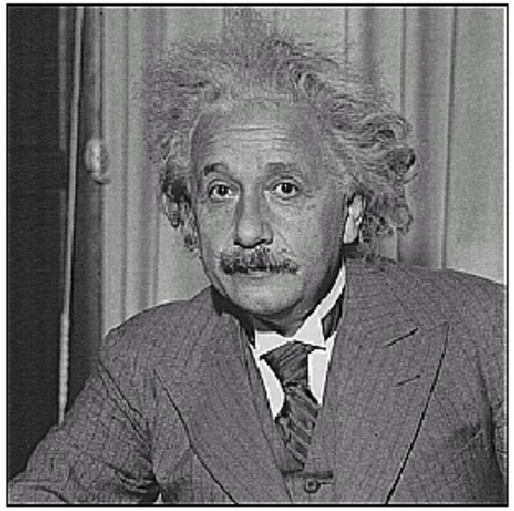
Sharpening filter
(accentuates edges)

Source: D. Lowe

Sharpening



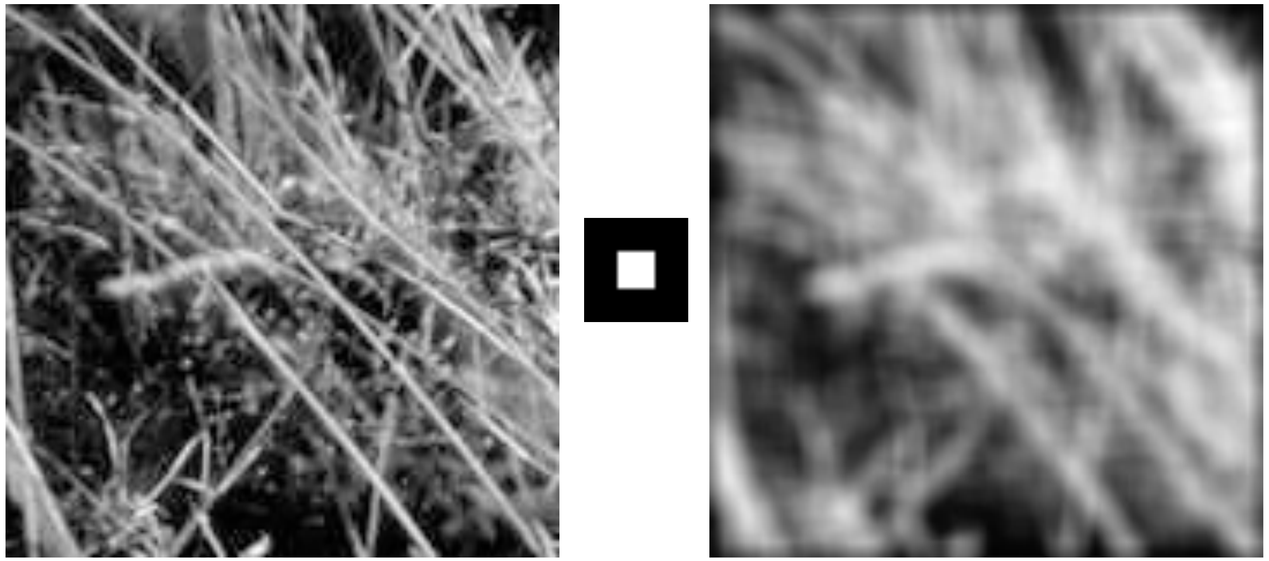
before



after

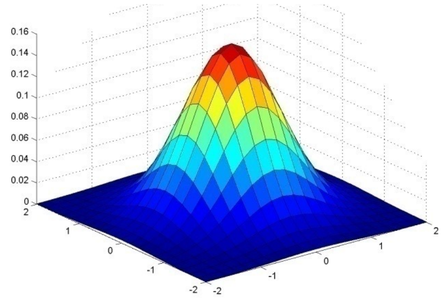
Source: D. Lowe

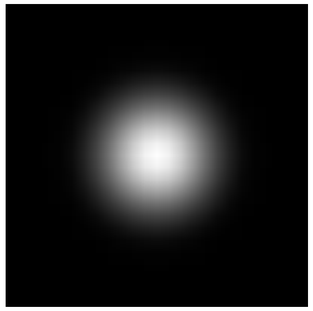
Smoothing with box filter revisited



Source: D. Forsyth

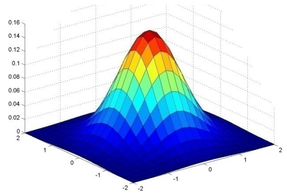
Gaussian kernel

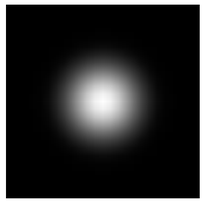




$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Gaussian kernel





Here is an example of a simple 3×3 Gaussian matrix:

0.075 0.124. 0.075
0.124 0.204 0.124
0.075 0.124 0.075

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

.

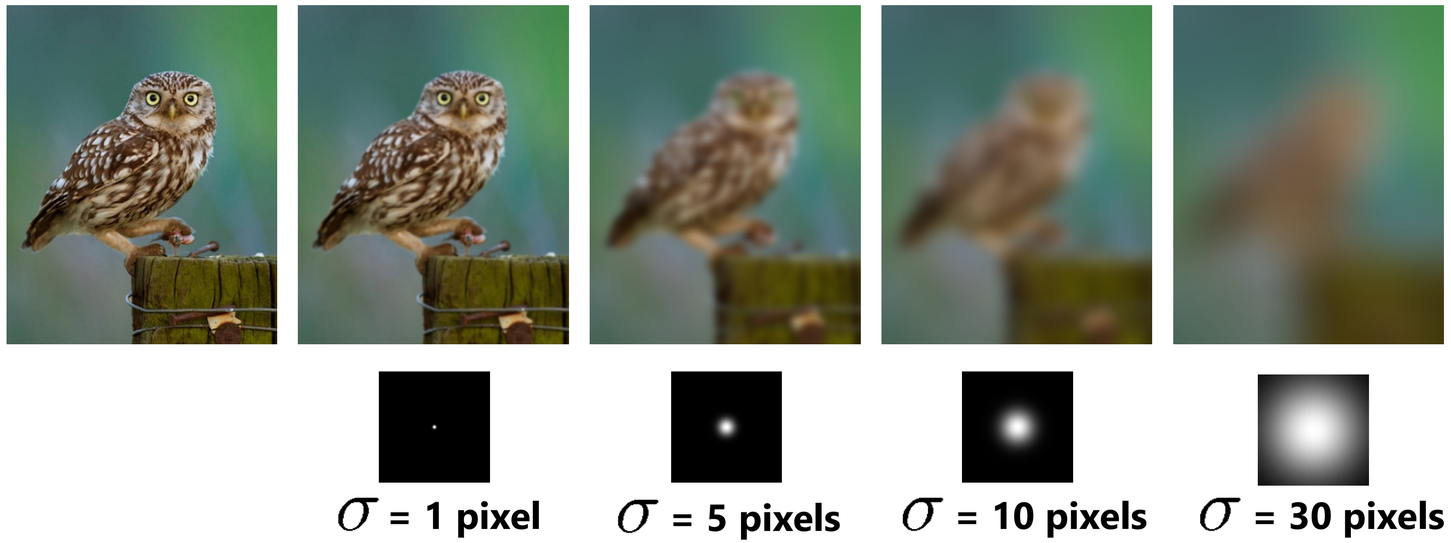
Source: C. Rasmussen

- x,y : Coordinates in 2D space.
- σ : Standard deviation, controlling the spread of the Gaussian function.
- e : Base of the natural logarithm.

The Gaussian function has the following properties:

- Symmetry: It is symmetric around its mean.
- Bell Shape: It has the shape of a "bell" which is determined by its standard deviation (σ).
- Non-Zero Everywhere: It is technically non-zero everywhere, though it approaches zero as you move away from the mean.
- Controlled by σ: A higher σ means a wider "bell," and a lower σ means a narrower "bell."

Gaussian filters

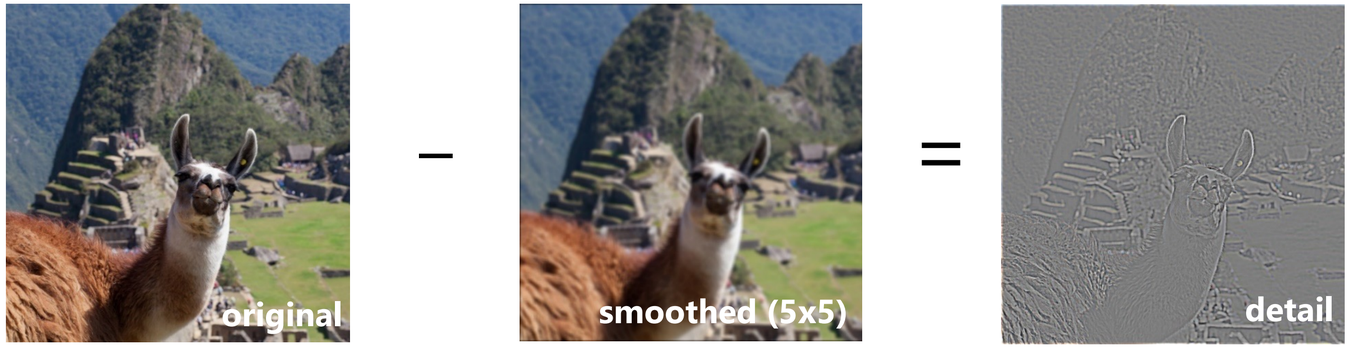


Gaussian filter

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

Sharpening revisited

- What does blurring take away?



(This "detail extraction" operation is also called a ***high-pass filter***)

Let's add it back:

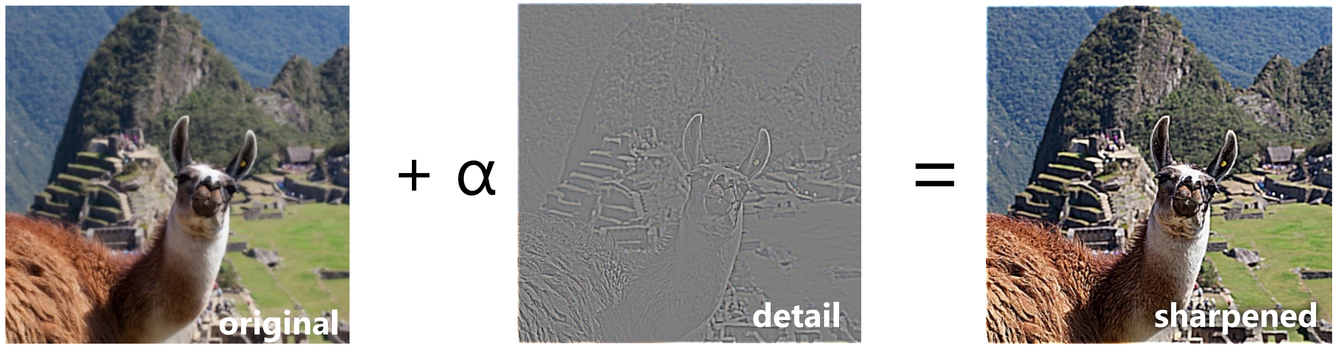


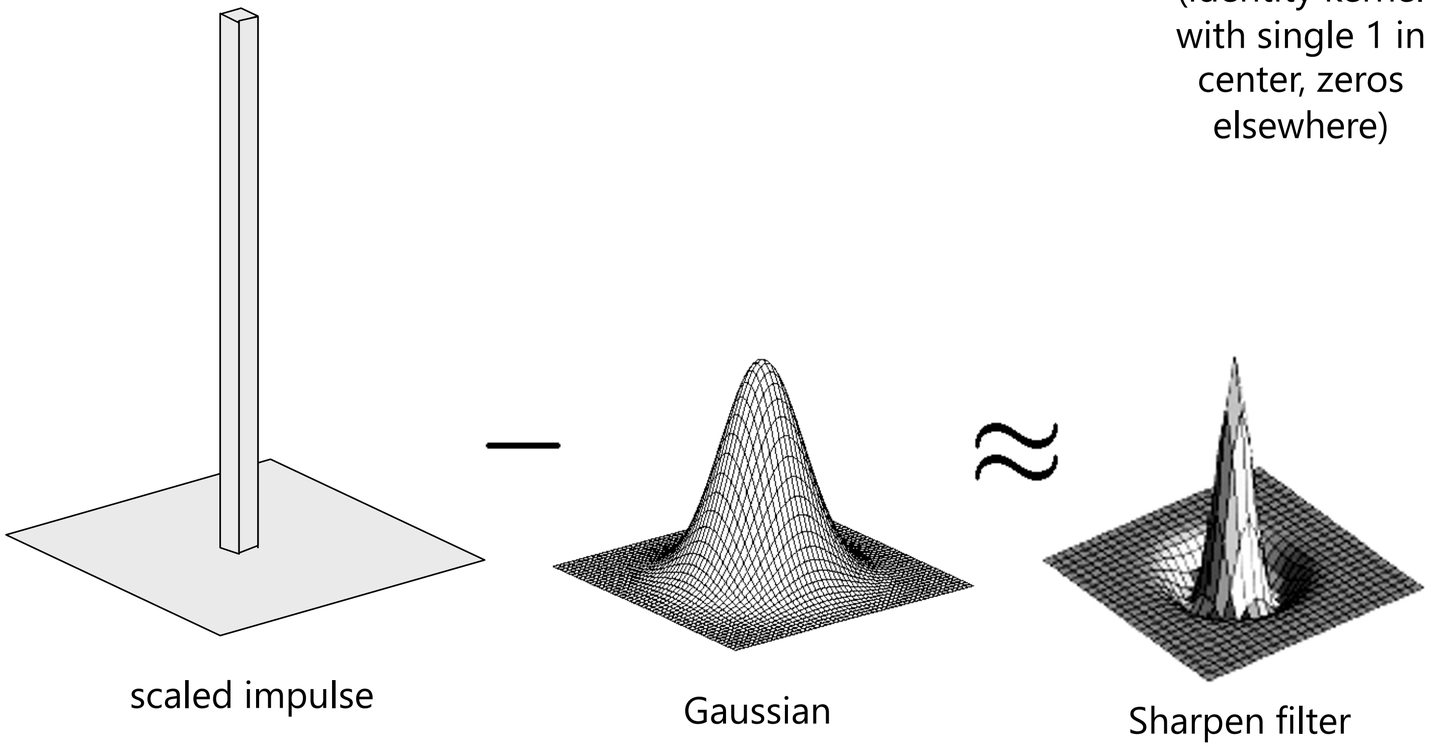
Photo credit: https://www.flickr.com/photos/geezaweezer/16089096376/

Sharpen filter

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

image

unit impulse (identity kernel with single 1 in center, zeros elsewhere)



scaled impulse — Gaussian ≈ Sharpen filter

Sharpen filter

“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: https://www.diyphotography.net/diy_create_your_own_bokeh/

Filters: Thresholding

$$g(m,n) = \begin{cases} 255, & f(m,n) > A \\ 0 & otherwise \end{cases}$$

Can thresholding be implemented with a linear filter?