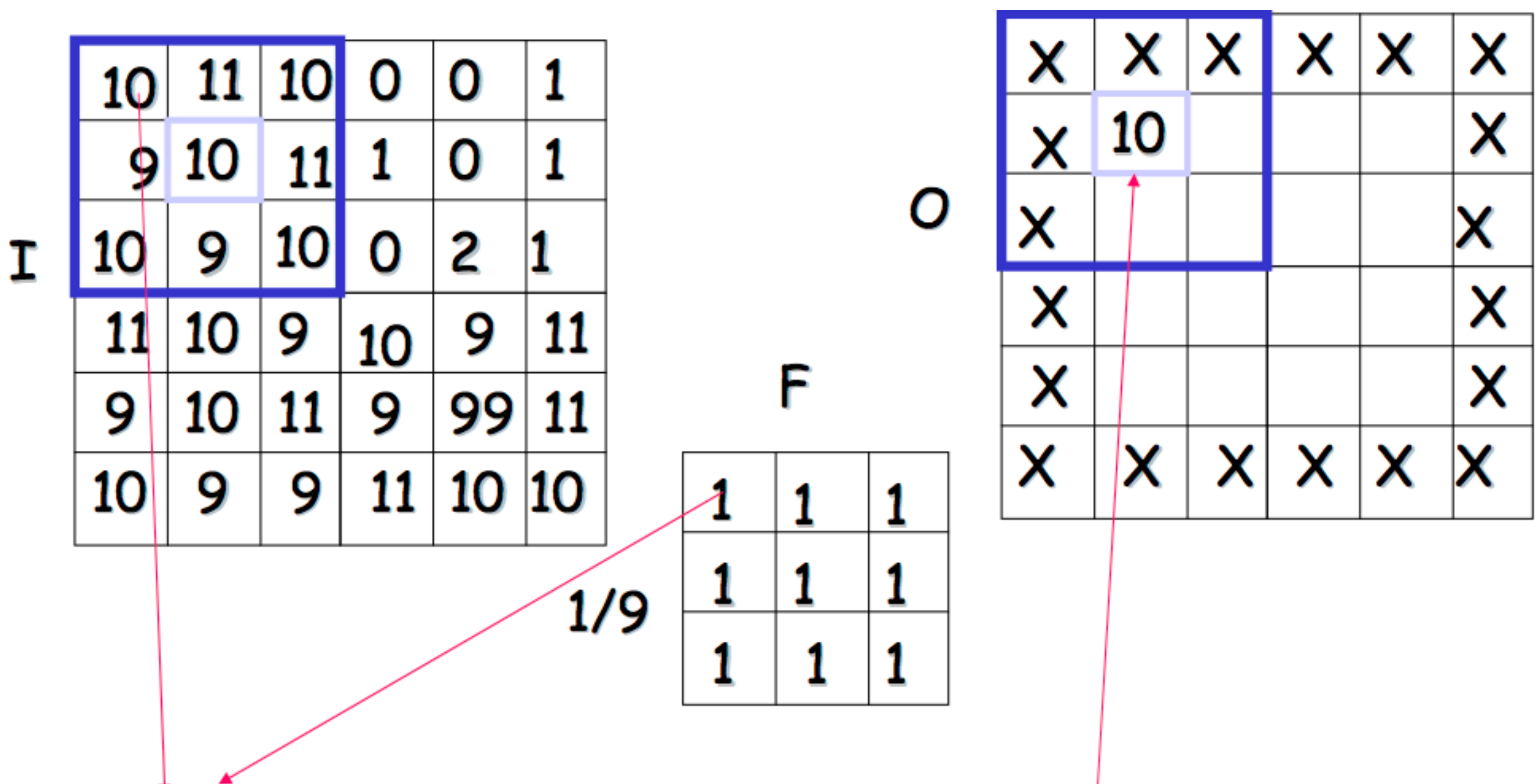
Linear Filtering

Linear Filters

- Linear filtering:
  - Form a new image whose pixels are a weighted sum of the original pixel values, using the same set of weights at each point.

Slide credit: David Lowe

Linear Filtering



1/9.(10x1 + 11x1 + 10x1 + 9x1 + 10x1 + 11x1 + 10x1 + 9x1 + 10x1) =
1/9.(90) = 10

Slide credit: Christopher Rasmussen

Linear Filtering (warm up slide)

Original

0	0	0
0	1	0
0	0	0

?

Slide credit: David Lowe

Linear Filtering (warm up slide)

Original

0	0	0
0	1	0
0	0	0

Filtered
(no change)

Slide credit: David Lowe

Linear Filtering

Original

0	0	0
0	0	1
0	0	0

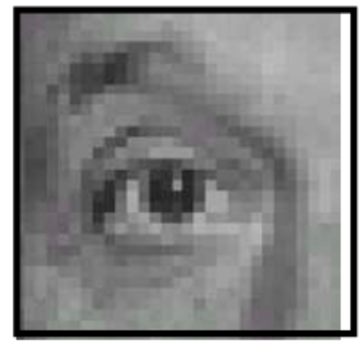
?

Slide credit: David Lowe

Linear Filtering

Original

0	0	0
0	0	1
0	0	0



Shifted left
By 1 pixel

Slide credit: David Lowe

Linear Filtering

Original

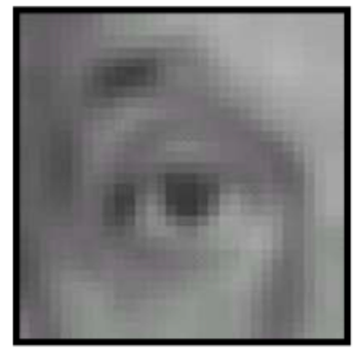
$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

?

Slide credit: David Lowe

Original

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



Blur (with a box filter)

Slide credit: David Lowe

Linear Filtering

Original

$$\begin{bmatrix}0&0&0\\0&2&0\\0&0&0\end{bmatrix} - \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix} \quad ?$$

(Note that filter sums to 1)

Slide credit: David Lowe

Linear Filtering

Original

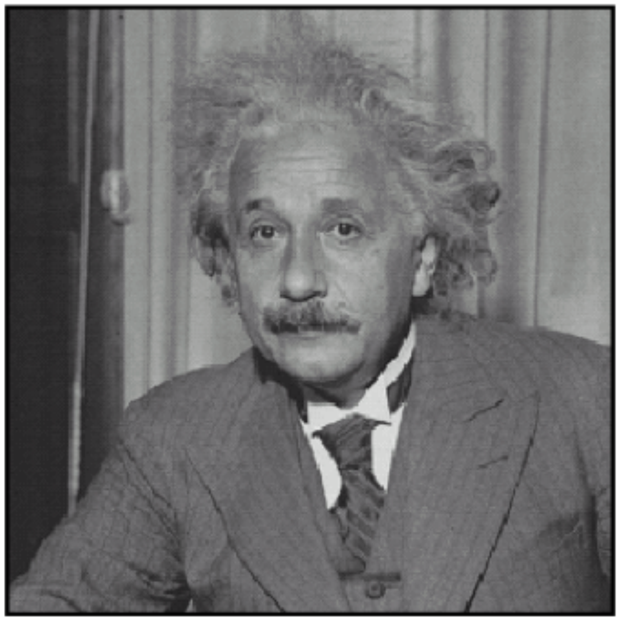
$$\begin{bmatrix}0&0&0\\0&2&0\\0&0&0\end{bmatrix} - \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

Sharpening filter

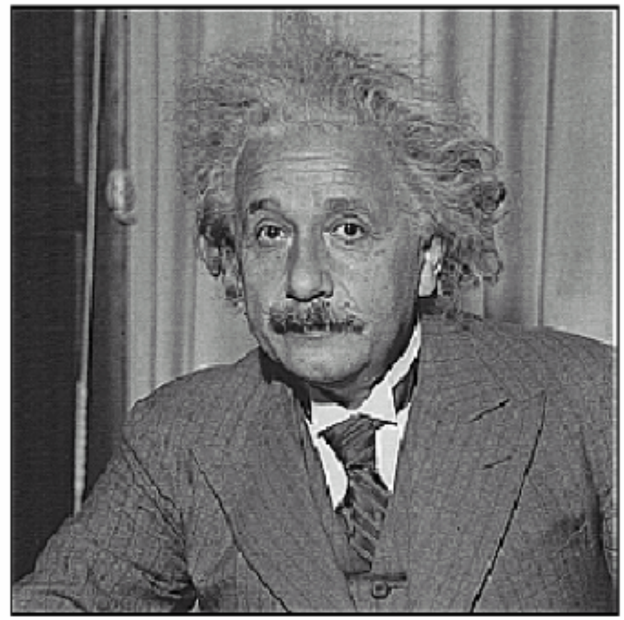
- Accentuates differences with local average
- Also known as Laplacian

Slide credit: David Lowe

Sharpening



before



after

Slide credit: David Lowe

Average Filter (box filter)

- Mask with positive entries, that sum to 1.
- Replaces each pixel with an average of its neighborhood.
- If all weights are equal, it is called a **box** filter.

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

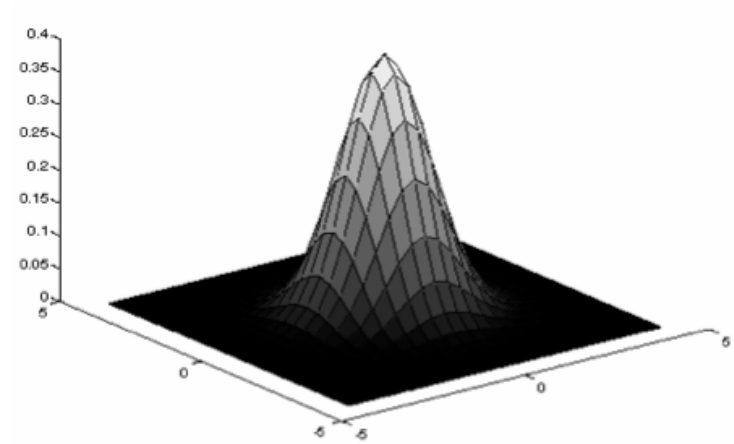
Slide credit: David Lowe

Example: Smoothing with a box filter

Slide credit: David Lowe

Smoothing with a Gaussian

- Smoothing with a box actually doesn't compare at all well with a defocussed lens
- Most obvious difference is that a single point of light viewed in a defocussed lens looks like a fuzzy blob; but the box filter would give a little square.

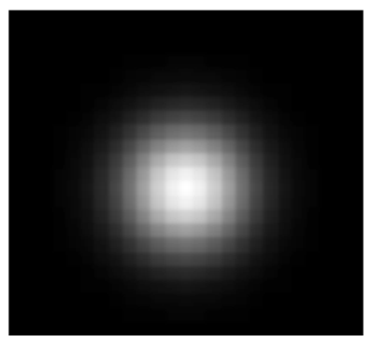


- A Gaussian gives a good model of a fuzzy blob
- It closely models many physical processes (the sum of many small effects)

Slide credit: David Lowe

Smoothing with box filter revisited

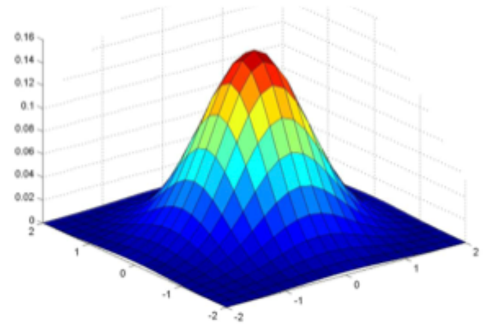
- Smoothing with an average actually doesn't compare at all well with a defocused lens
- Most obvious difference is that a single point of light viewed in a defocused lens looks like a fuzzy blob; but the averaging process would give a little square
- Better idea: to eliminate edge effects, weight contribution of neighborhood pixels according to their closeness to the center, like so:

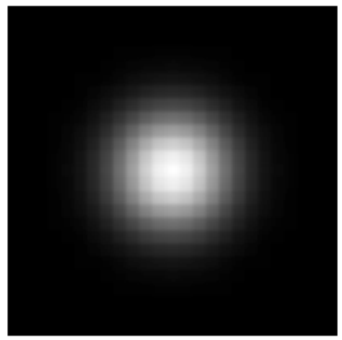


Slide credit: David Lowe

Gaussian Kernel

- Idea: Weight contributions of neighboring pixels by nearness





0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

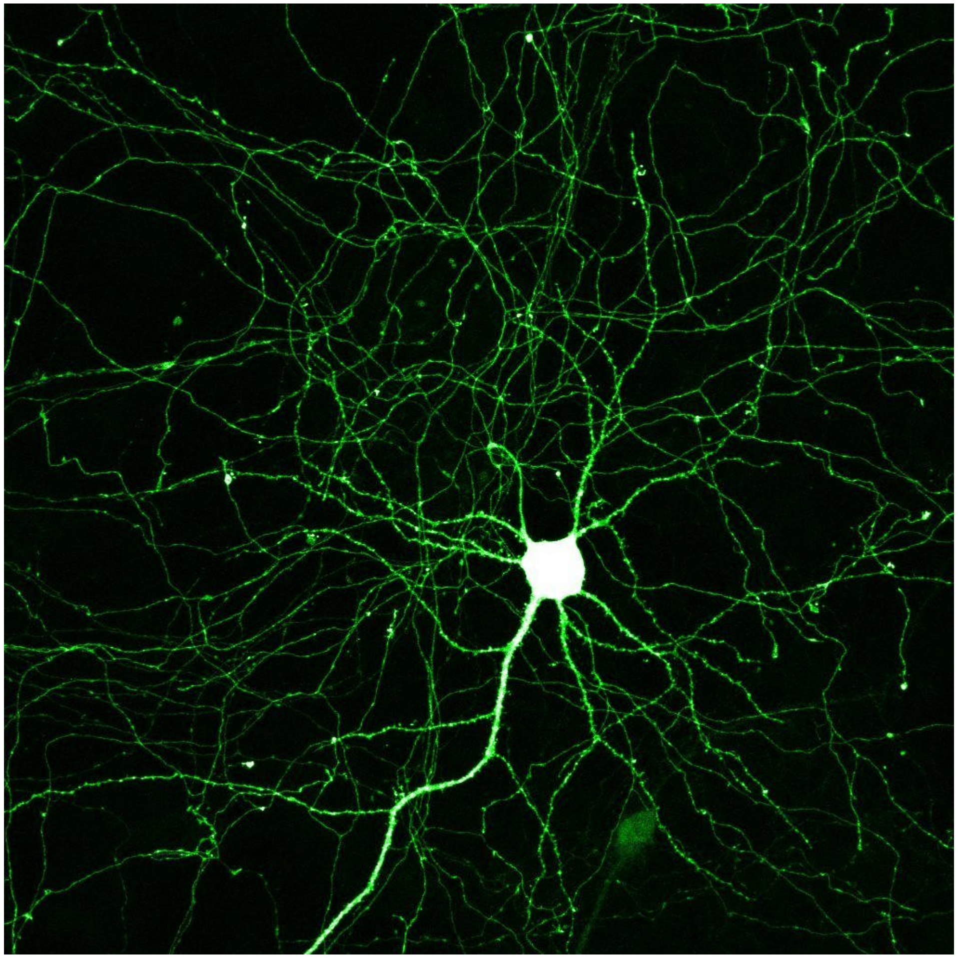
5 x 5, $\sigma = 1$

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

- Constant factor at front makes volume sum to 1 (can be ignored, as we should re-normalize weights to sum to 1 in any case).

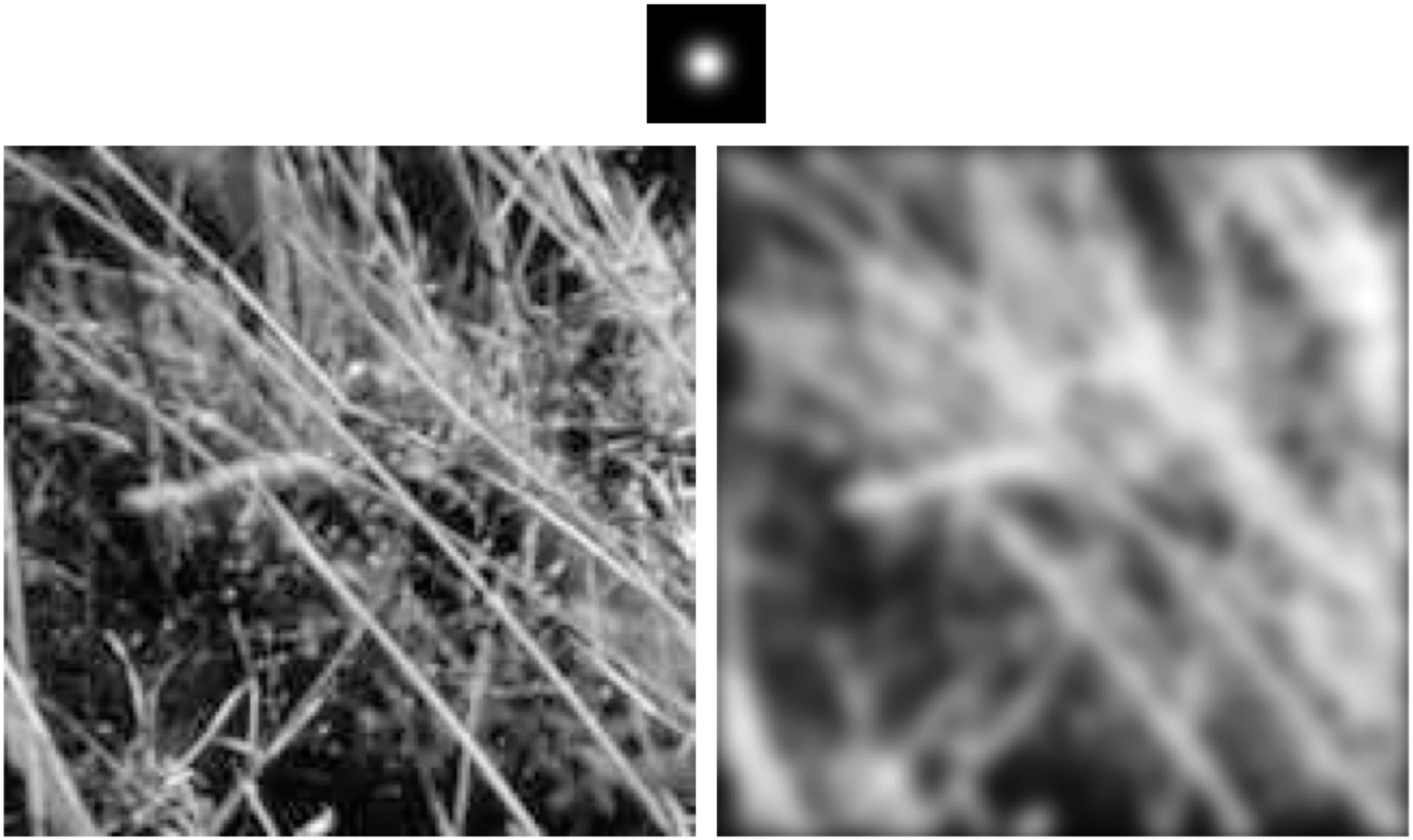
Slide credit: Christopher Rasmussen

A real neuron



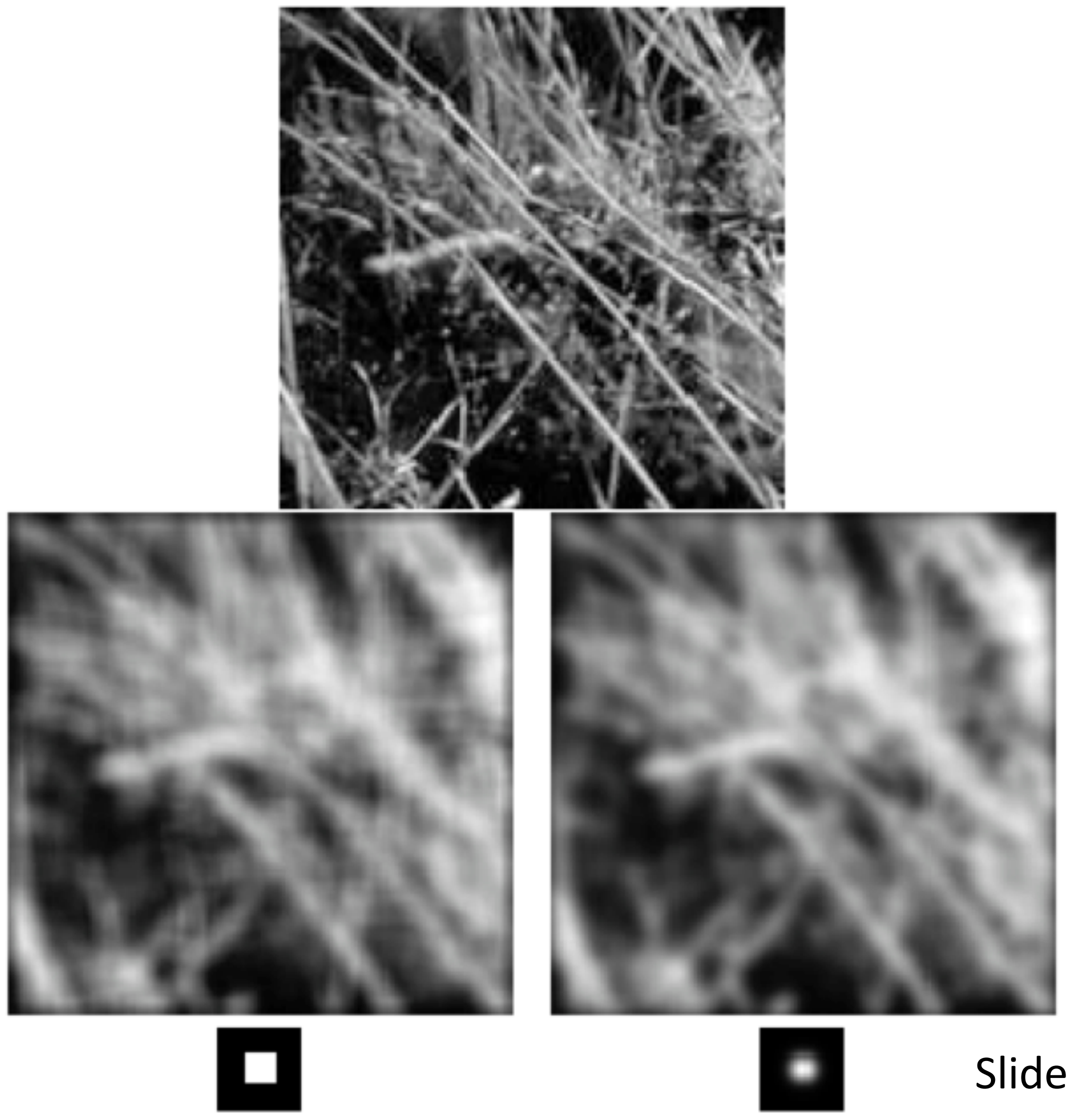
Location: Paul De Koninck Lab
Source: greenspine.ca

Smoothing with a Gaussian



Slide credit: David Lowe

Mean vs. Gaussian filtering



Slide credit: David Lowe

Efficient Implementation

- Both the BOX filter and the Gaussian filter are **separable** into two 1D convolutions:
  - First convolve each row with a 1D filter
  - Then convolve each column with a 1D filter.
  - (or vice-versa)

Slide credit: David Lowe

Associativity of Filtering

- Let “*” be linear filtering.
- A*B*C = (A*B)*C = A*(B*C)
  - Linear operators are associative.
  - Examples:
    - Addition
    - Integration
    - Matrix multiplication
    - Filtering

Separability example

2D convolution
(center location only)

$$\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} * \begin{bmatrix}2 & 3 & 3\\ 3 & 5 & 5\\ 4 & 4 & 6\end{bmatrix}$$

The filter factors
into a product of 1D
filters:

$$\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} = \begin{bmatrix}1\\ 2\\ 1\end{bmatrix} \times \begin{bmatrix}1 & 2 & 1\end{bmatrix}$$

Perform convolution
along rows:

$$\begin{bmatrix}1 & 2 & 1\end{bmatrix} * \begin{bmatrix}2 & 3 & 3\\ 3 & 5 & 5\\ 4 & 4 & 6\end{bmatrix} = \begin{bmatrix} & 11 & \\ & 18 & \\ & 18 & \end{bmatrix}$$

Followed by convolution
along the remaining column:

Source: K. Grauman

Separability of the Gaussian filter

For example, recall the 2D Gaussian

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

$$= \left(\frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{x^2}{2\sigma^2}} \right) \left(\frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{y^2}{2\sigma^2}} \right)$$

The 2D Gaussian can be expressed as the product of two functions, one a function of *x* and the other a function of *y*

Slide credit: David Lowe

Differentiation and Filtering

- Recall, for 2D function, f(x,y):

$$\frac{\partial f}{\partial x} = \lim_{\varepsilon \to 0}\left(\frac{f(x+\varepsilon, y)}{\varepsilon} - \frac{f(x,y)}{\varepsilon}\right)$$

- This is linear and shift invariant, so must be the result of a convolution.

- We could approximate this as

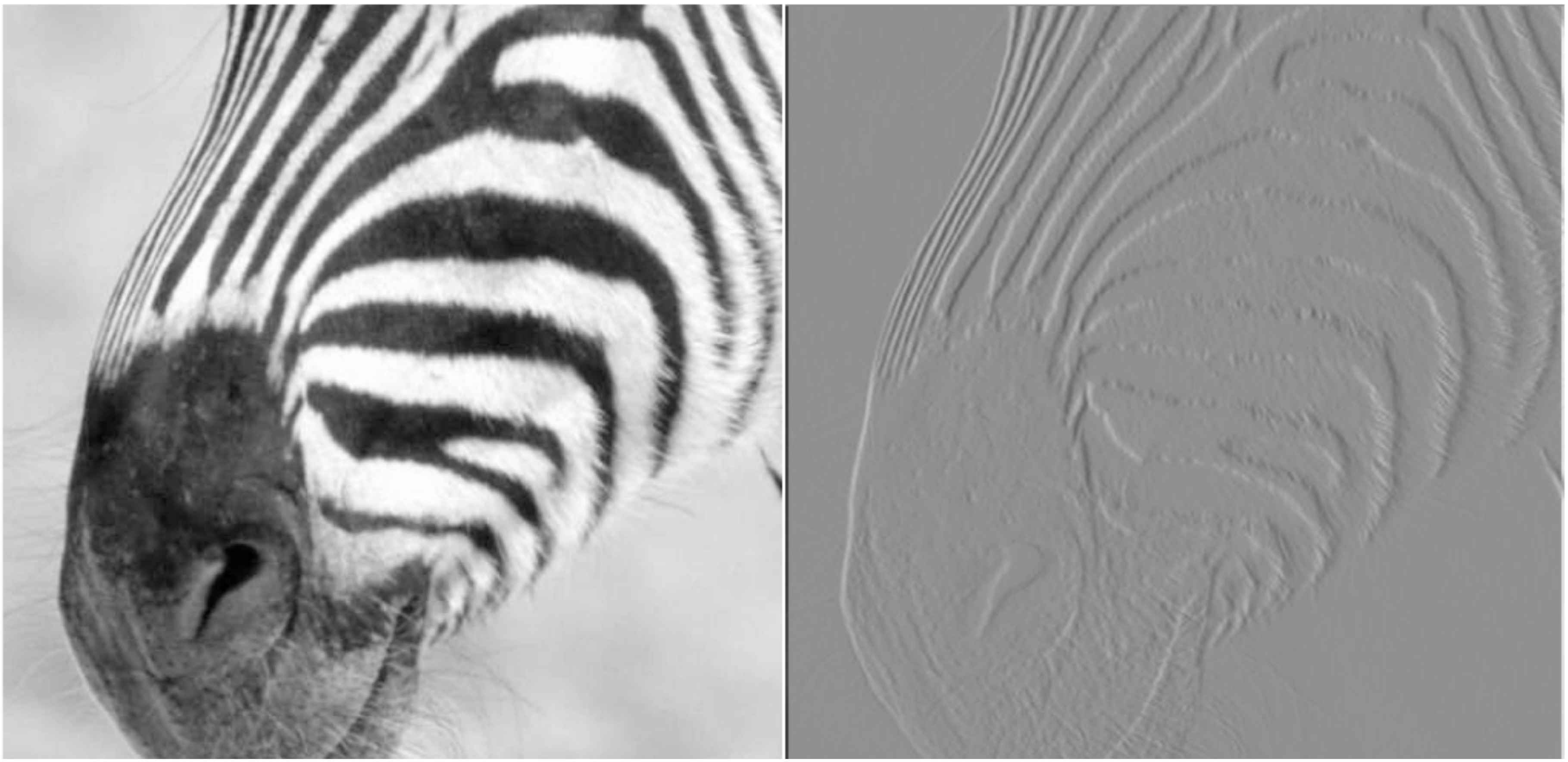
$$\frac{\partial f}{\partial x} \approx \frac{f(x_{n+1}, y) - f(x_n, y)}{\Delta x}$$

(which is obviously a convolution)

-1	**1**

Slide credit: David Lowe

Vertical gradients from finite differences

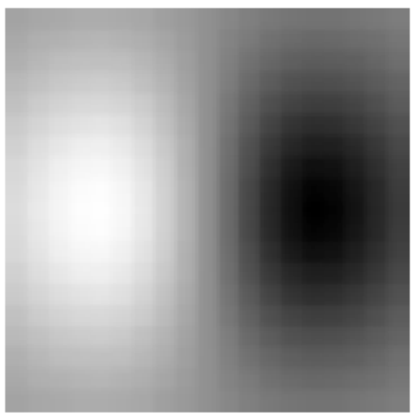


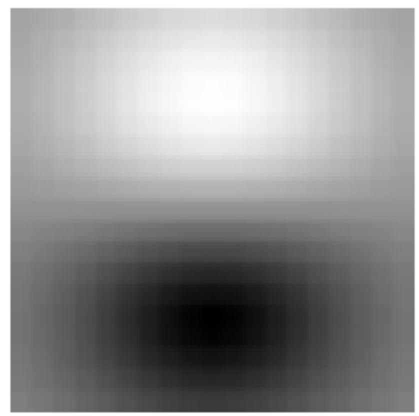
Slide credit: David Lowe

Filters are templates

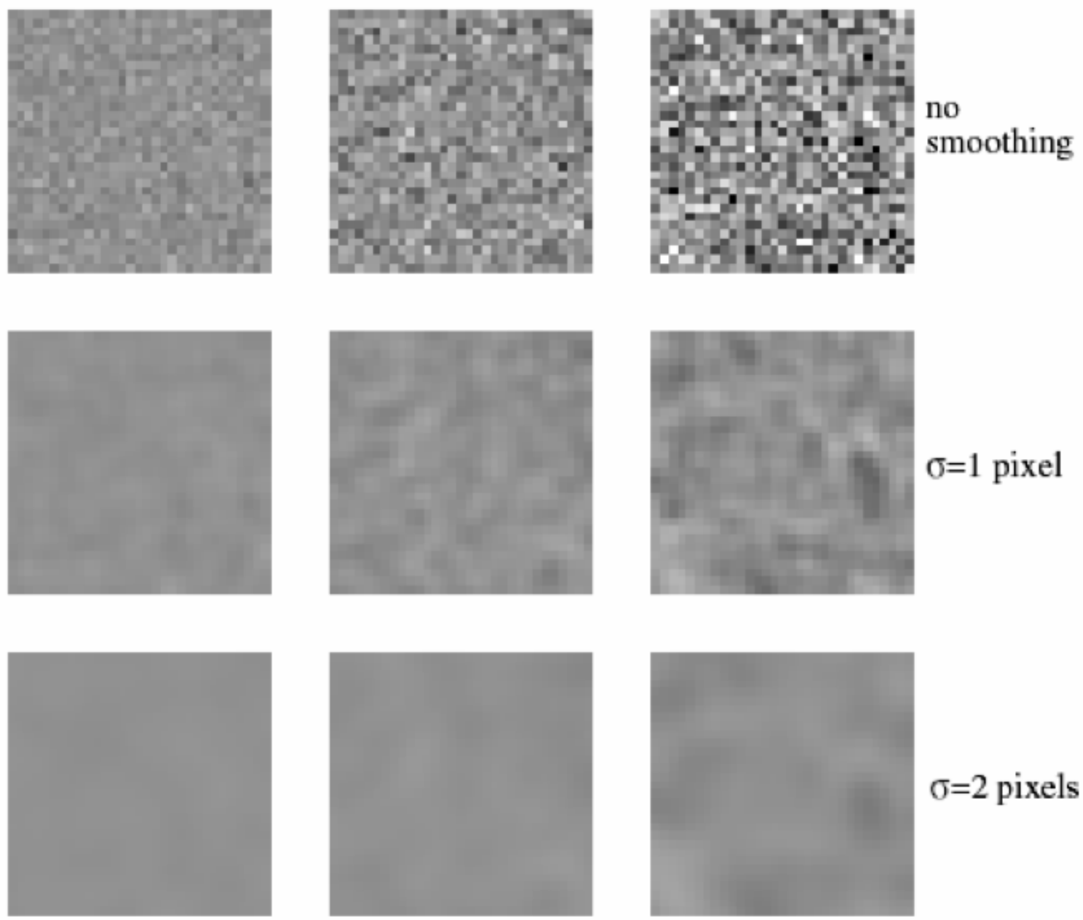
- Applying a filter at some point can be seen as taking a dot-product between the image and some vector
- Filtering the image is a set of dot products

- Insight
  - filters look like the effects they are intended to find
  - filters find effects they look like





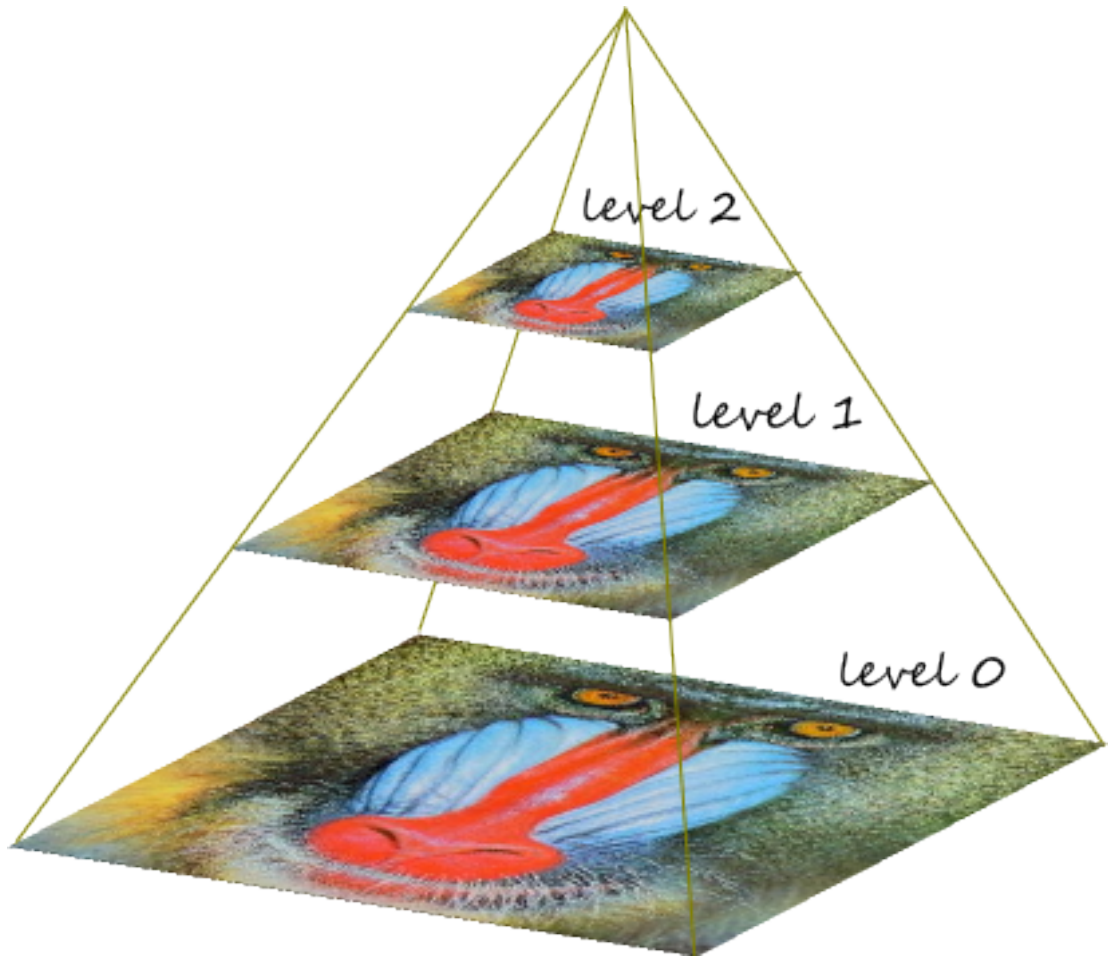
"Noise" reduction



Smoothing reduces pixel noise:

Each row shows smoothing with Gaussians of different width; each column shows different amounts of Gaussian noise.

Image Pyramids



Pyramid Recipe

- Filter with Gaussian
- Downsample
- Repeat