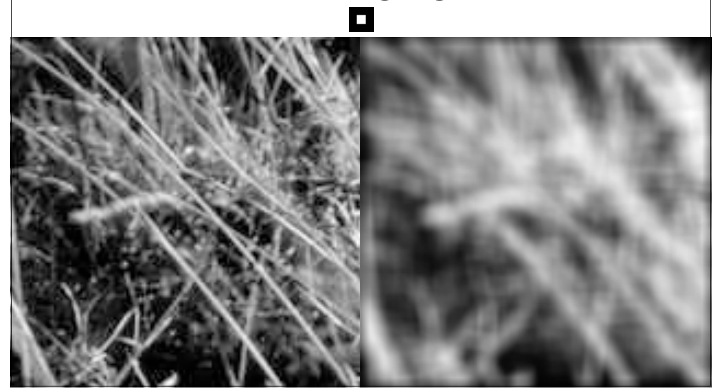
Correlation

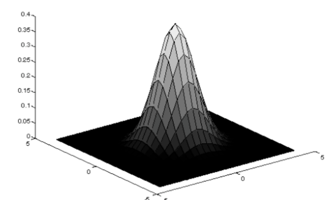
- Similar to convolution (no flips)
- Implements convolution (if a flip is used) or vice versa
- Finds things in images that “look like” the kernel
- The kernel is also referred to as a “mask”, especially in application oriented discussion (both in convolution and correlation).

Example: Smoothing by Averaging



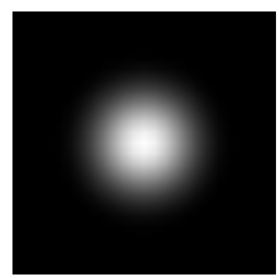
Smoothing with a Gaussian

- Smoothing with an average actually doesn’t really make sense because points close to the center should count more.
- Also, it does not 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.



- A Gaussian gives a good model of a fuzzy blob

An Isotropic Gaussian

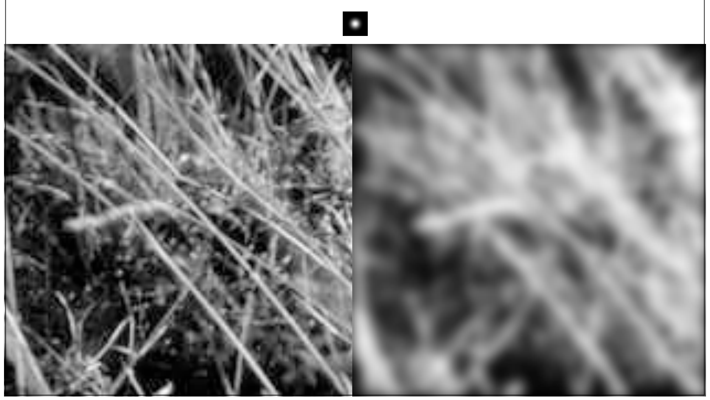


- The picture shows a smoothing kernel proportional to

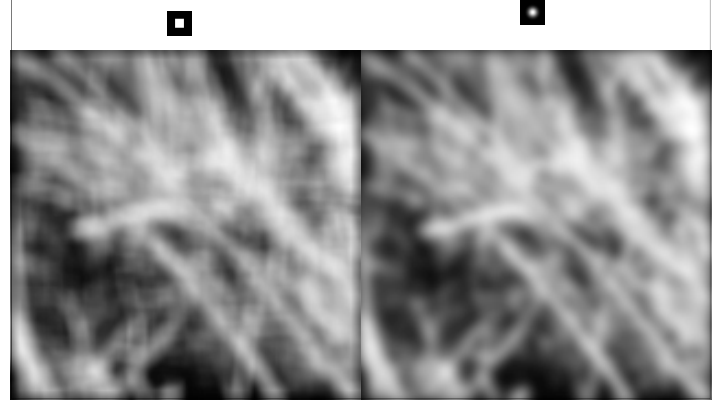
$$\exp\left(-\left(\frac{x^2+y^2}{2\sigma^2}\right)\right)$$

(a reasonable model of a circularly symmetric fuzzy blob)

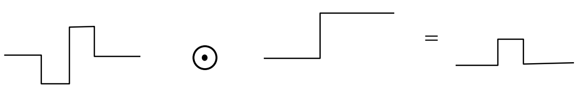
Smoothing with a Gaussian



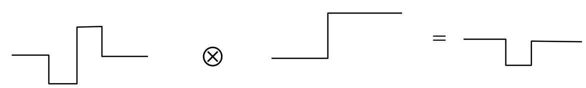
Block Averaging Gaussian



Correlation example



Convolution example



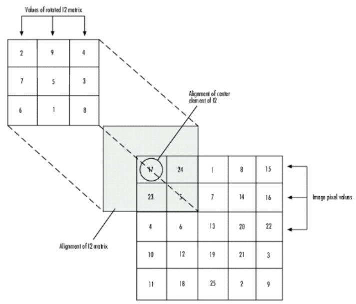
Convolution example two (from MathWorks website)

For example, suppose the image is

```
*

A = [17  24   1   8  15
     23   5   7  14  16
      4   6  13  20  22
     10  12  19  21   3
     11  18  25   2   9]
```

and the convolution kernel is

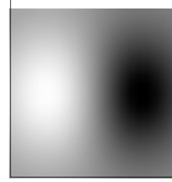
```
*

h = [8   1   6
     3   5   7
     4   9   2]
```

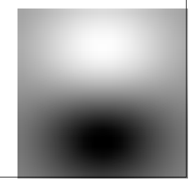


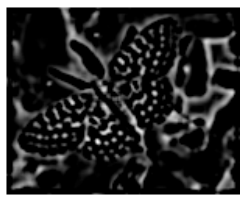
(Note two flips of kernel==rotate 180 degrees)

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
- Useful intuition
  - filters look like the effects they are intended to find
  - filters find effects that look like them

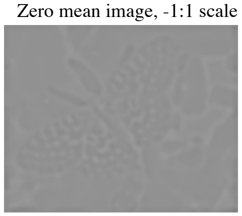




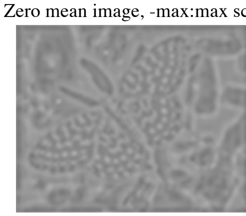


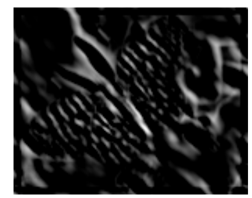
Positive responses

Zero mean image, -1:1 scale



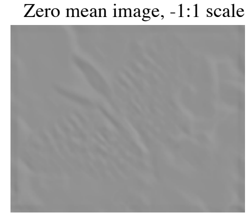
Zero mean image, -max:max scale



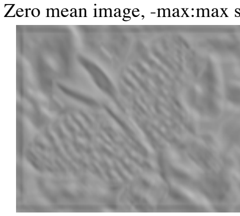


Positive responses

Zero mean image, -1:1 scale



Zero mean image, -max:max scale



Normalized correlation

- Think of filters of a dot product
  - **problem**: brighter parts give bigger results even if the structure is same (often not what you want)
  - **normalized** correlation output is filter output, divided by root sum of squares of values over which filter lies

$$\frac{\mathbf{h} \bullet \mathbf{f}}{|\mathbf{f}|} \qquad \text{(}\mathbf{f}\text{ is limited to where h is non zero)}$$

  - Can think in terms of angle between vectors. Recall

$$\cos(\theta) = \frac{\mathbf{h} \bullet \mathbf{f}}{|\mathbf{h}||\mathbf{f}|} \qquad \text{(}|\mathbf{h}|\text{ is not relevant to this problem)}$$

Finding Edges

- Edges reveal much about images
- Edge representations can be seen as information compression (because boundary is fewer pixels than the inside)
- Edges are the result of many different things
  - simple material change (step edge, corners)
  - illumination change (often soft, but not always)
  - shading edges and bar edges in inside corners
- An edge is basically where the images changes---hence finding images is studying changes (differentiation)

Differentiation and convolution

- Recall

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

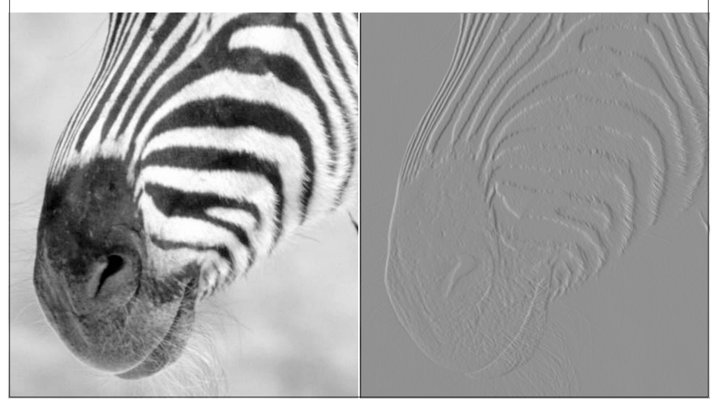
- Now this is linear and shift invariant, so must be the result of a convolution.

- We could approximate this as

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

- Obviously a convolution, but not a good way as we shall see.

Finite differences (x-direction)



Noise

- Simplest noise model
  - independent stationary additive Gaussian noise
  - the noise value at each pixel is given by an independent draw from the same normal probability distribution

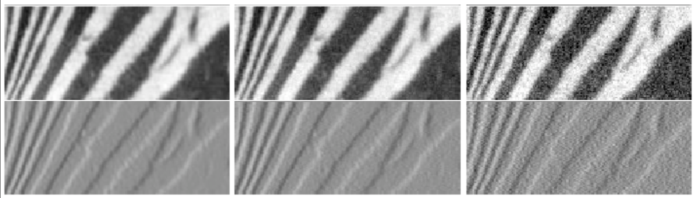
image with added Gaussian noise (sigma=1)

image with added Gaussian noise (sigma=16)

Finite differences and noise

- Finite difference filters respond strongly to noise
  - Noise is not correlated across adjacent pixels, but the pixels tend to be correlated
  - Thus differences lock onto the noise!
- Generally, the larger the noise the bigger such a response

Finite differences responding to noise



Increasing noise ----->

(zero mean additive gaussian noise)

Smoothing reduces noise

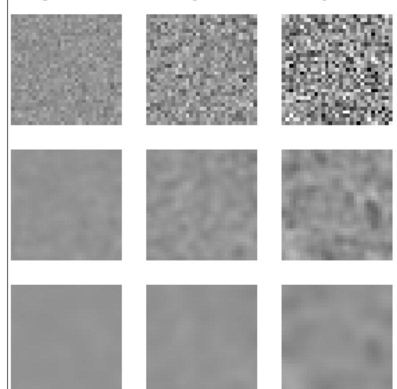
- Generally expect pixels to “be like” their neighbours
  - surfaces turn slowly
  - relatively few reflectance changes
- Generally expect noise processes to be **independent** from pixel to pixel
- Implies that some kind of averaging or smoothing suppresses noise, for appropriate noise models

Smoothing reduces noise

- Degree of smoothing <==> scale
  - the parameter in the symmetric Gaussian
  - as this parameter goes up, more pixels are involved in the average
  - and the image gets more blurred
  - and noise is more effectively suppressed

Noise sigma

σ=0.05 σ=0.1 σ=0.2



no smoothing

σ=1 pixel

σ=2 pixels

The effects of smoothing
Each row shows smoothing with gaussians of different width; each column shows different realizations of an image of gaussian noise.

Smoothing sigma

Median Filtering

- Using a Gaussian to remove noise assumes a well behaved noise process (sensitive to outliers).
- A more robust method is to replace a pixel with the median of the ones in a window (median filtering)
- This filter is non-linear!
  - We give up lots of nice properties