Differentiation and convolution

- Recall

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

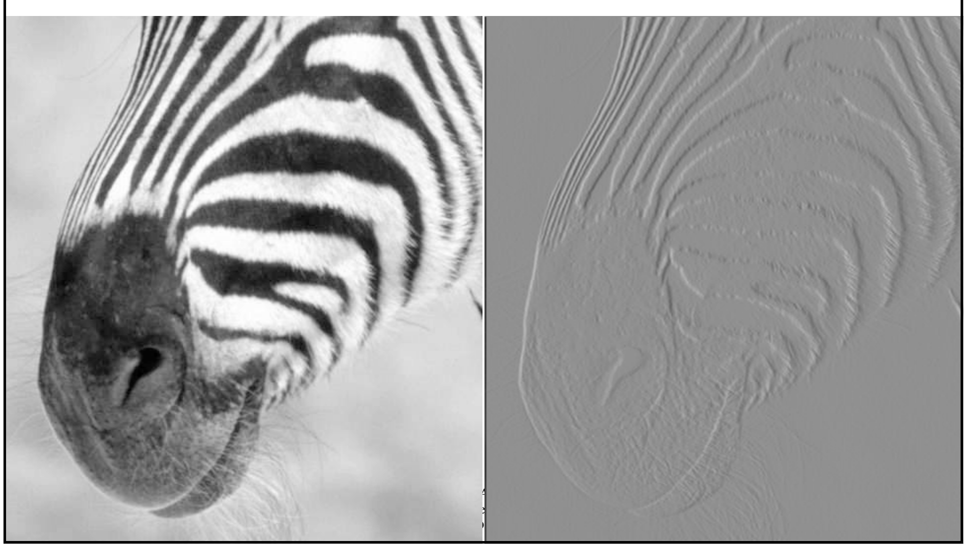
- Now this is linear and shift invariant, so must be the result of a convolution.

- We could approximate this as

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

(which is obviously a convolution; it's not a very good way to do things, as we shall see)

Finite differences

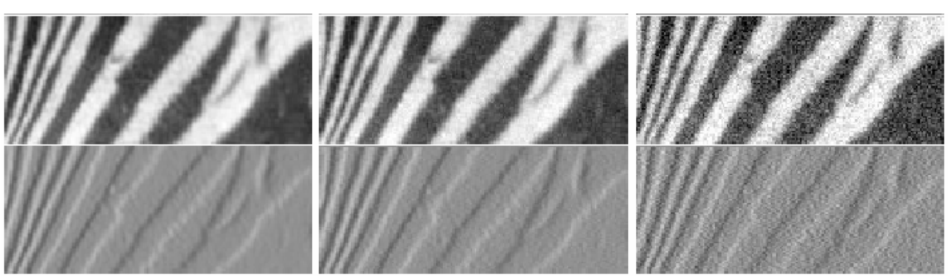


Finite differences and noise

- Finite difference filters respond strongly to noise
  - obvious reason: image noise results in pixels that look very different from their neighbours
- Generally, the larger the noise the stronger the response
- What is to be done?
  - intuitively, most pixels in images look quite a lot like their neighbours
  - this is true even at an edge; along the edge they're similar, across the edge they're not
  - suggests that smoothing the image should help, by forcing pixels different to their neighbours (=noise pixels?) to look more like neighbours

Computer Vision - A Modern Approach
Set: Linear Filters
Slides by D.A. Forsyth

Finite differences responding to noise

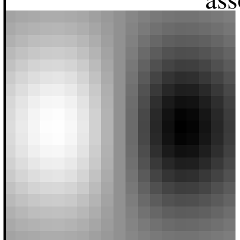


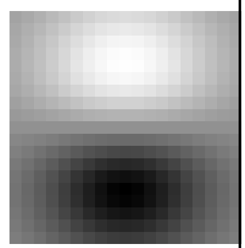
Increasing noise ->
(this is zero mean additive gaussian noise)

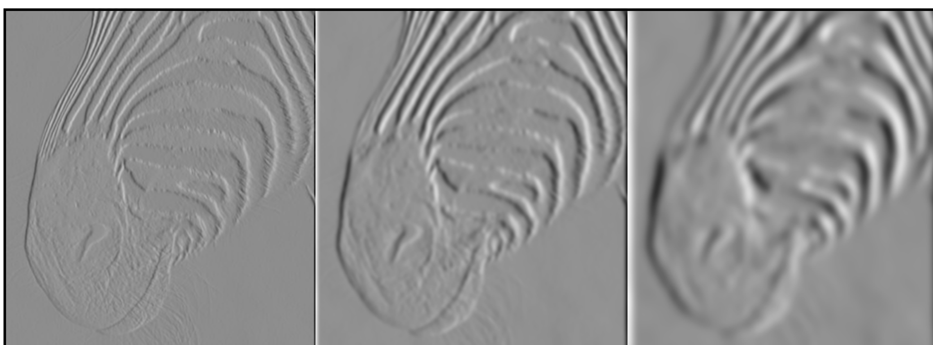
Computer Vision - A Modern Approach
Set: Linear Filters
Slides by D.A. Forsyth

Smoothing and Differentiation

- Issue: noise
  - smooth before differentiation
  - two convolutions to smooth, then differentiate?
  - actually, no - we can use a derivative of Gaussian filter
    - because differentiation is convolution, and convolution is associative







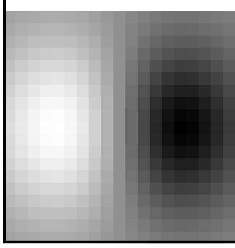
1 pixel 3 pixels 7 pixels

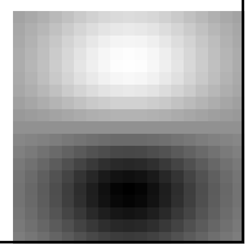
The scale of the smoothing filter affects derivative estimates, and also the semantics of the edges recovered.

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



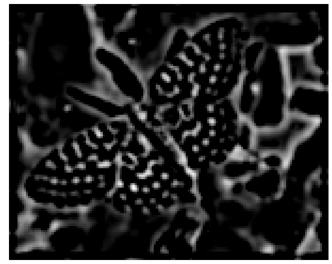


Normalized correlation

- Think of filters of a dot product
  - now measure the angle
  - i.e normalised correlation output is filter output, divided by root sum of squares of values over which filter lies

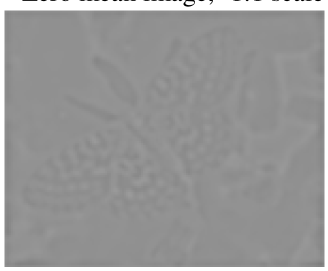
- Tricks:
  - ensure that filter has a zero response to a constant region (helps reduce response to irrelevant background)
  - subtract image average when computing the normalizing constant (i.e. subtract the image mean in the neighbourhood)
  - absolute value deals with contrast reversal

Computer Vision - A Modern Approach
Set: Linear Filters
Slides by D.A. Forsyth

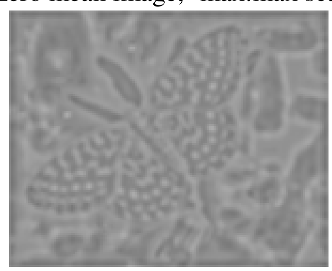


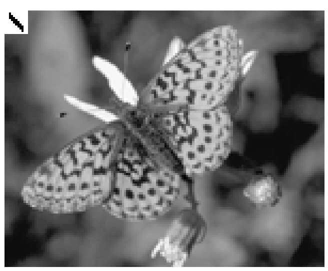
Positive responses

Zero mean image, -1:1 scale



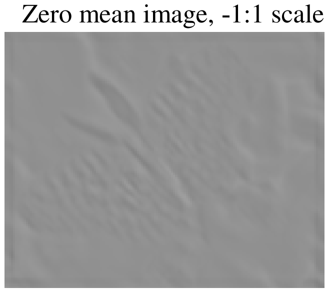
Zero mean image, -max:max scale





Positive responses

Zero mean image, -1:1 scale



Zero mean image, -max:max scale