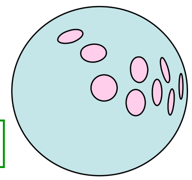
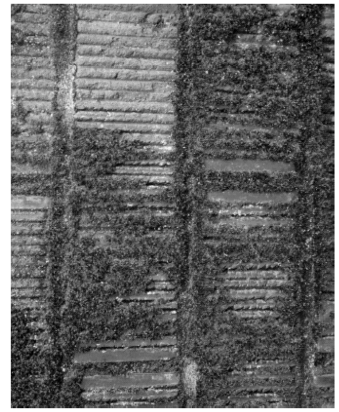
Texture

- Texture always has a scale (leaf -> bush -> forest)
- Key issue: representing texture
- Texture based matching
  - given a texture patch, can you find it in a database
- Texture segmentation
  - key issue: representing texture
- Texture synthesis
  - useful; also gives some insight into quality of representation
- Shape from texture
  - cover superficially

But you should be able to explain this figure covered in class.





Representing textures

- Textures are made up of quite stylized sub-elements
  - e.g. polka-dots
- Representation:
  - choose scale, quantify sub-elements, and represent their **statistics**
- But what are the sub-elements, and how do we quantify them?
  - recall (normalized) correlation
  - find evidence for sub-elements by applying filters
  - quantify using the magnitude of the filter response over the scale

Representing textures

- Begin with collections of responses to a variety of filters (filter bank)
- Generally need a collection of spots and bars at various scales and orientations (for the bars), but it is not so critical how one gets the spots and bars.
- Thus the filter banks are typically chosen based on other (often relatively arbitrary) considerations.

A typical filter bank

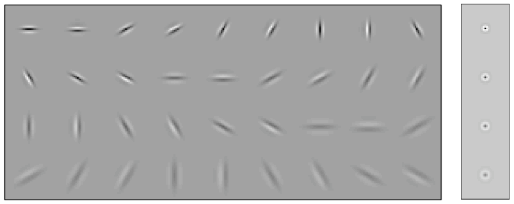
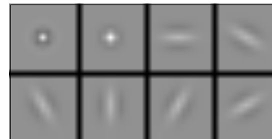
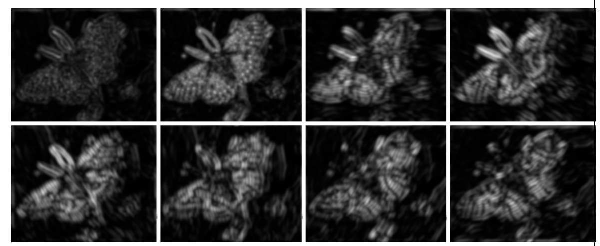


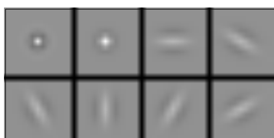
Figure 4. Left: Filter set f_i consisting of 2 phases (even and odd), 3 scales (spaced by half-octaves), and 6 orientations (equally spaced from 0 to π). The basic filter is a difference-of-Gaussian quadrature pair with $3:1$ elongation. Right: 4 scales of center-surround filters. Each filter is L_1-normalized for scale invariance.

From Malik et al., "Contour and texture analysis for image segmentation"

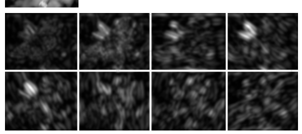
(We have an implementation for this filter bank, as part of the N-cuts software from Berkeley).

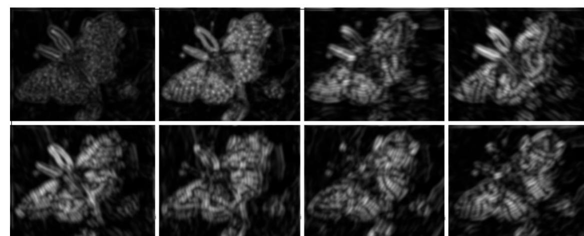




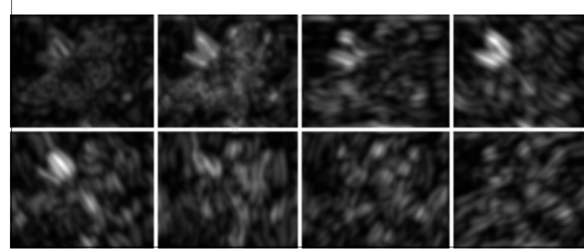


Next scale up



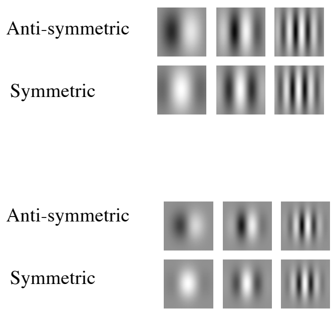


Smaller Scale



Larger Scale (Image from previous slide made larger to compare)

Yet another way to create filters



Gabor filters at different scales and spatial frequencies

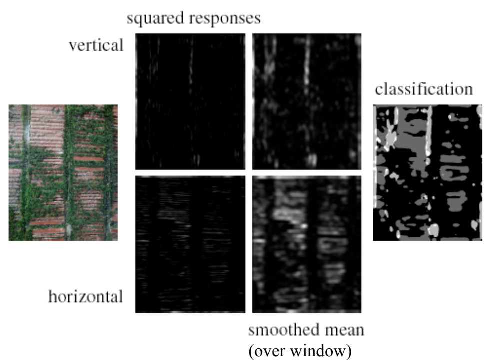
Basically a sine or cosine multiplied by a Gaussian to localize it.

Easy to make oriented versions

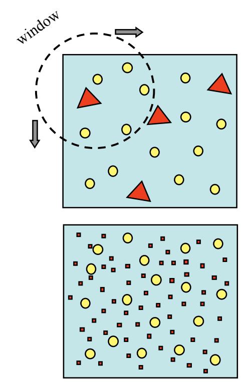
Bottom six are more localized (smaller Gaussian sigma)

Representing textures

- Associate texture with **statistics** of the conglomerate of responses over some scale (window size)
- Simplest statistic is mean (square) response for each filter
  - So, N filters gives a vector of dimension N
- Including standard deviation helps
  - Now, N filters gives a vector of dimension 2*N
- These simple methods ignore spatial correlation
  - Including spatial correlation increases data by a factor of the number of pixels in a window
  - Too many, and too noisy ==> cluster point data in "textons"
  - Texture in a window is a histogram of texton popularity



Example



Consider how a variety of filter shapes and scales provide numbers that can distinguish these textures and many others)

Consider how the statistics over the given window size capture the particulars of the repeating patterns.

exam

Texture synthesis

- Use image as a source of probability model
- Grab a section of the image at random for seeding
- Expand from unfilled edges by matching boundary sections to the image, and randomly sampling the unfilled value from the blocks matched
- (For details see pages 206-207 in text)

exam

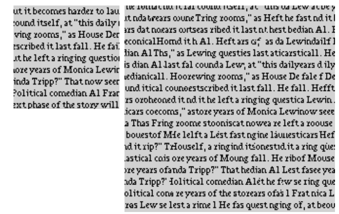


Figure from Texture Synthesis by Non-parametric Sampling, A. Efros and T.K. Leung, Proc. Int. Conf. Computer Vision, 1999 copyright 1999, IEEE