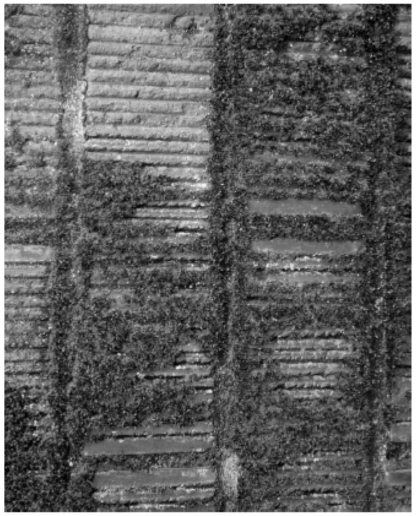
Texture

- 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 conceptually



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).
- The details of spots and bars do not matter much.
  - Thus the filter banks are typically chosen based on other (often relatively arbitrary) considerations.

A typical filter bank

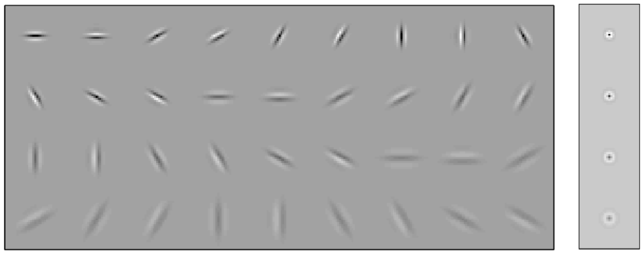
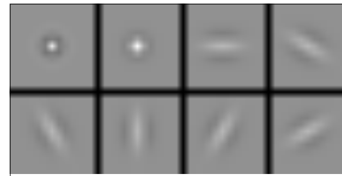
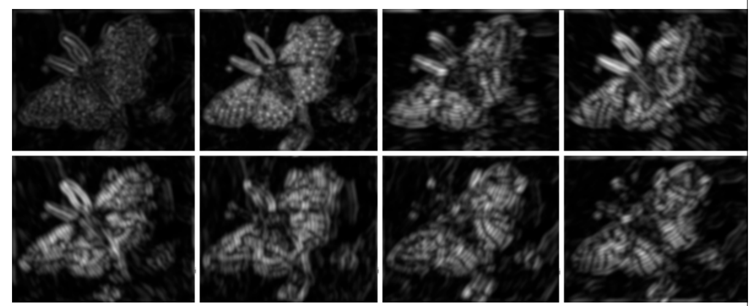


Figure 4. Left: Filter set f_i consisting of 2 phases (even and odd), 3 scales (spaced by half-octaves), and 6 orientations (equally spaced from 0 to π). The basic filter is a difference-of-Gaussian quadrature pair with $3:1$ elongation. Right: 4 scales of center-surround filters. Each filter is L_1-normalized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

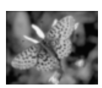
From Malik et al., "Contour and texture analysis for image segmentation"

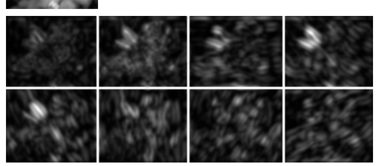
(We have an implementation for this filter bank, as part of the N-cuts software from Berkeley).

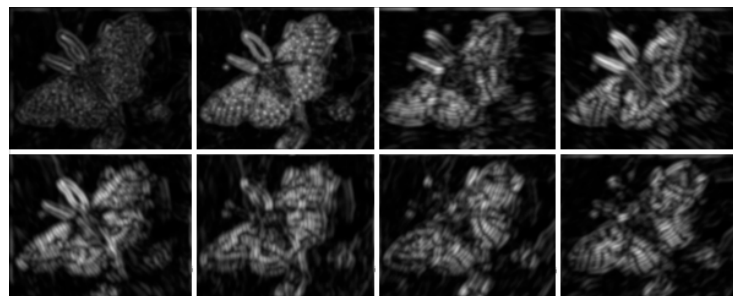




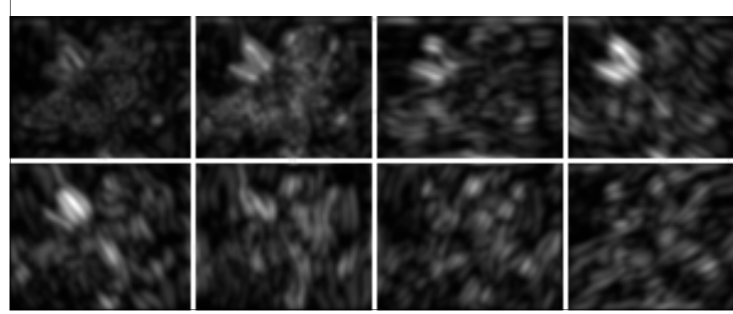
Next scale up







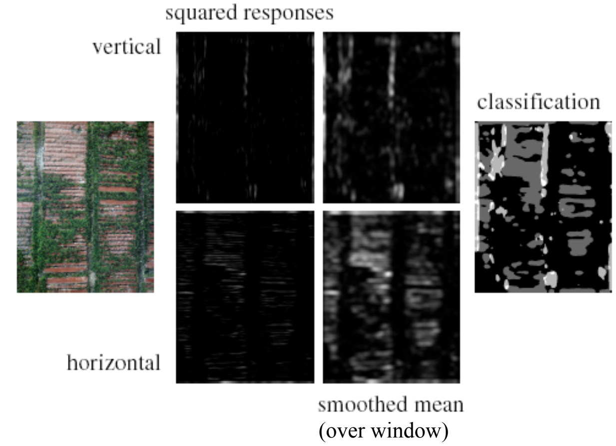
Smaller Scale



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

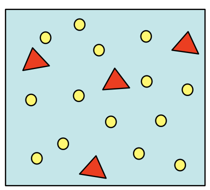
Representing textures

- 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



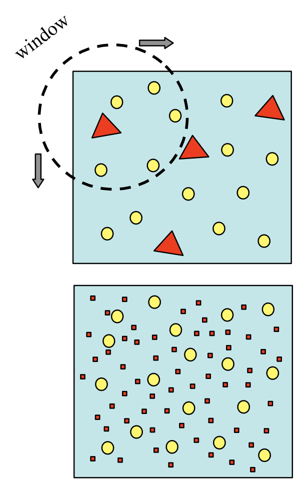
Textons

- Global statistics (e.g., mean of filter response magnitude) ignore spatial correlations
  - Some filters fire for the dots, other for the triangles.



- A complete representation of filter responses over windows will not work
  - expensive, represents textures that are never seen, does not exploit internal similarity in texture (e.g., multiple dots).
- One good solution is to **cluster** point data in "textons"
  - Texture in a window is a histogram of texton popularity.

Example



A variety of filter shapes and scales provide numbers that can distinguish these textures and many others.

However, simple statistics (e.g., filter response magnitude and variance) do not capture the spatial correlation well.

Commonly co-occurring filter responses can be represented by clusters (textons).

Textures can be represented by histograms over textons.

Texture (and color) segmentation

ex am

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)

ex am

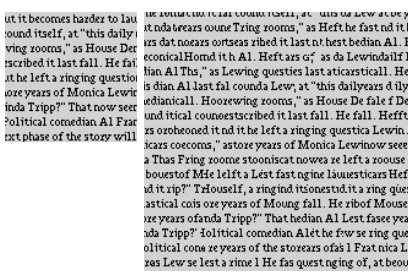


Figure from Texture Synthesis by Non-parametric Sampling, A. Efros and T.K. Leung, Proc. Int. Conf. Computer Vision, 1999 copyright 1999, IEEE