Syllabus Notes

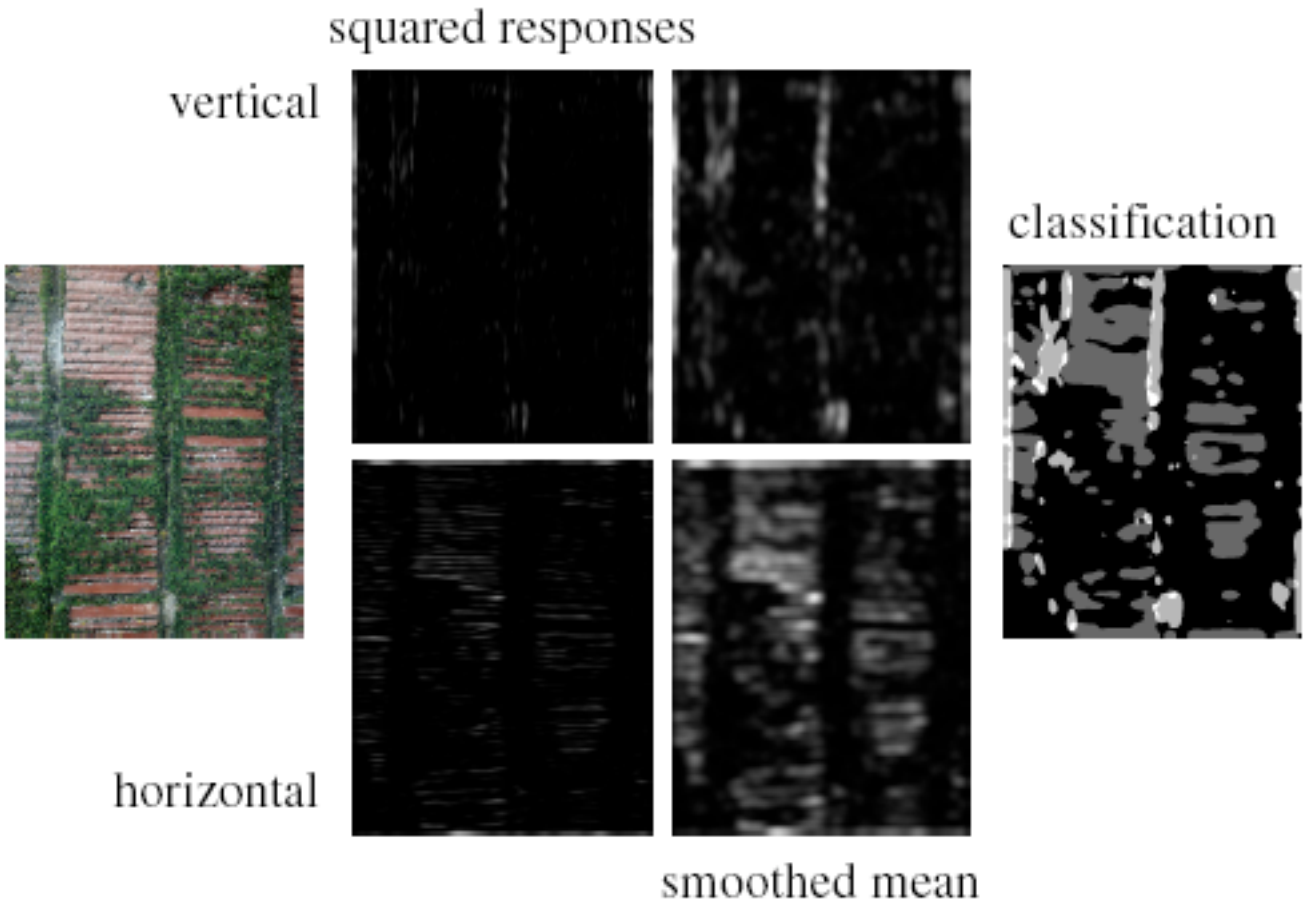
- We are finishing texture (New material from today's lecture will not surface in midterm).
- Recommended reading: Much of the texture chapter.

Representing textures (review)

- Collections of responses to a variety of filters
- 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.

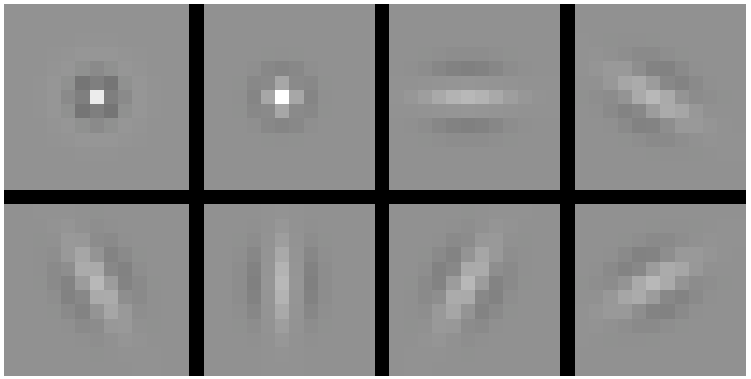
Representing textures (review)

- Associate texture with statistics of the conglomerate of responses over some scale (window size)
- Simplest statistic is mean (square) response for each filter
- Including standard deviation helps
- More sophisticated approaches include looking at histogram of responses over window (can often use fewer filters in this case)

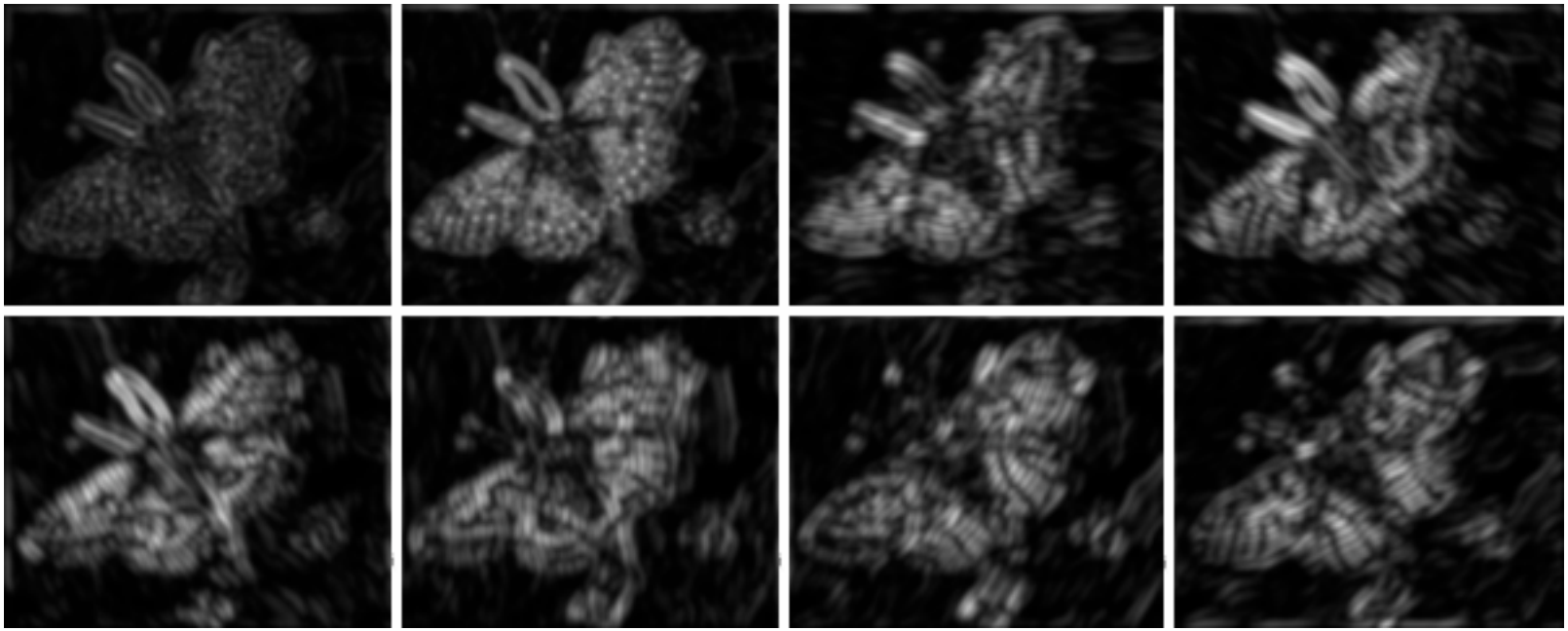


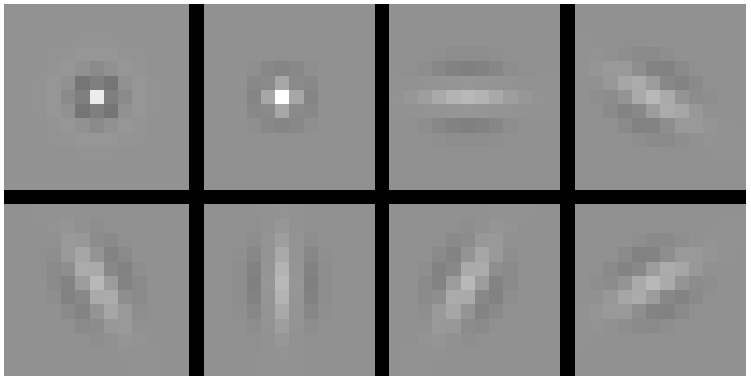
(scale implies a window)

Very simplified example (two filters)

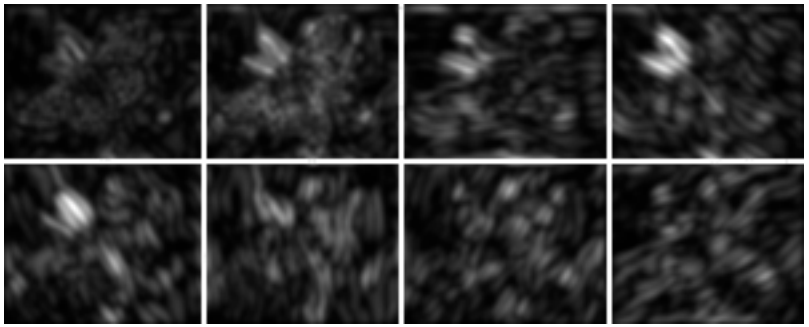


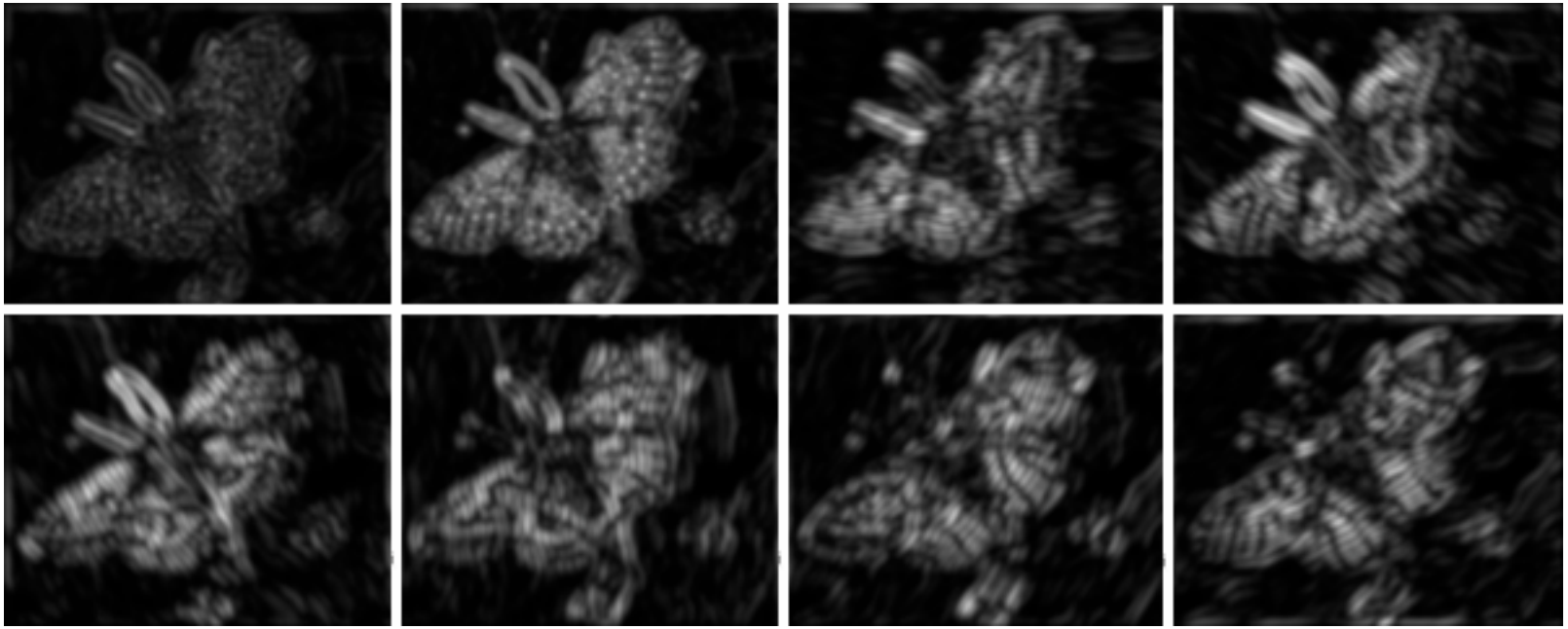
Spots and oriented bars at one scale (see page 191 for details of the construction of these filters).

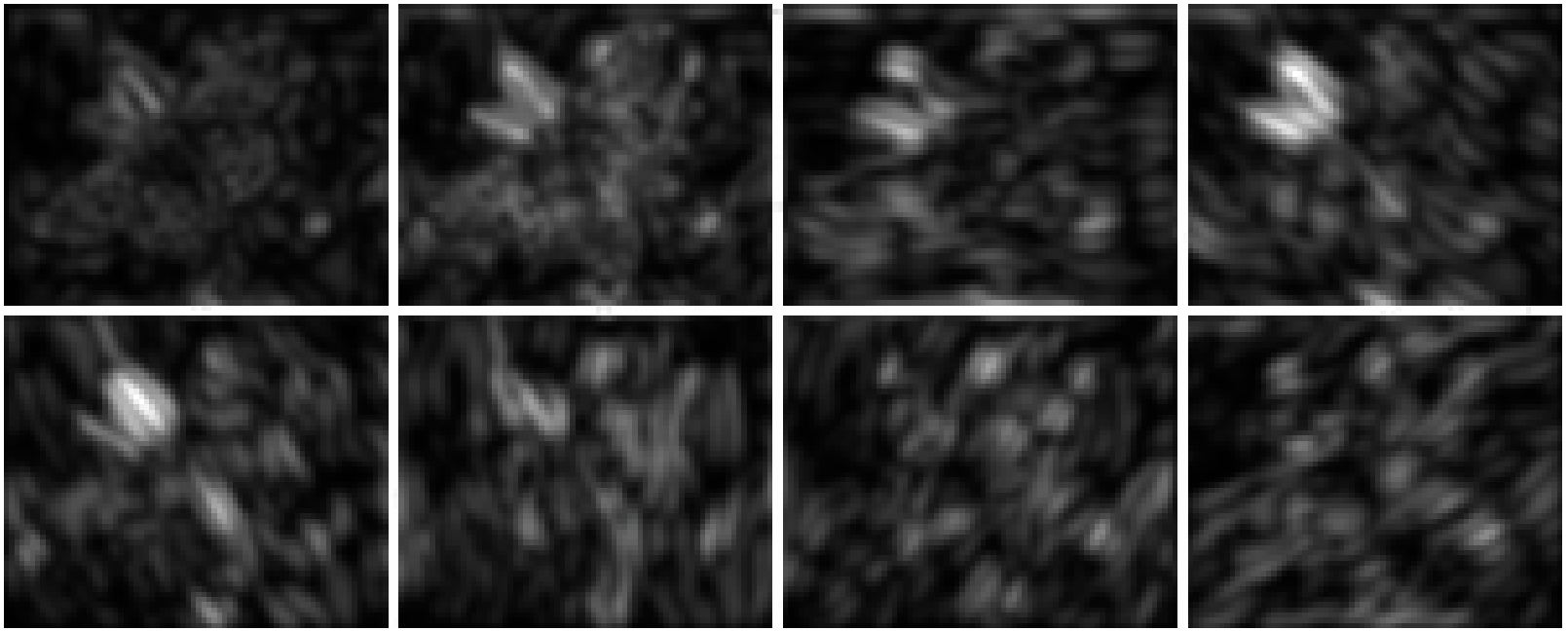




Next bigger scale







A typical filter bank

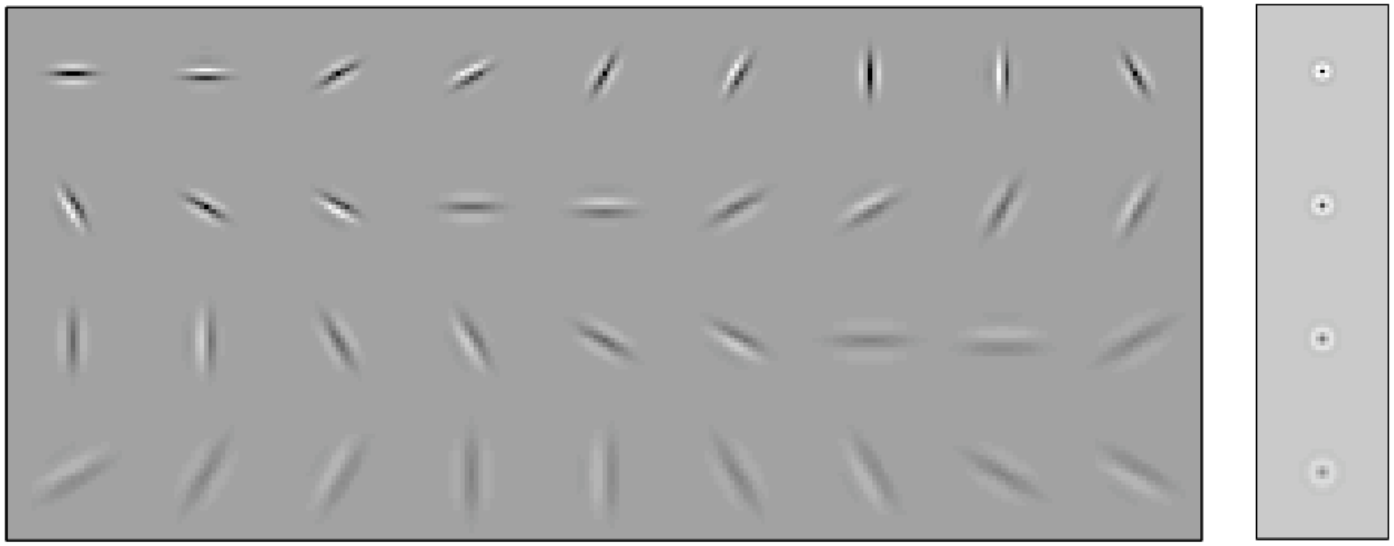


Figure 4. Left: Filter set f_i consisting of 2 phases (even and odd), 3 scales (spaced by half-octaves), and 6 orientations (equally spaced from 0 to π). The basic filter is a difference-of-Gaussian quadrature pair with 3 : 1 elongation. Right: 4 scales of center-surround filters. Each filter is L_1-normalized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

The Gaussian (scale) pyramid

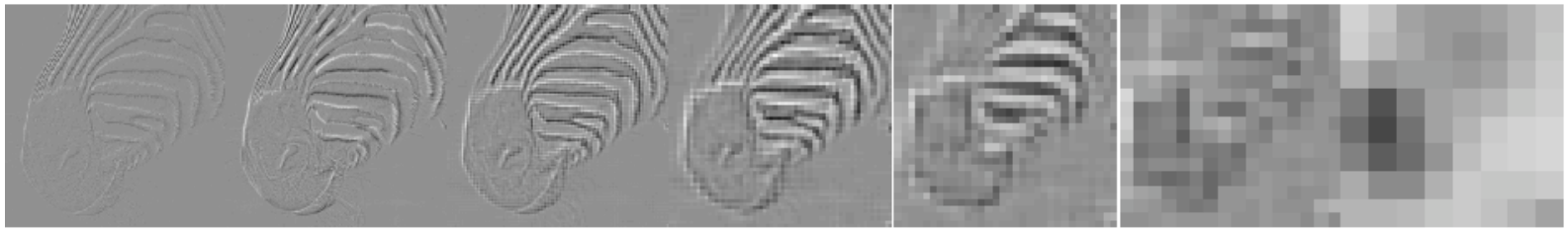
- Smooth with gaussians, because
  - a gaussian*gaussian=another gaussian
- Representation is redundant
- Allows easy analysis across scales

512 256 128 64 32 16 8

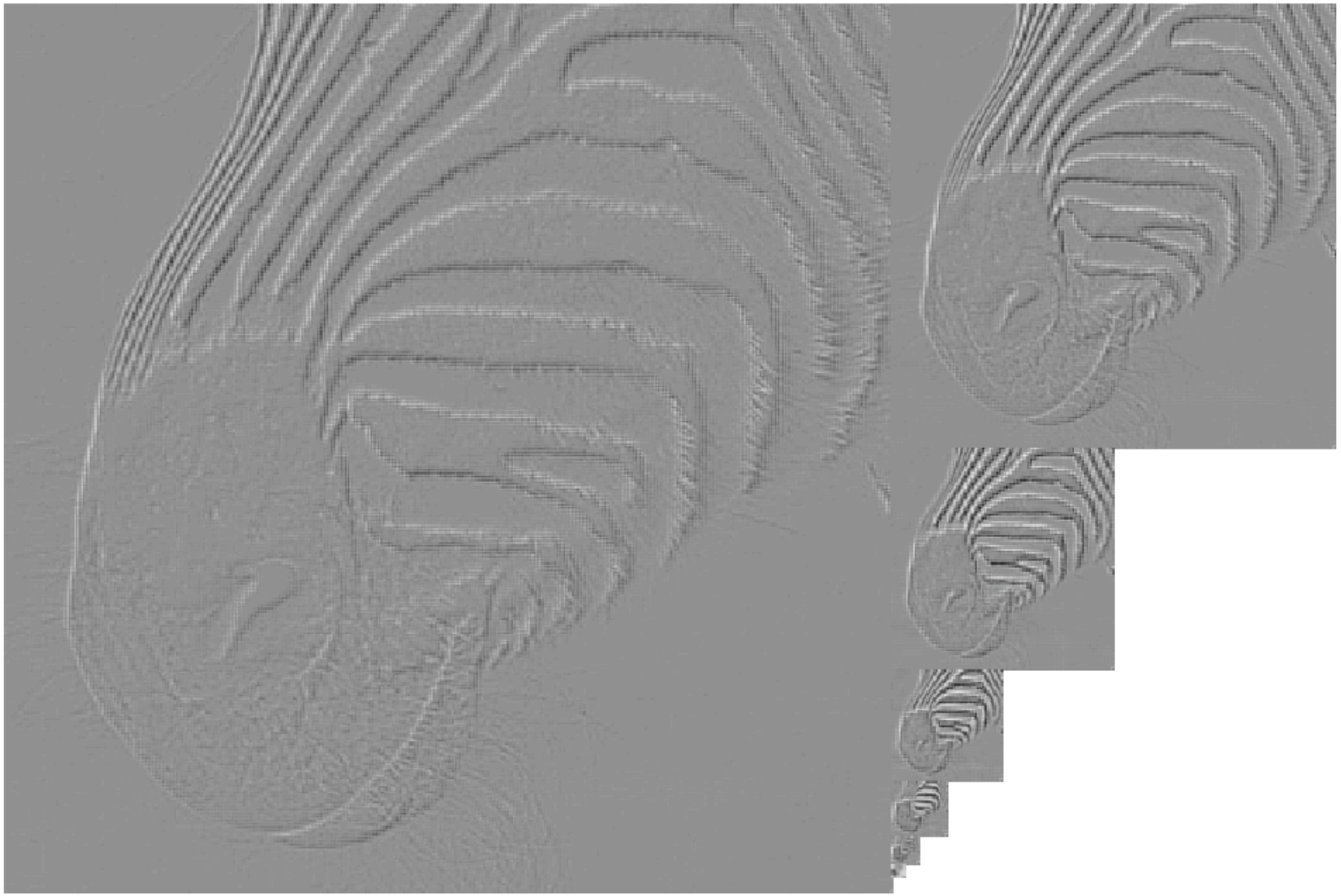
The Laplacian Pyramid (§9.2.1)

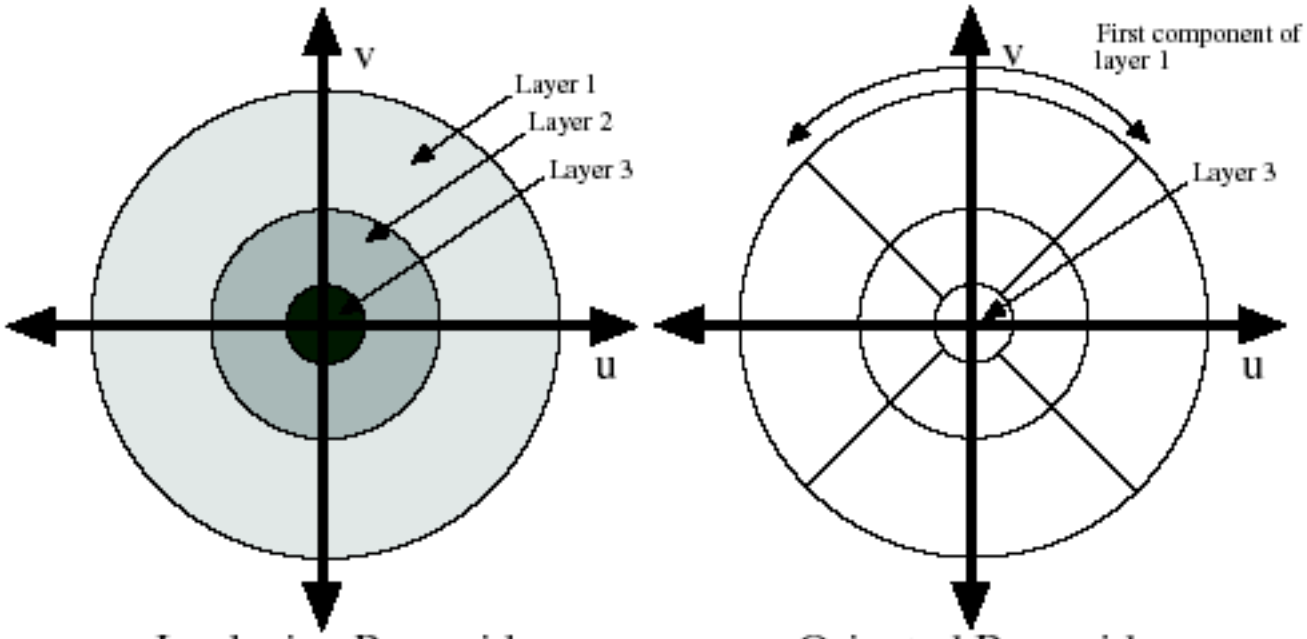
- Recall that the Laplacian of Gaussian (LOG) is approximated by difference of Gaussians (DOG)
- To recover the Gaussian pyramid we can “add up” the DOG pyramid
- In particular, we add lower layers to “up-sampled” versions of the current layer.
- Up sampling means we just duplicate the value
- Down sampling means we select a value from a block

512 256 128 64 32 16 8



512 256 128 64 32 16 8





Laplacian Pyramid

(Spot detectors)

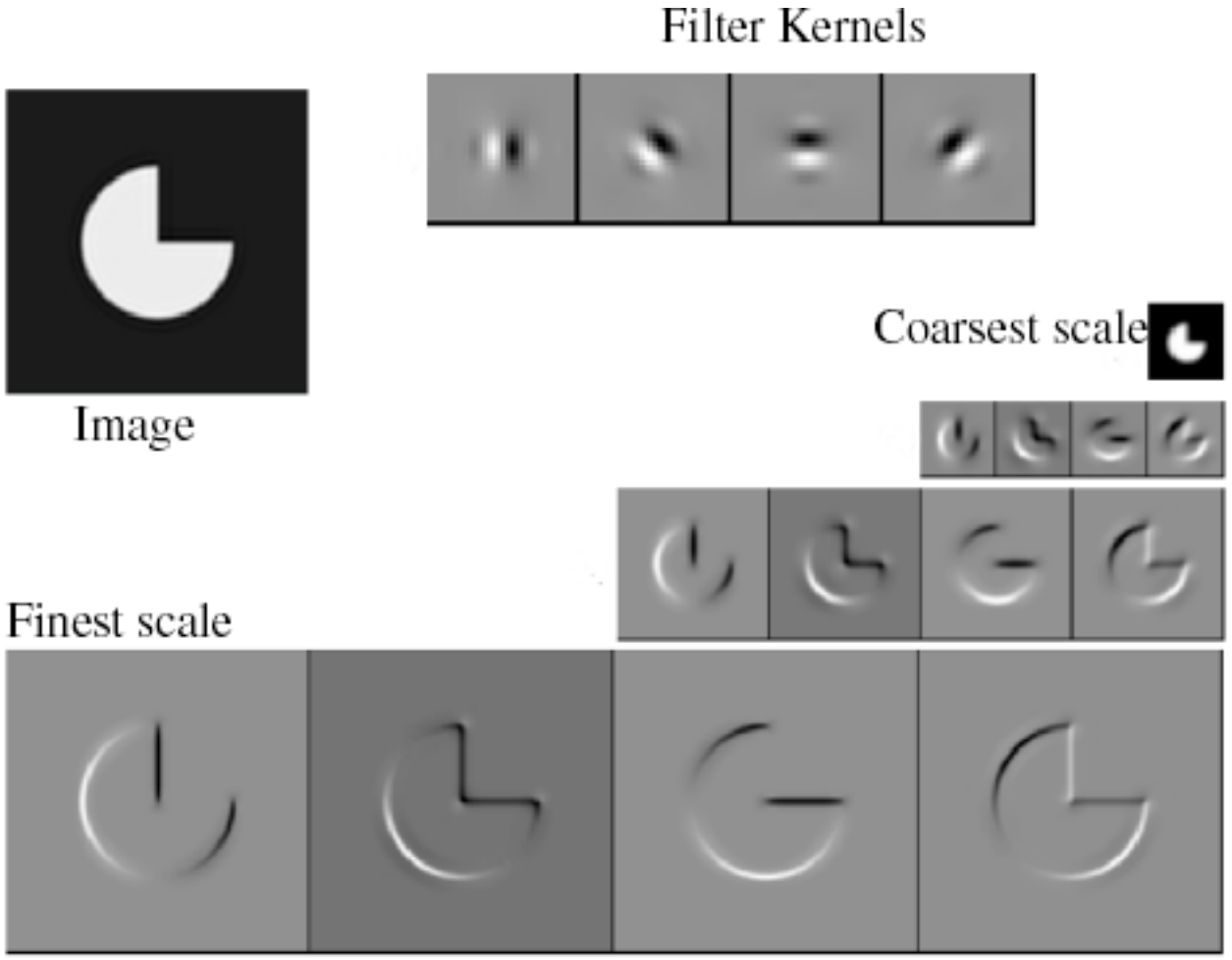
Oriented Pyramid

(Oriented bar detectors)

(Note that plots are in the frequency domain)

Oriented pyramids

- Laplacian pyramid is orientation independent
- Apply an oriented filter to determine orientations at each layer
  - by clever filter design, we can simplify synthesis
  - this represents image information at a particular scale and orientation

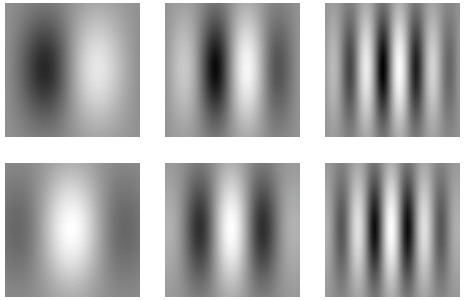


Reprinted from “Shiftable MultiScale Transforms,” by Simoncelli et al., IEEE Transactions on Information Theory, 1992, copyright 1992, IEEE

Yet another way to create filters

Anti-symmetric

Symmetric

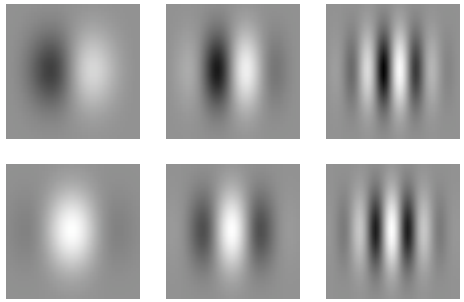


Gabor filters at different scales and spatial frequencies

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

Anti-symmetric

Symmetric



Easy to make oriented versions

Bottom six are more localized (smaller Gaussian sigma)

Final texture representation

- Form an oriented pyramid (or equivalent set of responses to filters at different scales and orientations).
- Square the output
- Take statistics of responses
  - simplest is mean of each filter output (are there lots of spots?)
  - next most convenient enhancement is to look at standard deviation of each filter output
  - more complicates schemes are important in practice

Texture synthesis

- Use image as a source of probability model
- Choose pixel values by matching neighbourhood, then filling in
- Matching process
  - look at pixel differences
  - count only synthesized pixels

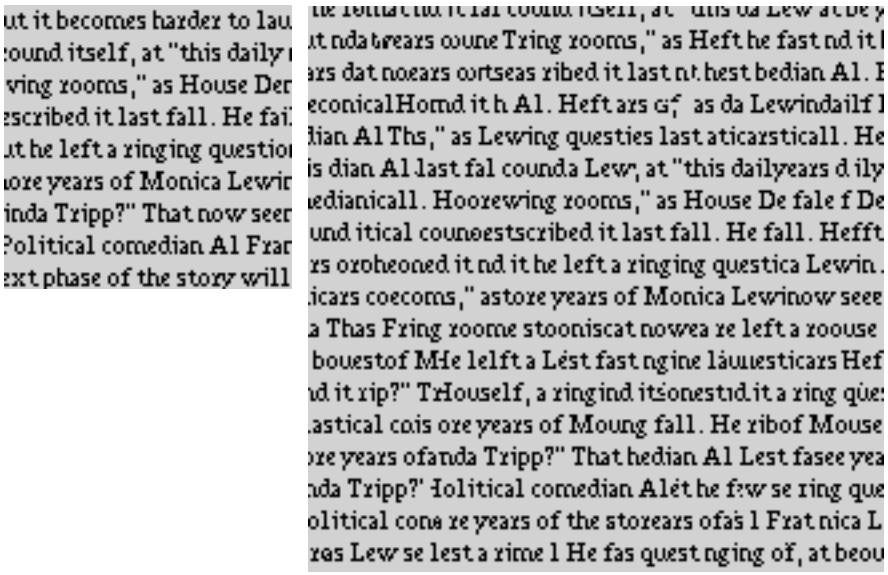


Figure from Texture Synthesis by Non-parametric Sampling, A. Efros and T.K. Leung, Proc. Int. Conf. Computer Vision, 1999 copyright 1999, IEEE