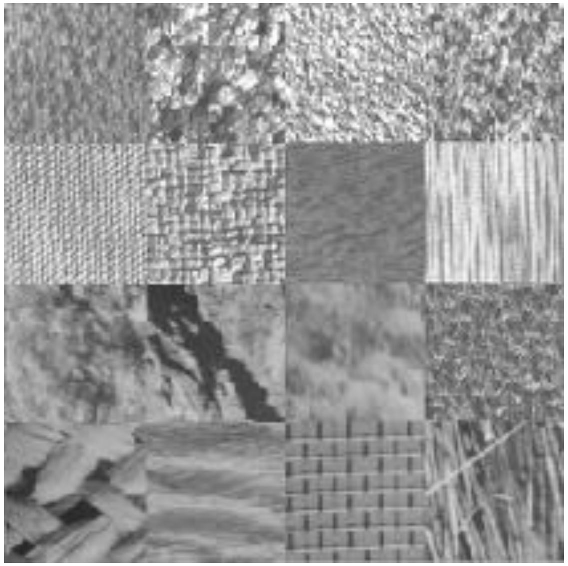
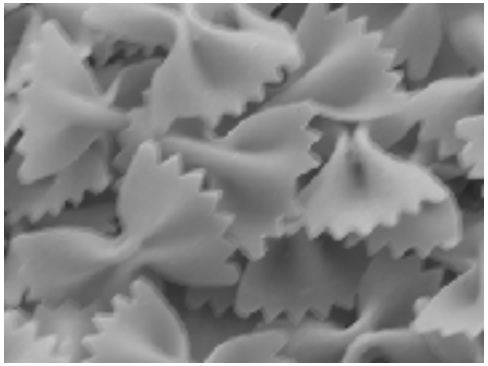
Texture

Some slides: courtesy of David Jacobs

Examples



Simple textures (material classification): Brodatz dataset



More general textures on unstructured scenes

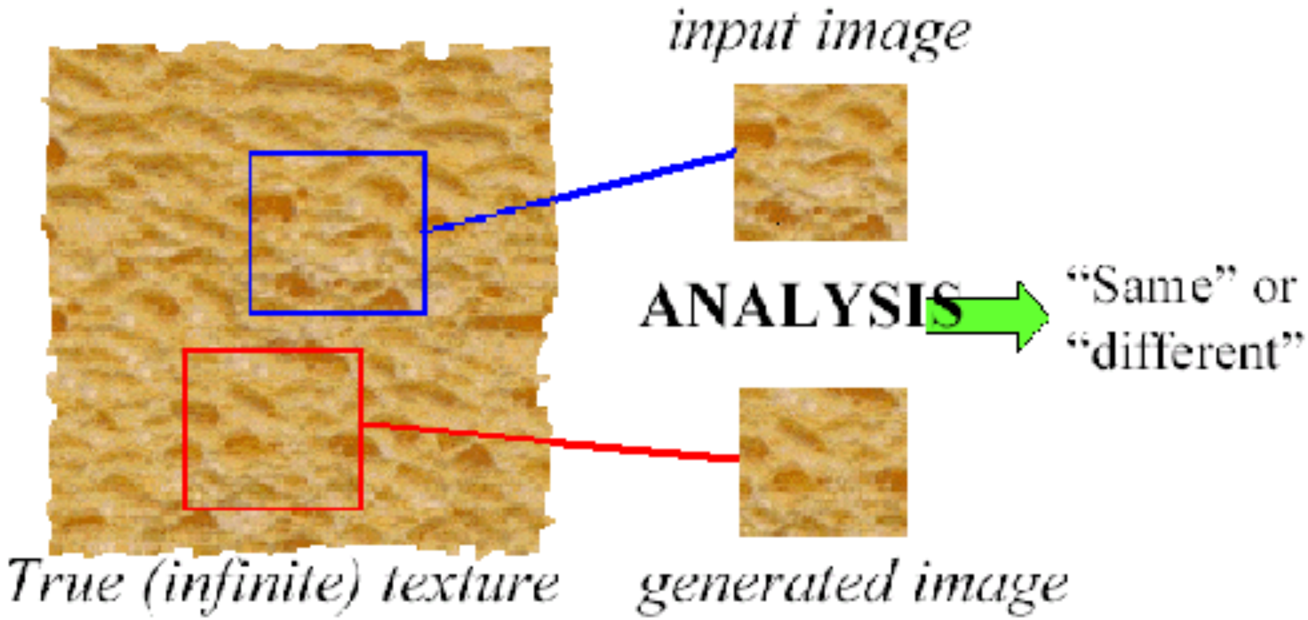
Dynamic textures

Applications

- Classification (Objects and Scene, Material)
- Segmentation: group regions with same texture
- Shape from texture: estimate surface orientation or shape from texture
- Synthesis (Graphics) : generate new texture patch given some examples

Issues: 1) Discrimination/Analysis

The Goal of Texture Analysis

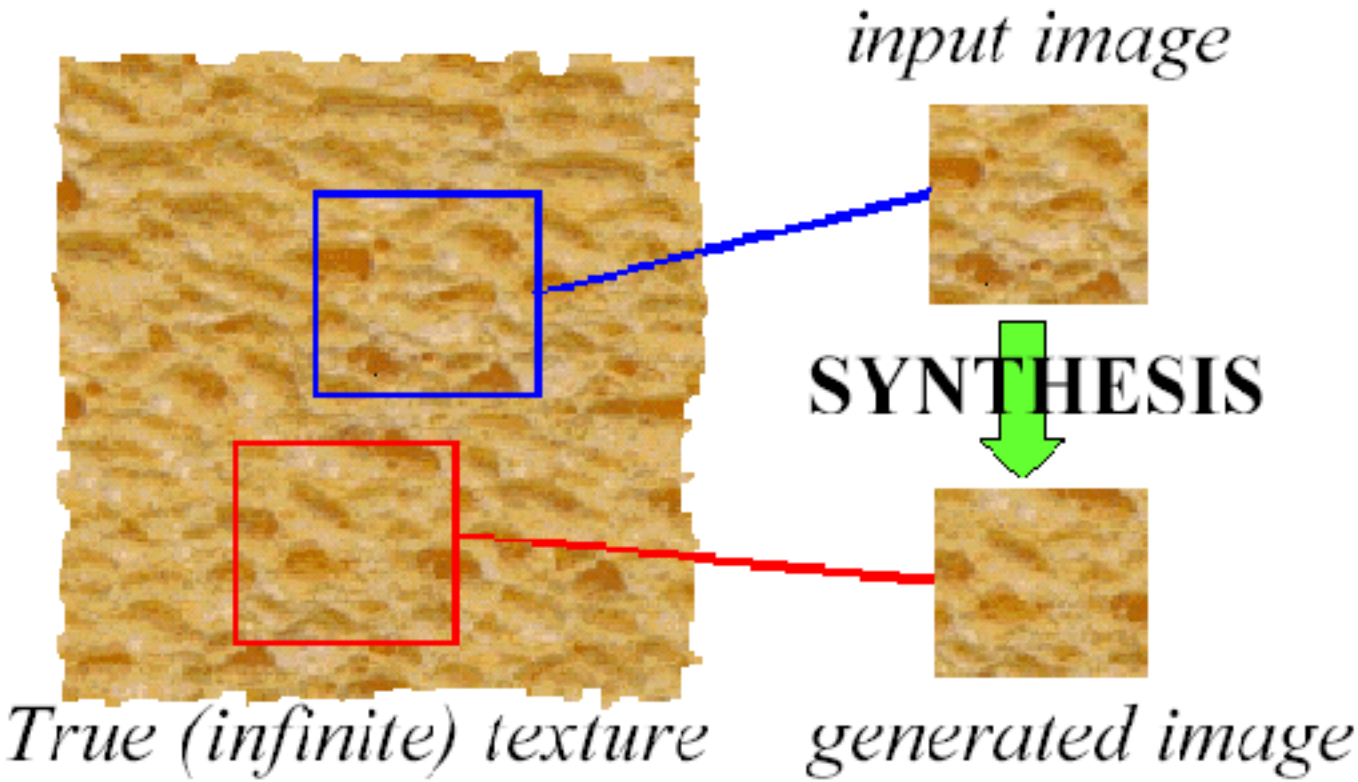


Compare textures and decide if they're made of the same "stuff".

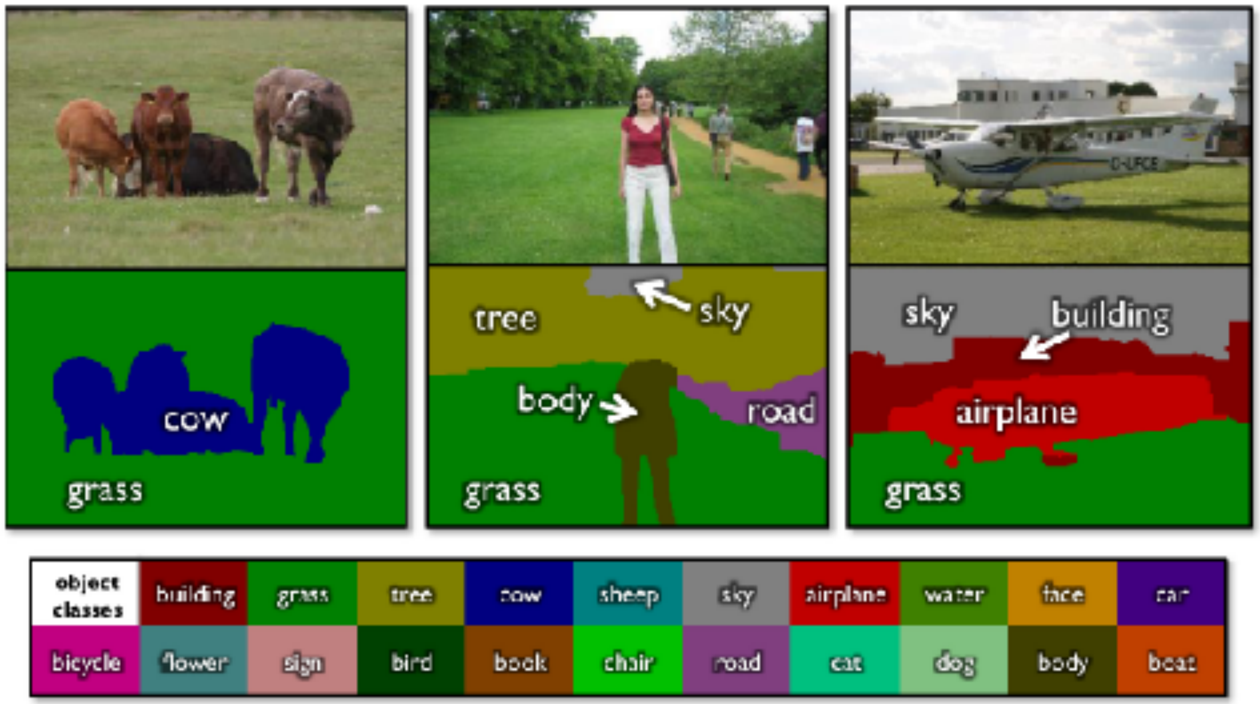
(Freeman)

2) Synthesis

The Goal of Texture Synthesis

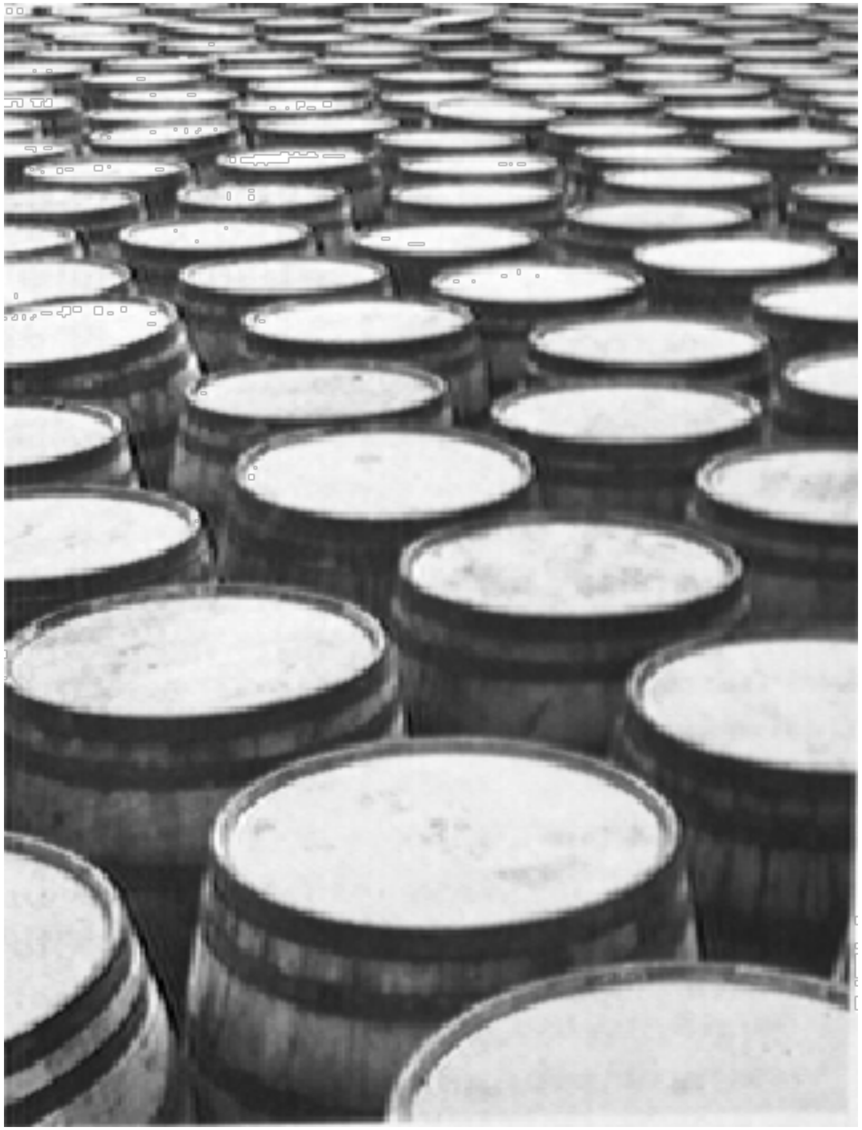


Texture for Scene classification



TextonBoost (Shotton et al. 2009 IJCV)

Texture provides shape information



Gradient in the spacing of the barrels (Gibson 1957)

Texture gradient associated to converging lines (Gibson1957)

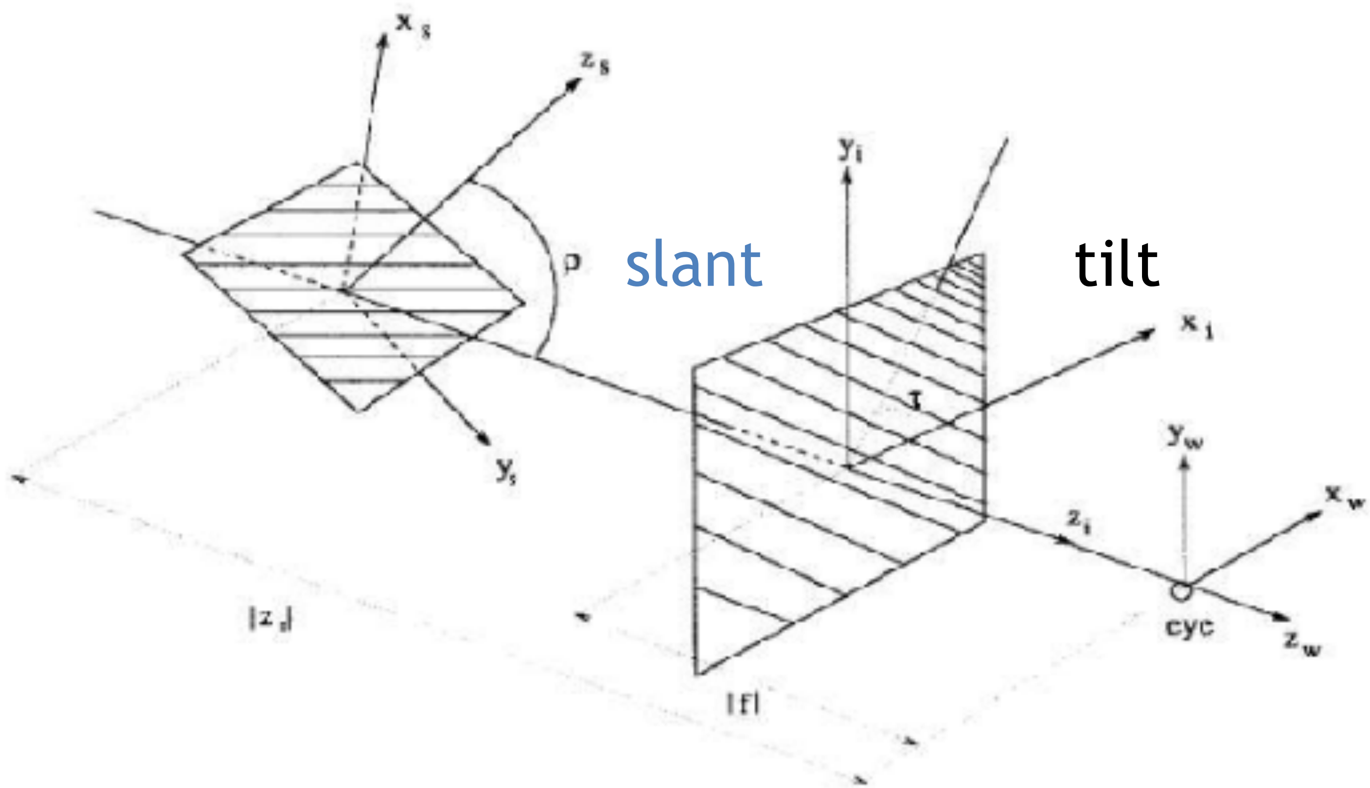
Shape from texture

- Classic formulation: Given a single image of a textured surface, estimate the shape of the observed surface from the distortion of the texture created by the imaging process
- Approaches estimate plane parameters,

  Require restrictive assumptions on the texture

❖ isotropy (same distribution in every direction) and orthographic projection

❖ Homogeneity (every texture window is same)



slant
tilt

Texture description for recognition

Two main issues

- **1. Local description:** extracting image structure with filters (blobs, edges, Gabors, or keypoint descriptors) ; different scales
- **2. Global description:**

  - statistical models (histograms or higher order statistics, MRF)

  - Some models (networks, ant systems, etc.)

Why computing texture?

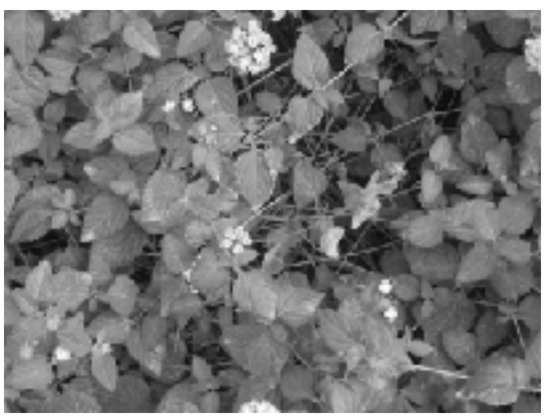
- Indicative of material property -> attribute description of objects
- Good descriptor for scene elements
- For boundary classification: need to distinguish between object boundary and texture edges

Overview

- Local: Filter selection and scale selection for local descriptors
- Global: Statistical description: histogram
- MFS (multifractal spectrum): invariant to geometric and illumination changes
- Edge classification using texture and applying it shadow detection

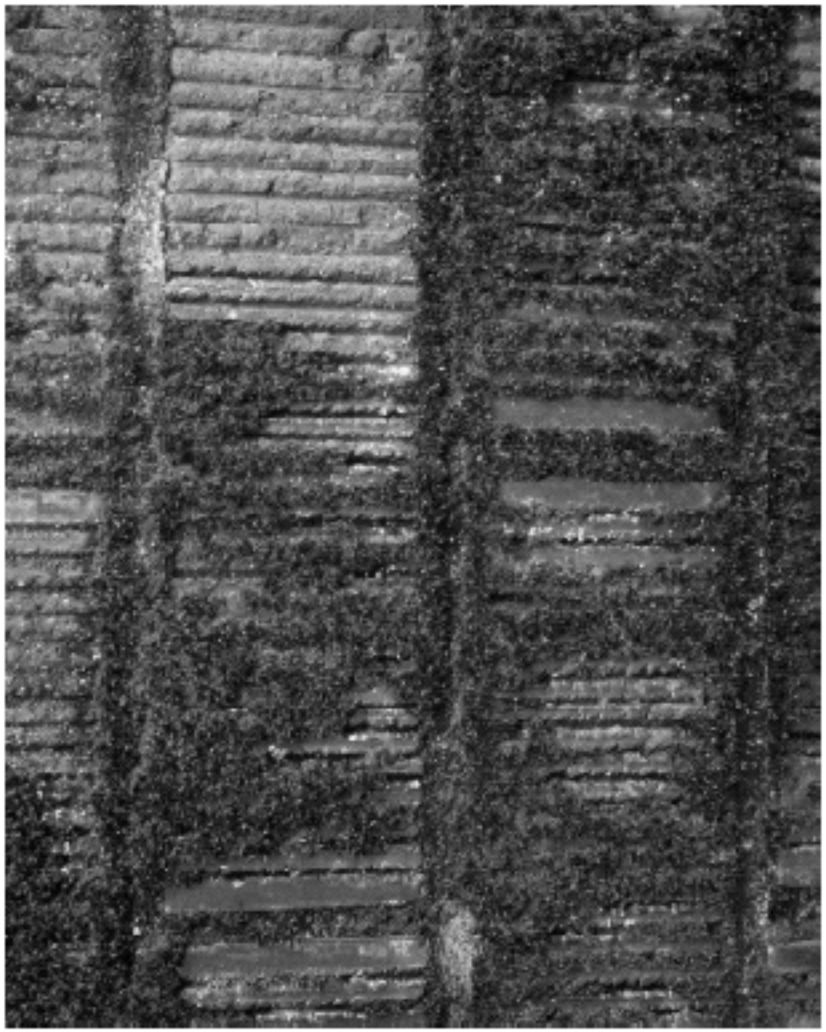
Local descriptors: motivation

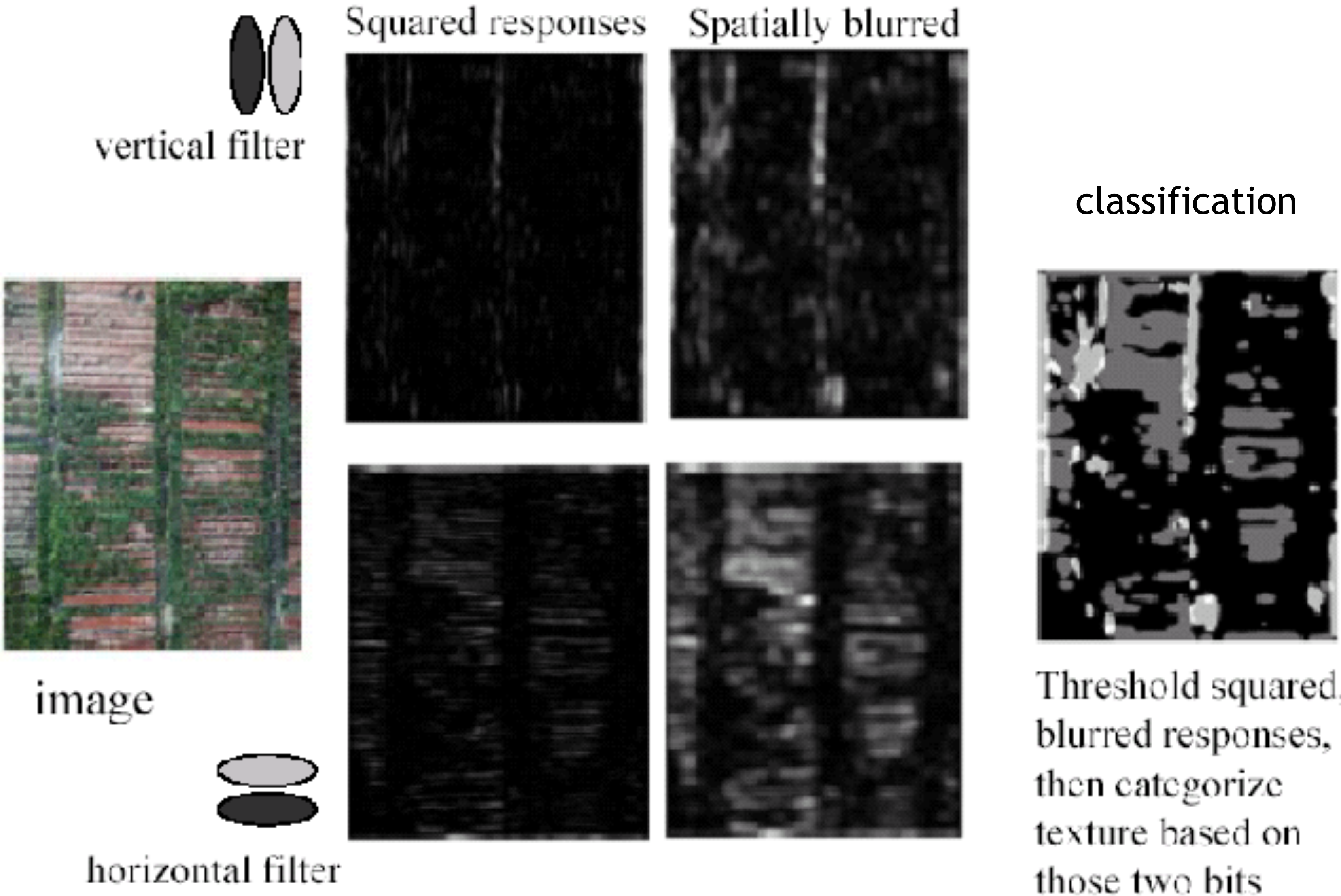
Ideally we think of texture as consisting of texture elements (**Textons**



Since in real images there are no canonical elements, we apply filter that pick up “**blobs**” and “**bars**”

Example (Forsyth & Ponce)




Squared responses
Spatially blurred
vertical filter
classification
image
horizontal filter
Threshold squared, blurred responses, then categorize texture based on those two bits

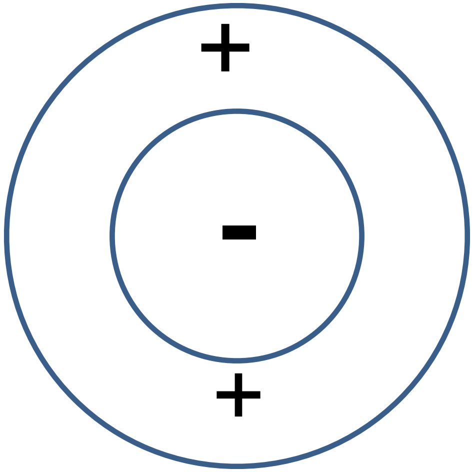
What are Right Filters?

- The more independent the better.
  - In an image, output of one filter should be independent of others.
  - Because our comparison assumes independence.
  - Wavelets seem to be best.

Blob detector

- A filter at multiple scales.
- The biggest response should be when the filter has the same location and scale as the blob.

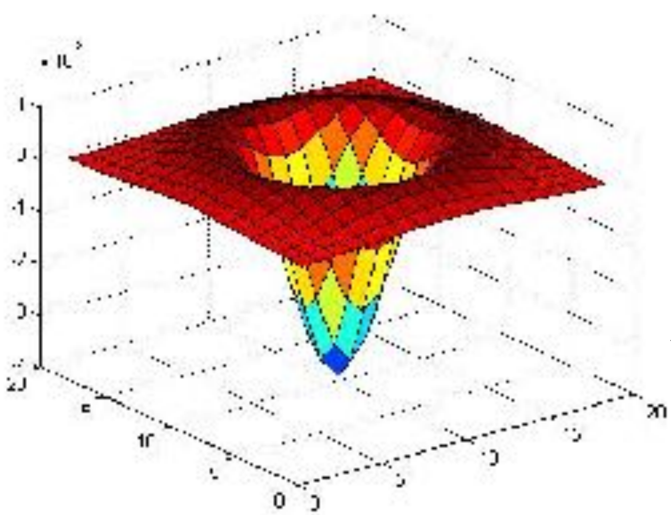
Center Surround filter

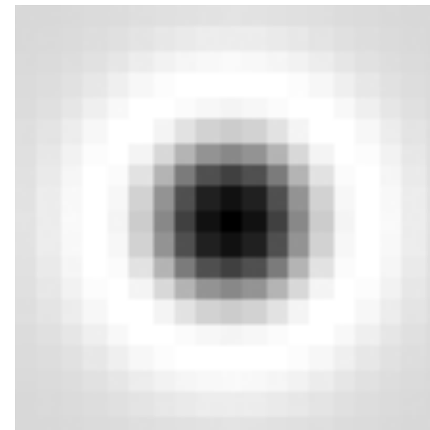


- When does this have biggest response?
- When inside is dark
- And outside is light
- Similar filters are in humans and animals

Blob filter

- Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D





$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

Need to scale-normalize: $$\nabla^2_{\text{norm}} g = \sigma^2 \left(\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$$

Efficient implementation

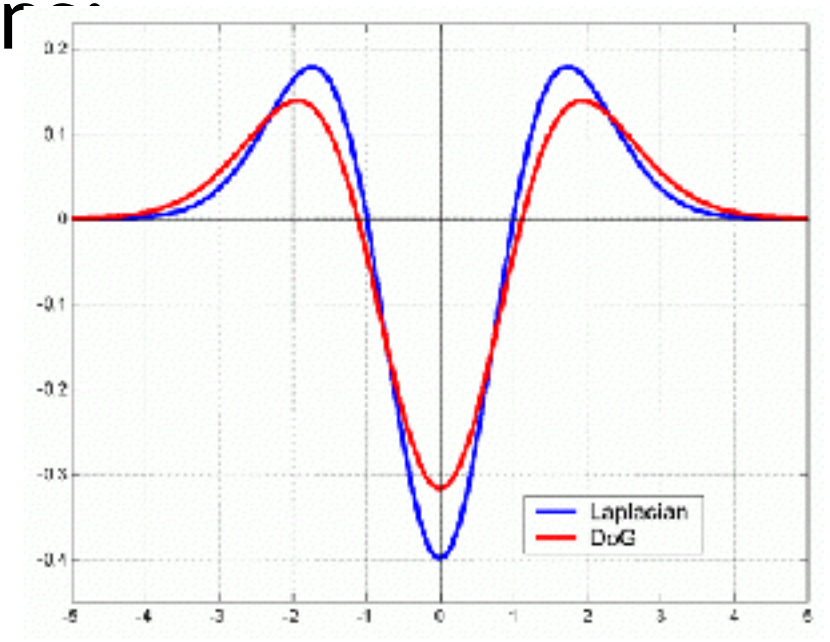
- Approximating the Laplacian with a difference of Gaussians:

$$L = \sigma^2 \left(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)$$

(Laplacian)

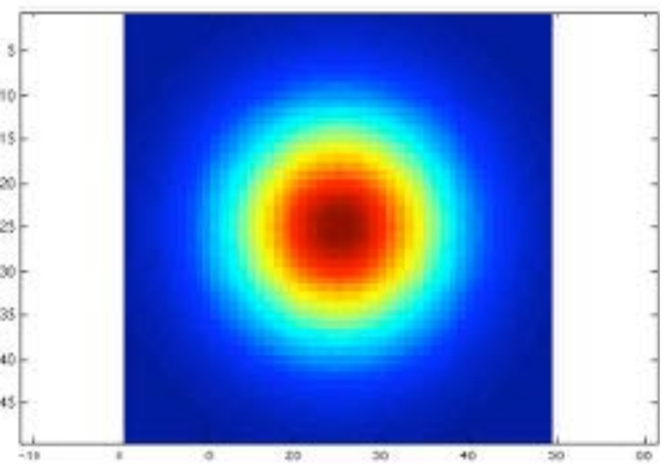
$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)

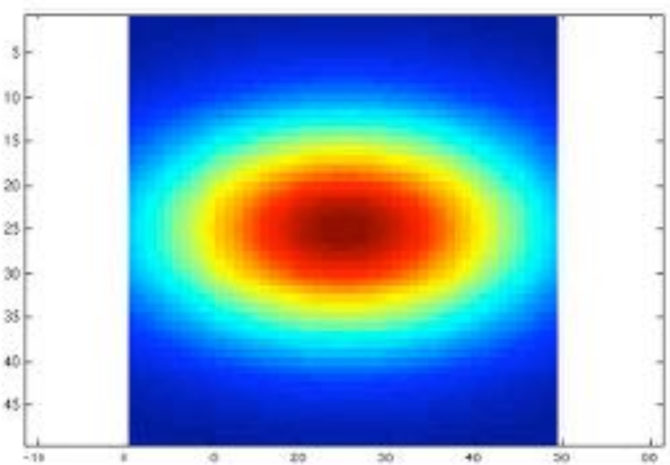


Multivariate Gaussian

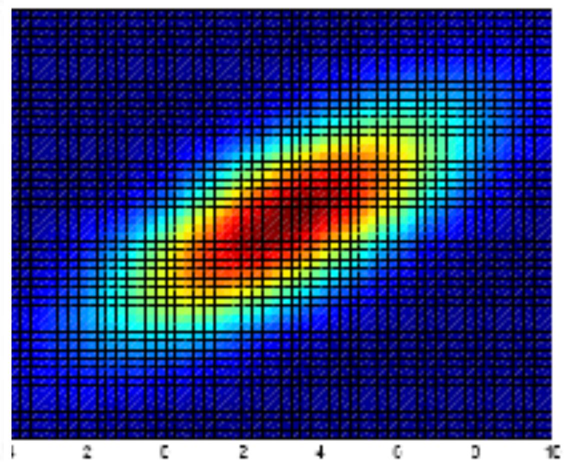
$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right).$$



$$\Sigma = \begin{bmatrix} 9 & 0 \\ 0 & 9 \end{bmatrix}$$

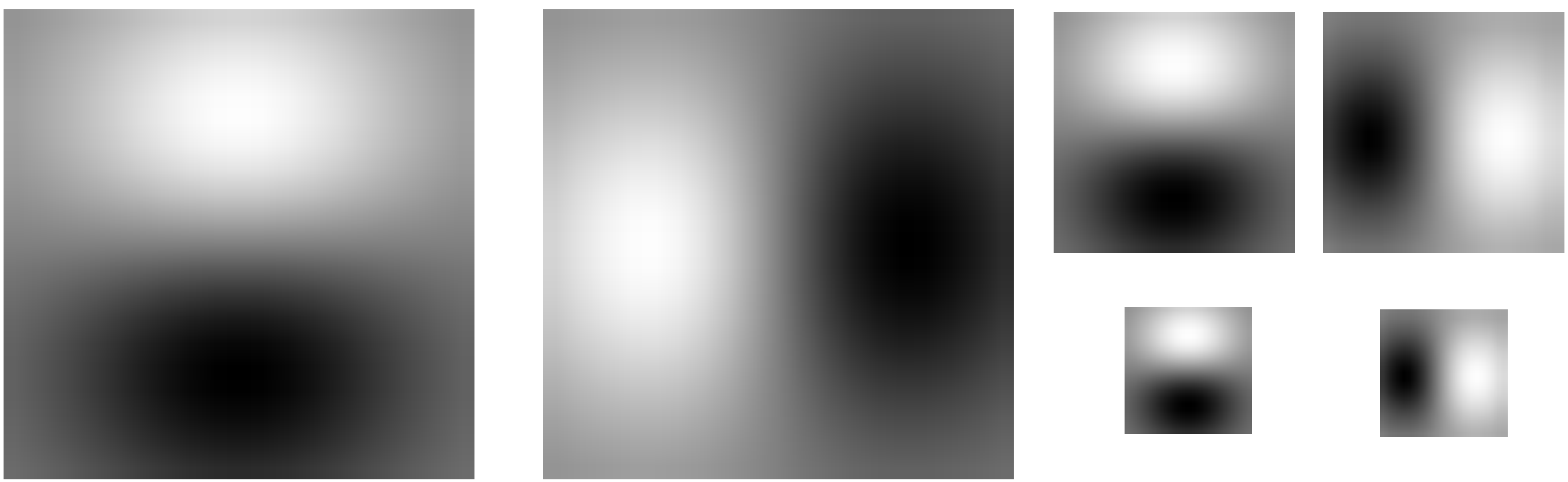


$$\Sigma = \begin{bmatrix} 16 & 0 \\ 0 & 9 \end{bmatrix}$$

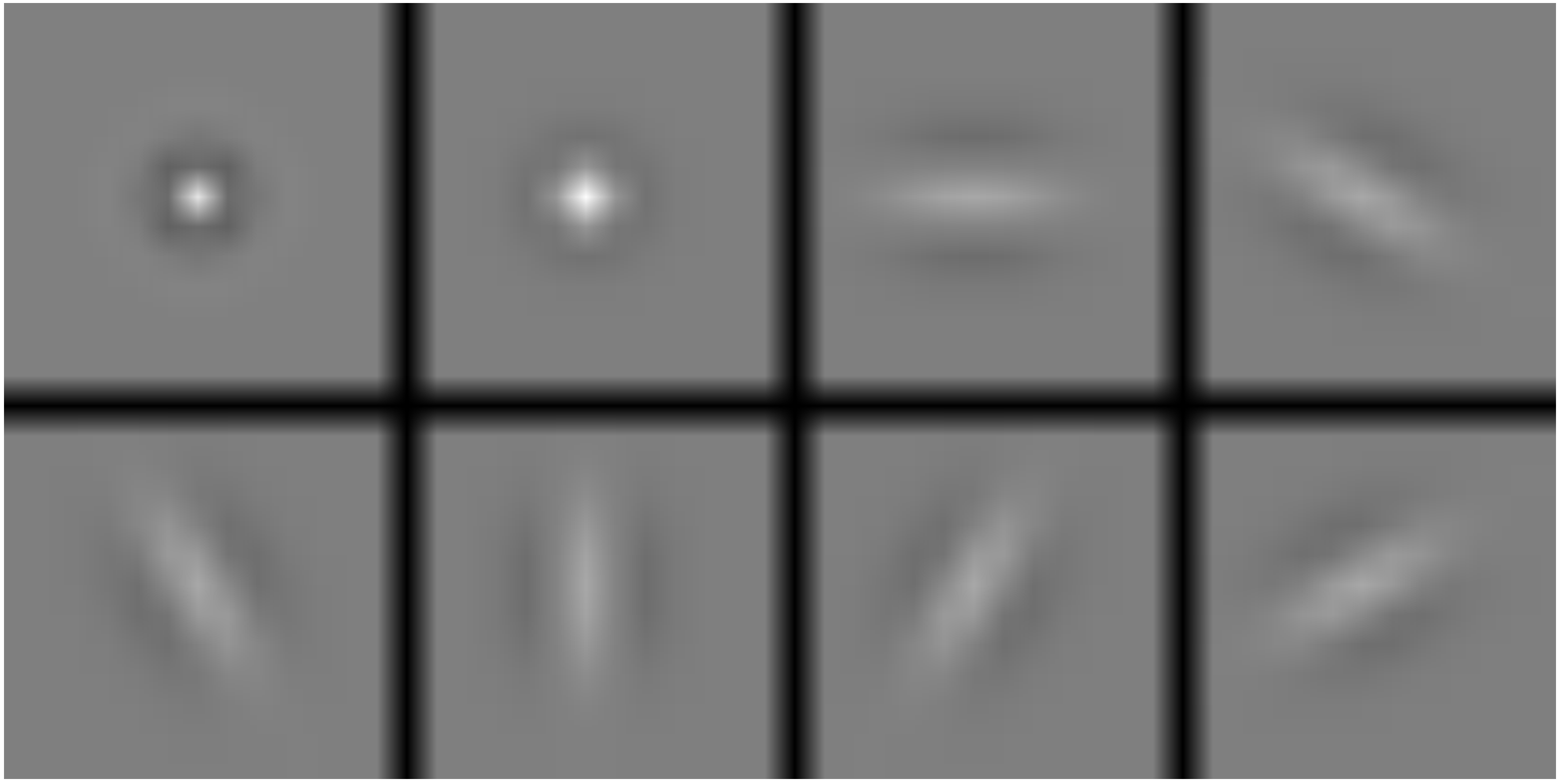


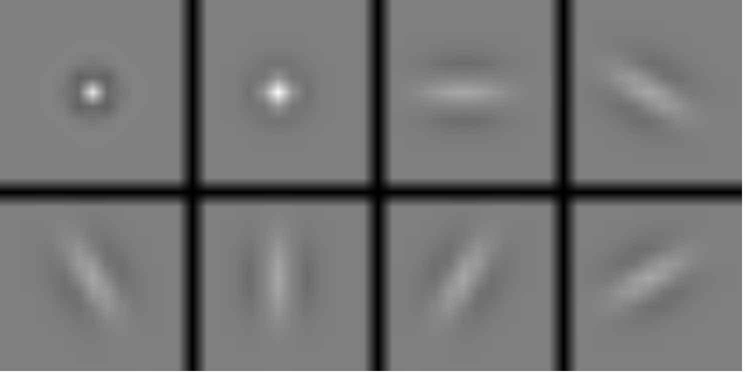
$$\Sigma = \begin{bmatrix} 10 & 5 \\ 5 & 5 \end{bmatrix}$$

Difference of Gaussian Filters

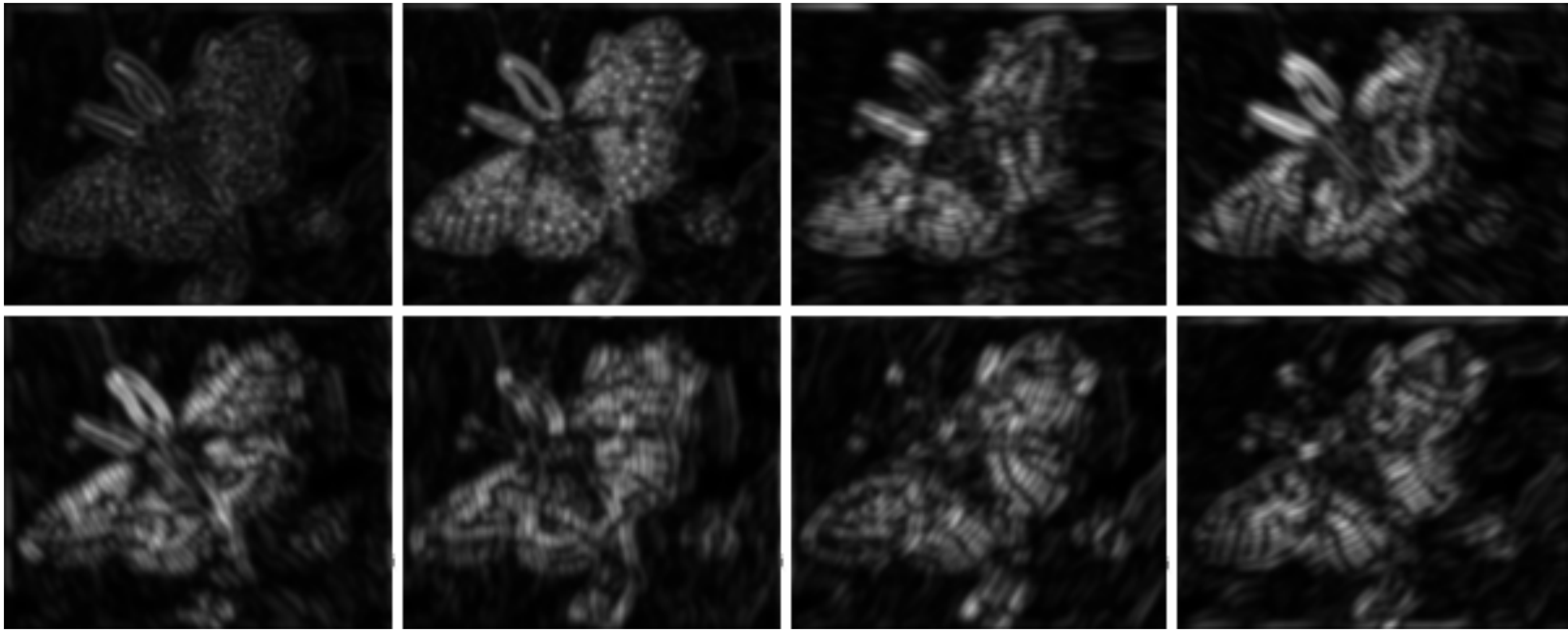


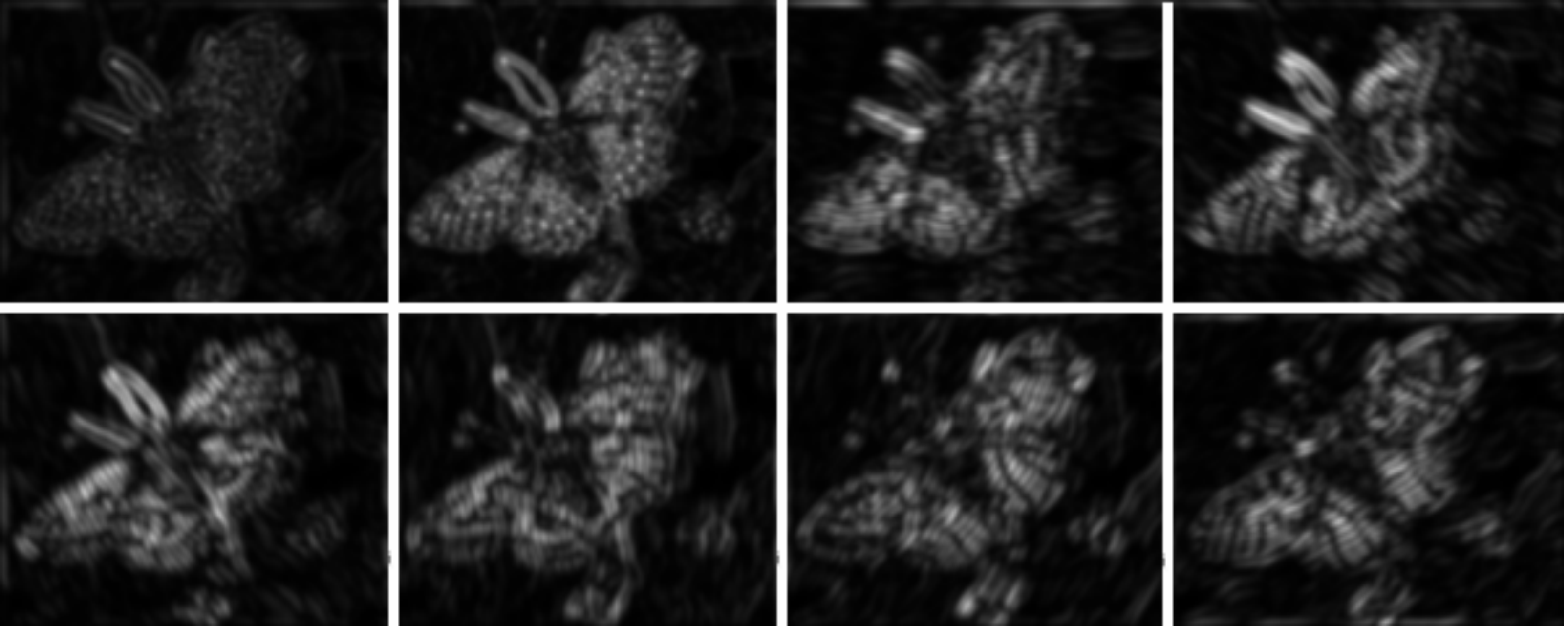
Spots and Oriented Bars (Malik and Perona)



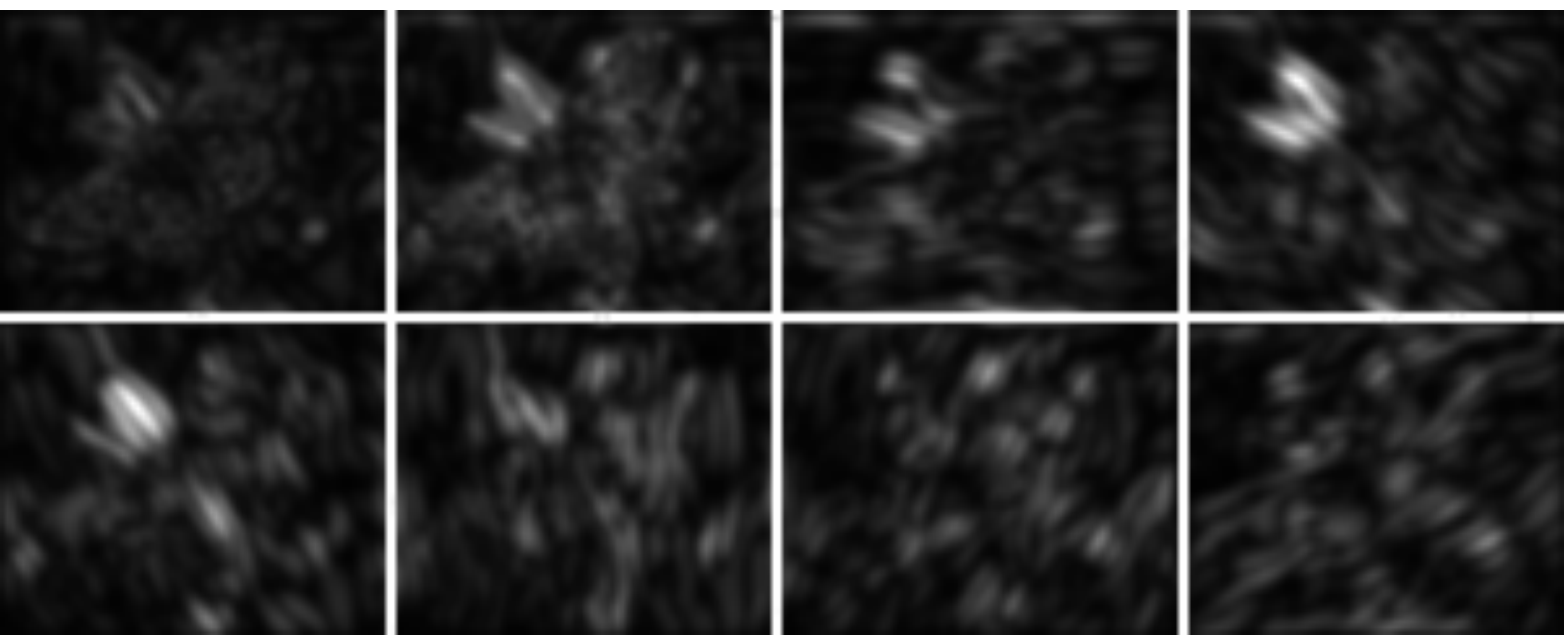


Applying these eight filters to the butterfly im





At fine scale

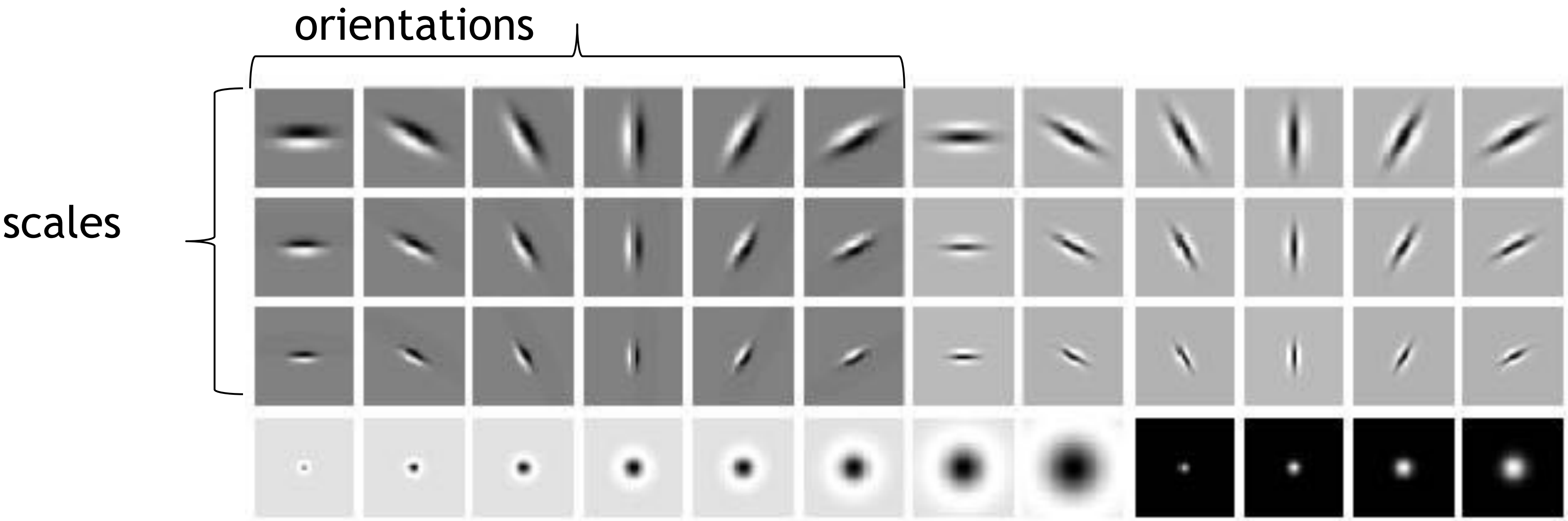


At coarser scal

Filter banks

- We apply a collection of multiple filters: a **filter bank**
- The responses to the filters are collected in **feature vectors,** which are multi-dimensional.
  - We can think of nearness, farness in feature space

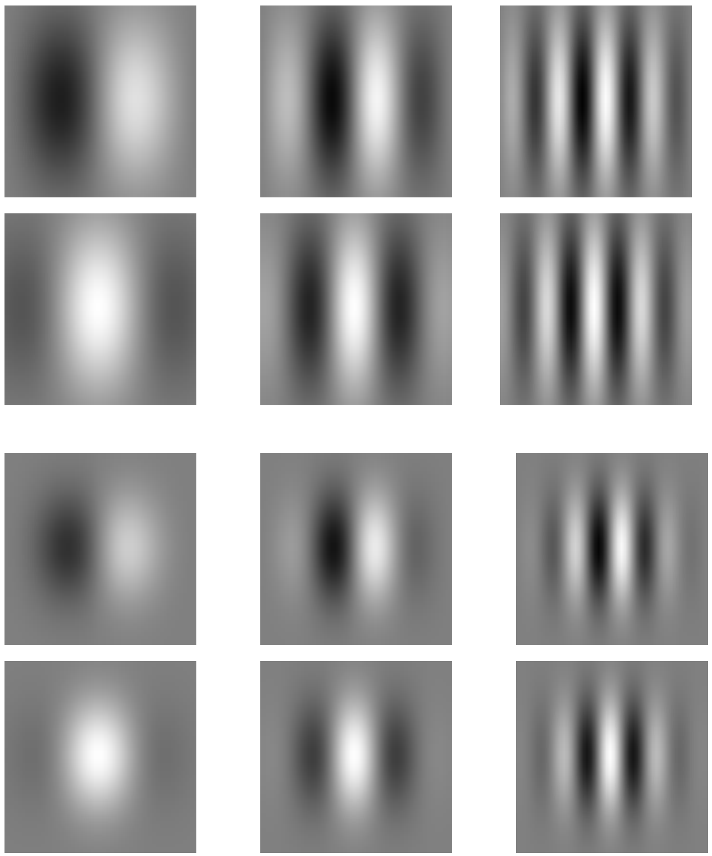
Filter banks



Leung Malik filterbank: 48 filters: **2 Gaussian derivative filters at 6 orienta and 3 scales, 8 Laplacian of Gaussian filters and 4 Gaussian filters.**

- What filters to put in the bank?
  - Typically we want a combination of scales and orientations, different types of patterns.

Gabor Filters



Gabor filters at different scales and spatial frequencies

top row shows anti-symmetric (or odd) filters, bottom row the symmetric (or even) filters.

$$\text{symmetric: } \cos(k_x x + k_y y)\exp - \left\{\frac{x^2 + y^2}{2\sigma^2}\right\}$$

$$\text{antisymmetric: } \sin(k_x x + k_y y)\exp - \left\{\frac{x^2 + y^2}{2\sigma^2}\right\}$$

Gabor filters are examples of Wavelets

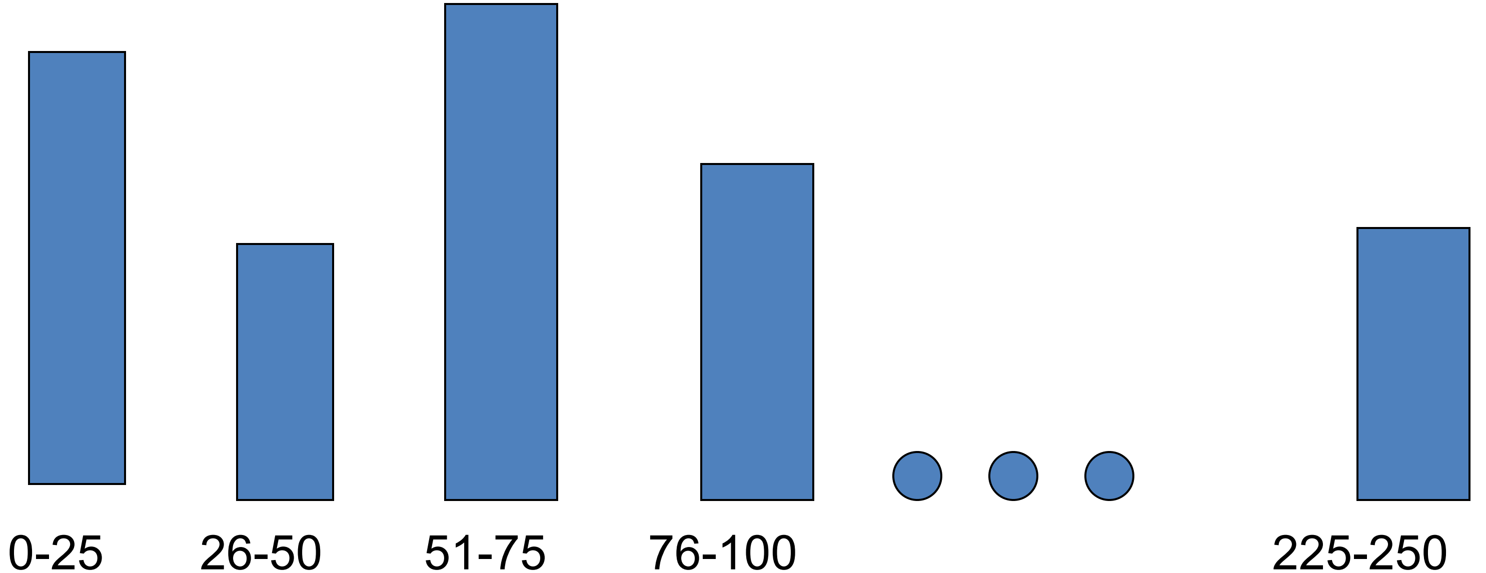
- We know two bases for images:
  - Pixels are localized in space.
  - Fourier are localized in frequency.
- Wavelets are a little of both.
- Good for measuring frequency locally.

Global: descriptions

- Simplest histograms

Global description: Simplest Texture Discrimination

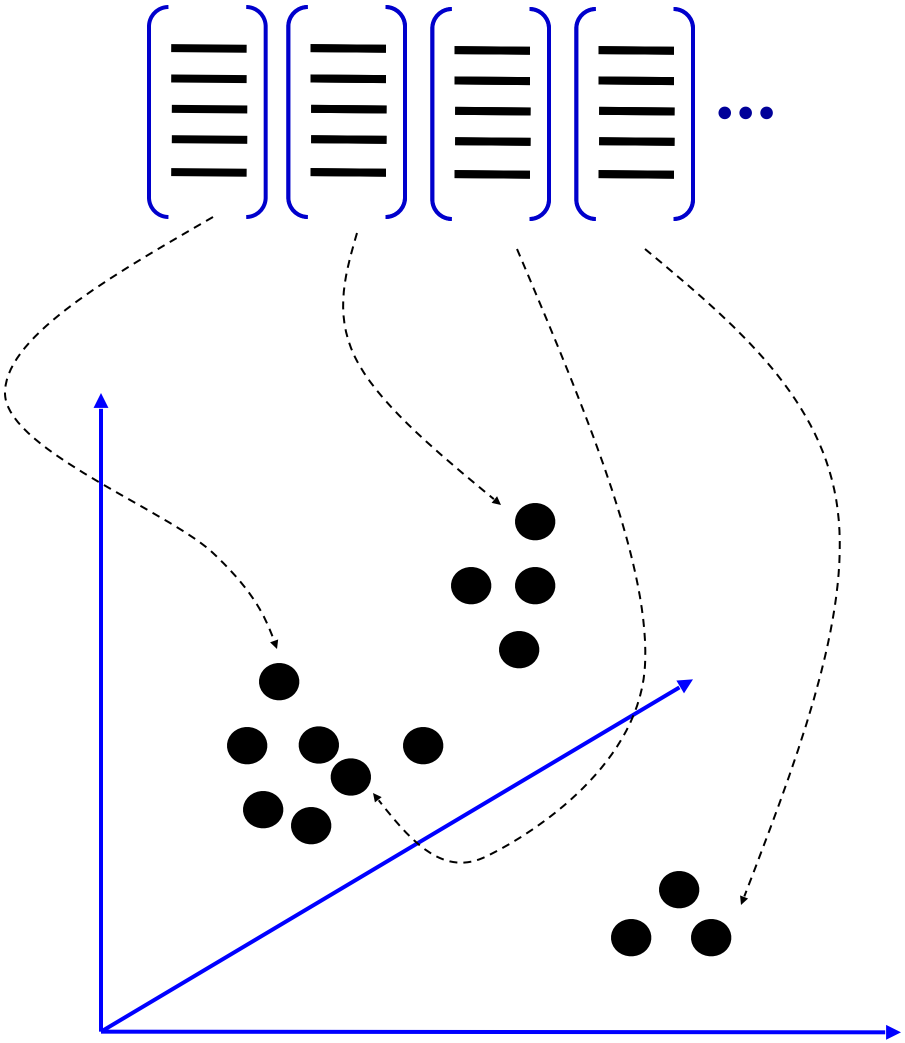
- Compare histograms.
  - Divide intensities into discrete ranges.
  - Count how many pixels in each range.



High-dimensional features

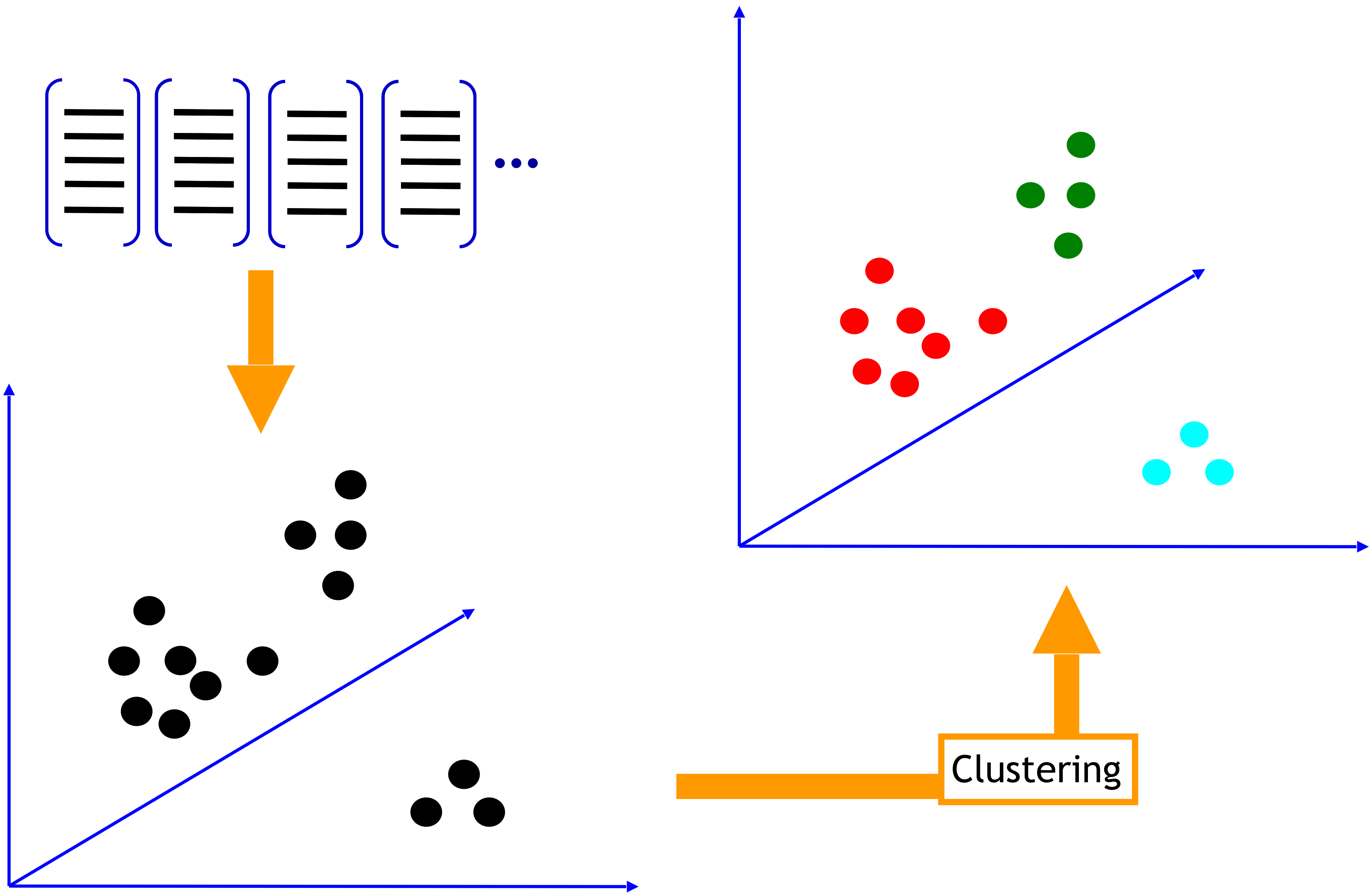
- Often a texture dictionary is learned first by clustering the feature vectors using K-mean clustering.
- Histogram, where each cluster is represented by cluster center, called textons
- Each pixel is assigned to closest texton
- Histograms are compared (often with Chi-square distance)

2. Learning the visual vocabulary



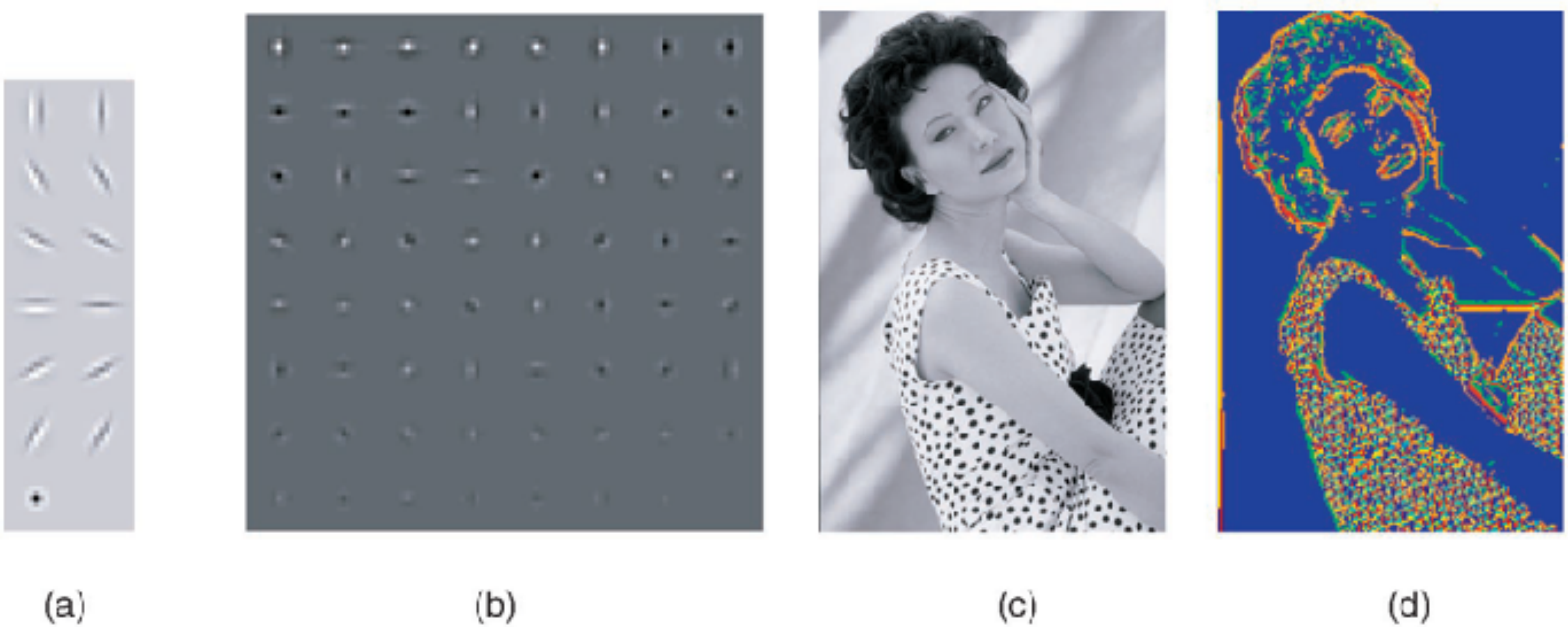
Slide credit: Josef Sivic

2. Learning the visual vocabulary



Slide credit: Josef Sivic

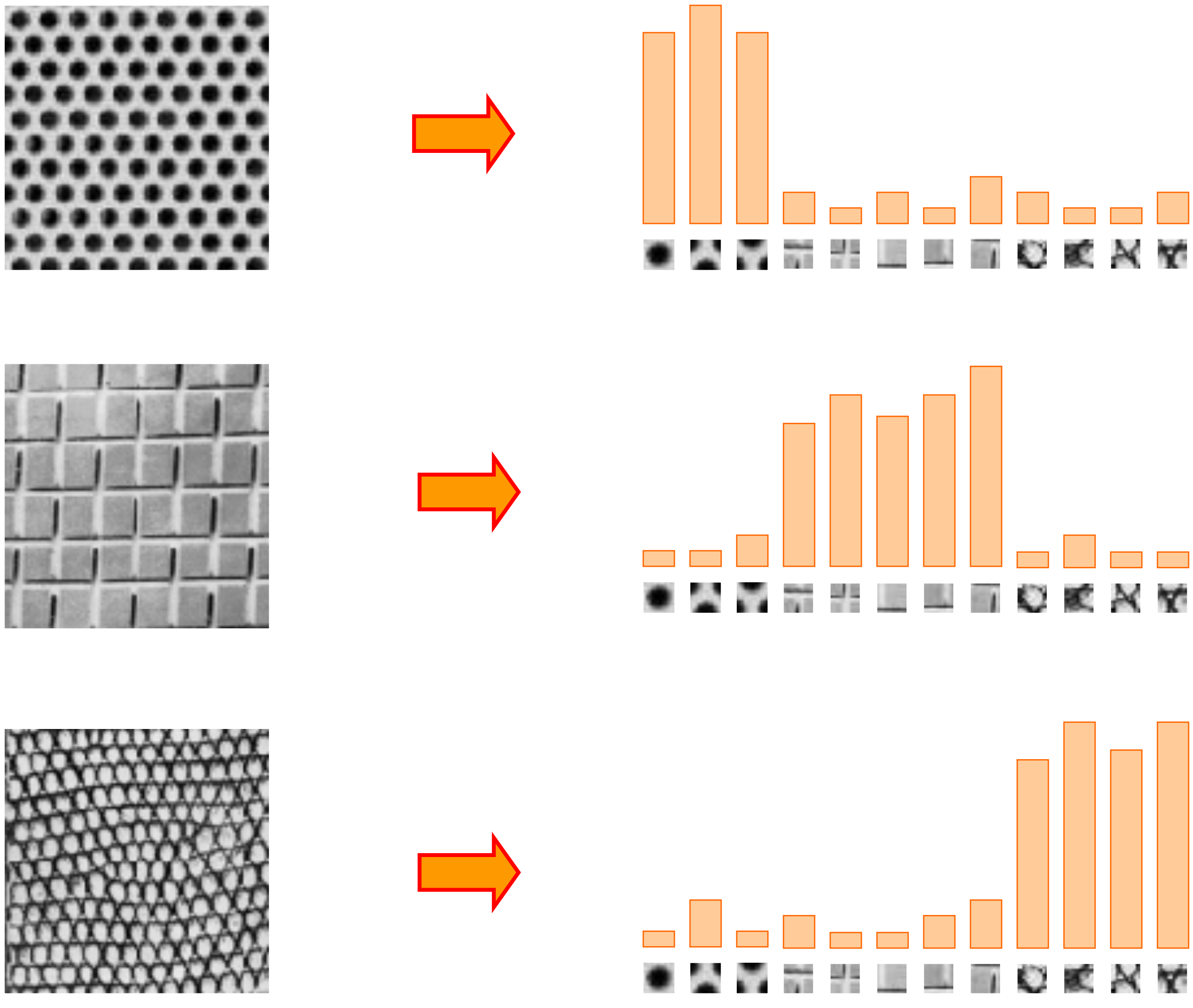
Example of texton dictionary



Filterbank (13 filters) Universal textons (64) Image Texton map (color-code

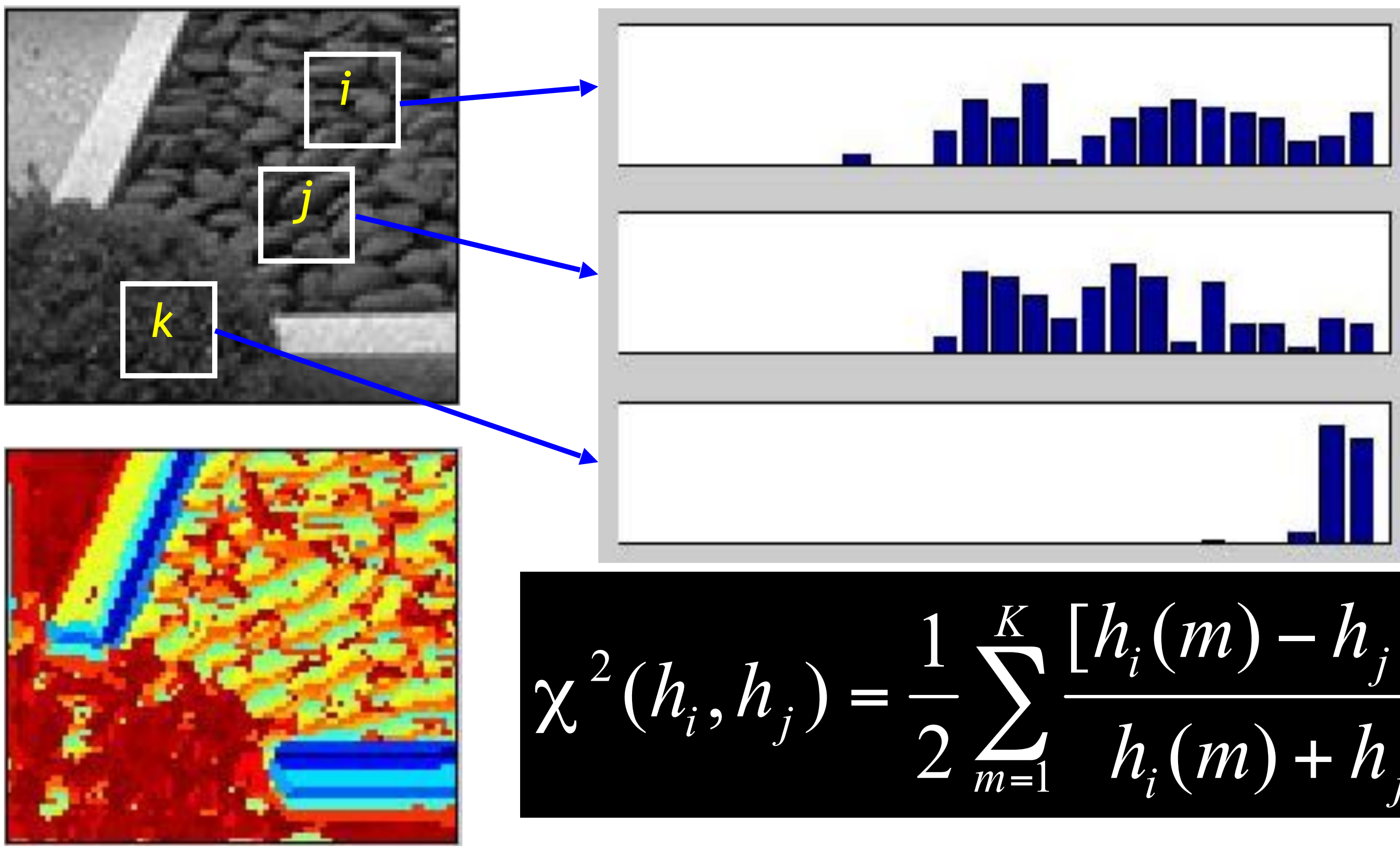
Martin, Fowlkes, Malik, 2004: Berkeley (Pb) edge detector

Texture recognition



Julesz 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

Chi square distance between texton histograms



$$\chi^2(h_i, h_j) = \frac{1}{2}\sum_{m=1}^{K}\frac{[h_i(m) - h_j(m)]^2}{h_i(m) + h_j(m)}$$

(Malik)

Different approaches

- Universal texton dictionaries vs

  Different dictionaries for each texture class
- Sparse features vs. dense features
- Different histogram comparisons

  e.g L1 distance $D(h_1, h_2) = \sum_{i=1}^{N} | h_1(i) - h_2(i) |$

  or EMD (earth mover's distance)