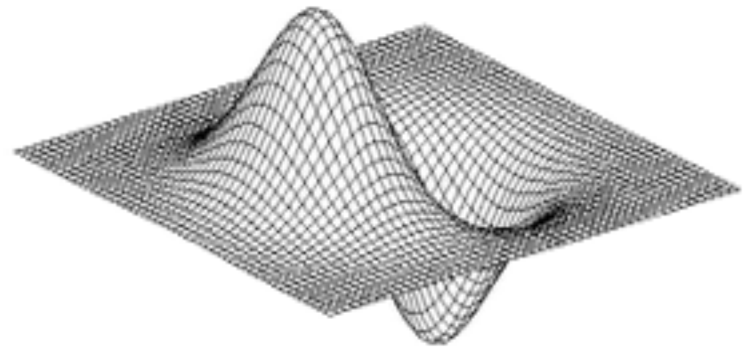
Filters + linear algebra

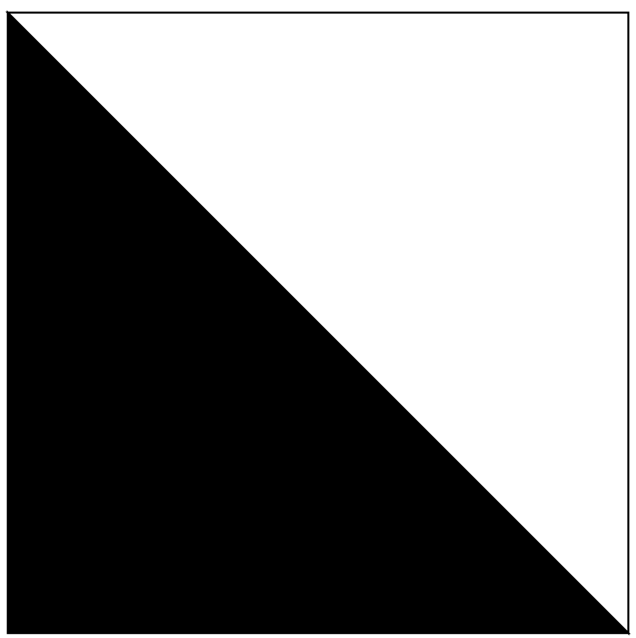
Outline

- Efficiency (pyramids, separability, steerability)
- Linear algebra
- Bag-of-words

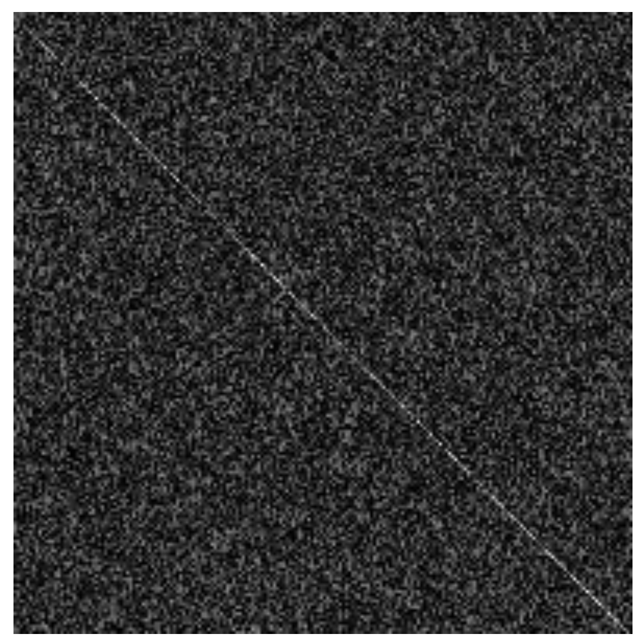
Recall: Canny



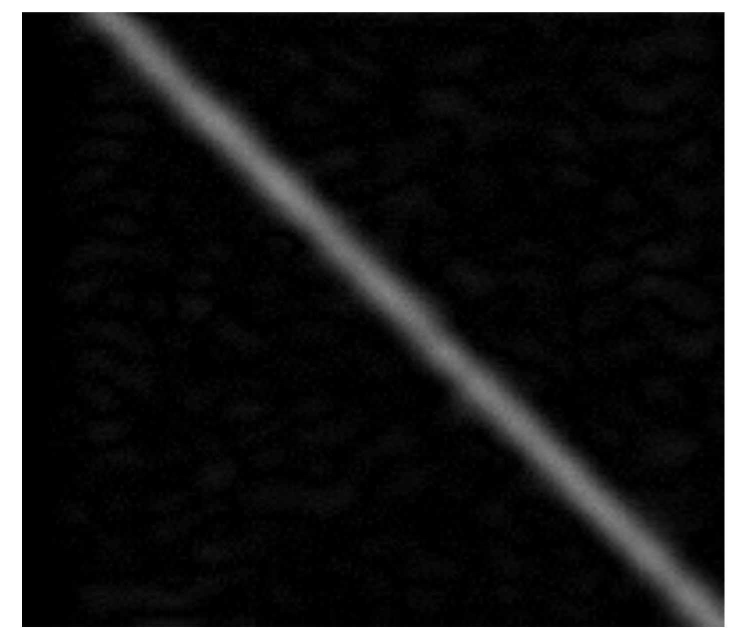
Derivative-of-Gaussian = Gaussian * [1 -1]



Ideal image



+noise

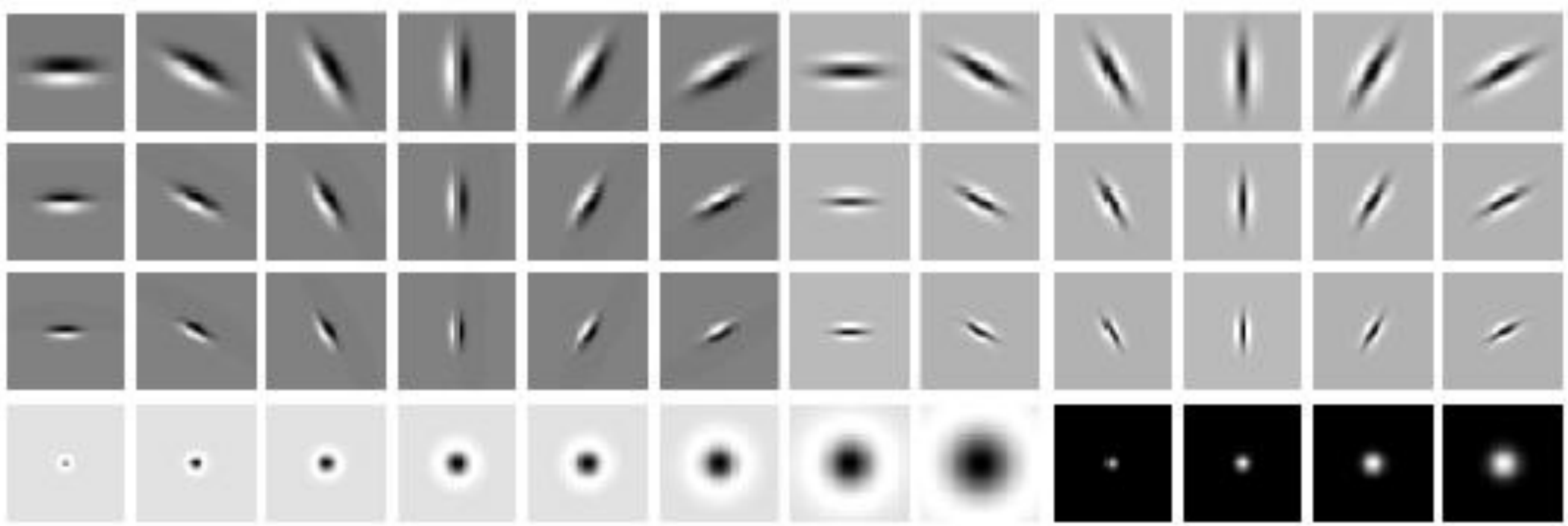


filtered

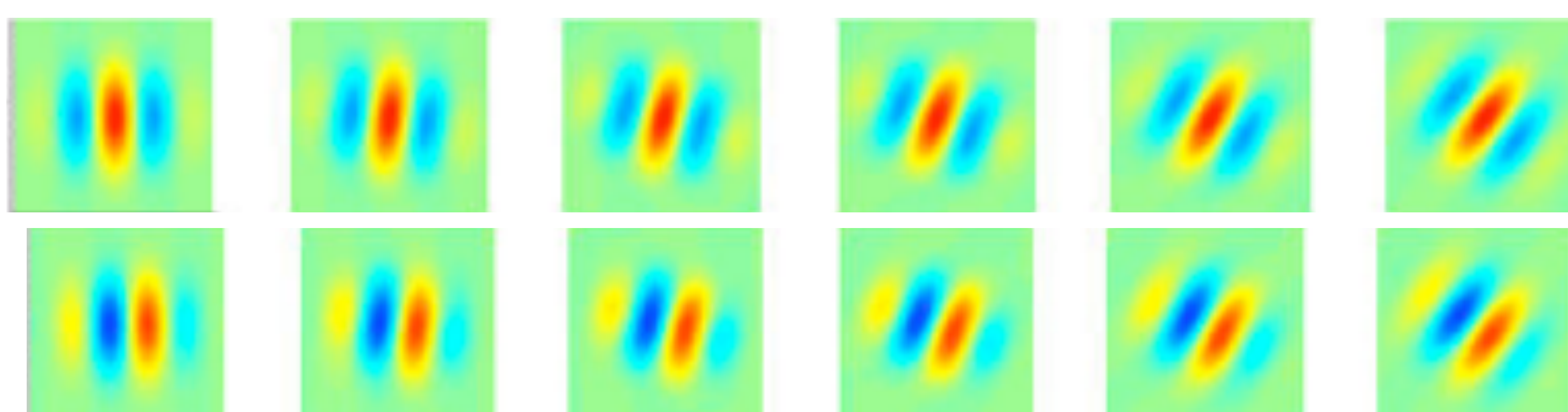
Fundamental tradeoff between good localization and noise reduction

soln 1: NMS

Other soln: oriented filter banks



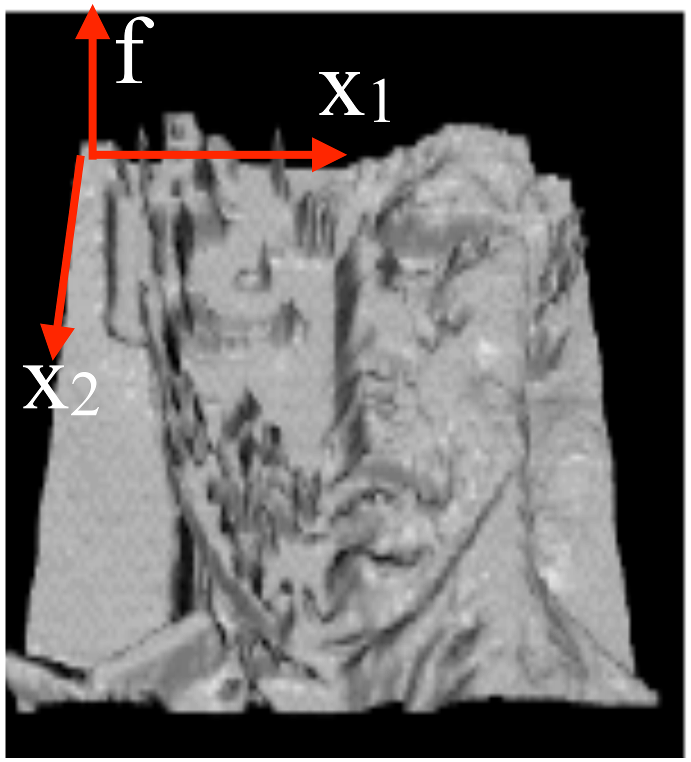
Leung Malik



Gabor filter bank

Revisiting orientations

https://en.wikipedia.org/wiki/Directional_derivative



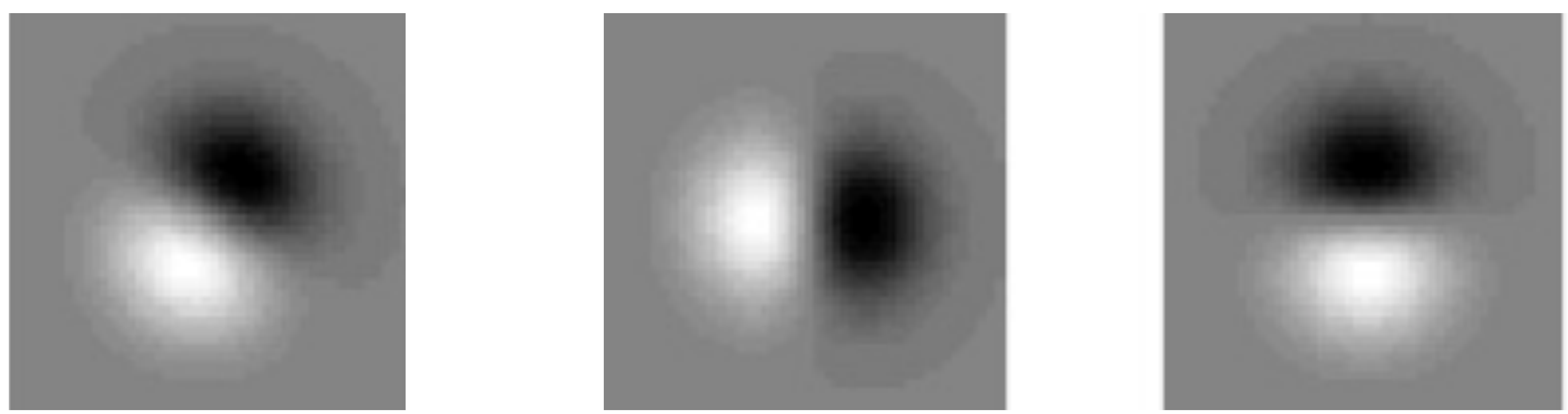
$$f(\mathbf{x}) \quad \text{where} \quad \mathbf{x} = (x_1, x_2), \mathbf{v} = (v_1, v_2)$$

$$\nabla_v f(\mathbf{x}) = \lim_{a \to 0} \frac{f(\mathbf{x} + a\mathbf{v}) - f(\mathbf{x})}{a} = \nabla f(\mathbf{x}) \cdot \mathbf{v}$$

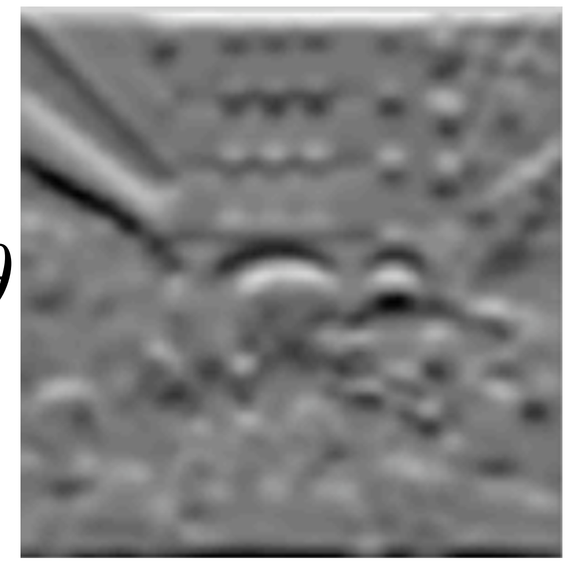
Steerability

- Steerability - the ability to synthesize a filter of any orientation from a linear combination of filters at fixed orientaton

$$\nabla_\theta G_\sigma(x,y) = \cos\theta \frac{\partial G_\sigma}{\partial x} + \sin\theta \frac{\partial G_\sigma}{\partial y}$$



Steerability

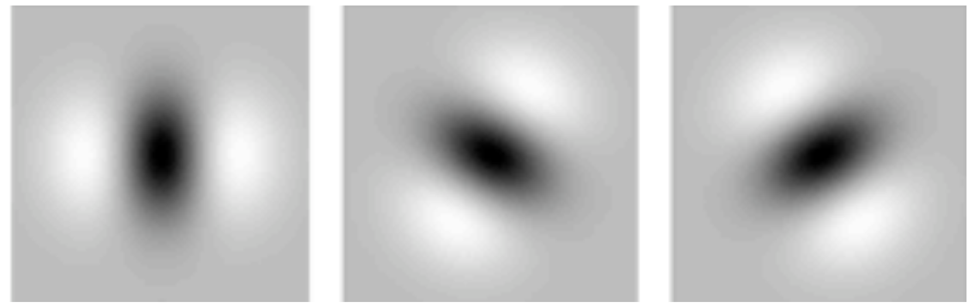
$\cos\theta$  $+\sin\theta$  = 

For a given (x,y) point, let's select the direction that maximizes the above. What's the value of this maximal directional gradient?

The (smoothed) gradient magnitude!

$$\max_{\theta} \nabla_{\theta} F(x,y) = ||\nabla F(x,y)||$$

Second case: Second-derivatives of Gaussians



$$G_\theta = \sum_i k_i(\theta) G_i$$

$G_{2a} = 0.9213(2x^2 - 1)e^{-(x^2+y^2)}$	$k_a(\theta) = \cos^2(\theta)$
$G_{2b} = 1.843xye^{-(x^2+y^2)}$	$k_b(\theta) = -2\cos(\theta)\sin(\theta)$
$G_{2c} = 0.9213(2y^2 - 1)e^{-(x^2+y^2)}$	$k_c(\theta) = \sin^2(\theta)$

When is this possible? Filters must be “smooth” in orientation space

W. Freeman, T. Adelson, “The Design and Use of Sterrable Filters”, IEEE Trans. Patt, Anal. and Machine Intell., vol 13, #9, pp 891-900, Sept 1991

Separability

Image of size N^2
Filter of size M^2

Complexity of filtering?

O(N^2M^2)

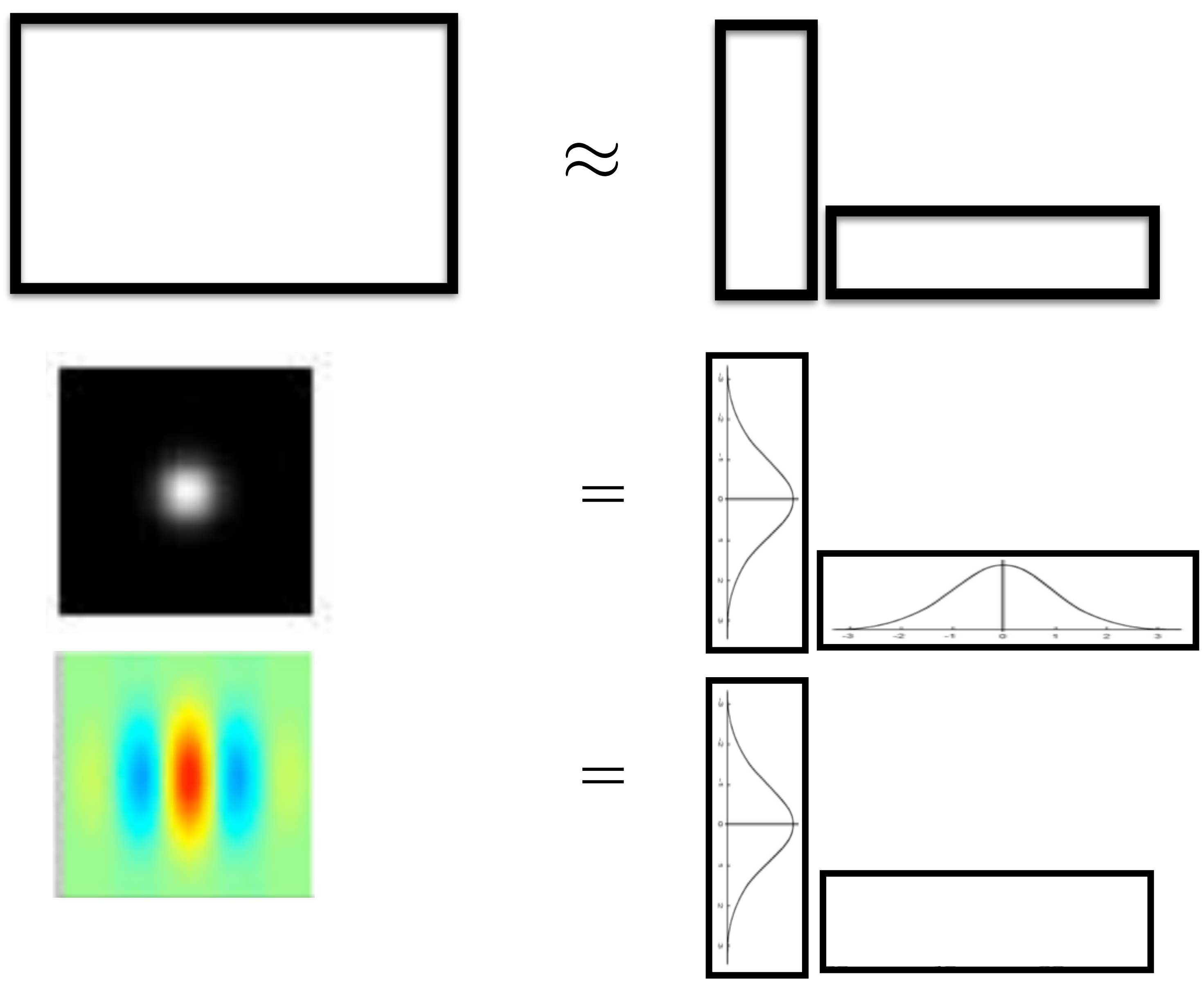
$$H[u, v] = H_x[u] H_y[v]$$

$$G[i, j] = \sum_u \sum_v H[u, v] F[i + u, j + v]$$

Separability

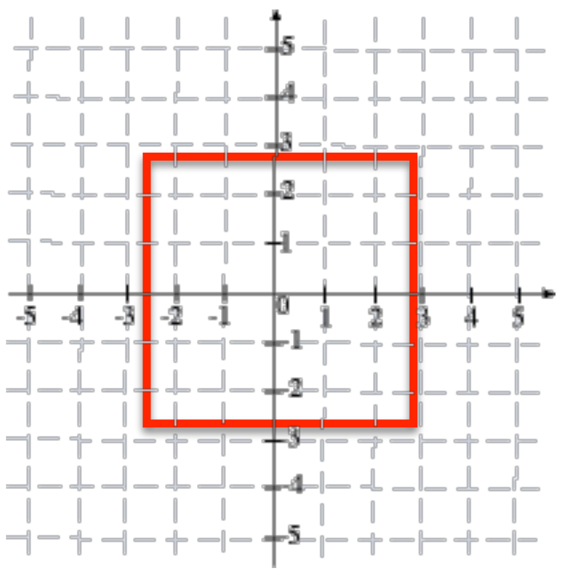
Given a filter, how can we come up with a good separable approximation?

$$H[u, v] \approx H_x[u] H_y[v]$$

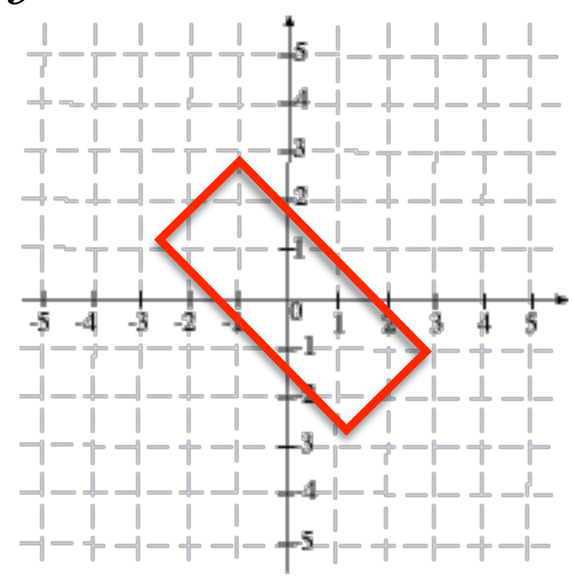


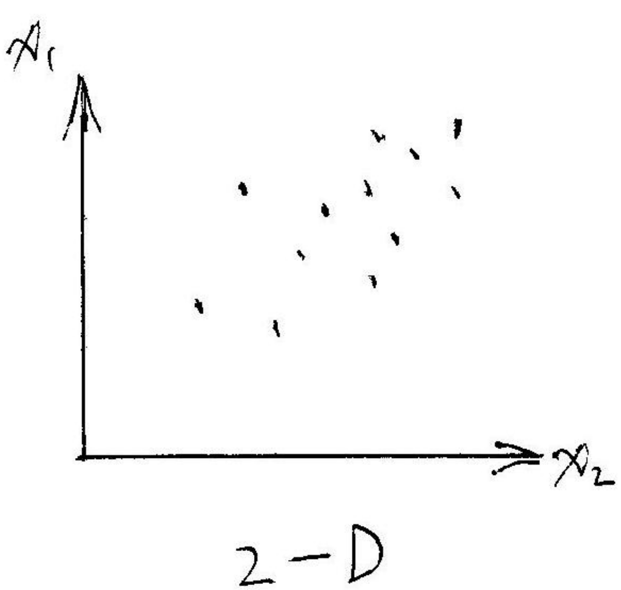
Linear algebra digression

Any matrix can be thought of as a *transformation*

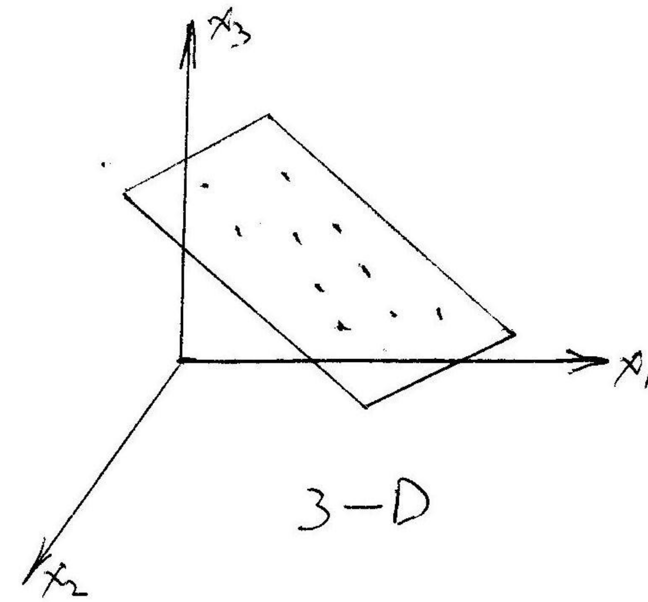


$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$





$$A = \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix}$$

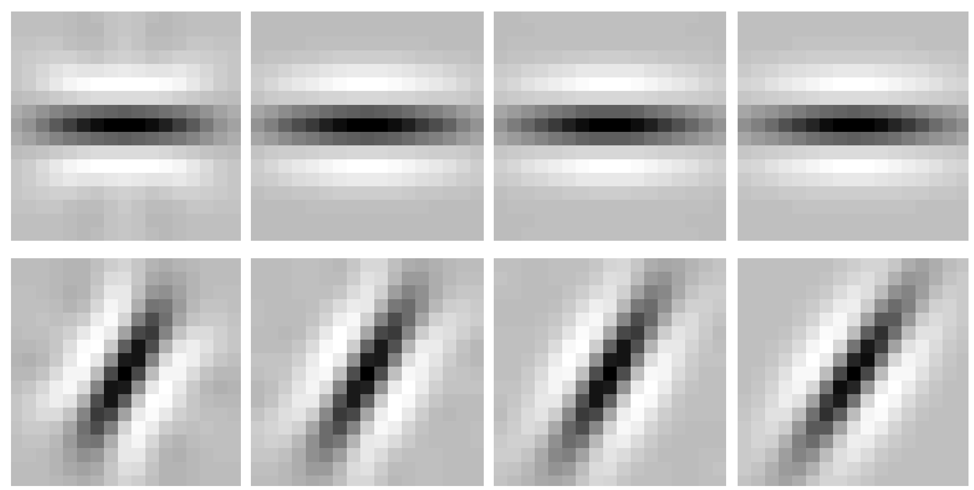


Change of basis

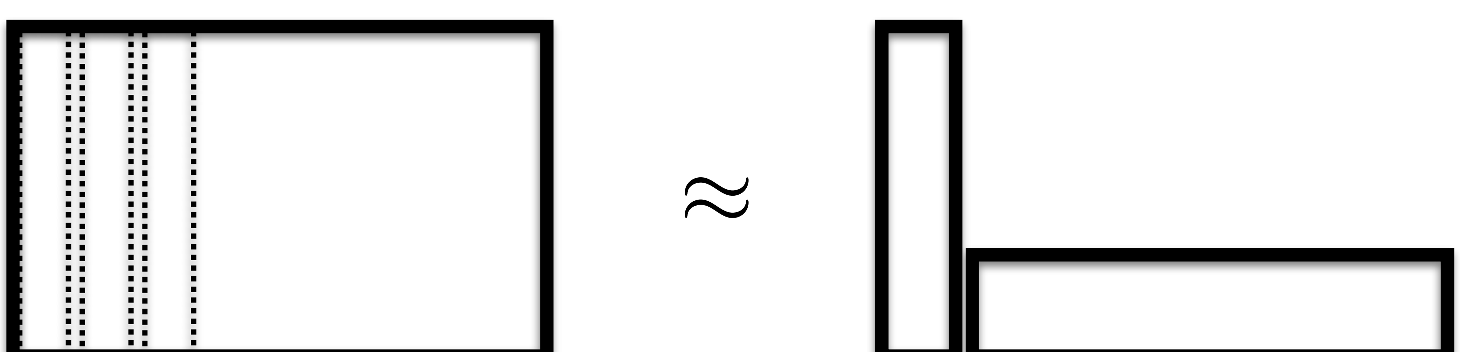
See handout

Least-squares method of steerability

Shy & Perona, CVPR94



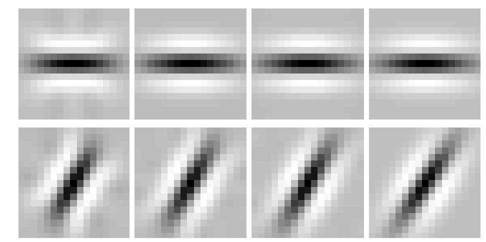
Stack bank of filters into a matrix
Apply SVD to generate low-rank approximation



Least-squares method of steerability

Shy & Perona, CVPR94

Rank 1 approximation



$$H[u,v,k] = H_s[u,v]c[k]$$

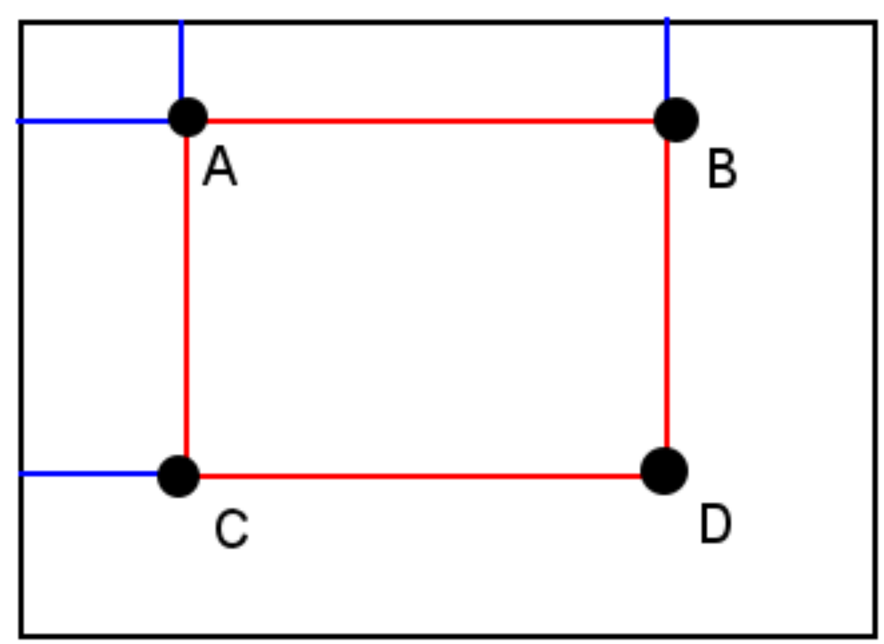
$$G[i,j,k] = \sum_u \sum_v H[u,v,k]F[i+u,j+v]$$

Final trick for efficient filters: box filtering with integral images

http://en.wikipedia.org/wiki/Summed_area_table

$$I(x,y) = \sum_{\substack{x' \le x \\ y' \le y}} i(x',y')$$

$$I(x,y) = i(x,y) + I(x-1,y) + I(x,y-1) - I(x-1,y-1)$$



Sum = D - B - C + A

Reduces $O(N^2M^2)$ to $O(N^2)$

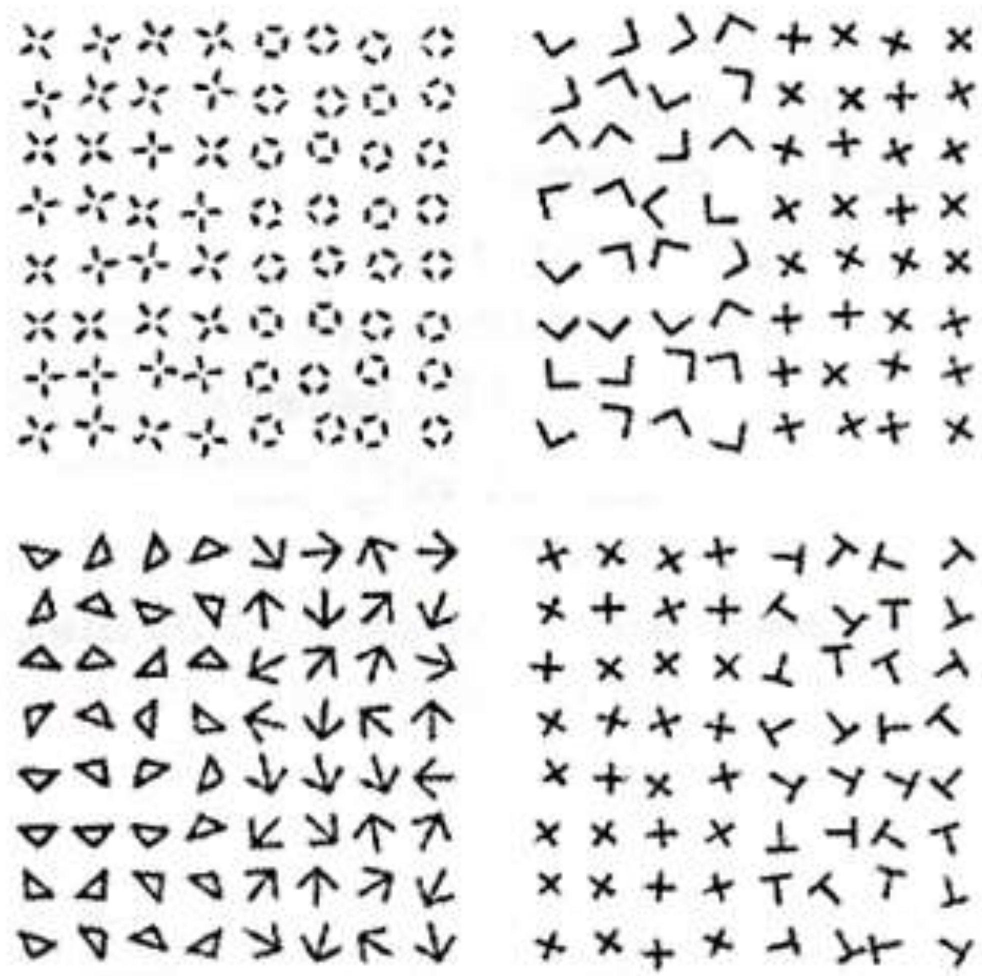
Outline

- Efficiency (pyramids, separability, steerability)
- Bag-of-words
- Frequency analysis

HW1: Scene Classification

Can we think about as different “textures”?

Pre-attentive texture discrimination



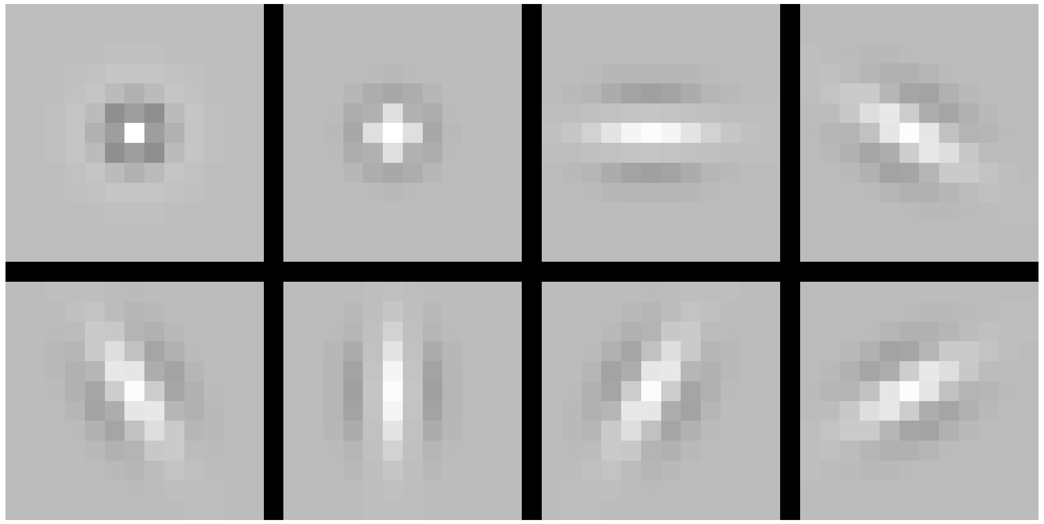
(Julesz,1981)

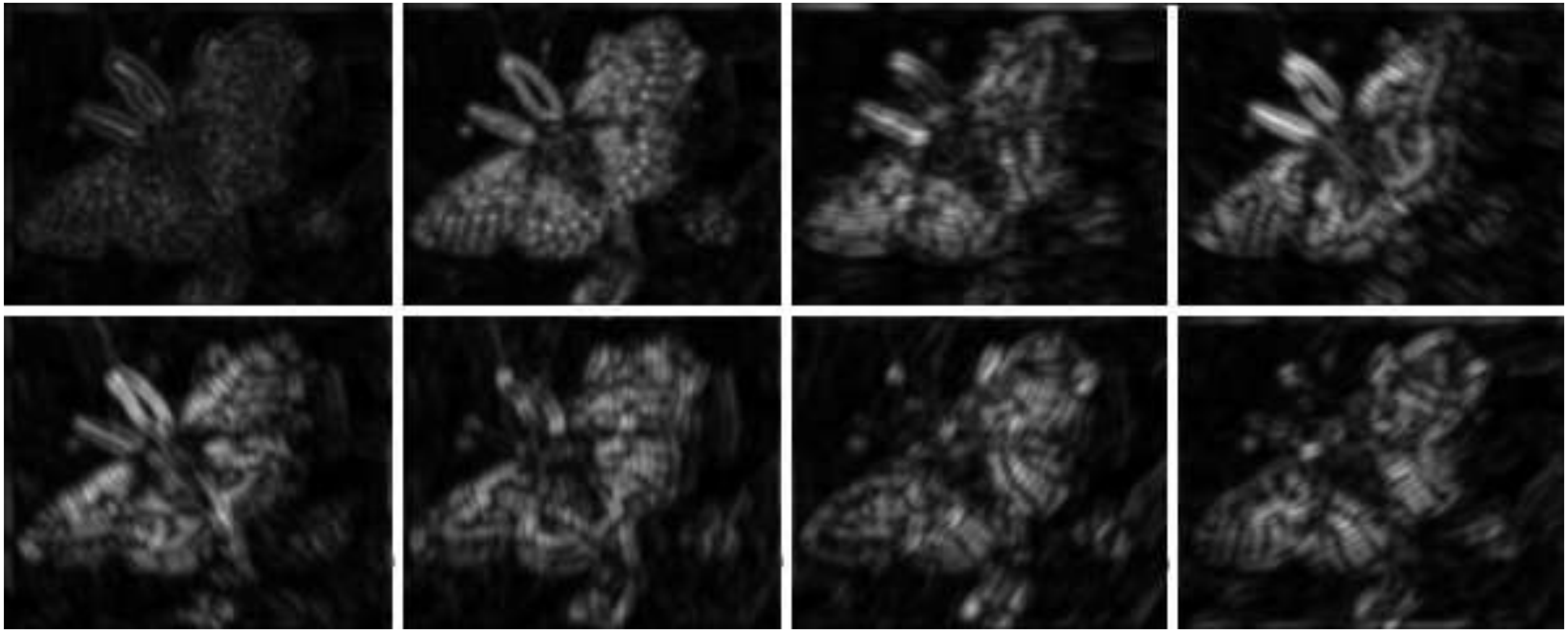
“textons”

160 ms, outside foveal gaze
Instantenous, or effortless texture discrimination

Representing textures

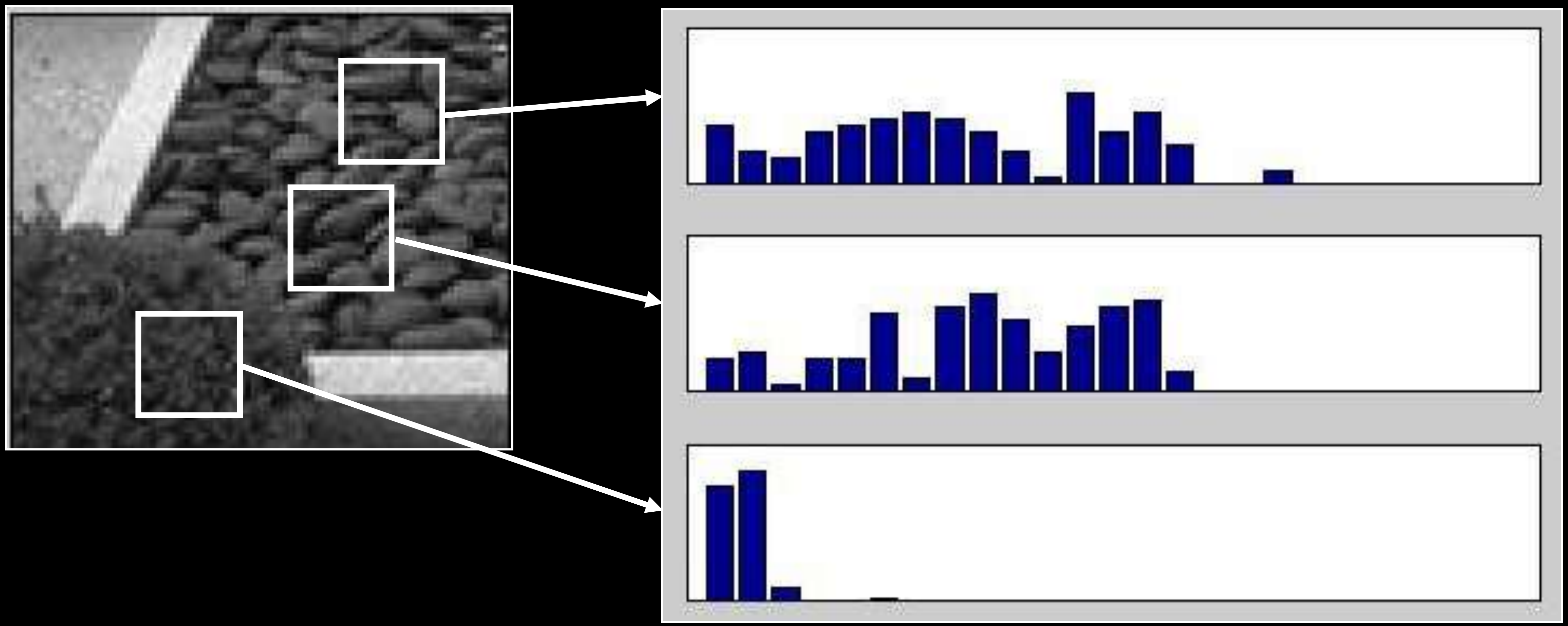
- Textures are made up of stylized subelements, repeated in meaningful ways
- Representation:
  - find the subelements, and represent their statistics
- But what are the subelements, and how do we find them?
  - find subelements by applying filters, looking at the magnitude of the response
- What statistics?
  - Mean, standard deviation, histograms of marginal statistics





Histogram of filter responses

P(I * )



Use collection of histograms for a set of filters to represent texture

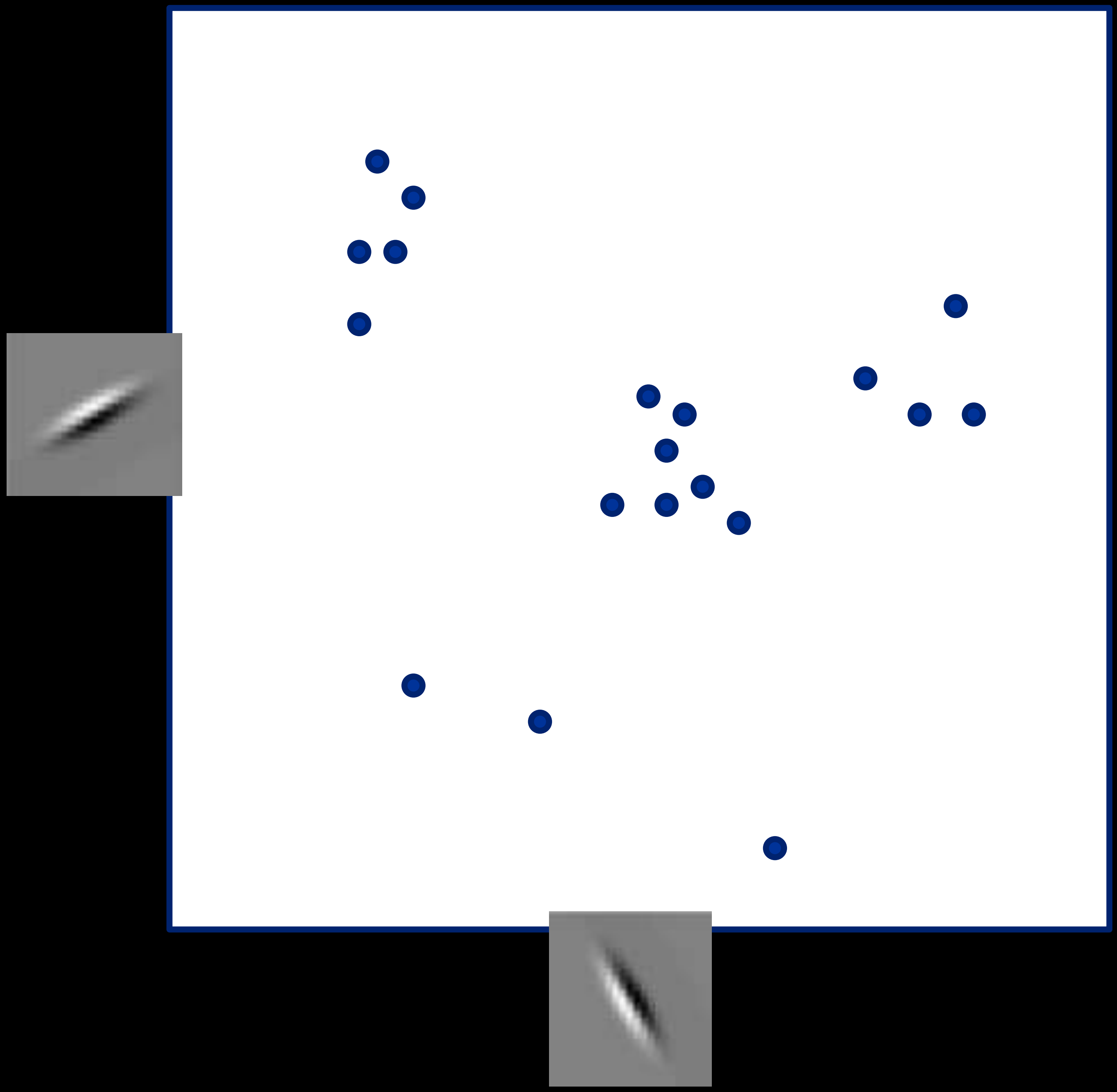
Joint vs marginals

- If we have M filters and discretize reponses in N possible values:

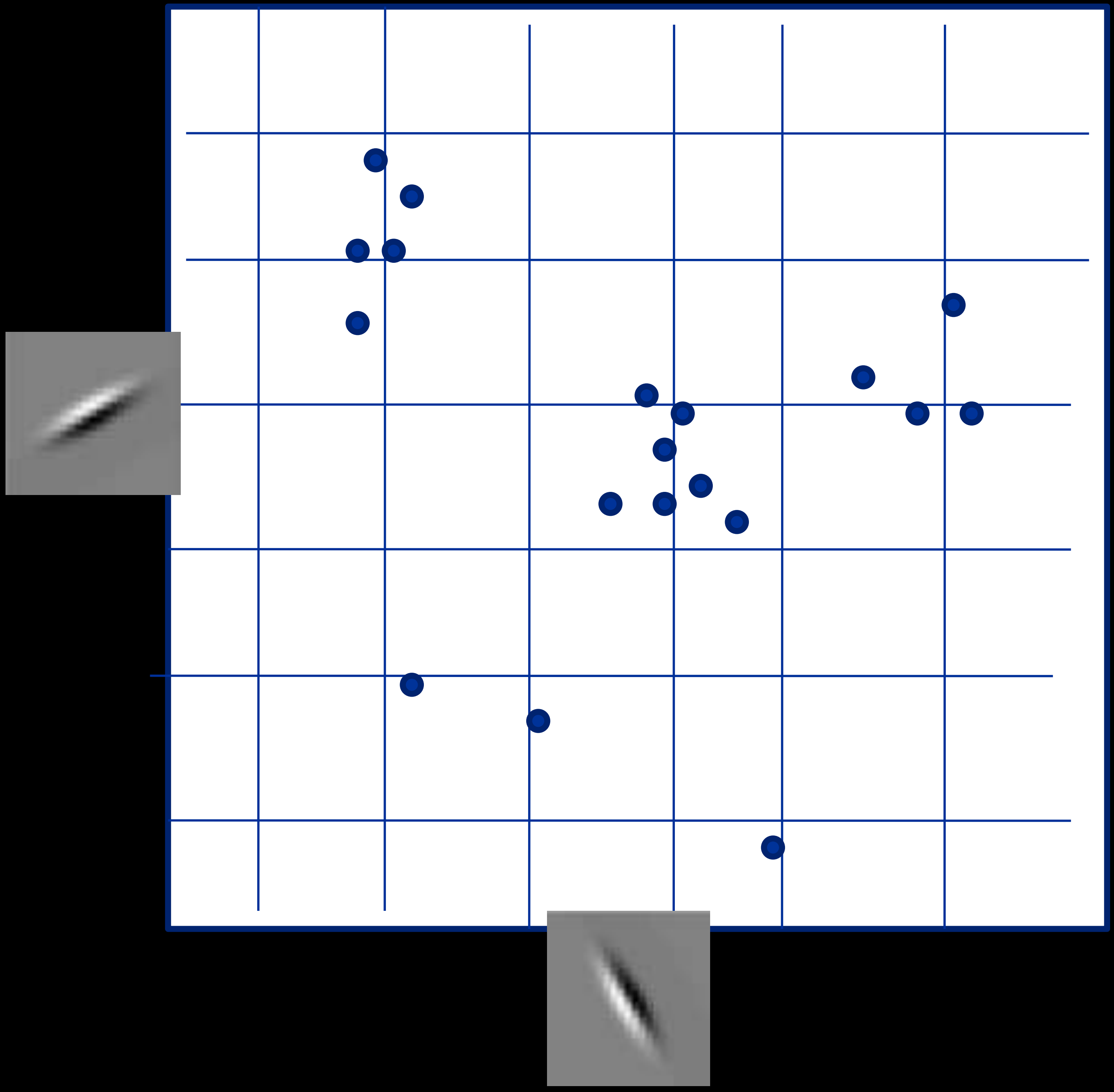
$$P(f1,f2,f3,\ldots) = N^M \text{ values}$$

$$P(f1)P(f2)P(f3)\ldots. = NM \text{ values}$$

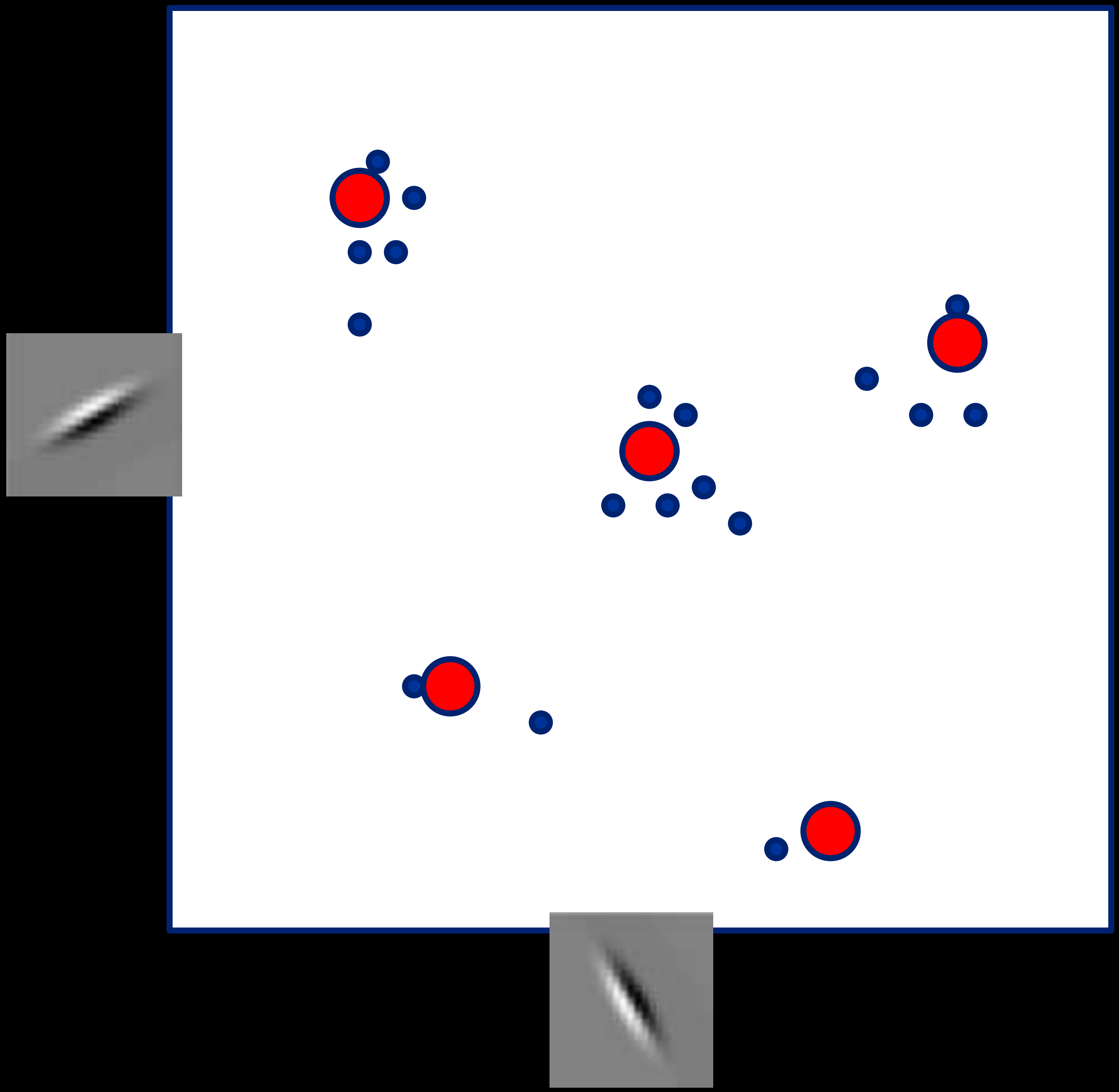
Let’s look at samples from joint



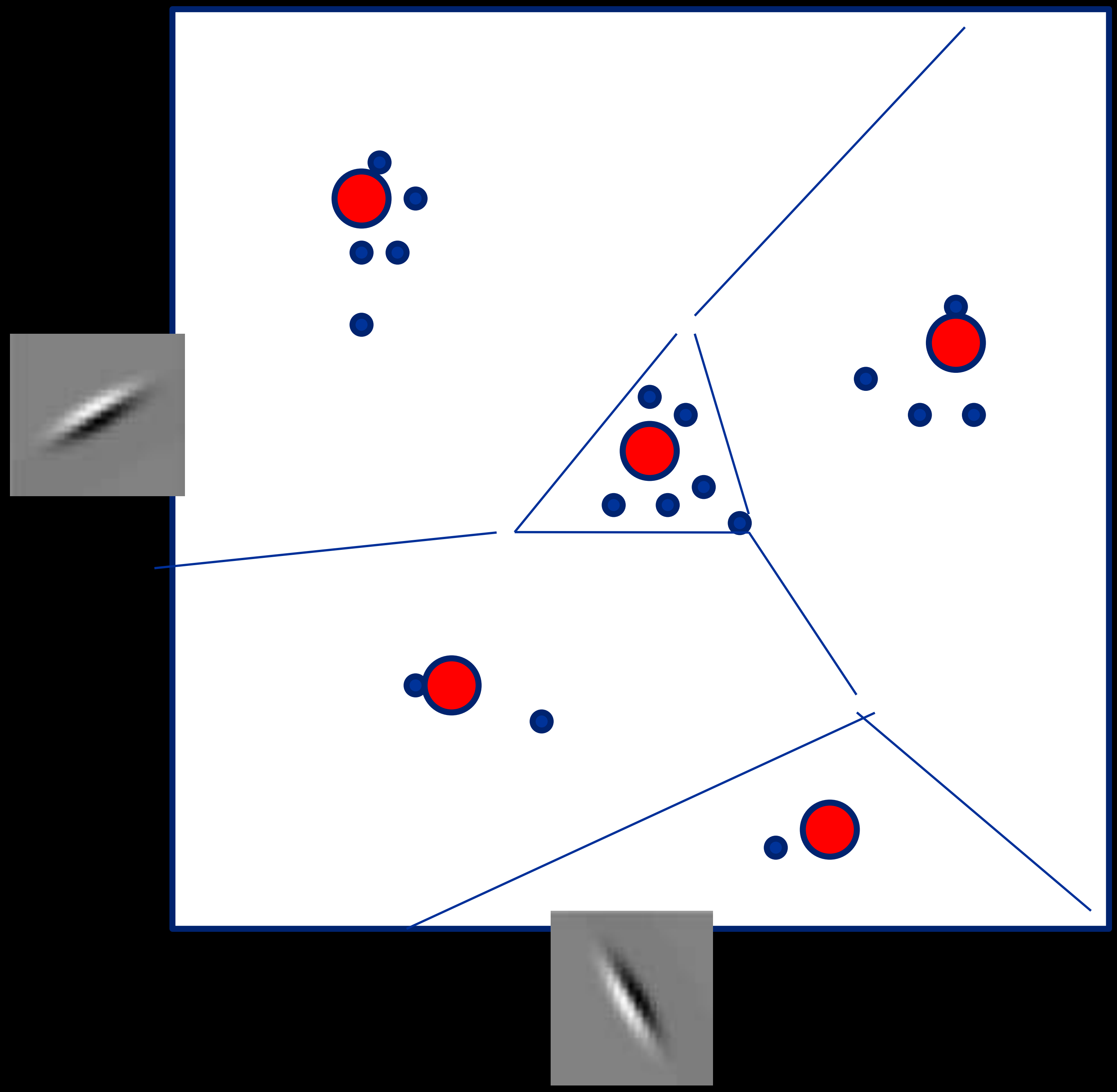
Counting bins is difficult because most will be zero



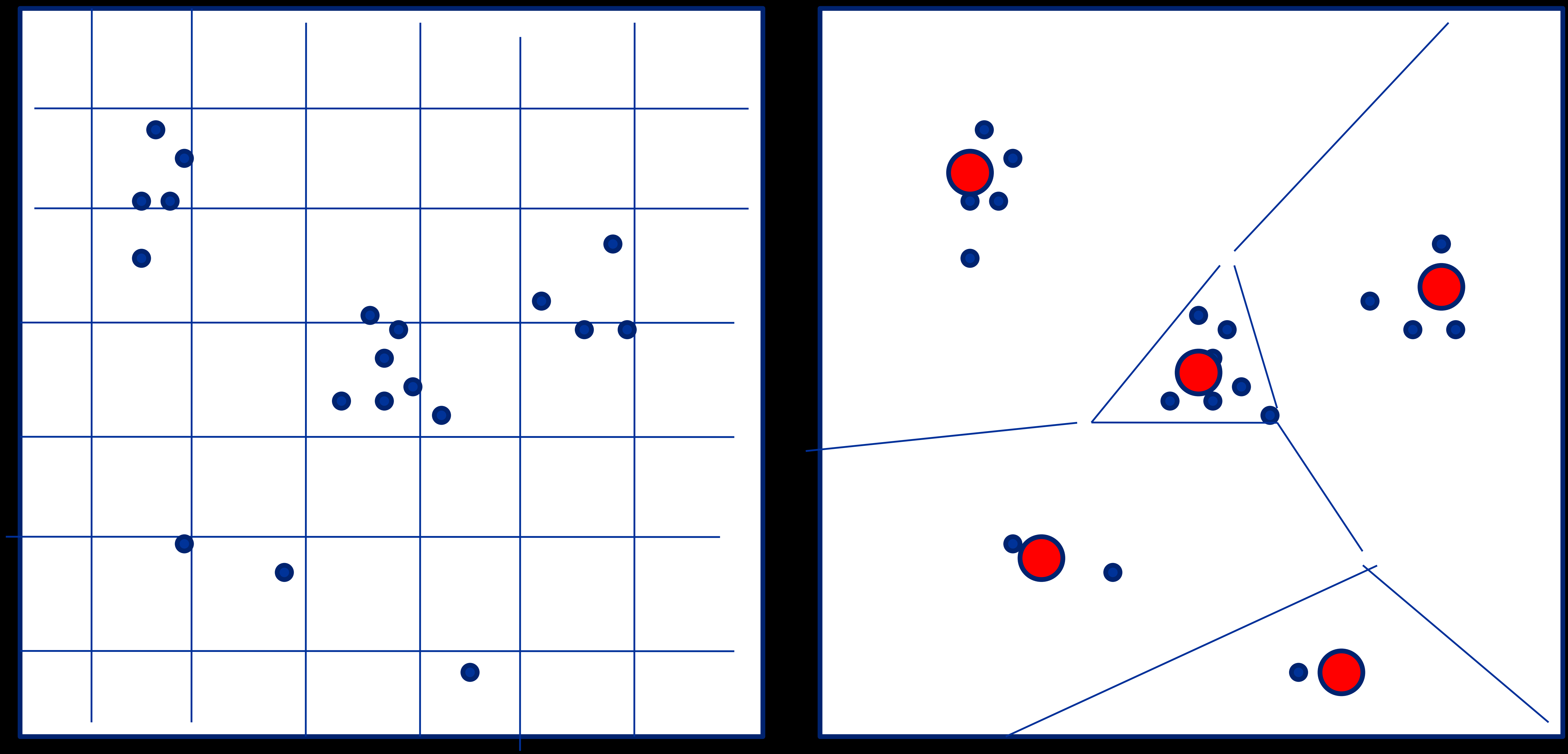
But reponses seem to cluster into groups….



Adaptive binning strategy



Capture joint statistics via histograms of vector quantized features



K-means algorithm

$$\min_{Z,D} C(Z,D,X) \quad \text{where} \quad C(Z,D,X) = \sum_i ||x_i - d_{z_i}||^2$$

$$x_i \in R^n$$

$$z_i \in \{1, 2, \ldots K\}$$

$$d_j \in R^n$$

K-means algorithm

$$\min_{Z,D} C(Z,D,X) \quad \text{where} \quad C(Z,D,X) = \sum_i ||x_i - d_{z_i}||^2$$

1. $\min_{Z} C(Z,D,X)$

2. $\min_{D} C(Z,D,X)$

K-means algorithm

$$\min_{Z,D} C(Z,D,X) \quad \text{where} \quad C(Z,D,X) = \sum_i ||x_i - d_{z_i}||^2$$

1. $\min_{Z} C(Z,D,X)$

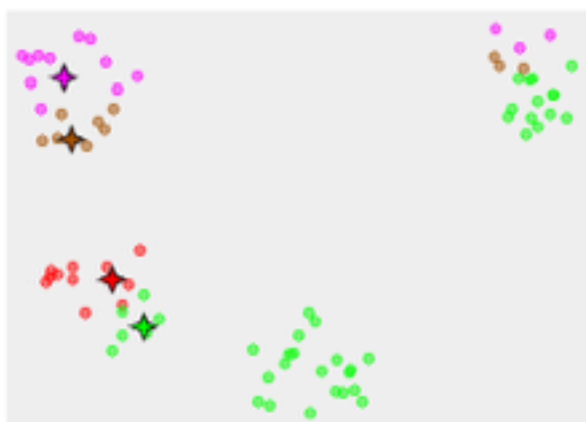
   $$z_i = \arg\min_k ||x_i - d_k||^2, \quad \forall i$$

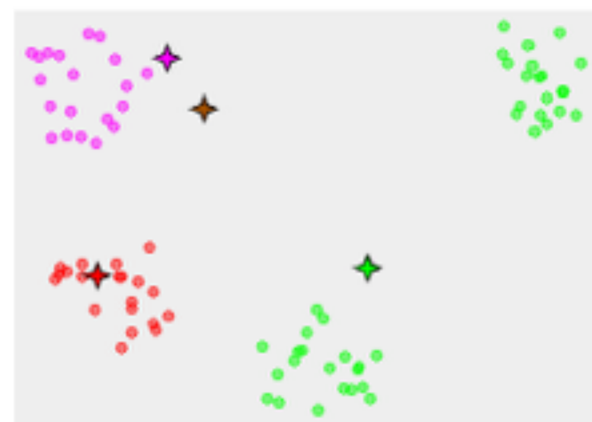
2. $\min_{D} C(Z,D,X)$

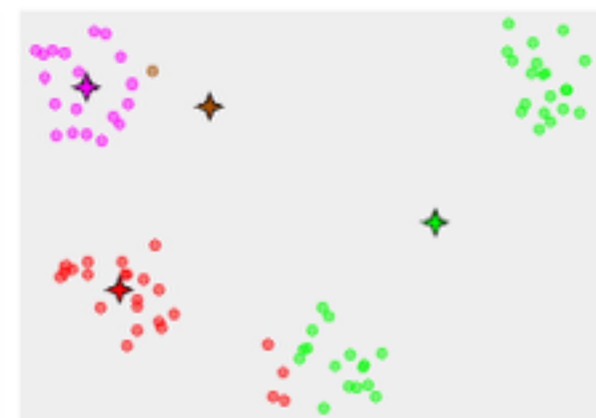
   $$d_k = \frac{1}{|S_i|} \sum_{i \in S_k} x_i, \quad S_k = \{i : z_i = k\}, \forall k$$

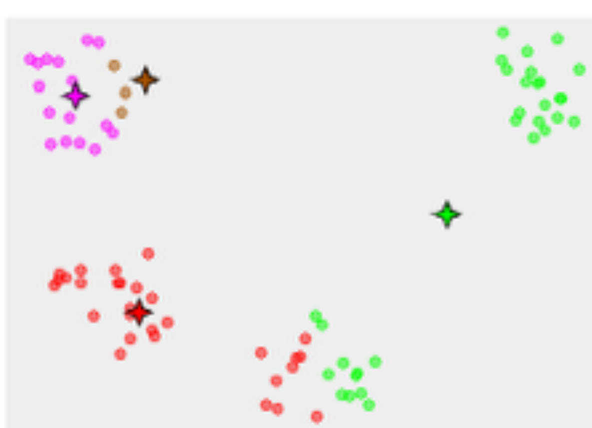
K-means

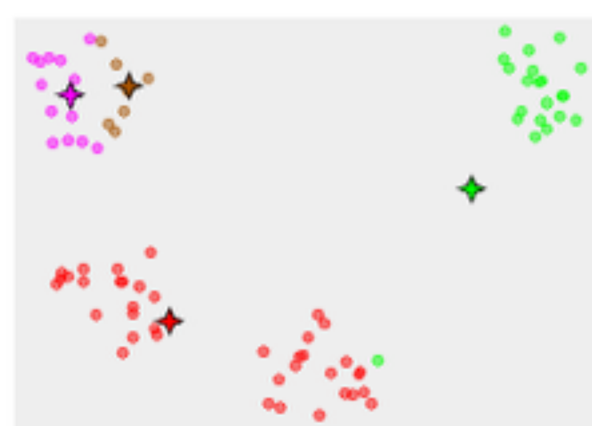
Do we globally optimize the objective function?

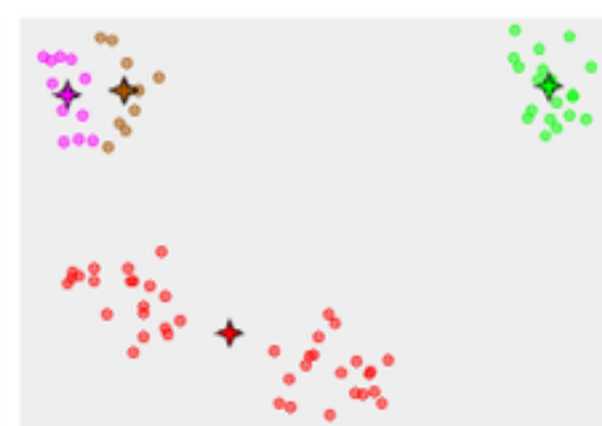

1


2


3


4


5

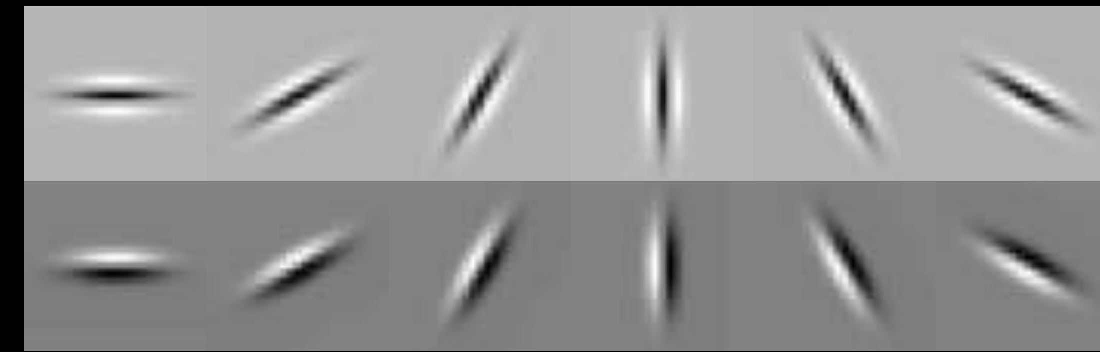

6

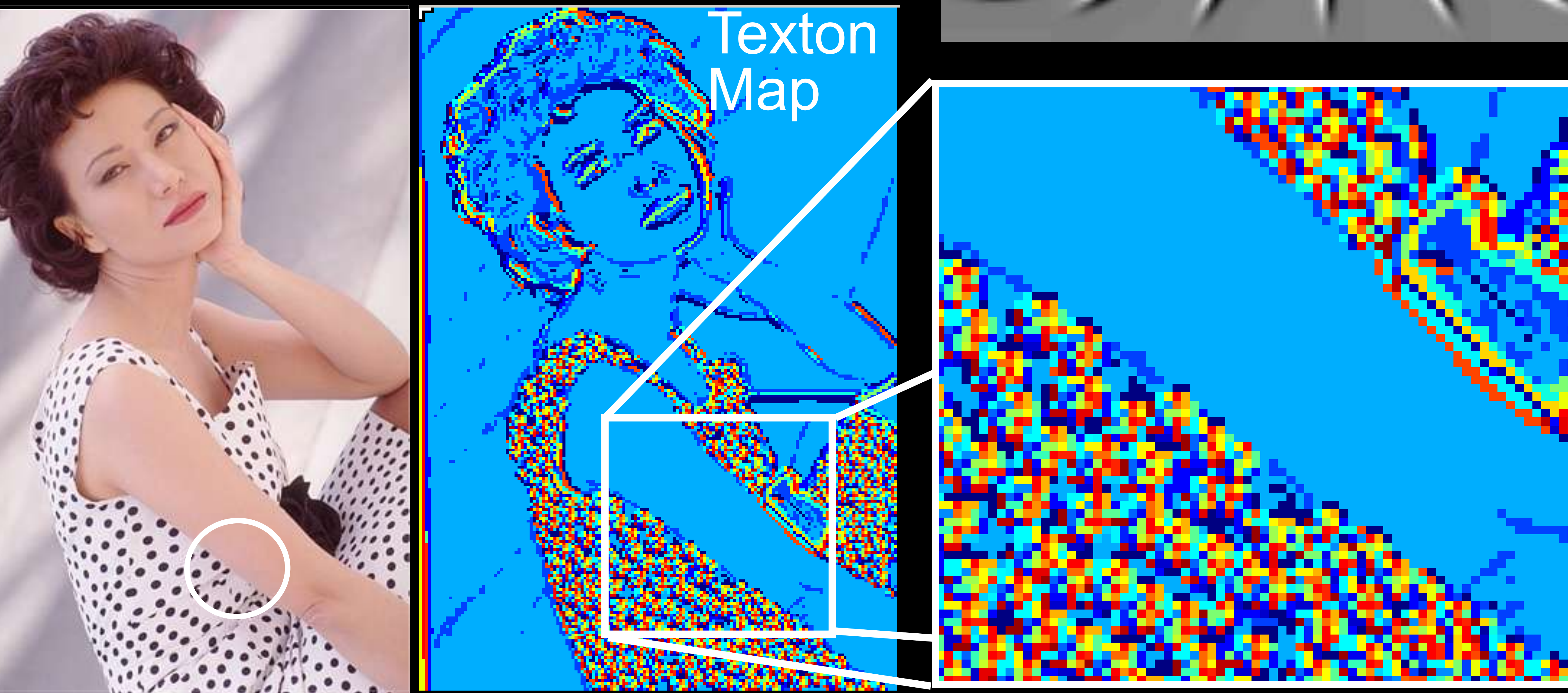
Training vs testing

Training: $\min_{Z,D} C(Z, D, X_{\text{train}})$

Testing: $\min_{Z} C(Z, D, X_{\text{test}})$

“Textons”





Textons are vector-quantized filter outputs.

Use k-means to cluster joint filter outputs, adaptively partition the space into histogram bins.

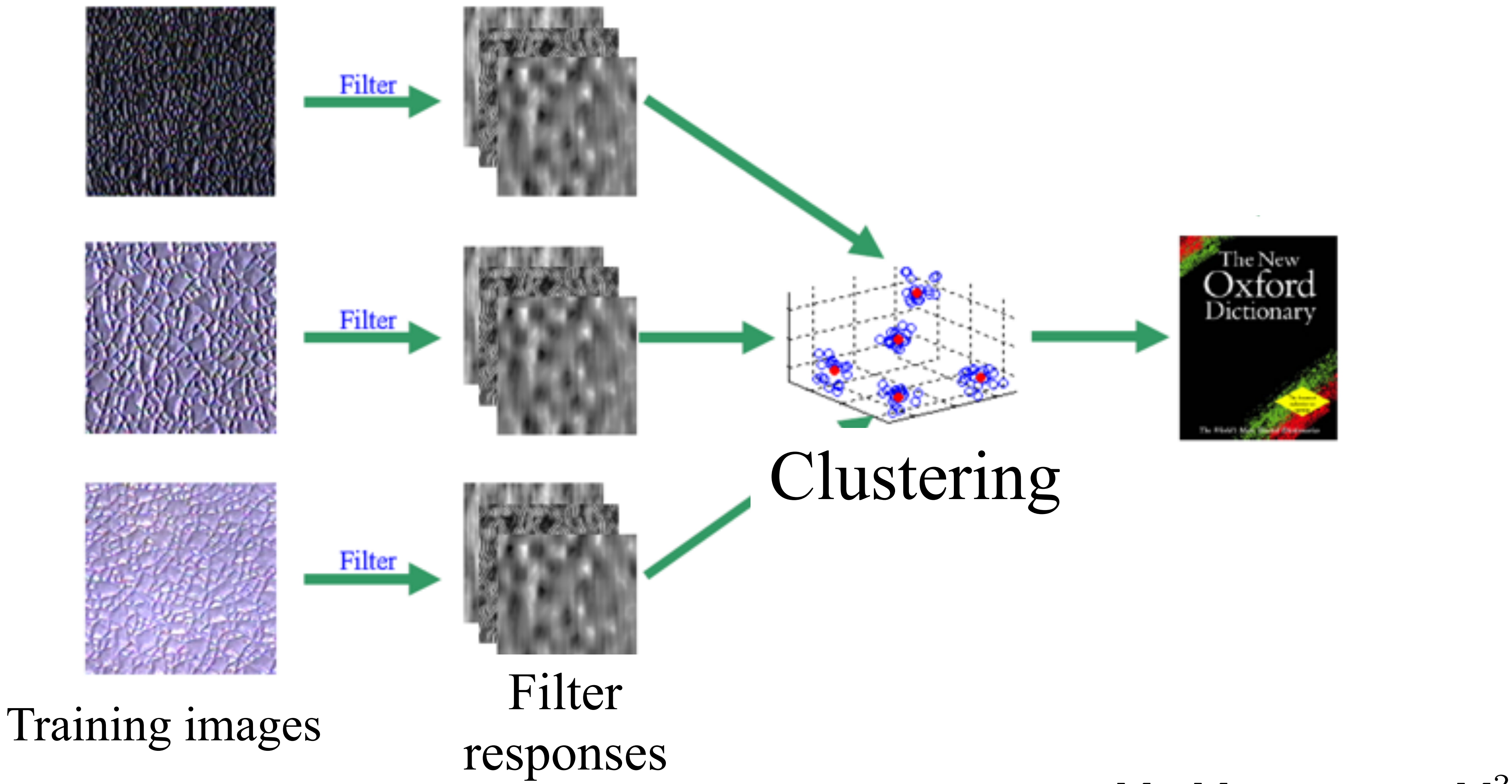
Inspiration from Text Analysis

Political observers say that the government of Zorgia does not control the political situation. The government will not hold elections …

The ZH-20 unit is a 200Gigahertz processor with 2Gigabyte memory. Its strength is its bus and high-speed memory……

How to compare the two articles?

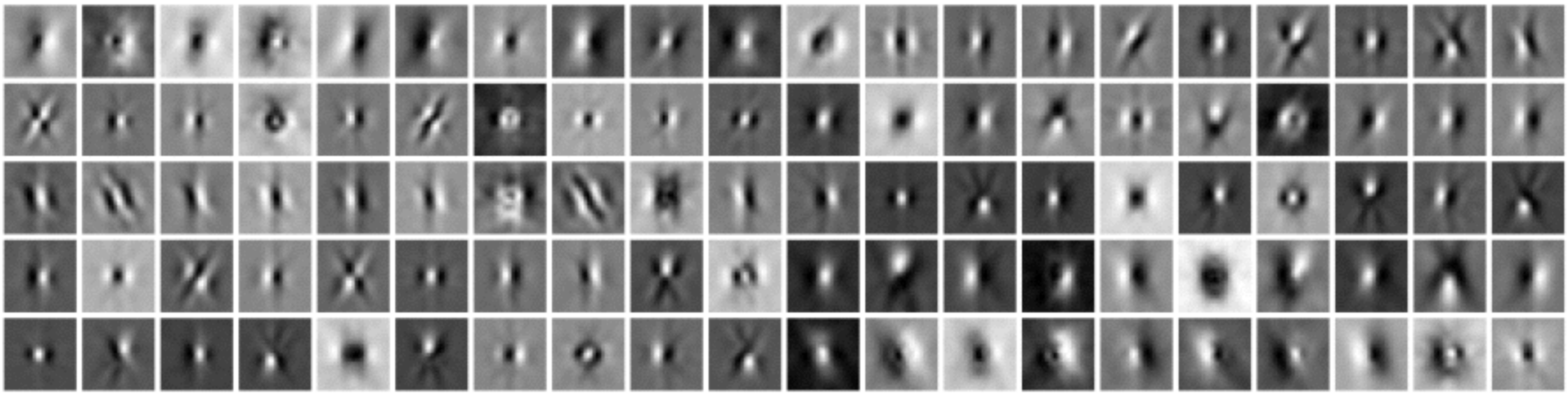
Bag-of-words



Given a large set of vectorized image patches: $x \in R^{M \times M} \Rightarrow x \in R^{M^2}$
and a bank of vectorized filters F = $[f_1, f_2, .. f_b]$

1. Project each patch into *basis* spanned by F: $y = F^T x, \quad y \in R^b$
   (does this basis span $R^{M\text{^}2}$? Is it orthonormal?)
2. Cluster patches in this projected space

Use pseudoinverse of filter bank to visualize cluster means in original space



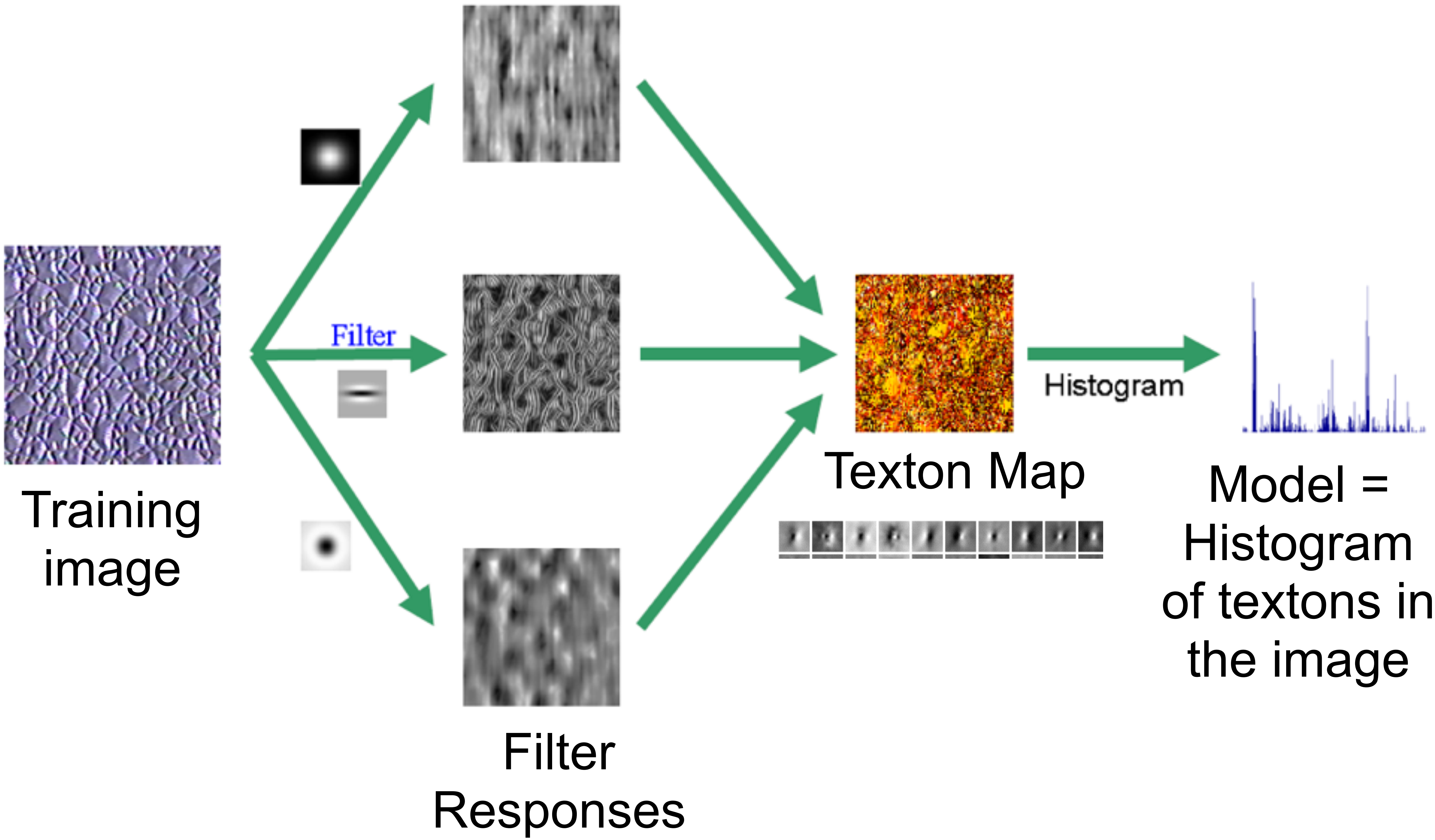
Given a M X M image patch ‘x’ (reshaped into a M^2 vector) and a filter bank of B filters, filter bank responses can be seen as a change of basis

$$y = F^T x, \quad x \in R^{M^2}, y \in R^B$$

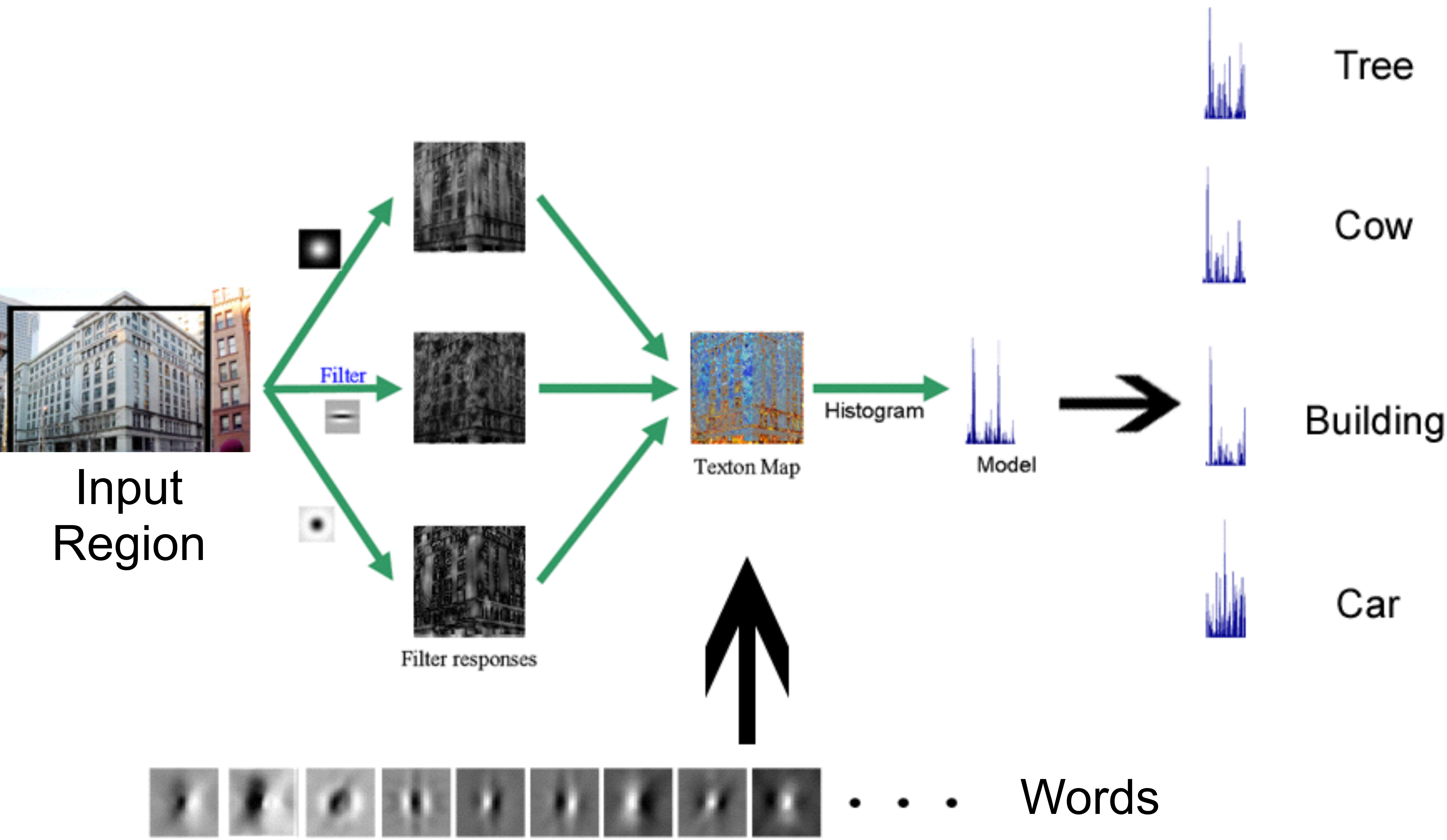
$$x \approx (F^T)^+ y$$

$$Vis(d_j) \approx (F^T)^+ d_j$$

Modeling Texture Distributions

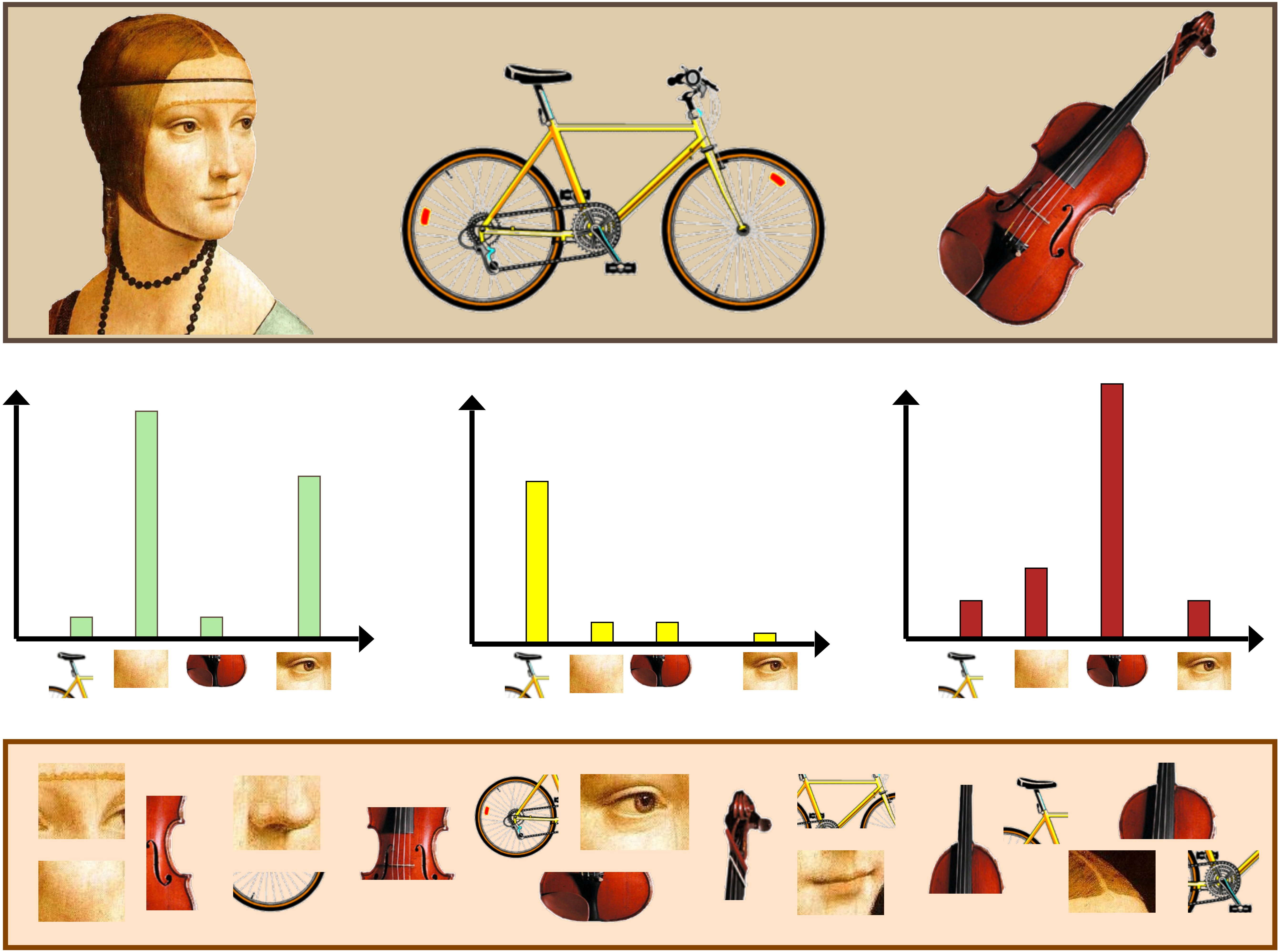


Example Classification

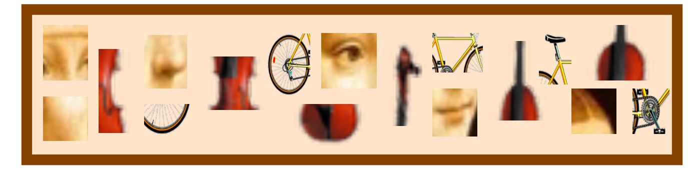


Object → **Bag of ‘words’**

ICCV 2005 short course, L. Fei-Fei



Bags of visual words



- Summarize entire image based on its distribution (histogram) of word occurrences.
- Analogous to bag of words representation commonly used for documents.

Image credit: Fei-Fei Li

Outline

- Efficiency (pyramids, separability, steerability)
- Linear algebra
- Bag-of-words
- Frequency analysis (don’t expect to get to)