Descriptors II

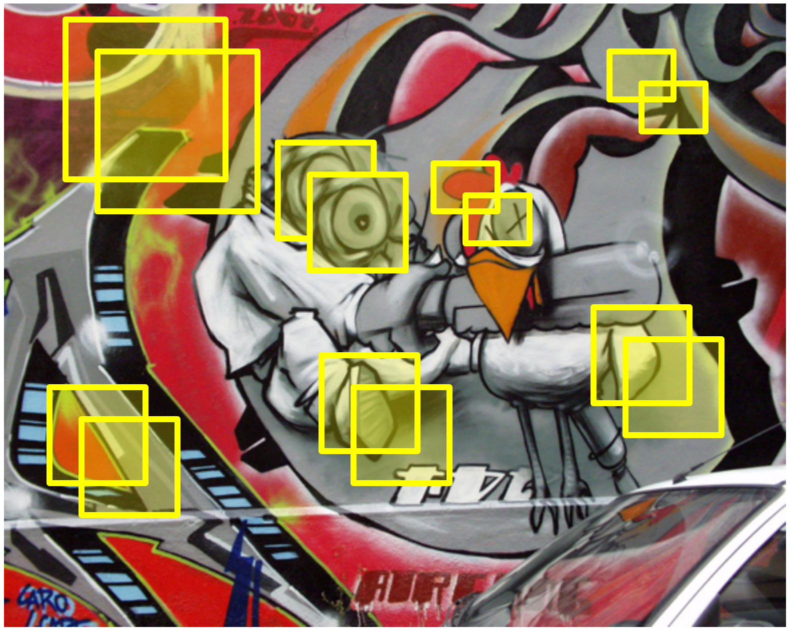
CSE 576

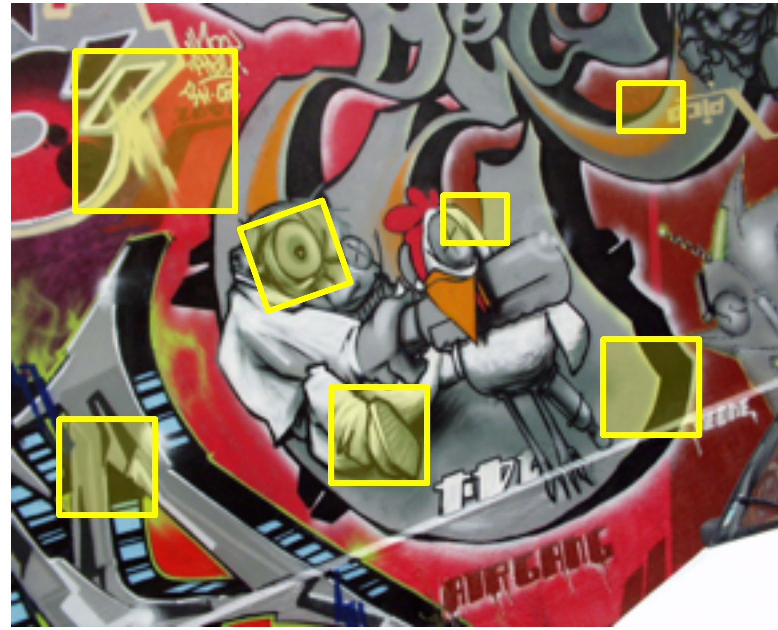
Ali Farhadi

Many slides from Larry Zitnick, Steve Seitz

How can we find corresponding points?

How can we find correspondences?



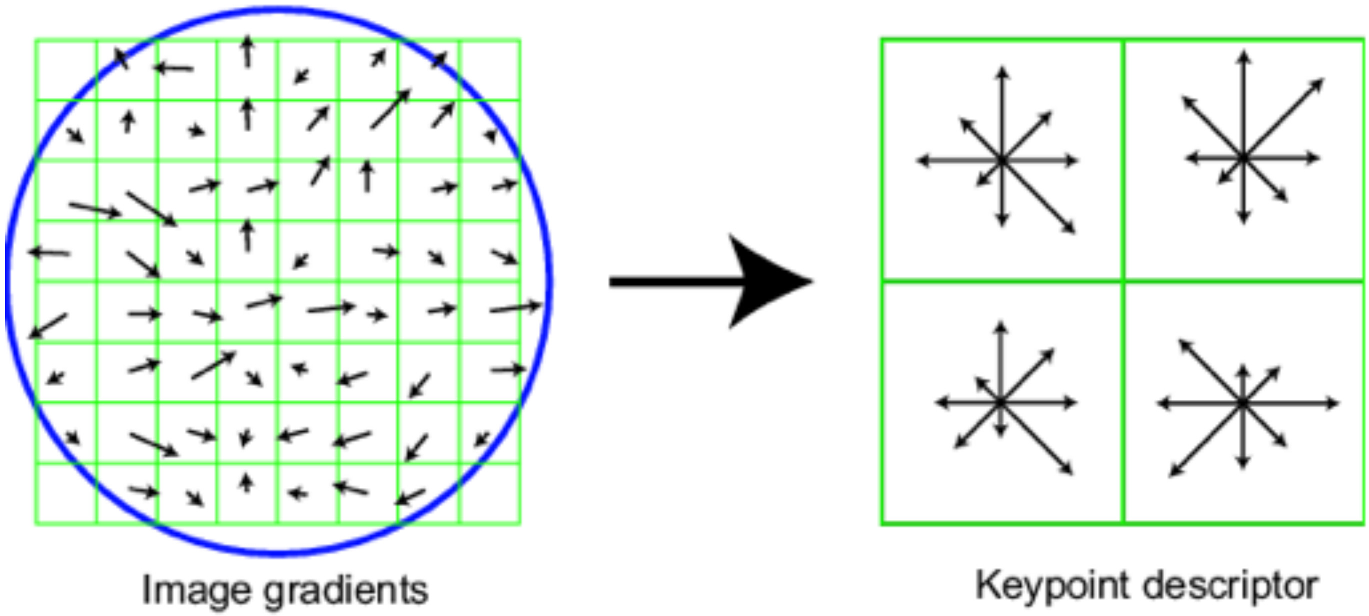


SIFT descriptor

Full version

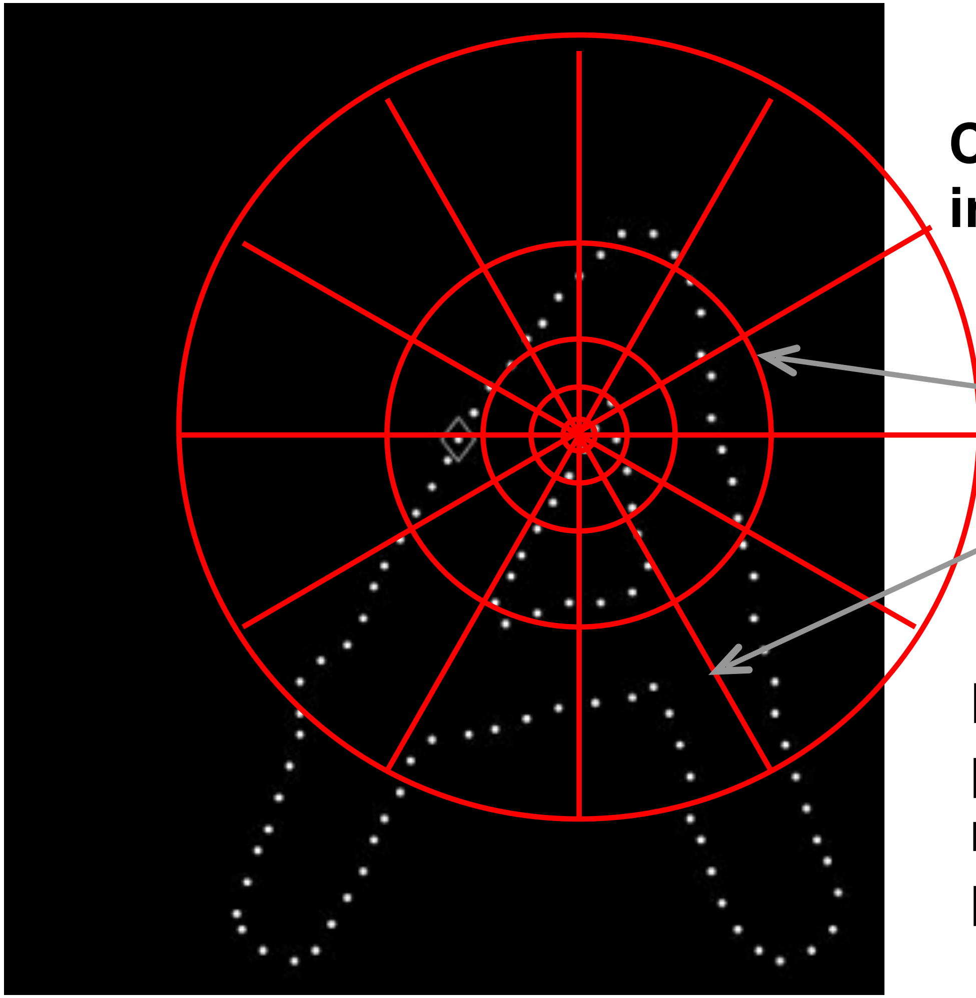
- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor

Vector



Adapted from slide by David Lowe

Local Descriptors: Shape Context



Count the number of points inside each bin, e.g.:

Count = 4

⋮

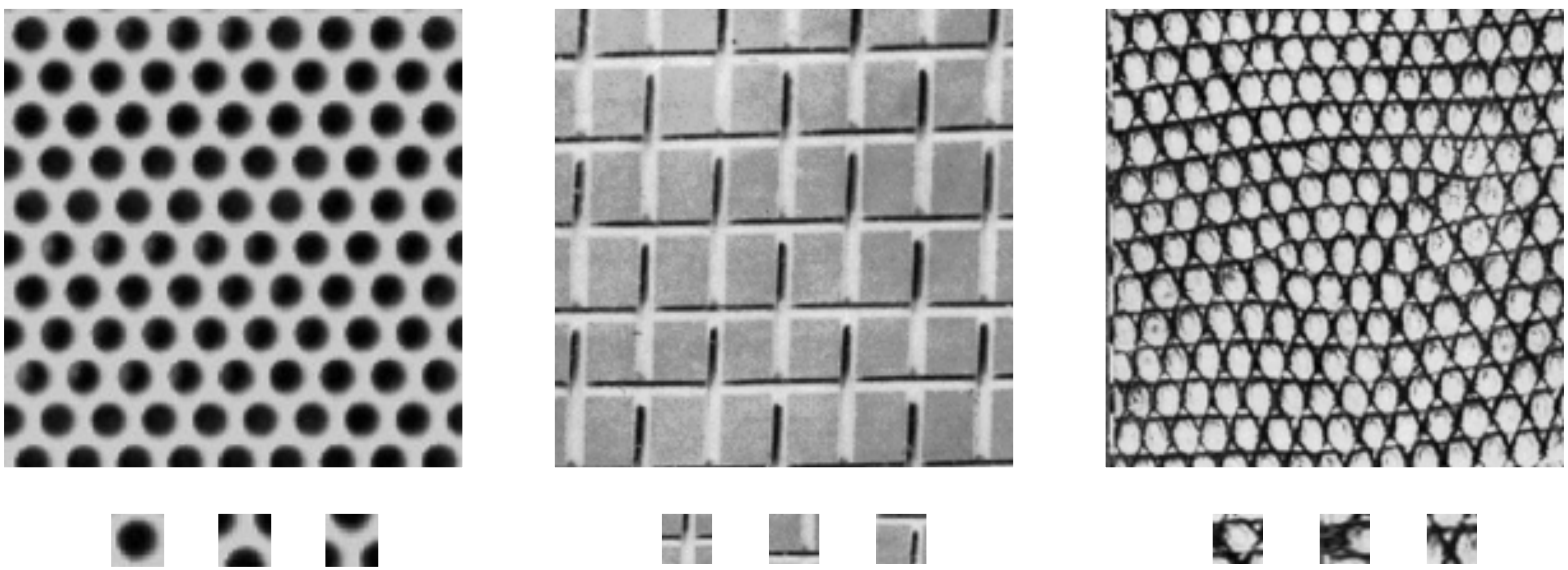
Count = 10

Log-polar binning: more precision for nearby points, more flexibility for farther points.

Belongie & Malik, ICCV 2001

Texture

- Texture is characterized by the repetition of basic elements or *textons*
- For stochastic textures, it is the identity of the textons, not their spatial arrangement, that matters



Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary   Salton & McGill (1983)

Bag-of-words models

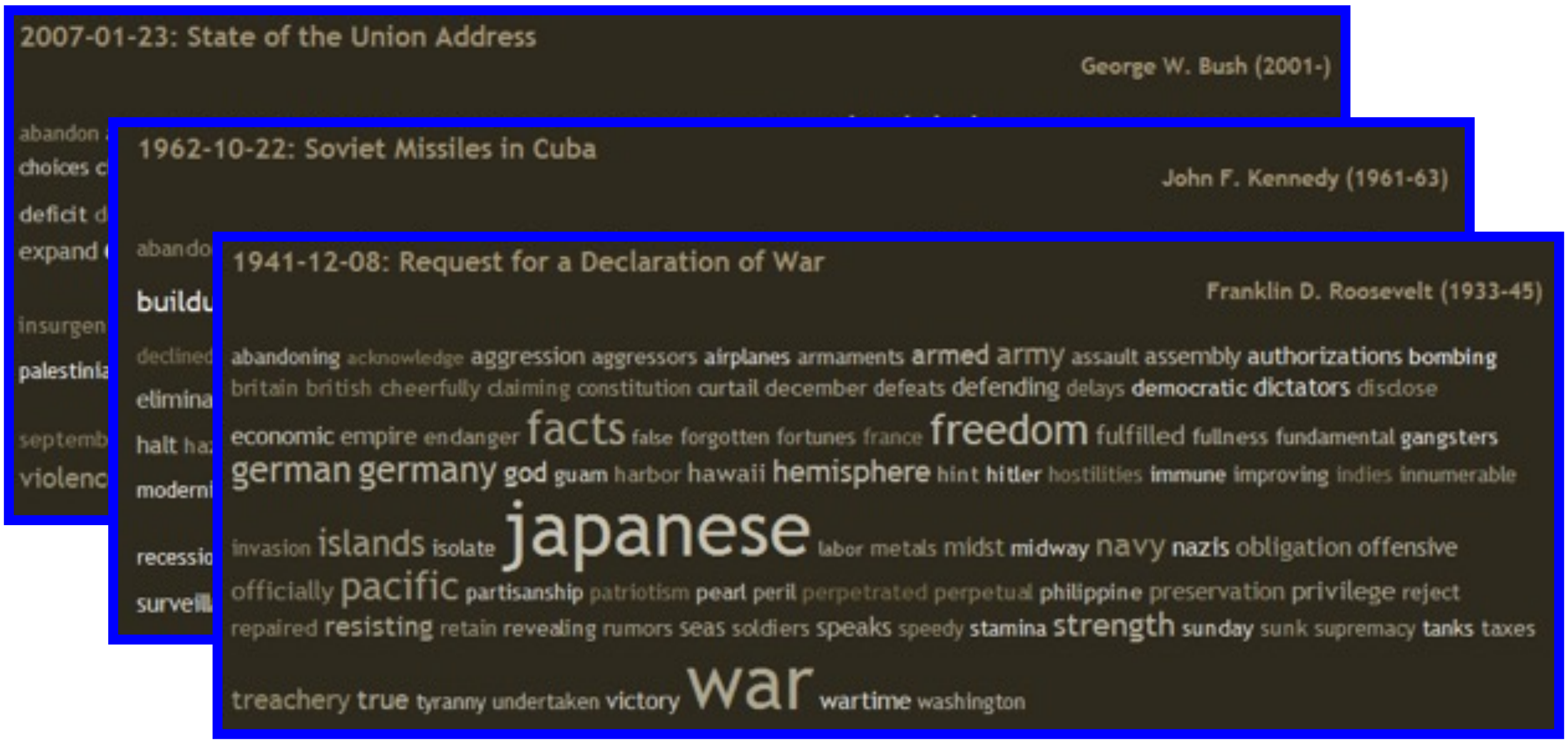
- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

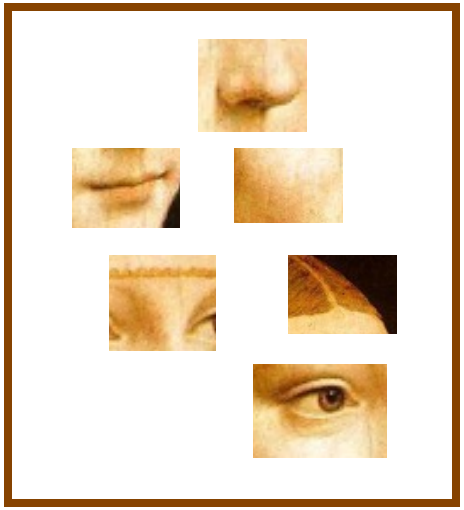
Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)



Bags of features for image classification

1. Extract features



Bags of features for image classification

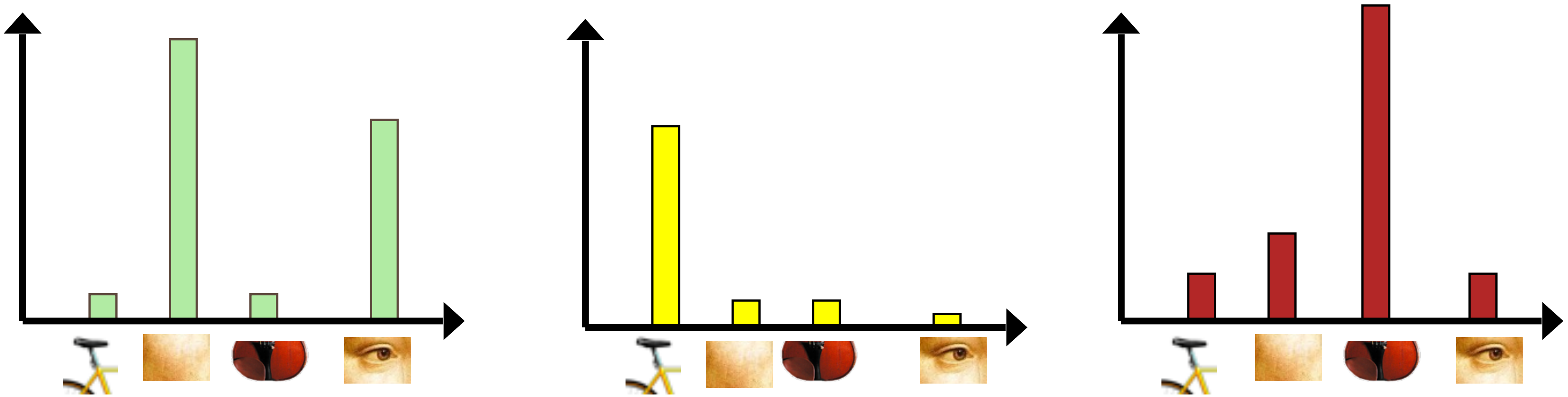
1. Extract features
2. Learn "visual vocabulary"

Bags of features for image classification

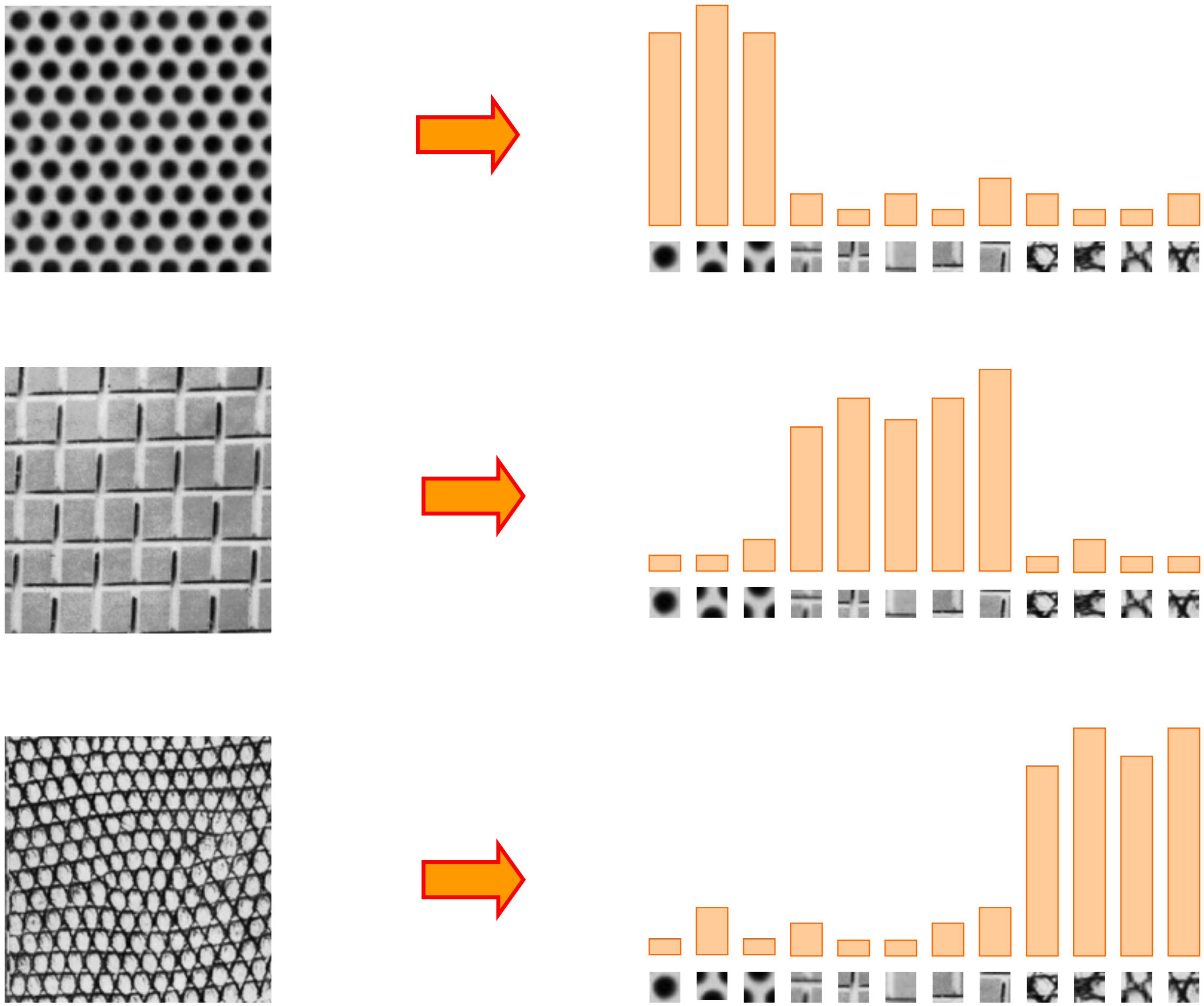
1. Extract features
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary

Bags of features for image classification

1. Extract features
2. Learn "visual vocabulary"
3. Quantize features using visual vocabulary
4. Represent images by frequencies of "visual words"

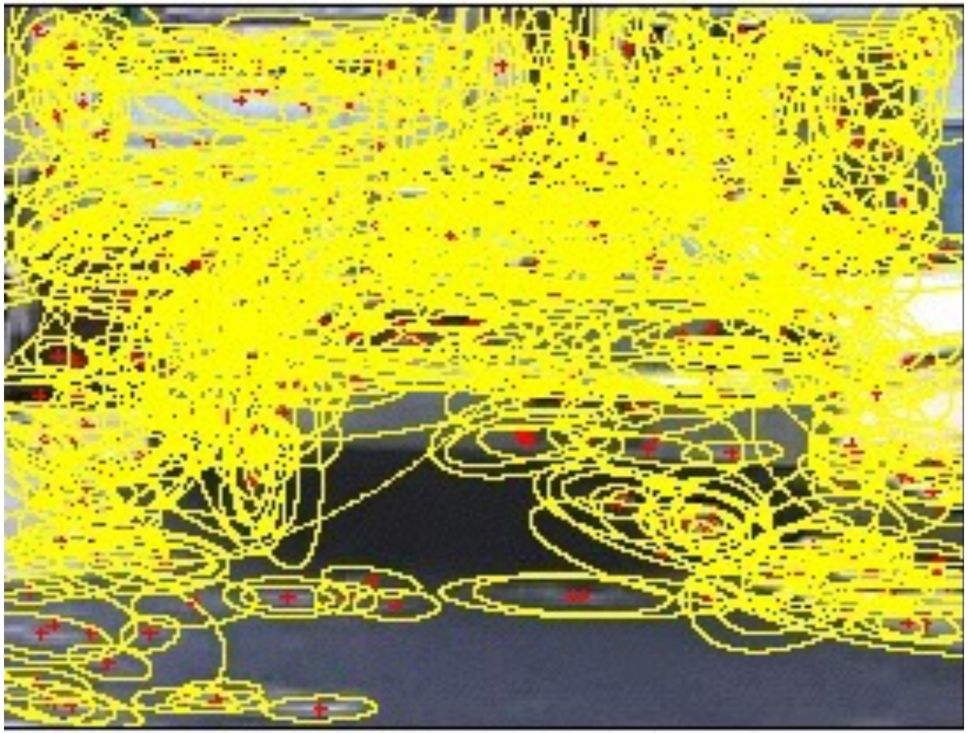


Texture representation



1. Feature extraction

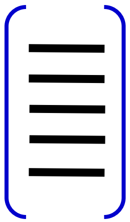
- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
- Interest point detector
  - Csurka et al. 2004
  - Fei-Fei & Perona, 2005
  - Sivic et al. 2005



1. Feature extraction

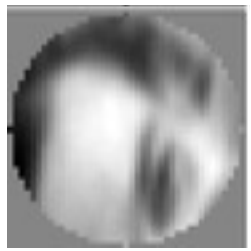
- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
- Interest point detector
  - Csurka et al. 2004
  - Fei-Fei & Perona, 2005
  - Sivic et al. 2005
- Other methods
  - Random sampling (Vidal-Naquet & Ullman, 2002)
  - Segmentation-based patches (Barnard et al. 2003)

1. Feature extraction

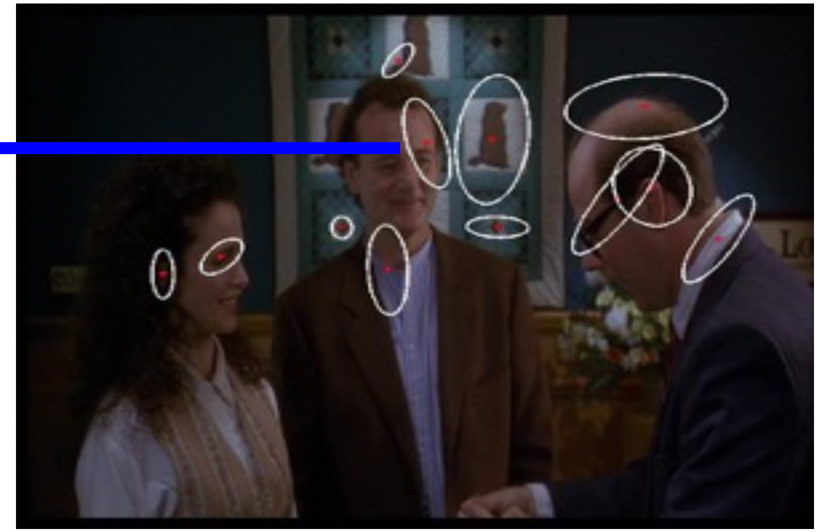


Compute SIFT descriptor

[Lowe'99]



Normalize patch



Detect patches

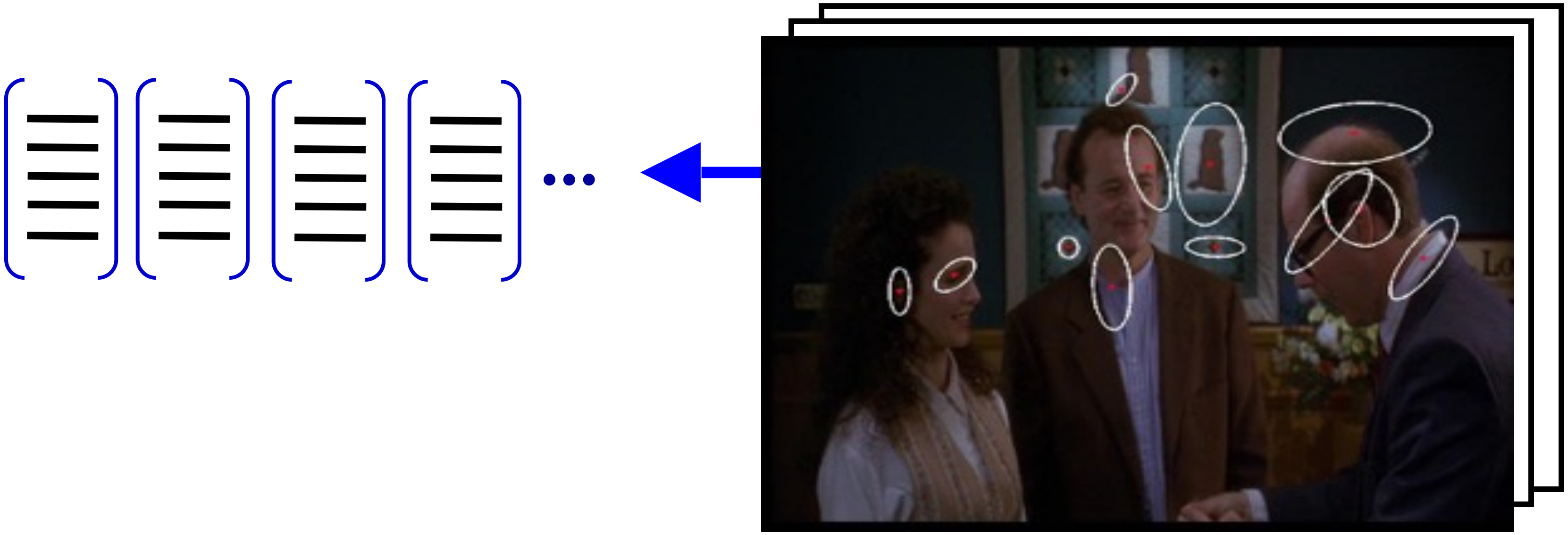
[Mikojaczyk and Schmid '02]

[Mata, Chum, Urban & Pajdla, '02]

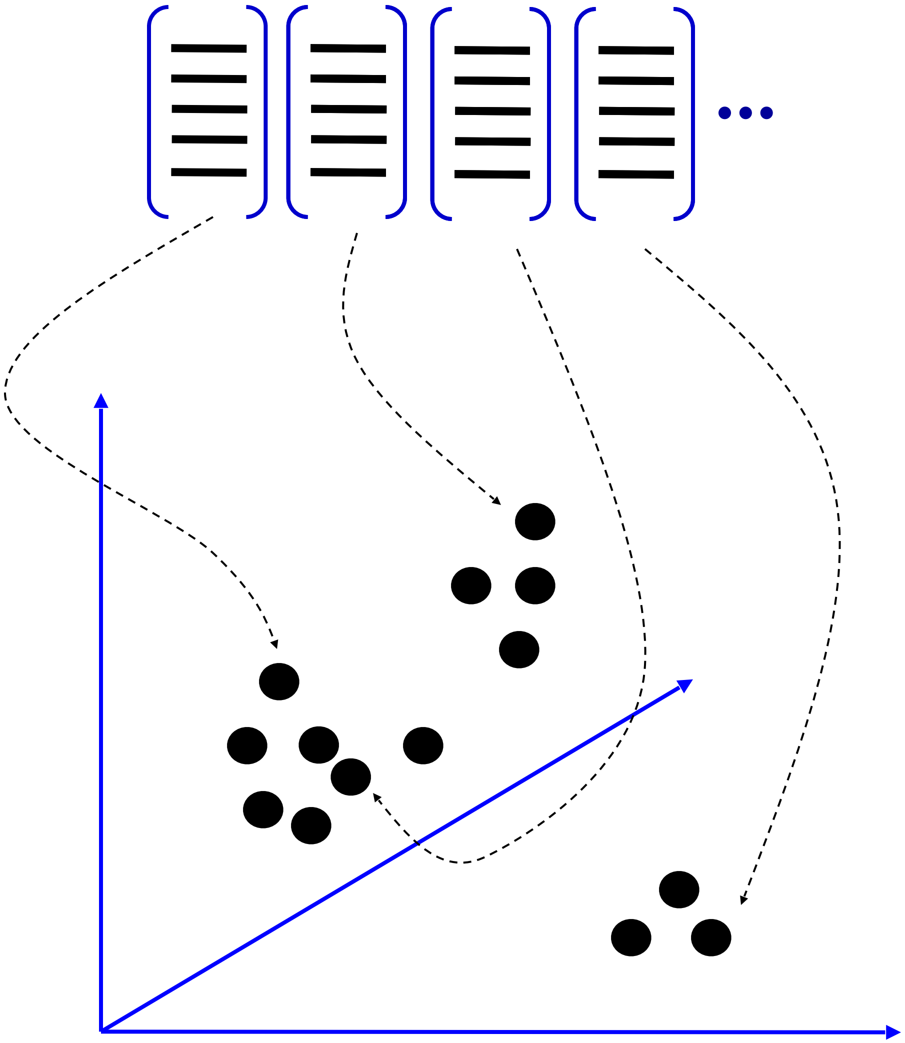
[Sivic & Zisserman, '03]

Slide credit: Josef Sivic

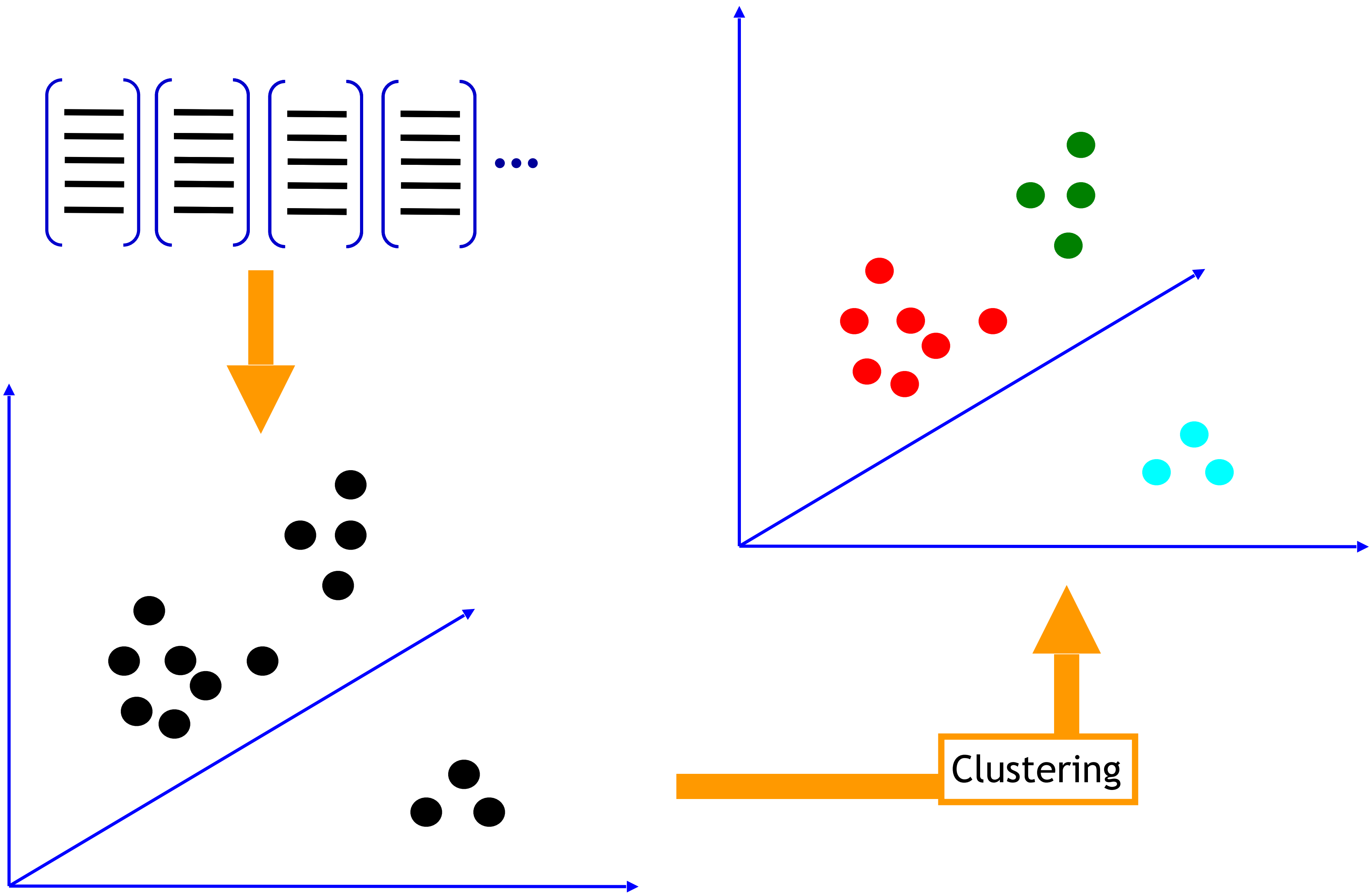
1. Feature extraction



2. Discovering the visual vocabulary

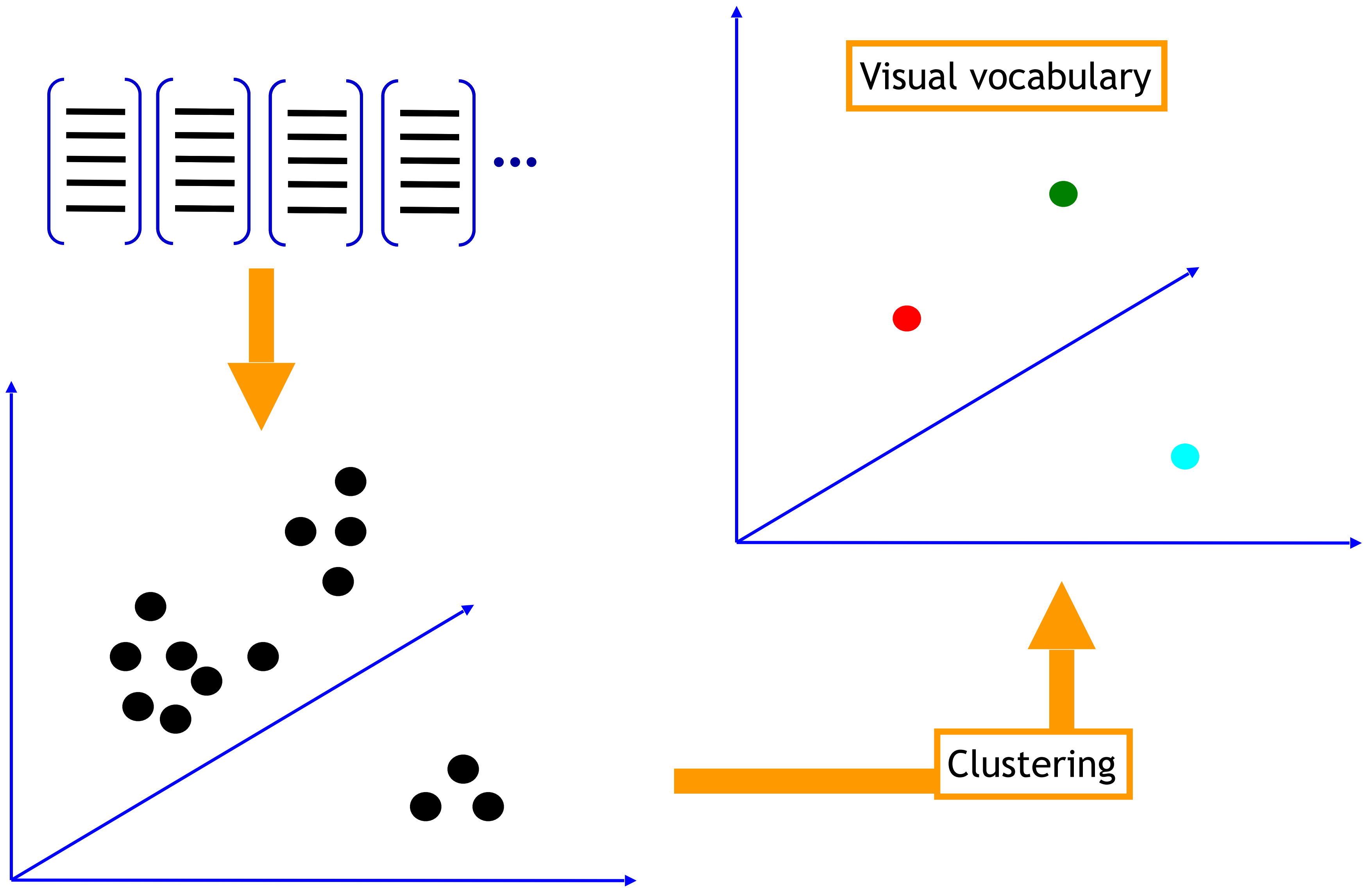


2. Discovering the visual vocabulary



Slide credit: Josef Sivic

2. Discovering the visual vocabulary



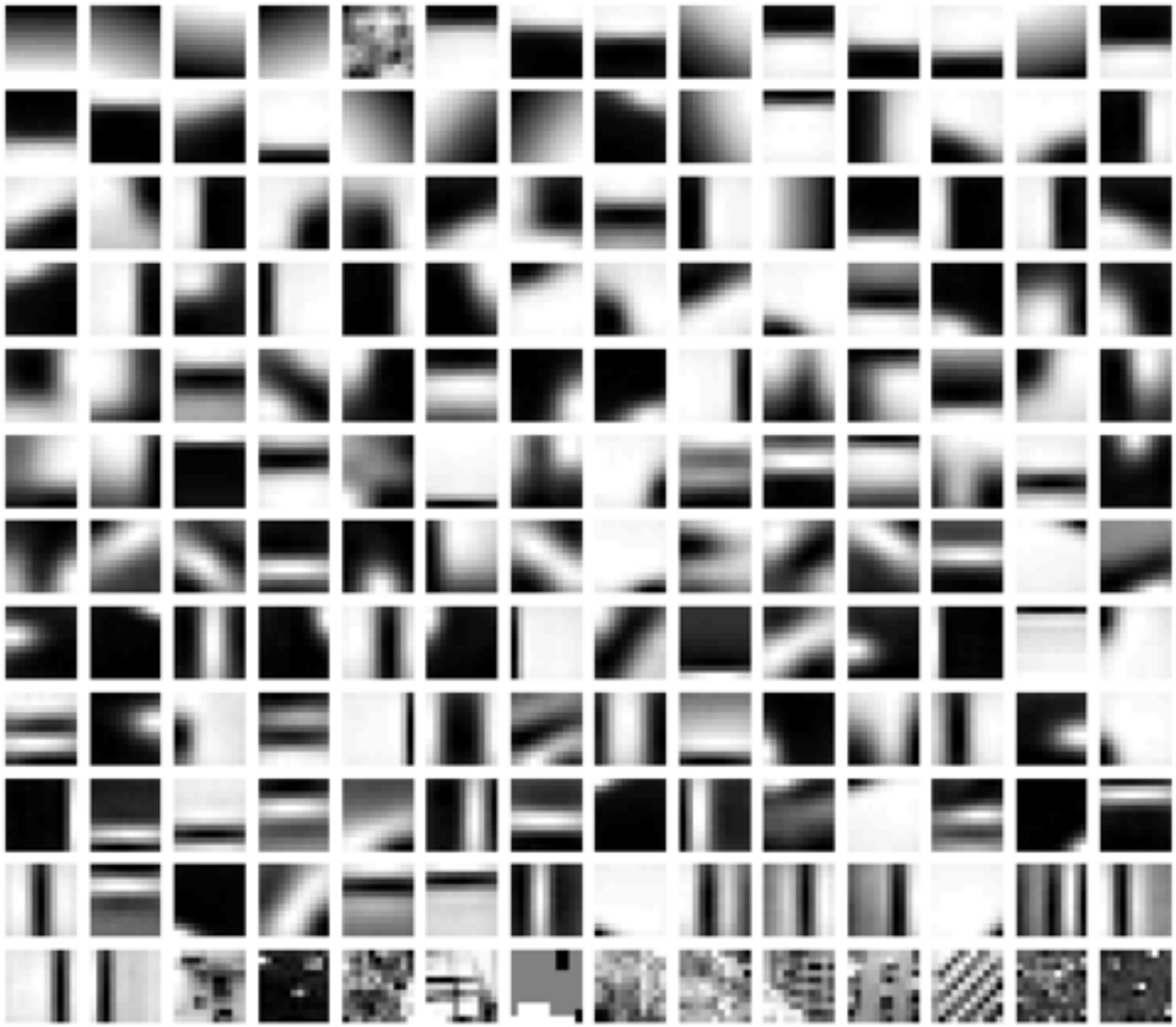
Slide credit: Josef Sivic

Clustering and vector quantization

- Clustering is a common method for learning a visual vocabulary or codebook
  - Unsupervised learning process
  - Each cluster center produced by k-means becomes a codevector
  - Codebook can be learned on separate training set
  - Provided the training set is sufficiently representative, the codebook will be “universal”

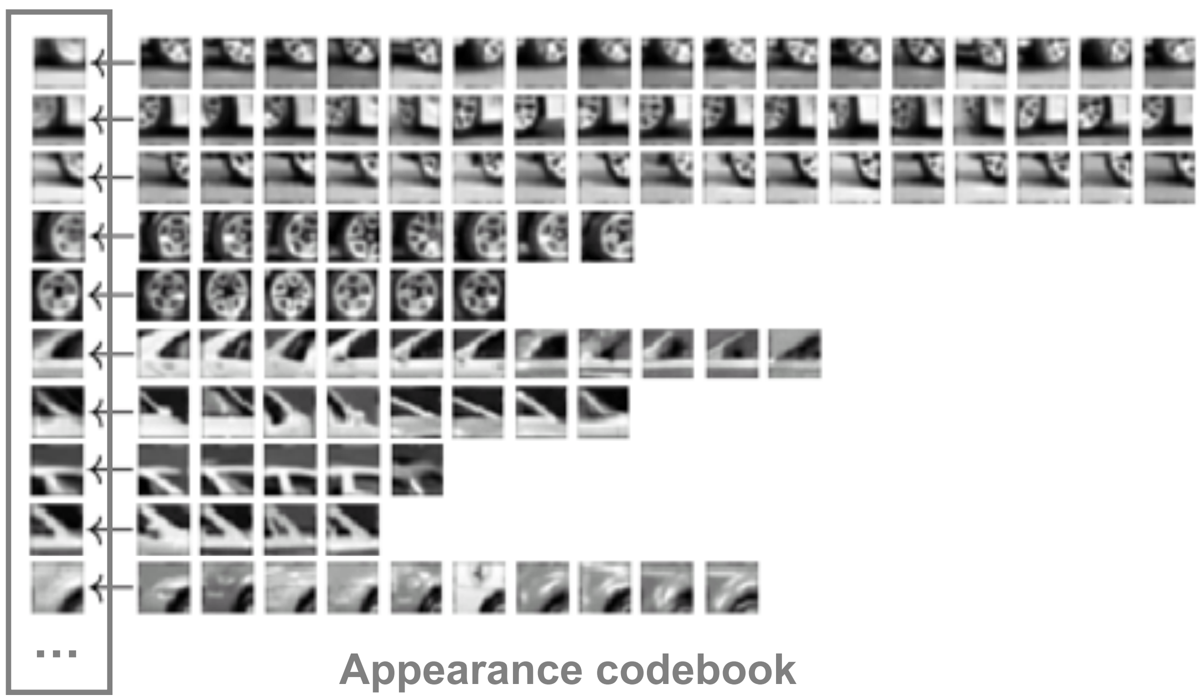
- The codebook is used for quantizing features
  - A *vector quantizer* takes a feature vector and maps it to the index of the nearest codevector in a codebook
  - Codebook = visual vocabulary
  - Codevector = visual word

Example visual vocabulary



Fei-Fei et al. 2005

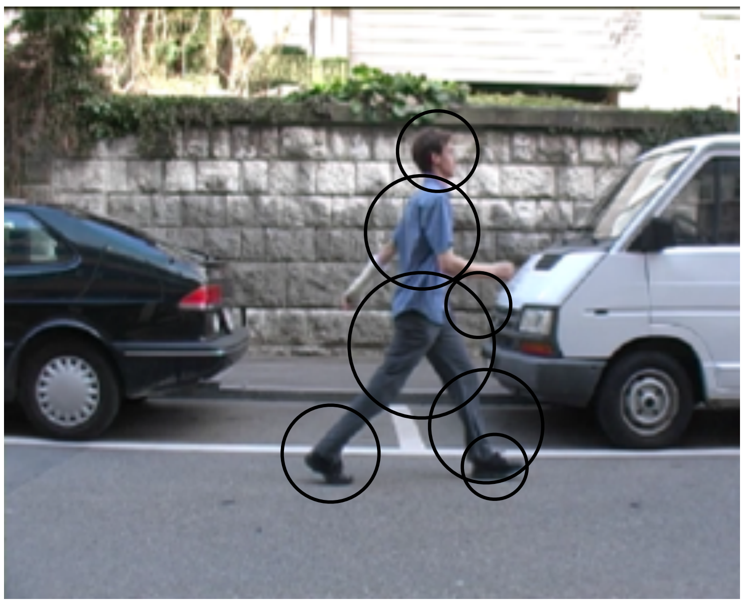
Example codebook

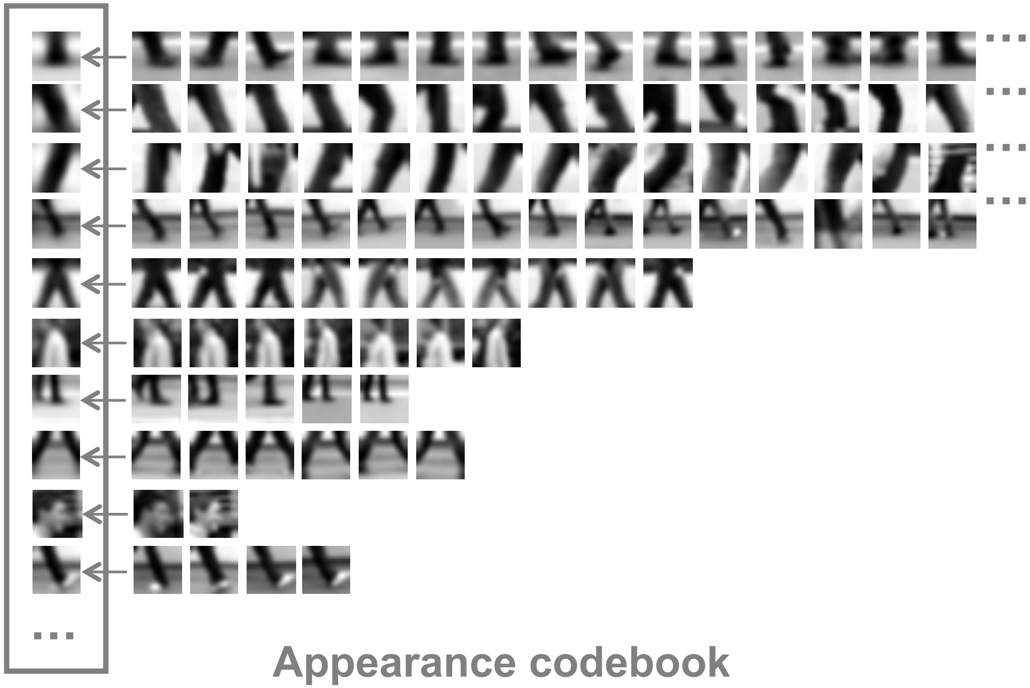


Appearance codebook

Source: B. Leibe

Another codebook

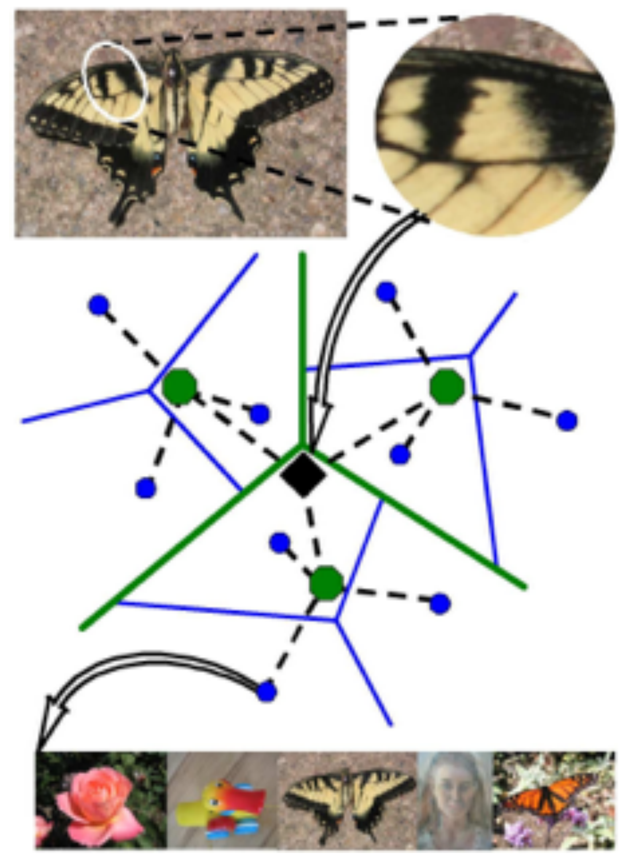




Source: B. Leibe

Visual vocabularies: Issues

- How to choose vocabulary size?
  - Too small: visual words not representative of all patches
  - Too large: quantization artifacts,
  overfitting
- Computational efficiency
  - Vocabulary trees
    (Nister & Stewenius, 2006)



3. Image representation

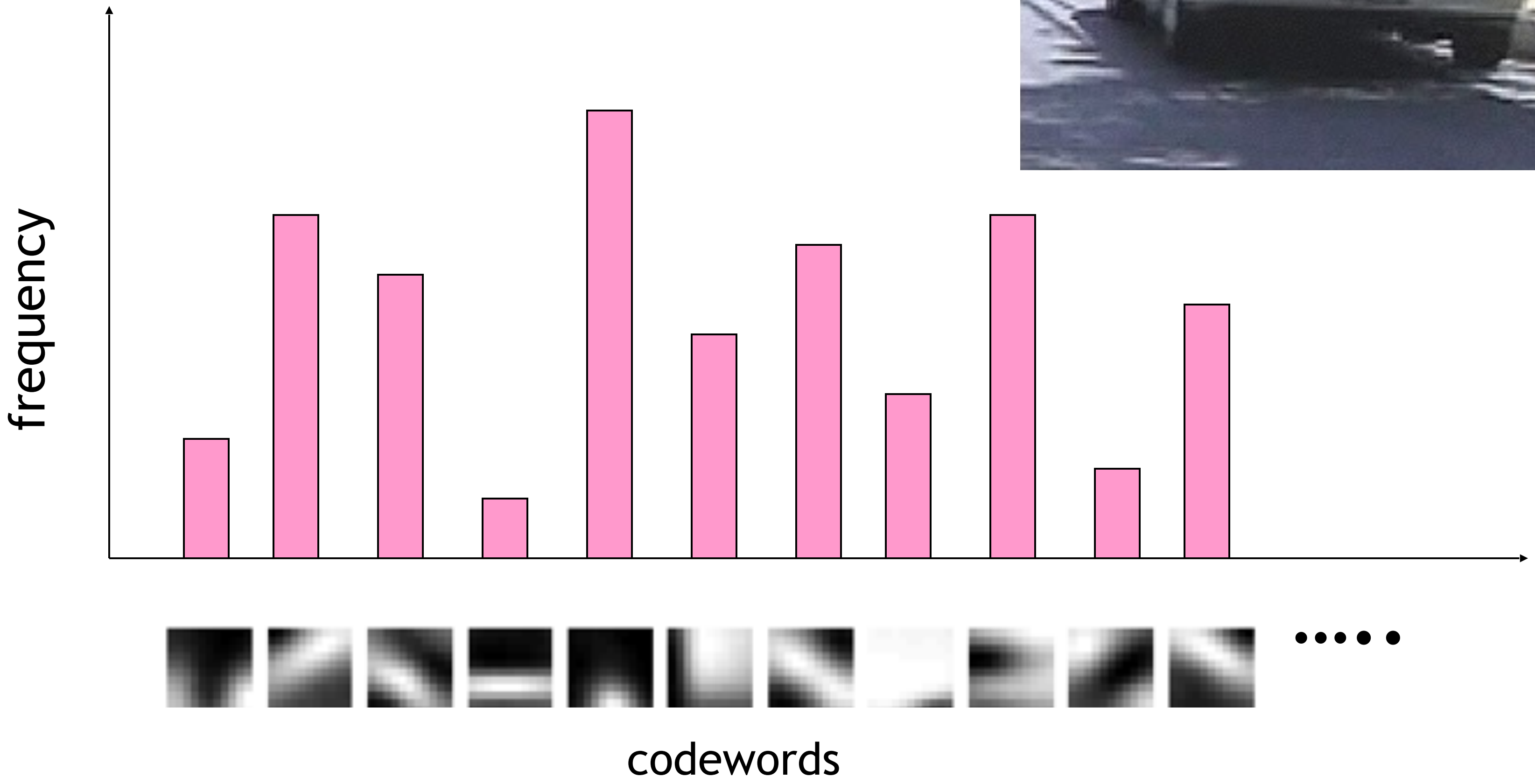
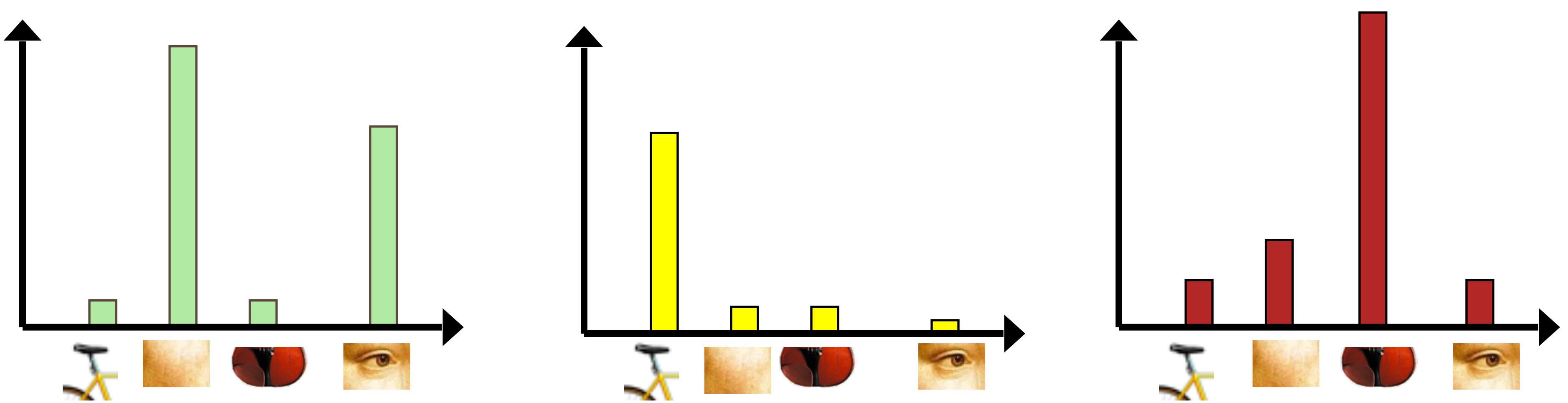


Image classification

- Given the bag-of-features representations of images from different classes, learn a classifier using machine learning



Another Representation: Filter bank

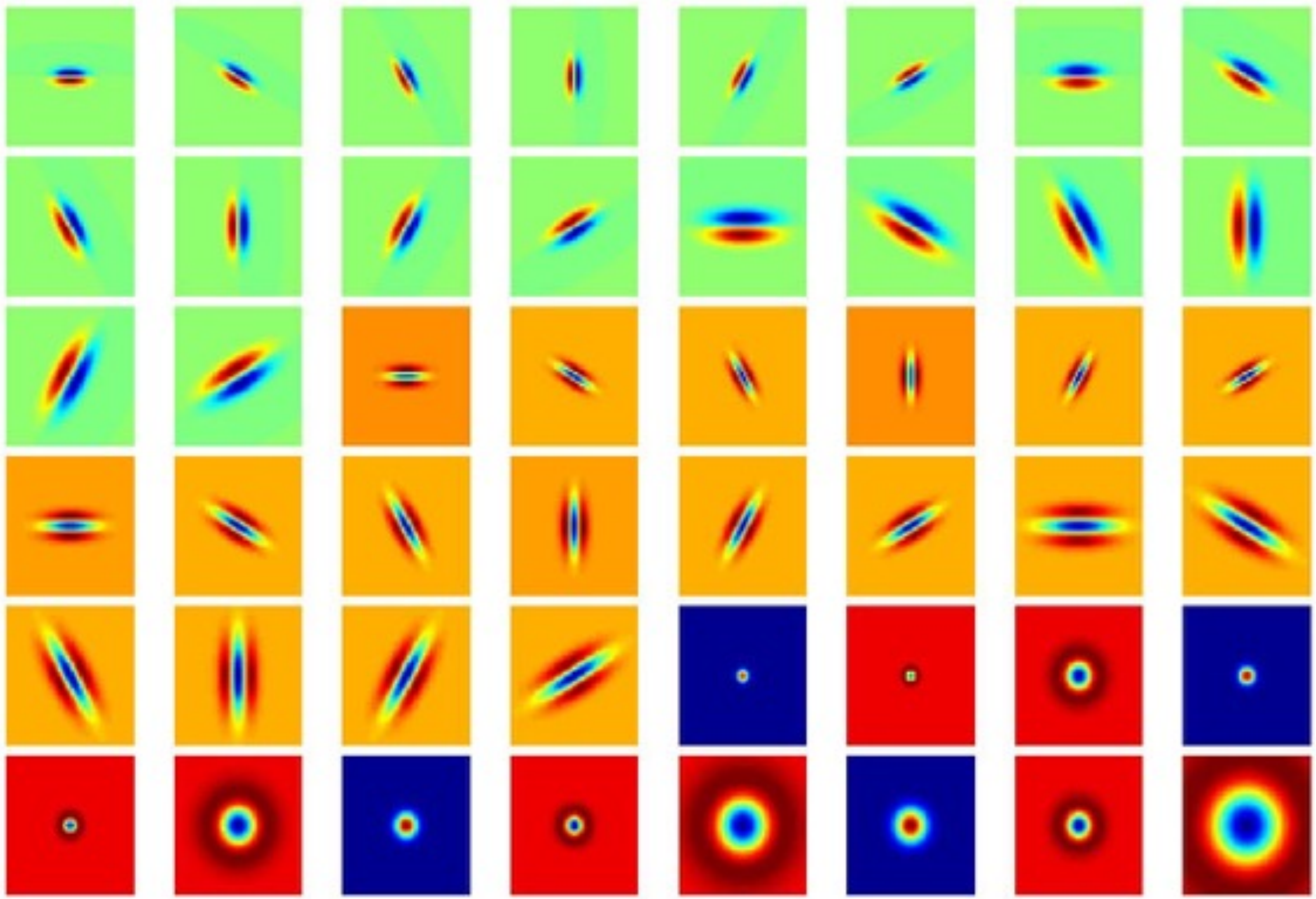
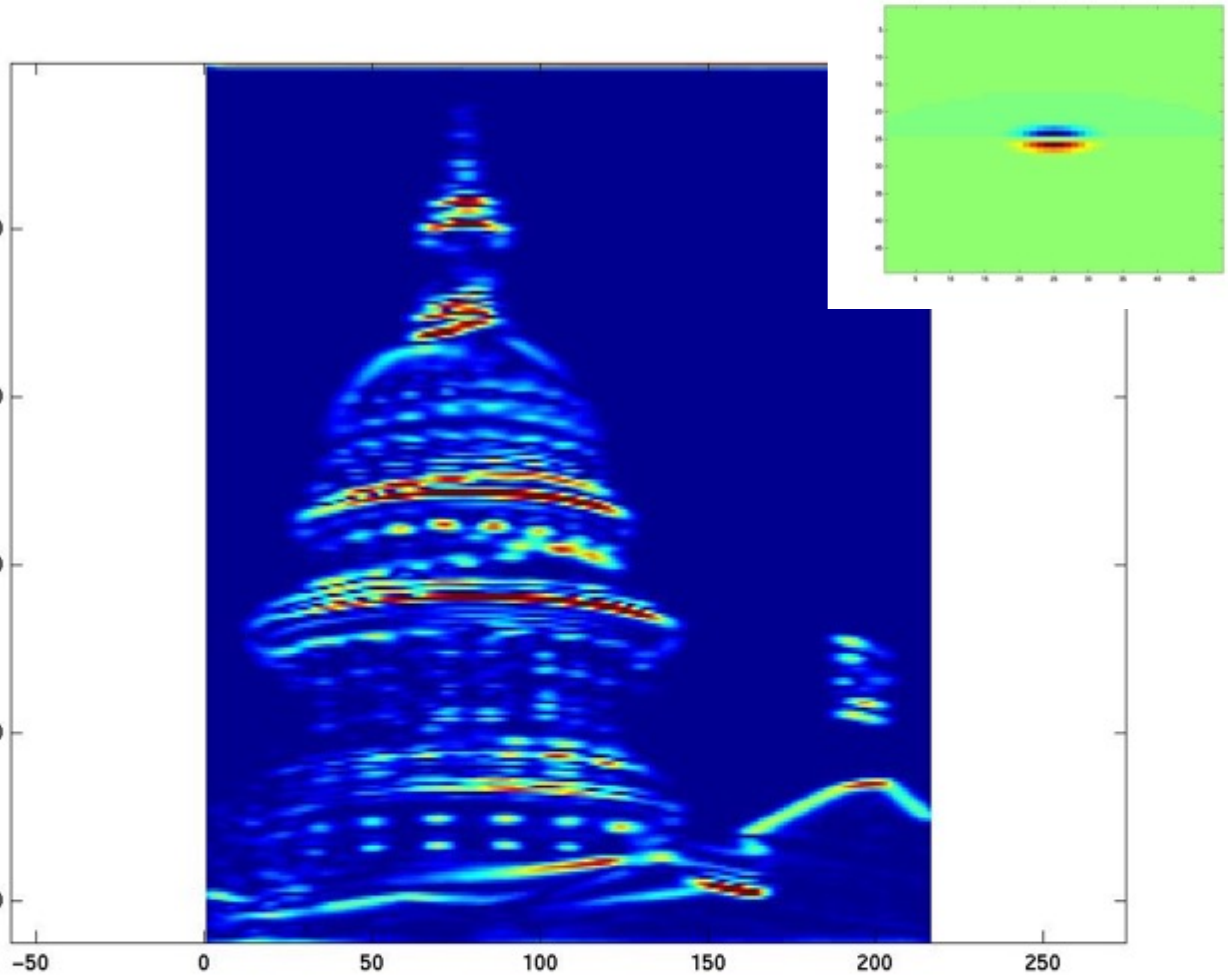
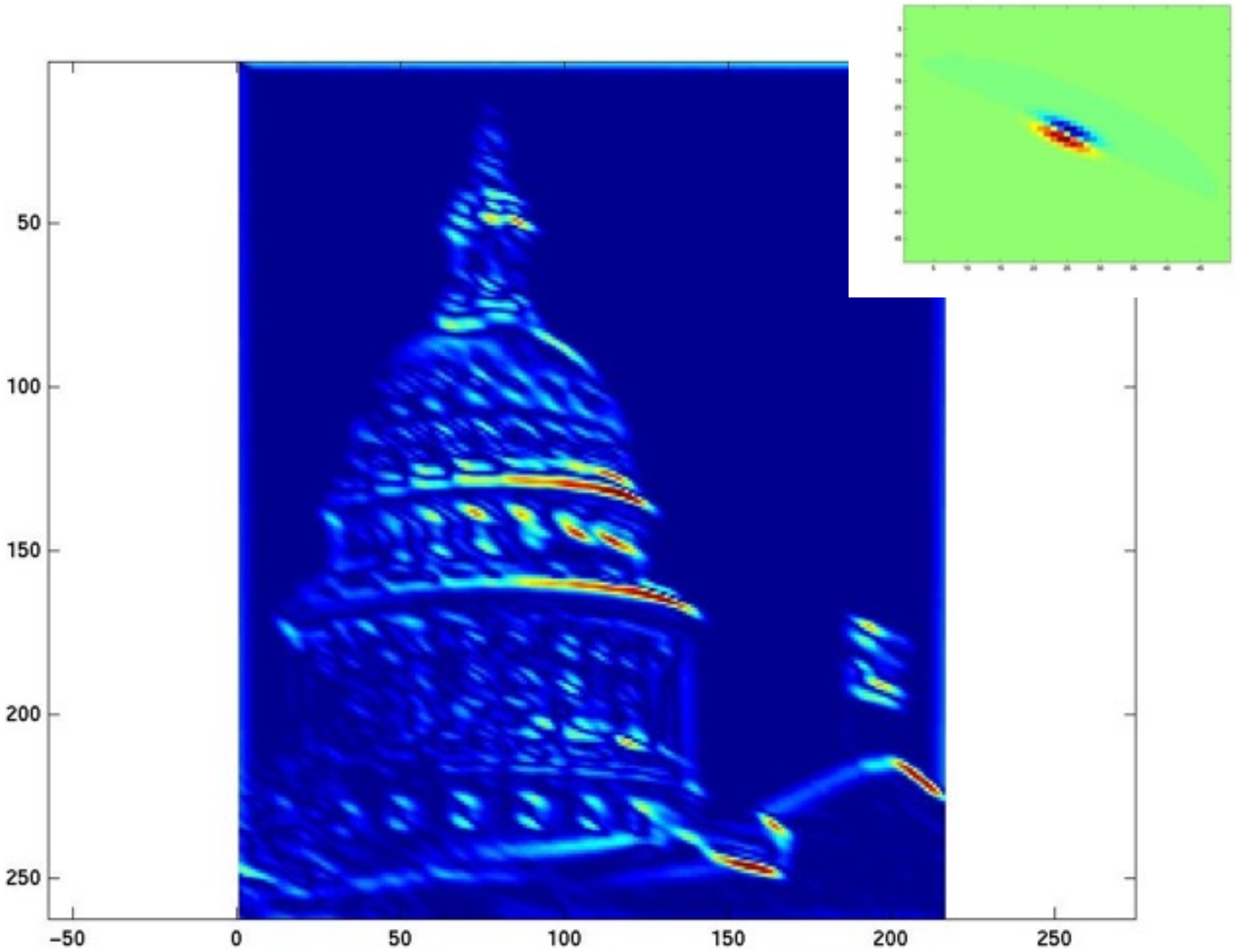


Image from http://www.texasexplorer.com/austincap2.jpg

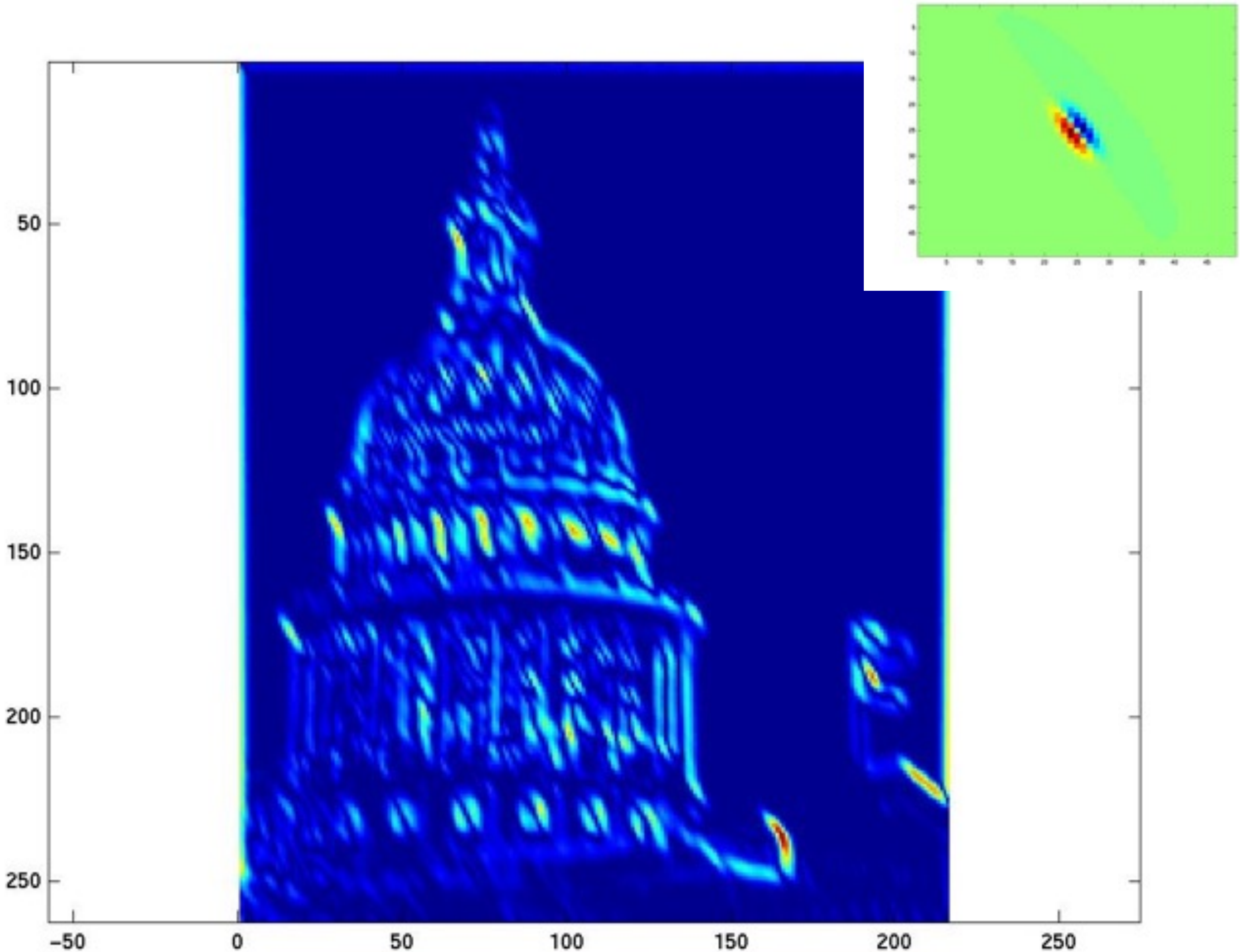
Kristen Grauman



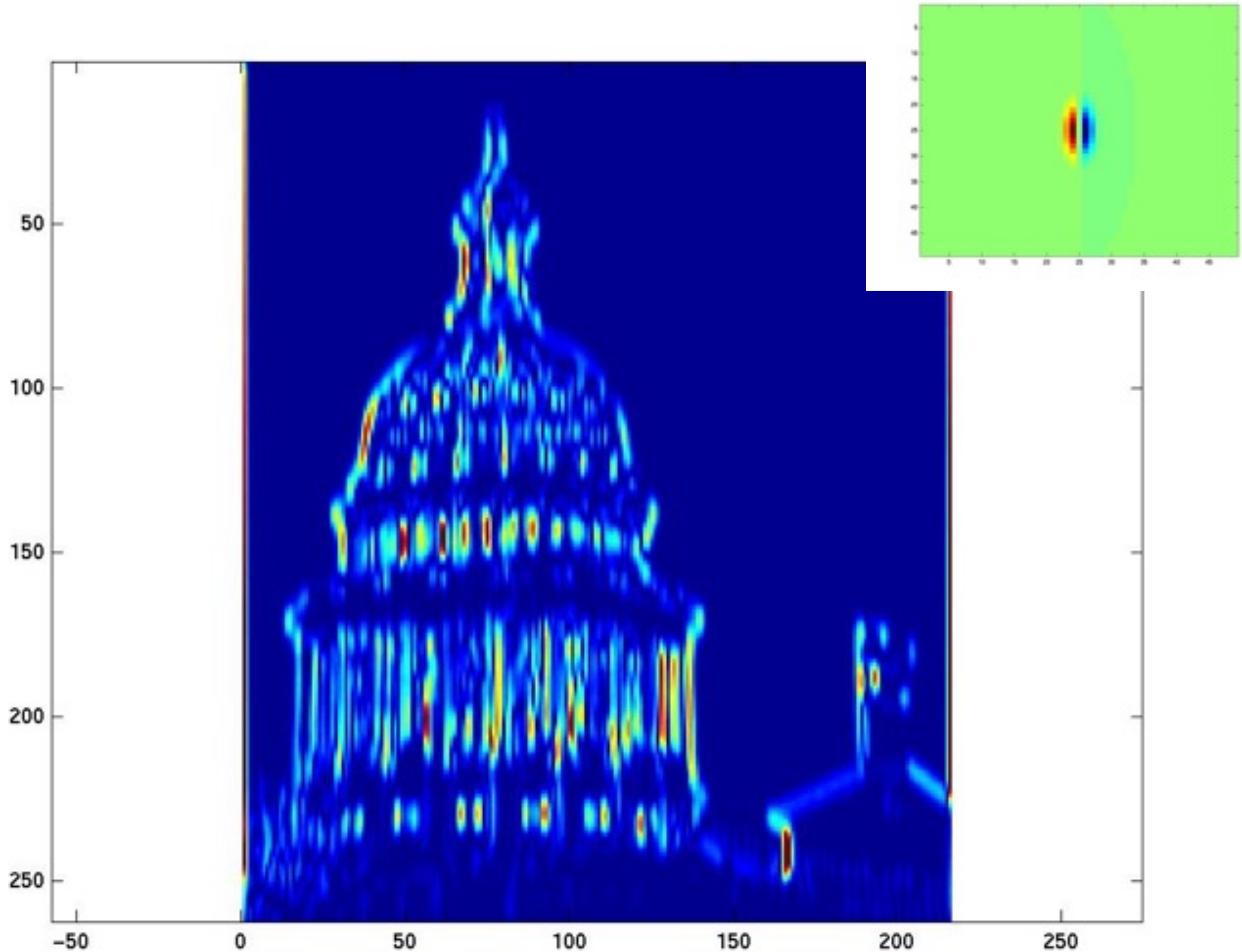
Kristen Grauman



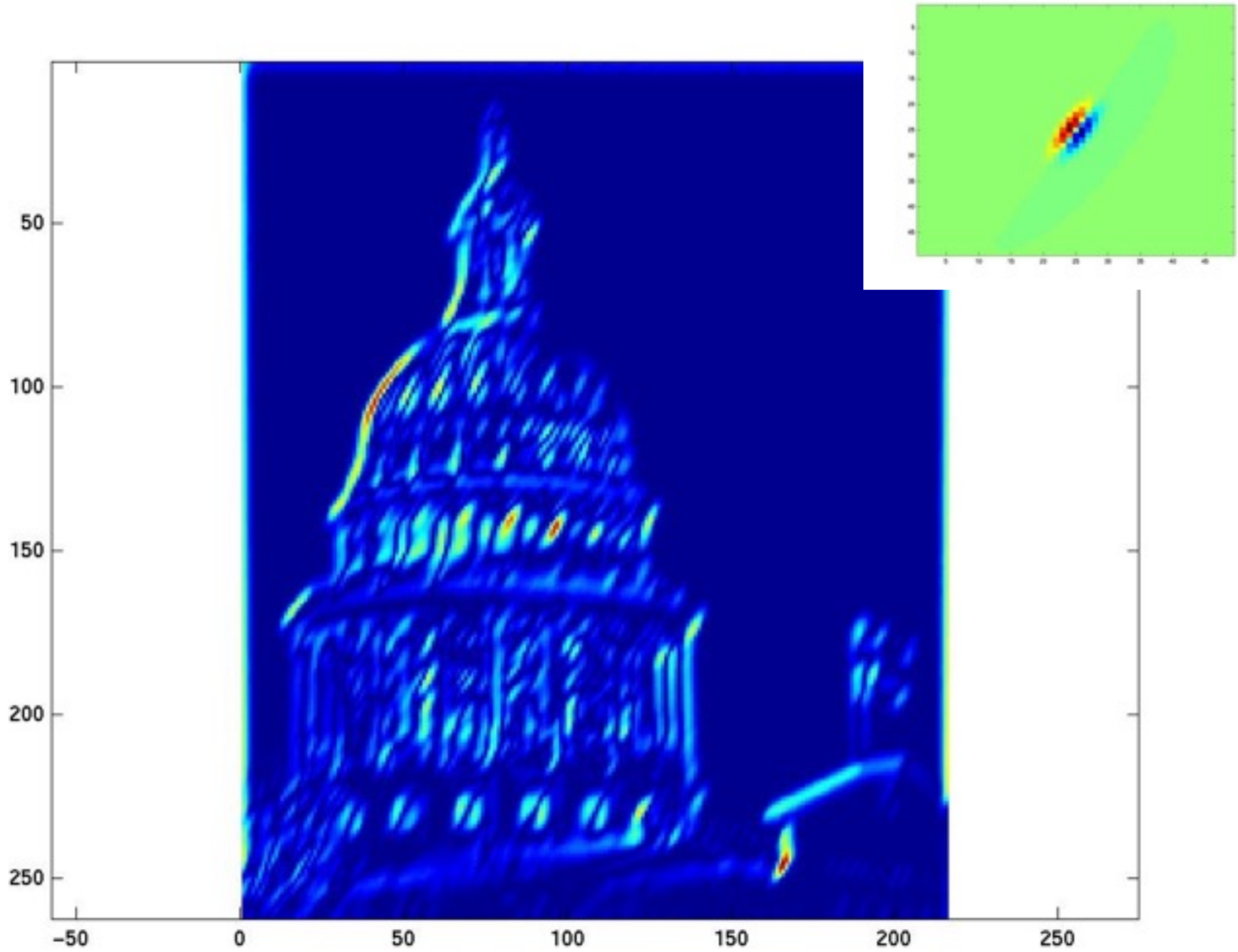
Kristen Grauman



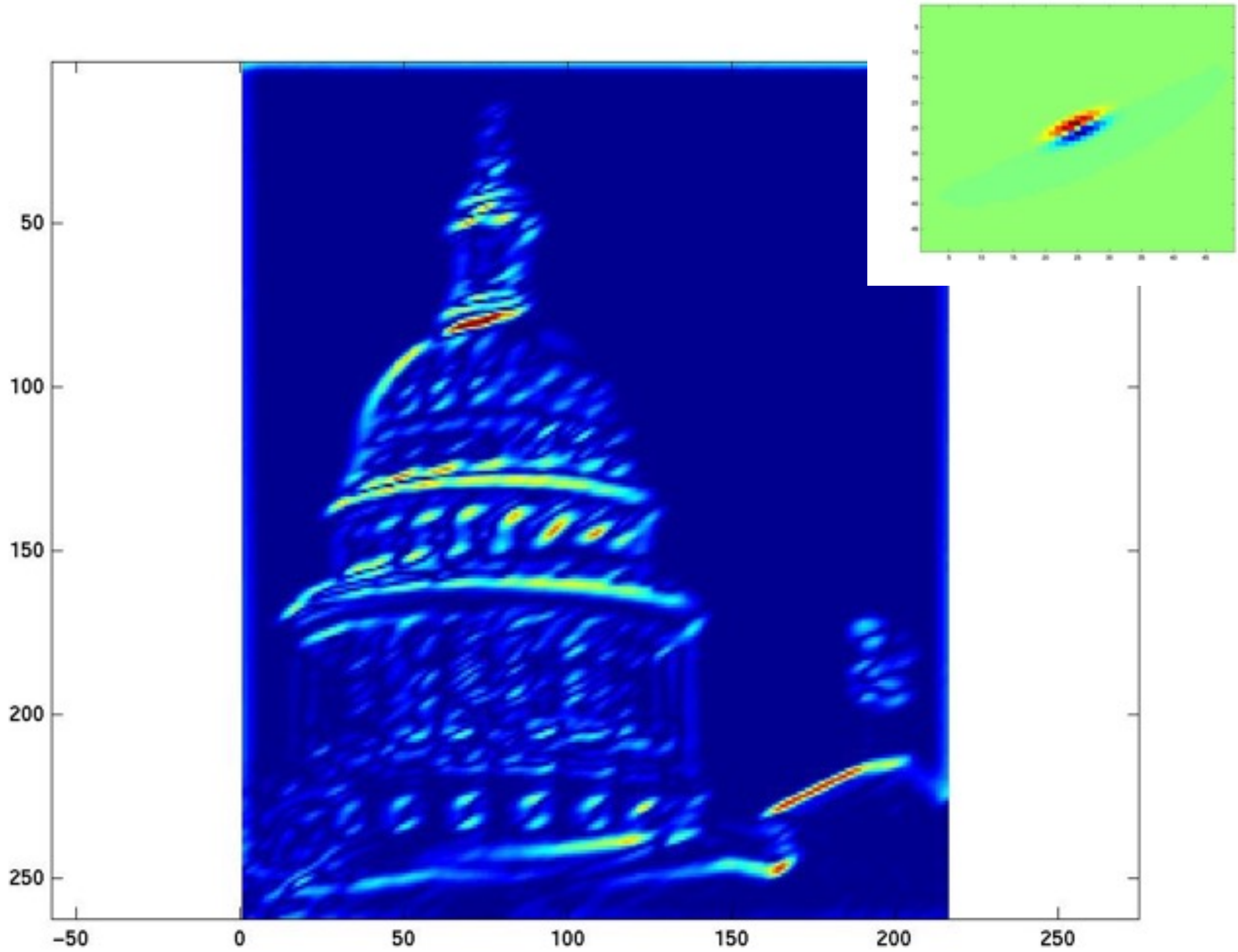
Kristen Grauman



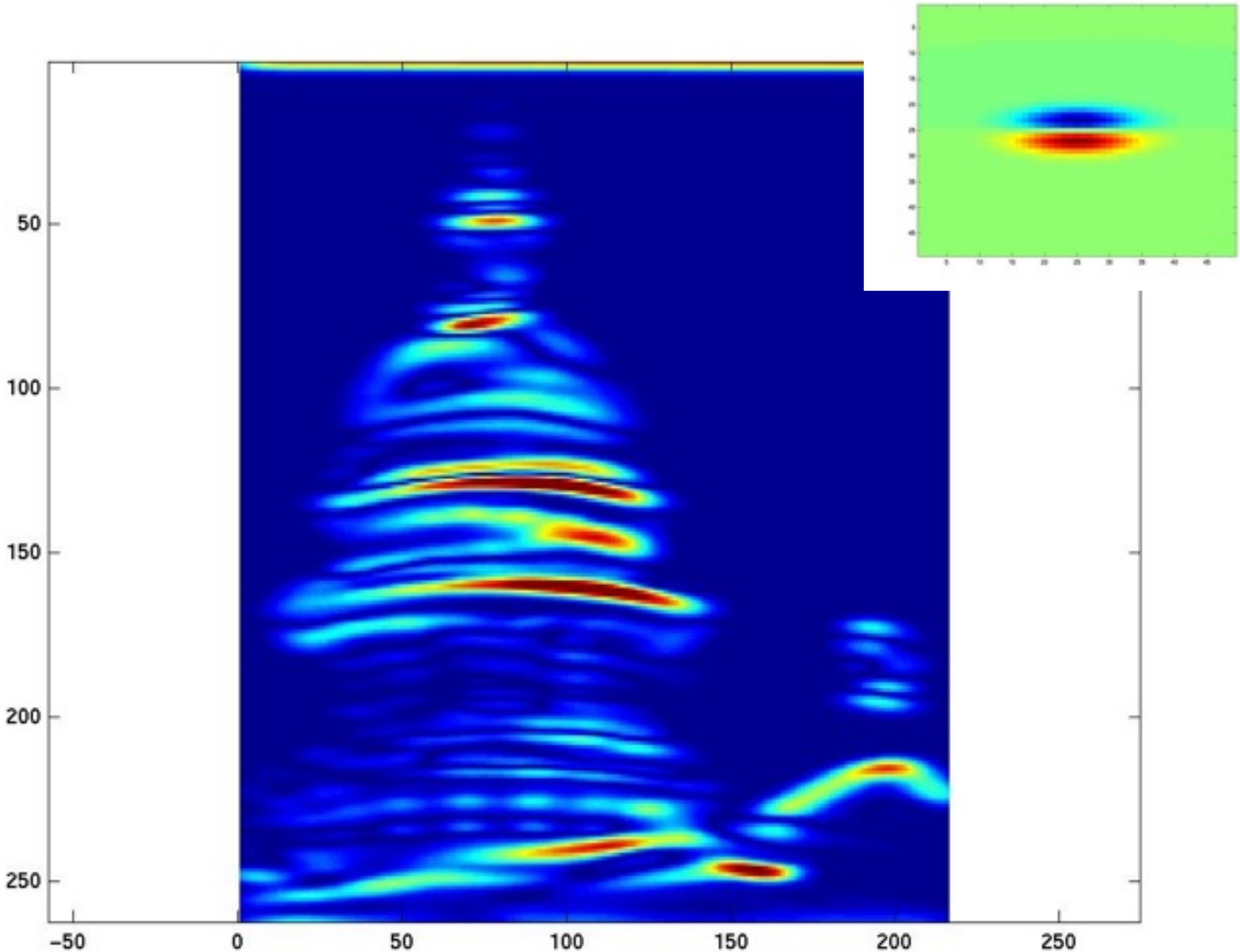
Kristen Grauman



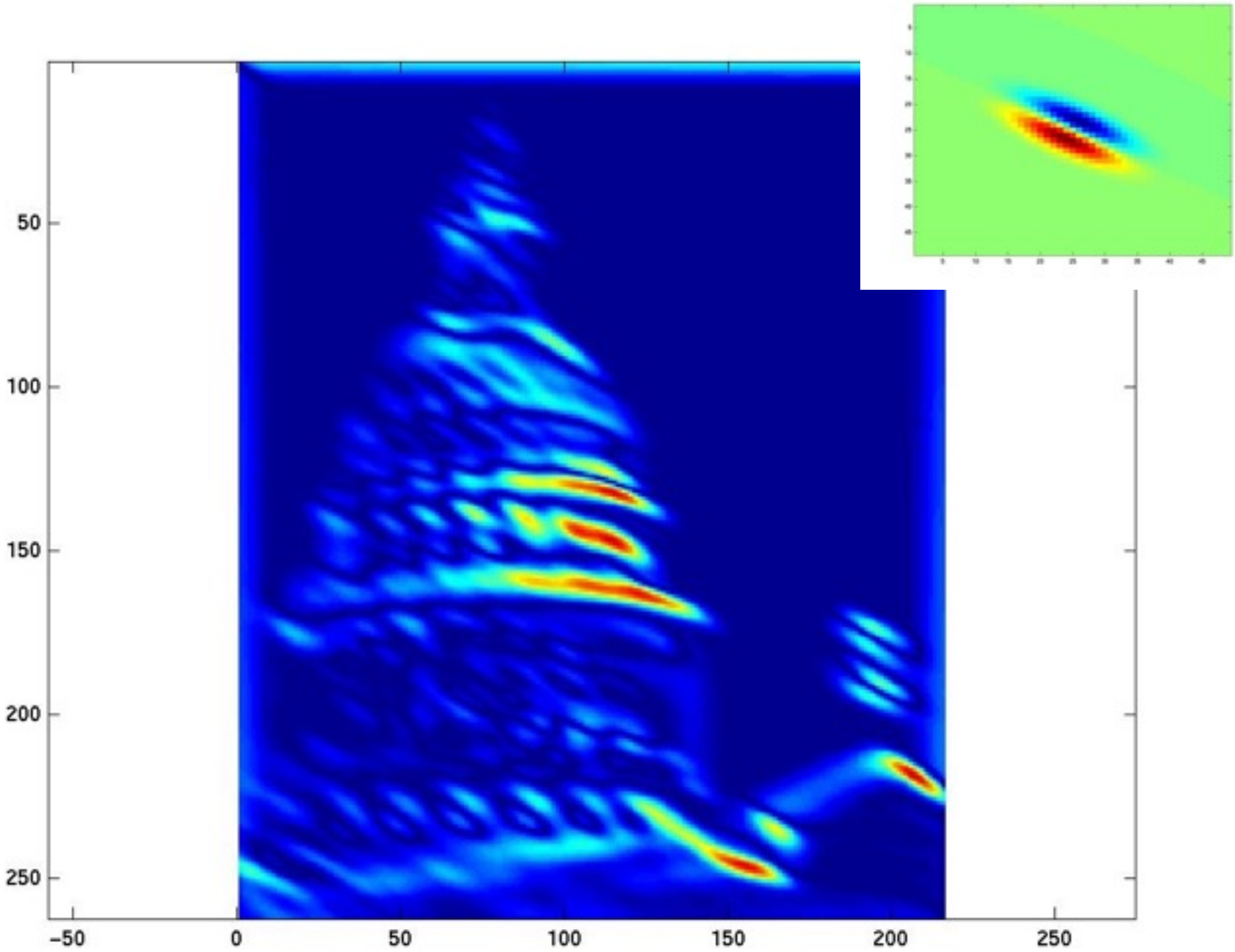
Kristen Grauman



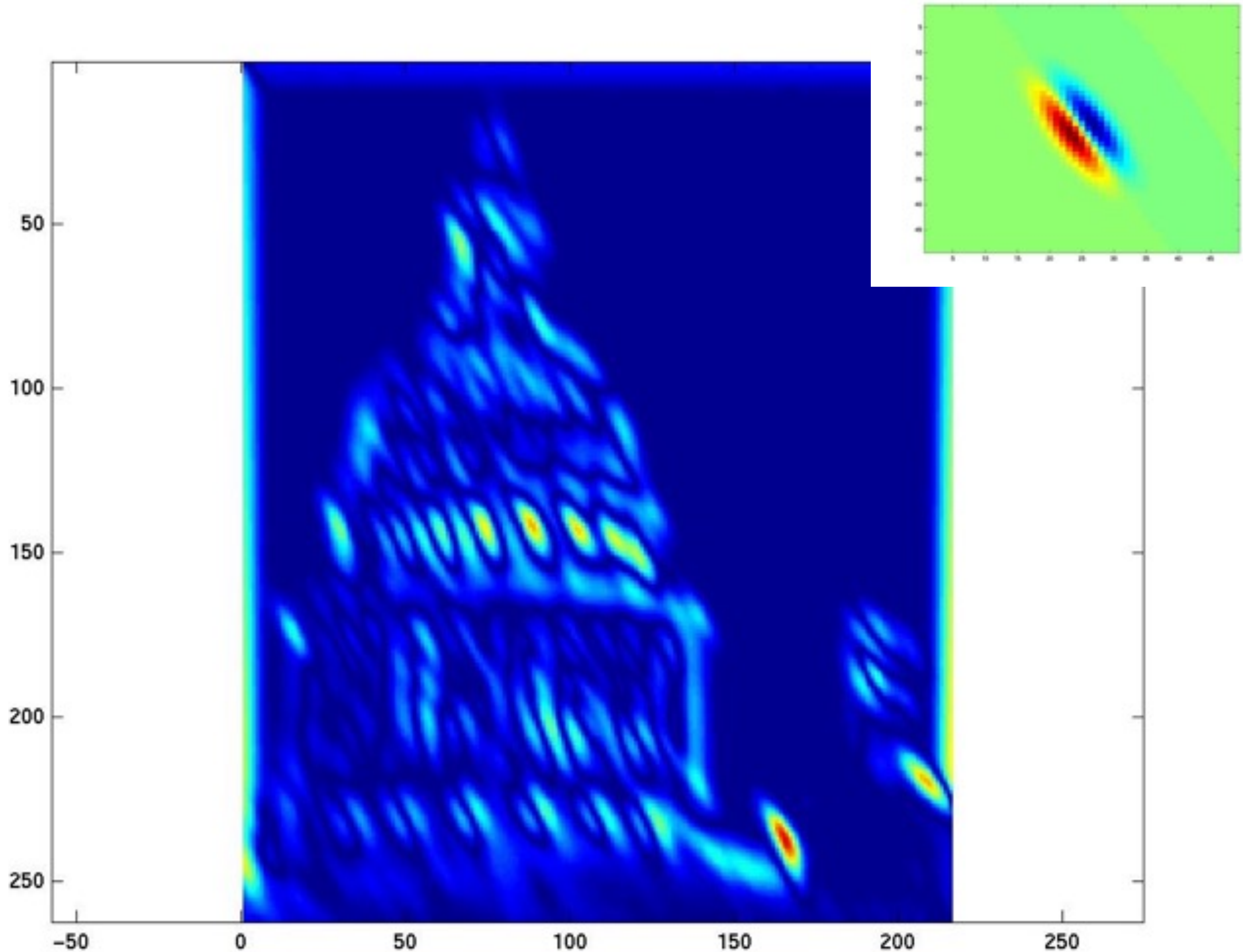
Kristen Grauman



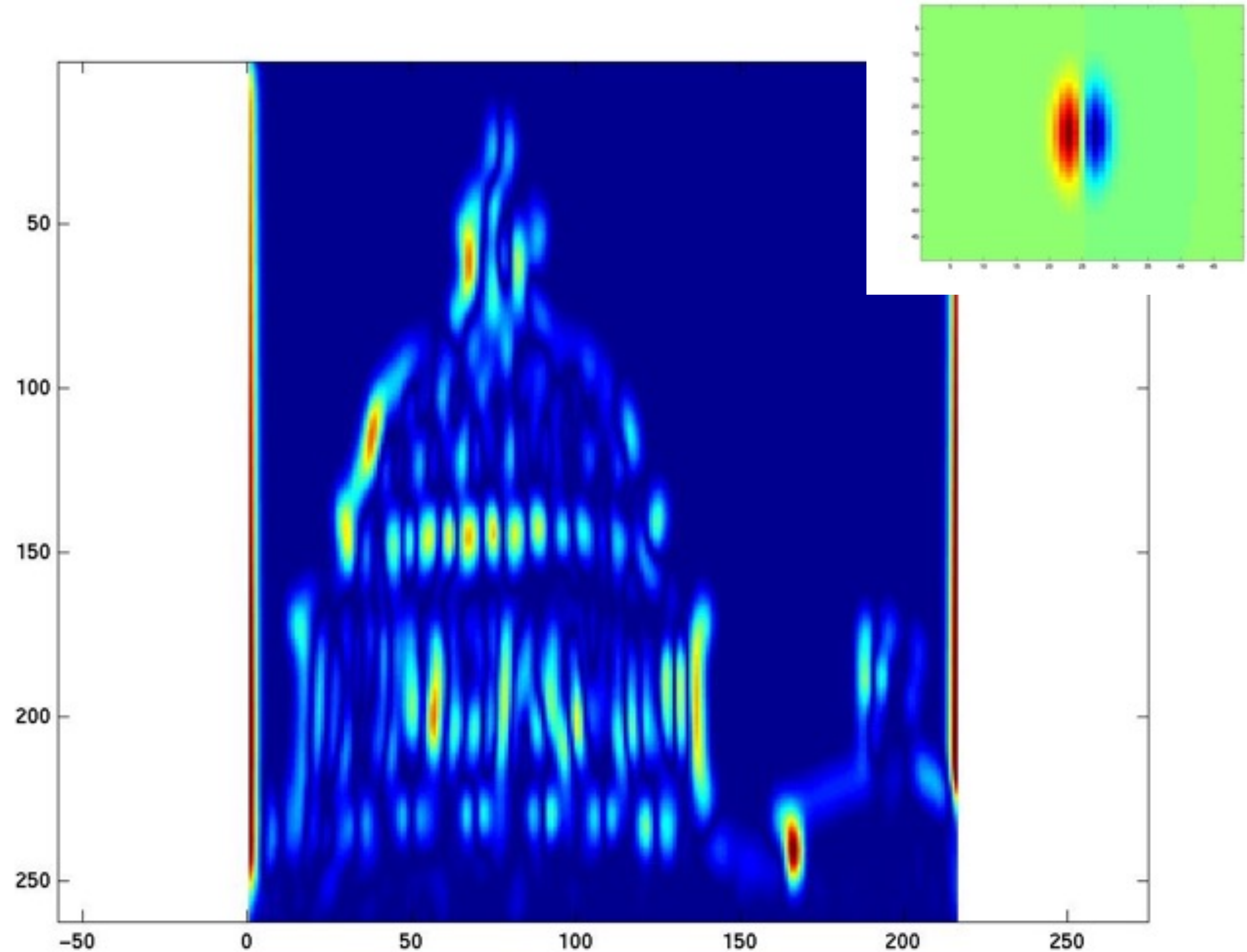
Kristen Grauman



Kristen Grauman



Kristen Grauman

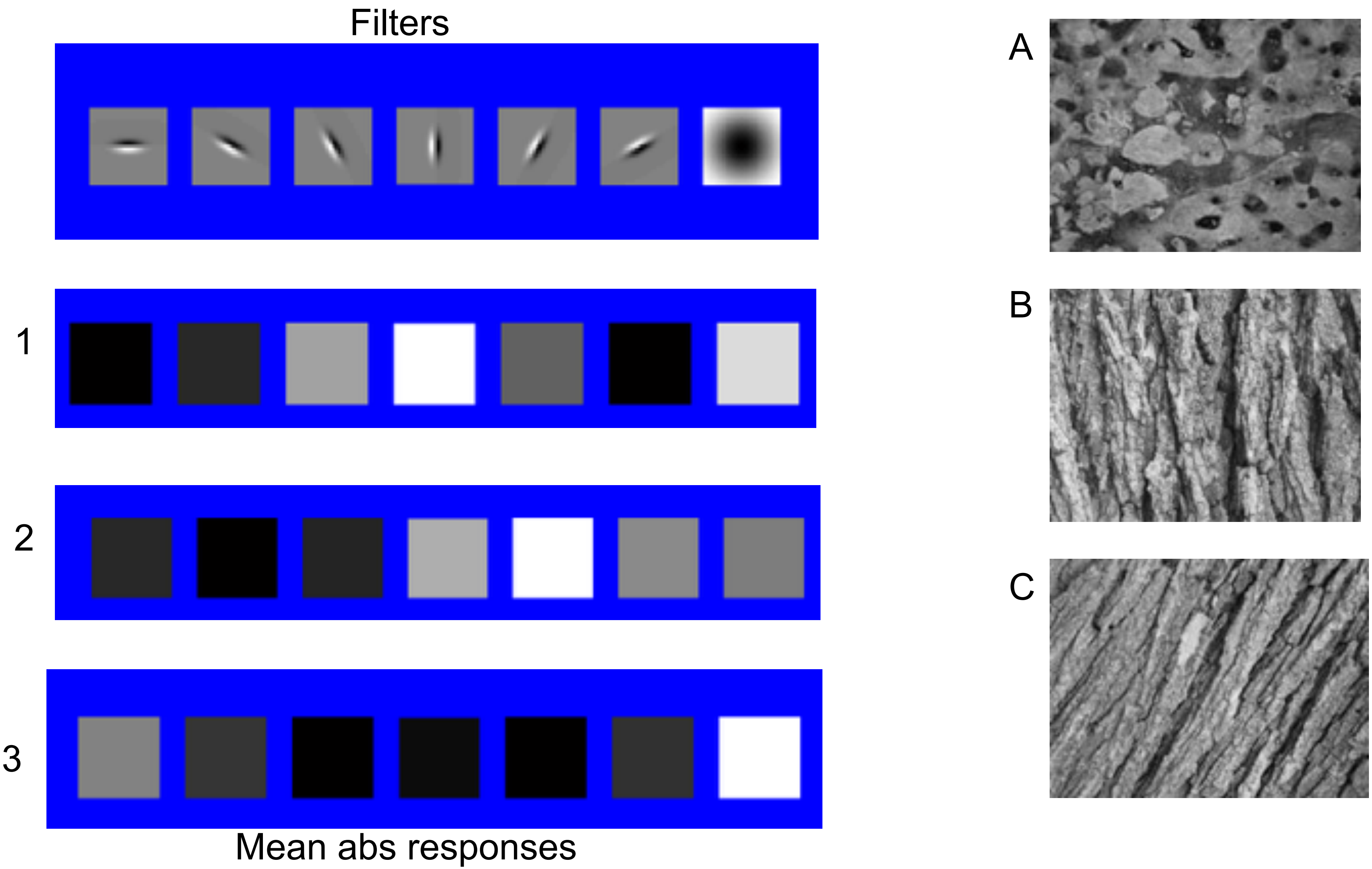


Kristen Grauman

How can we represent texture?

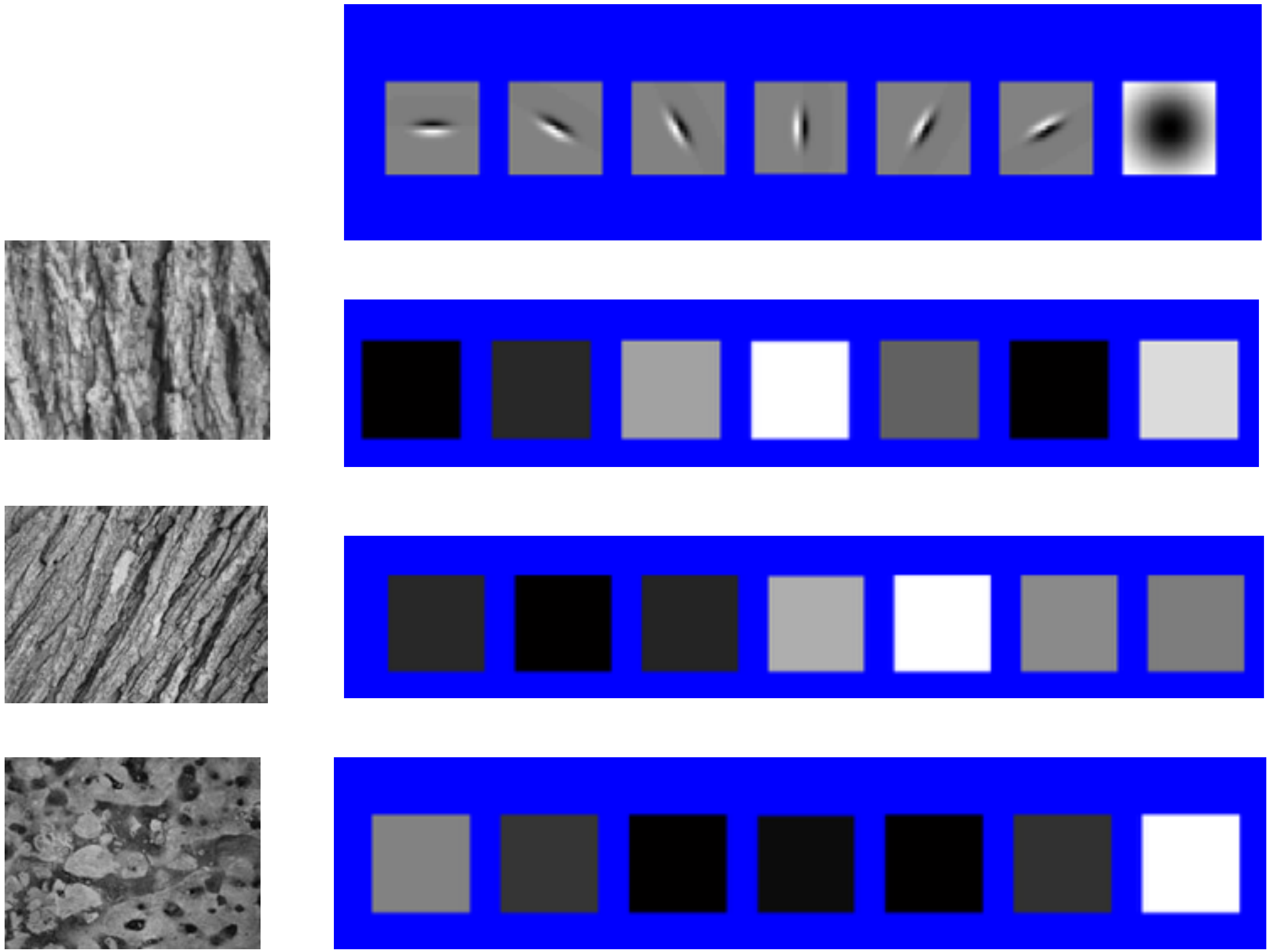
- Measure responses of various filters at different orientations and scales
- Idea 1: Record simple statistics (e.g., mean, std.) of absolute filter responses

Can you match the texture to the response?



Representing texture by mean abs response

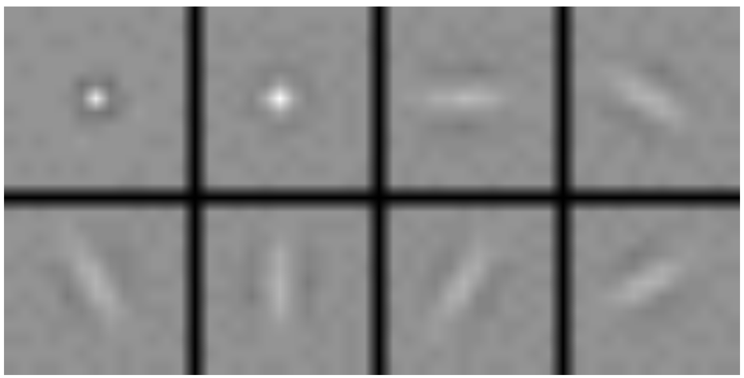
Filters

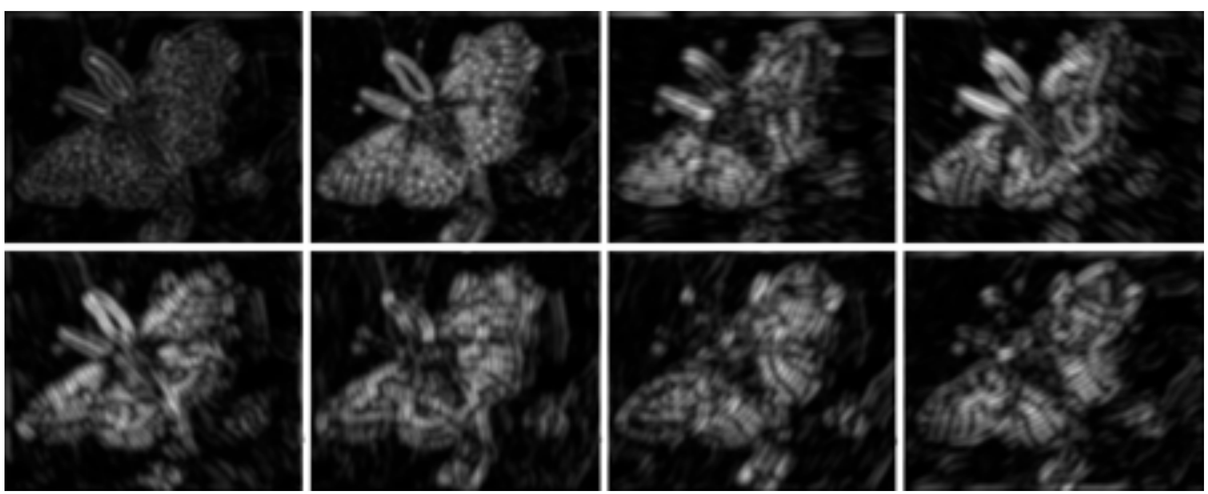


Mean abs responses

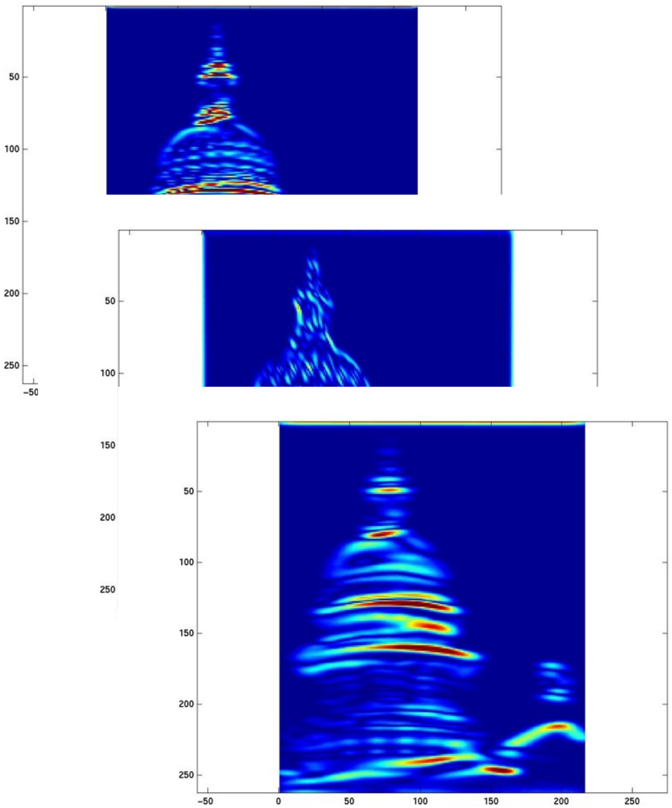
Representing texture

- Idea 2: take vectors of filter responses at each pixel and cluster them, then take histograms

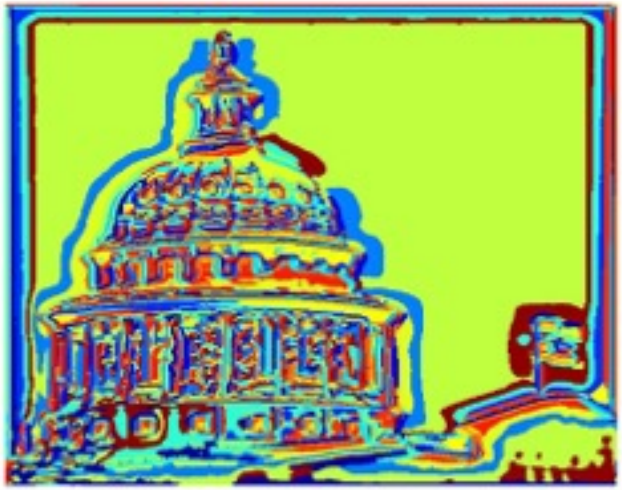




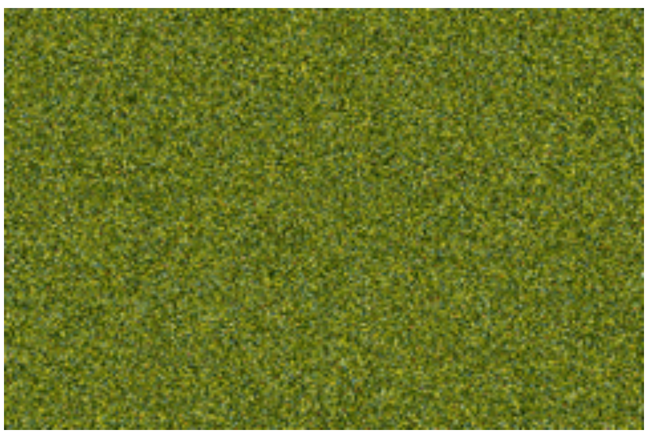
Representing texture

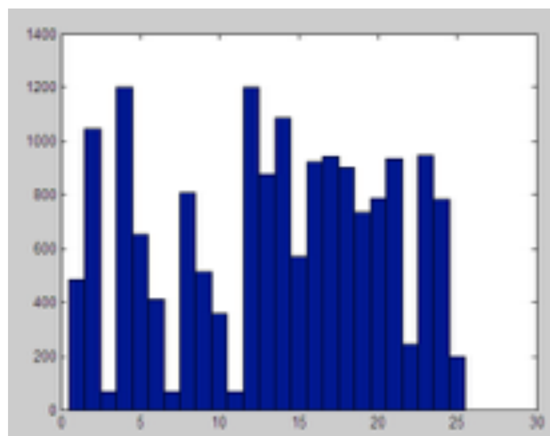


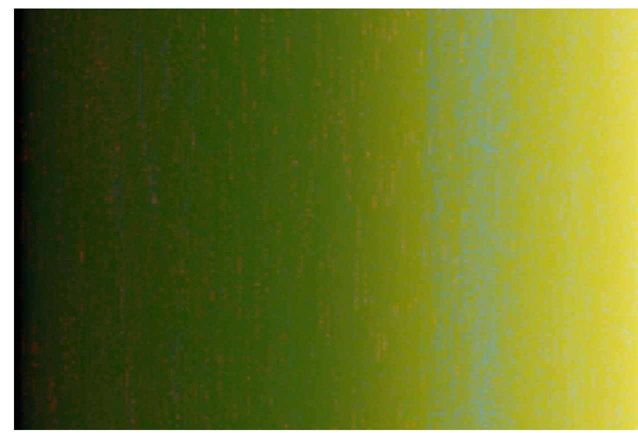
clustering



But what about layout?



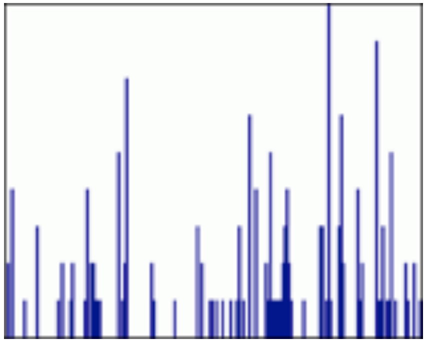




All of these images have the same color histogram

Spatial pyramid representation

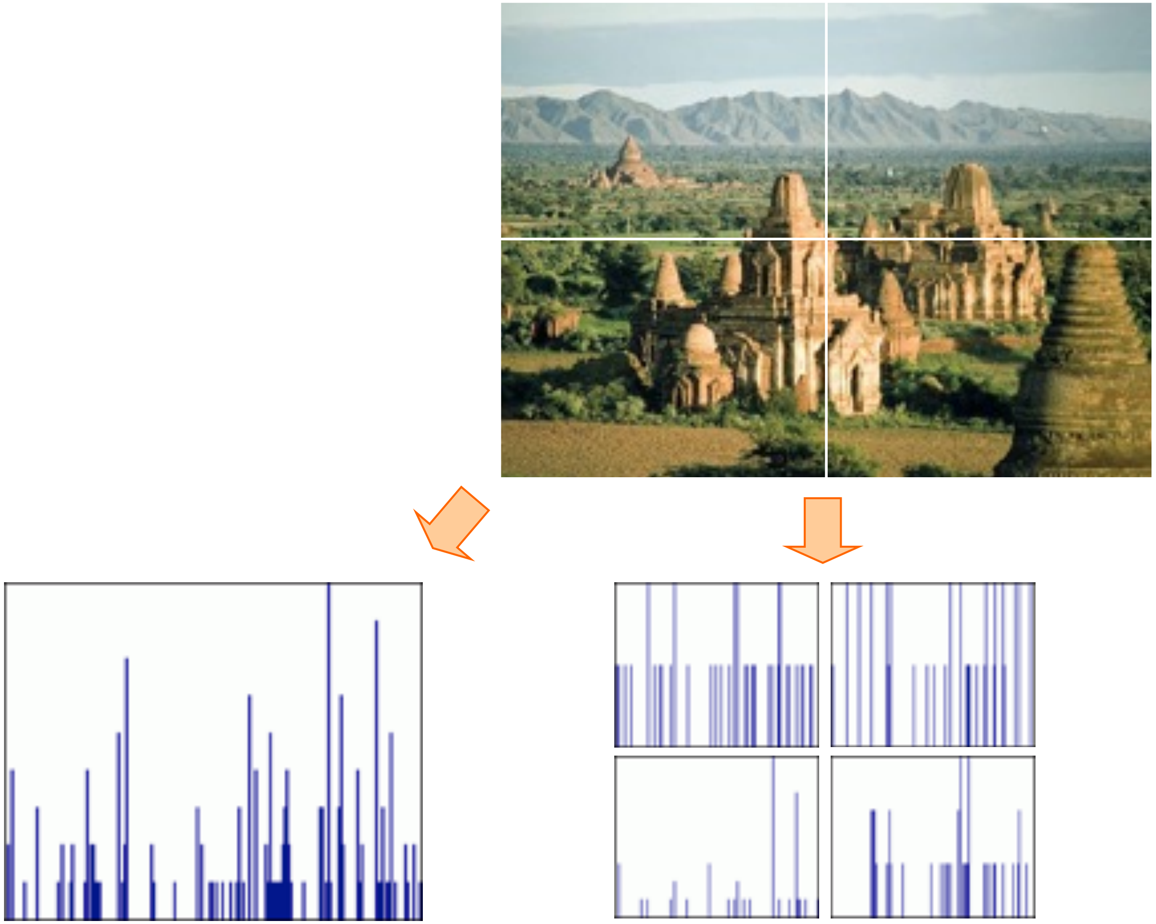
- Extension of a bag of features
- Locally orderless representation at several levels of resolution



Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution



Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution

Lazebnik, Schmid & Ponce (CVPR 2006)

What about Scenes?