Image representations

Subhransu Maji

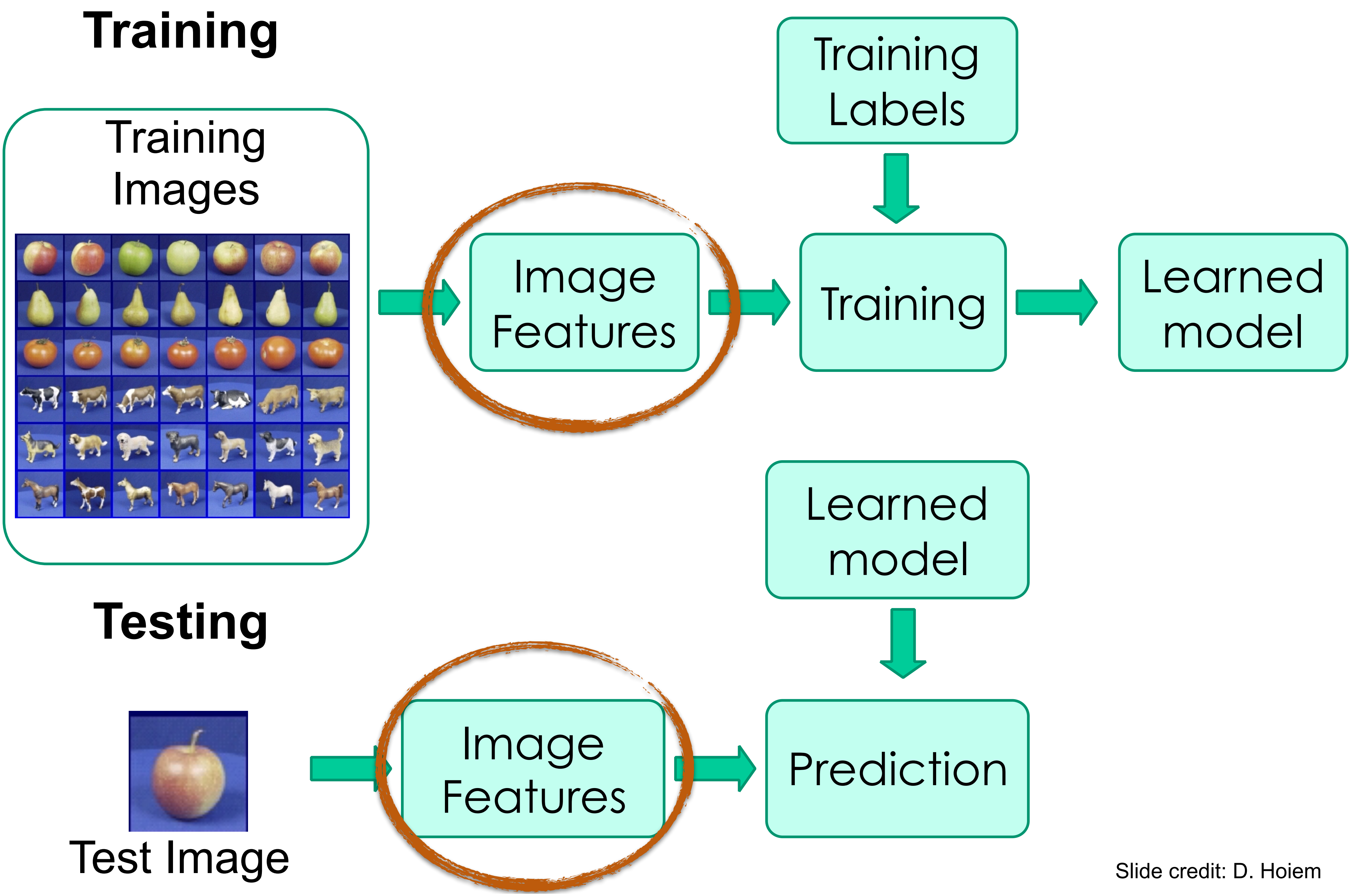
CMPSCI 670: Computer Vision

October 25/27, 2016

Administrativia

- Has everyone submitted a project abstract?
  - I'll take a look at these over the weekend
  - Expect some comments if you have not talked to me already

Recall: Steps



Slide credit: D. Hoiem

What is an image feature?

- Any transformation of an image into a new representation
- <u>Example:</u> transform an image into a binary edge map

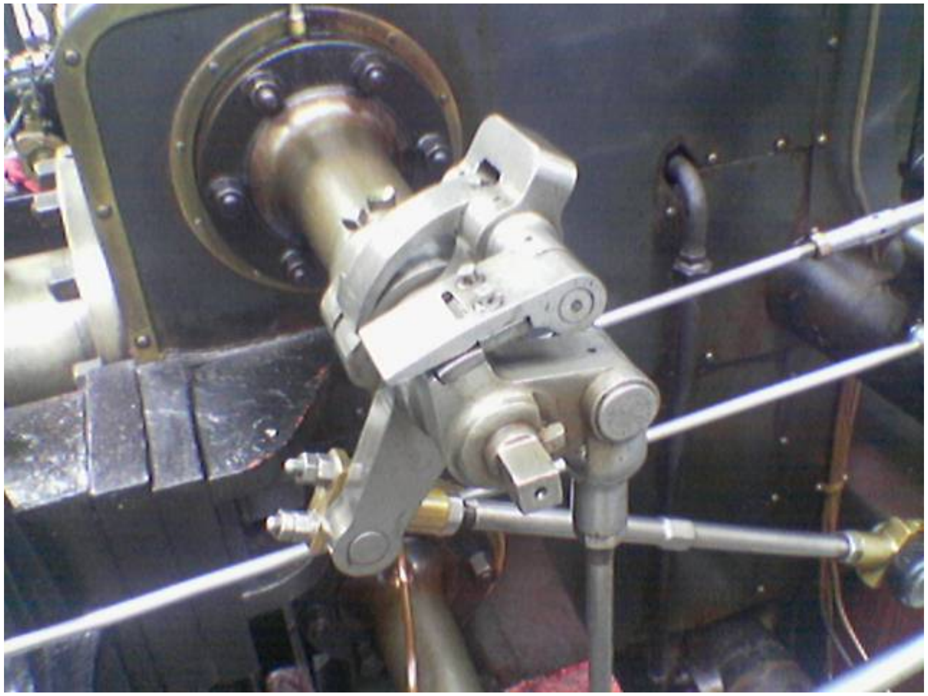
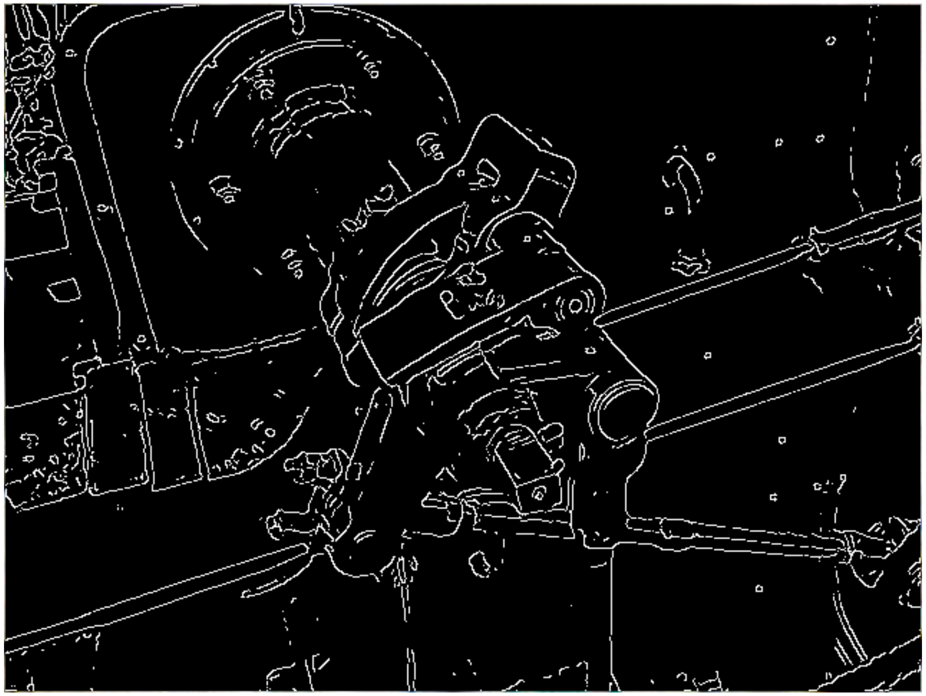
 

Image source: wikipedia

Goals of a feature map

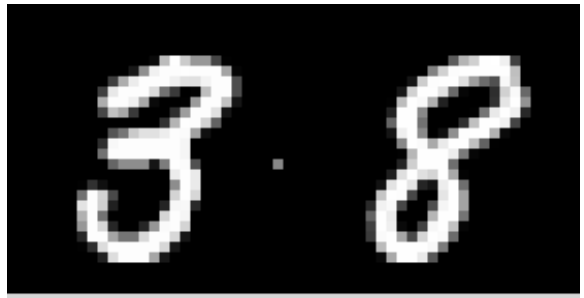
- Introduce invariance: illumination, deformations, position
- Preserve useful properties: shape, texture, color
- Make the subsequent learning easier
  - Ability to learn from a few examples
  - Can use simpler classifiers (prevent overfitting)

Figure 1.3: Variation in appearance due to a change in illumination

Image: [Fergus05]

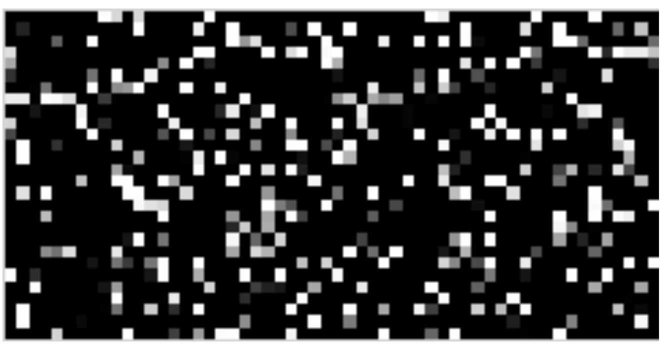
The importance of good features

- Most learning methods are invariant to feature permutation
  - E.g., patch vs. pixel representation of images



permute pixels

permute patches



bag of pixels

bag of patches

can you recognize the digits?

Hand-crafting features

- In general the optimal feature depends on
  - the nature of the recognition task
  - the choice of subsequent classifier
    - "Shallow" learning — hand-crafted features + simple classifiers
    - "Deep" learning — end-to-end mapping of pixels to labels
- Two families of features that work well with simple classifiers
  - Histogram of oriented gradients — captures overall shape
  - Bag of visual words — captures local shape and texture

shape

texture

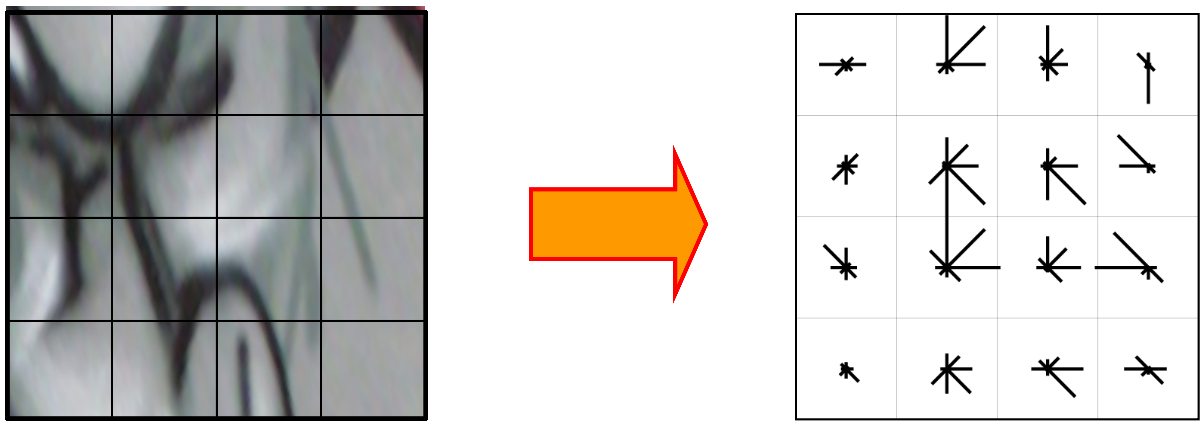
Motivation

- Recall the feature matching step in image alignment
- Problem with pixel values as a feature representation
  - illumination changes, small deformations

- How can we design a feature that is robust to these changes?

SIFT features

- Descriptor computation:
  - Divide patch into 4x4 sub-patches
  - Compute histogram of gradient orientations (8 reference angles) inside each sub-patch
  - Resulting descriptor: 4x4x8 = 128 dimensions
  - Additional step: normalize the descriptor to unit length



David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), pp. 91-110, 2004.

SIFT features

- Descriptor computation:
  - Divide patch into 4x4 sub-patches
  - Compute histogram of gradient orientations (8 reference angles) inside each sub-patch
  - Resulting descriptor: 4x4x8 = 128 dimensions
  - Additional step: normalize the descriptor to unit length

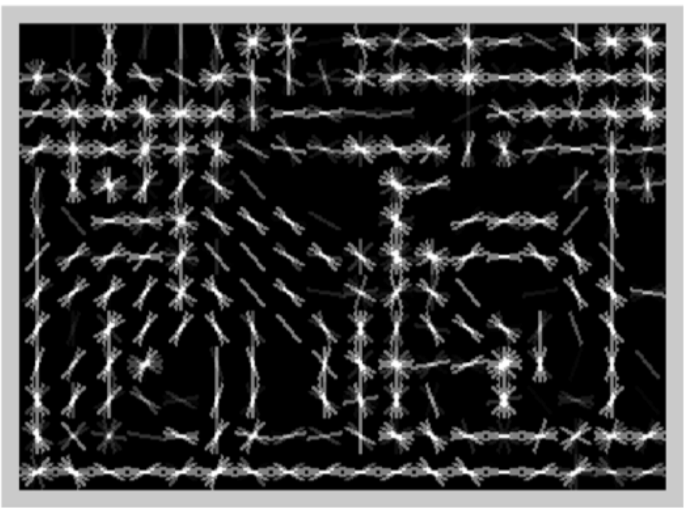
- Advantage over raw vectors of pixel values
  - Gradients less sensitive to illumination change
  - Pooling of gradients over the sub-patches achieves robustness to small shifts, but still preserves some spatial information

David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), pp. 91-110, 2004.

Histogram of Oriented Gradients

- Can apply the same idea to the whole image
  - Preserves the overall structure of the image
  - Provides robustness to illumination and small deformations

- Introduced by Dalal and Triggs (CVPR 2005) for pedestrian detection



HOG feature

HOG feature: basic idea

- Divide the image into blocks
- Compute histograms of gradients for each regions

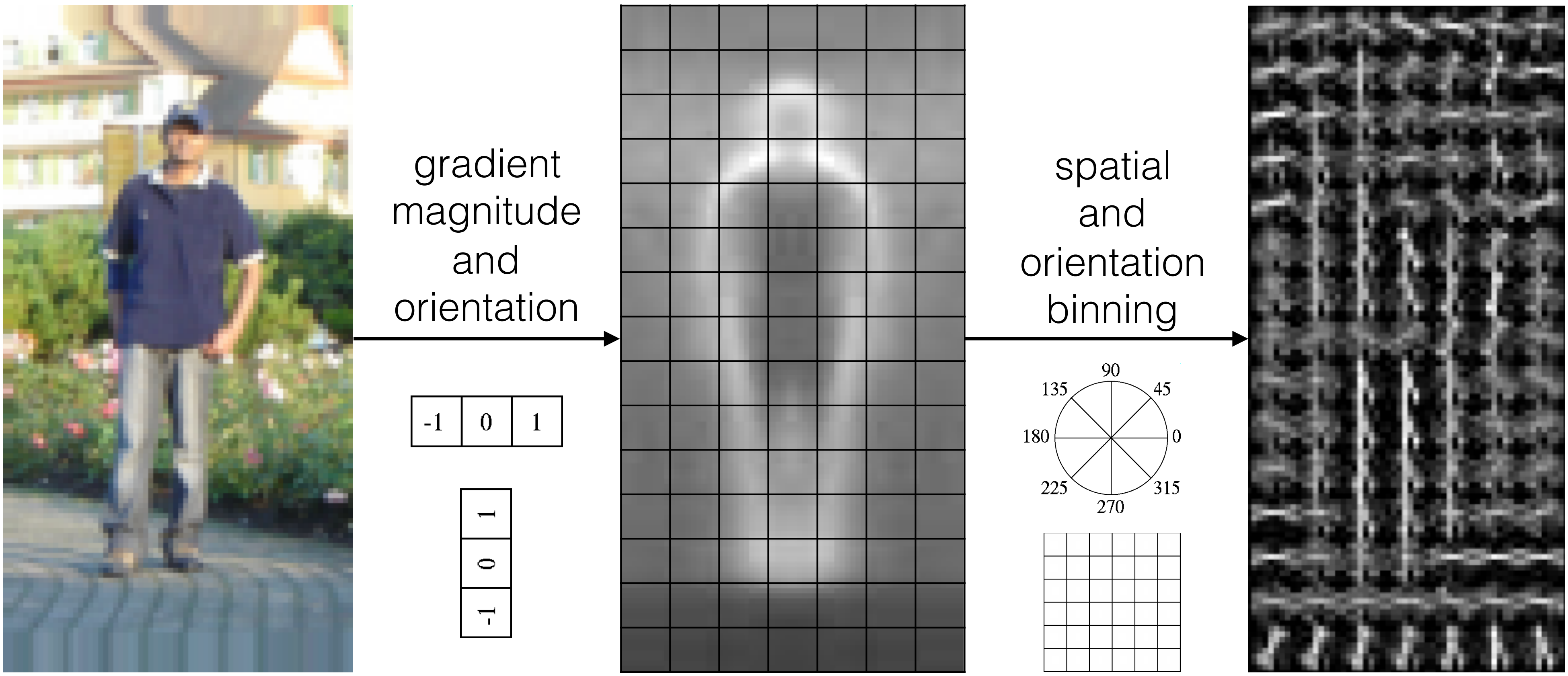
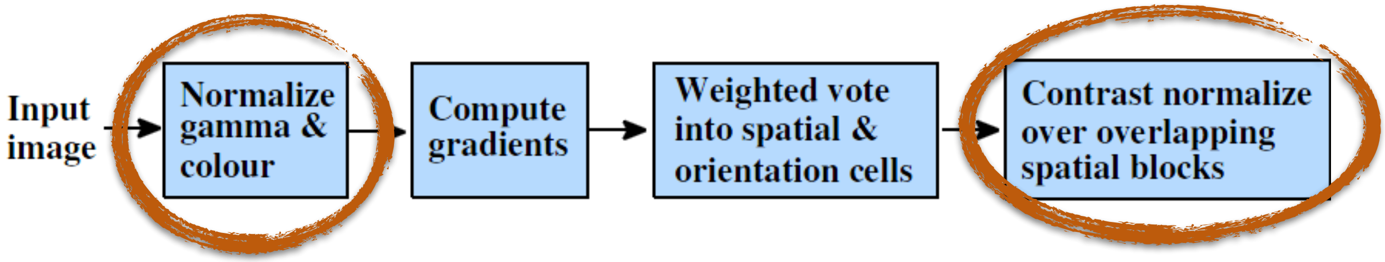
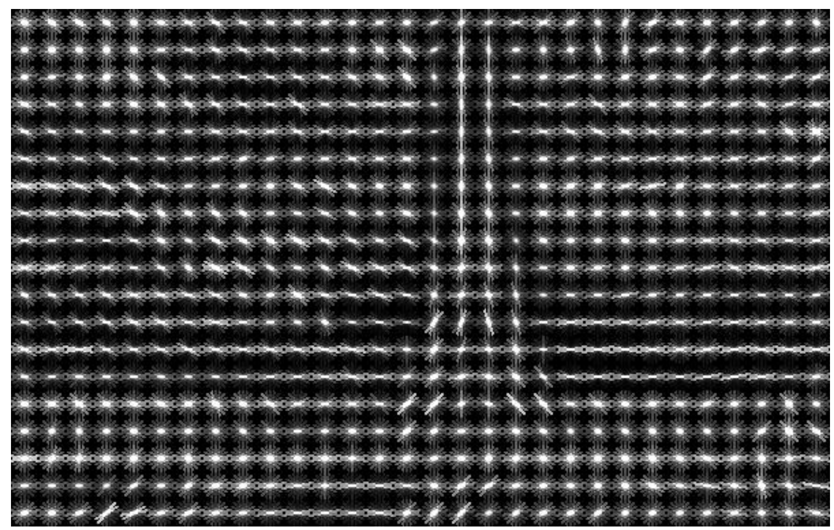


image Gradient norm HOG feature

HOG feature: additional steps

- Additional steps for more invariance
  - Logarithm of the intensity values
  - Local contrast normalization

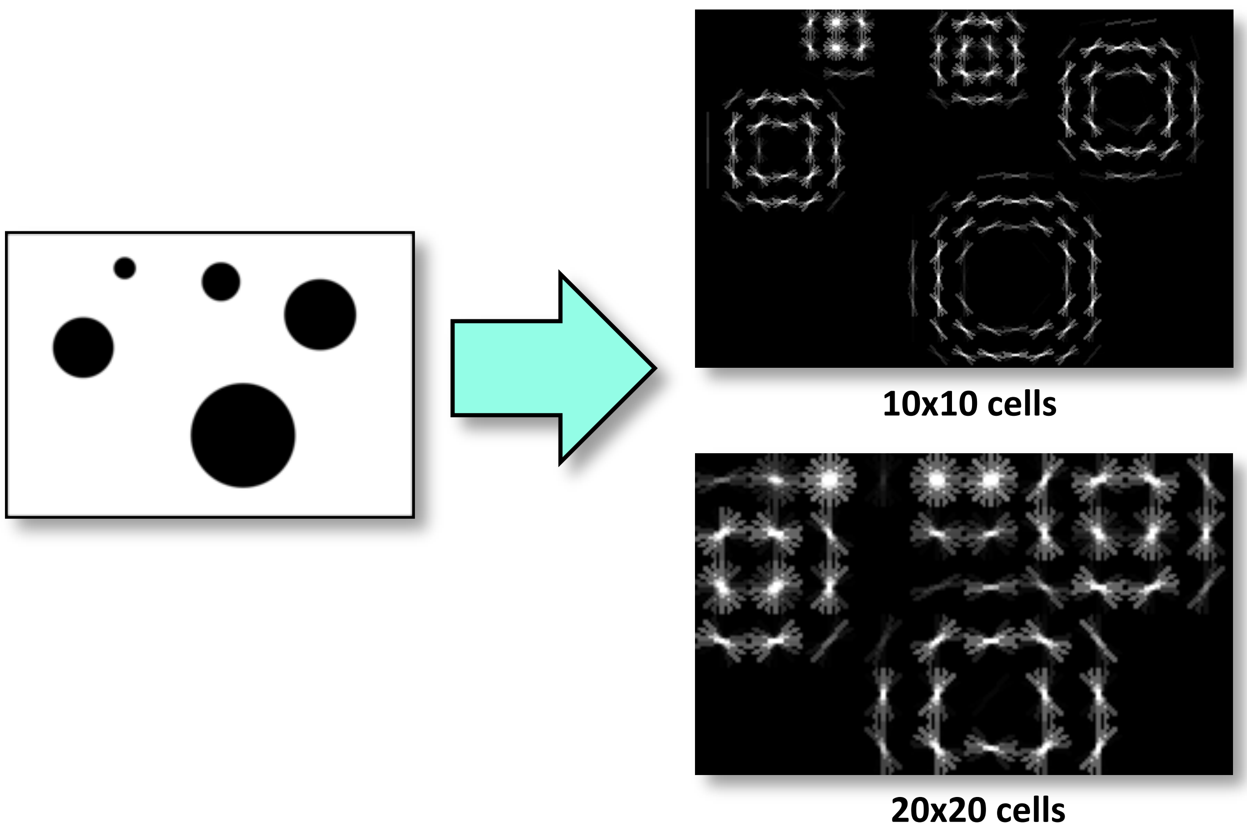




N. Dalal and B. Triggs, [Histograms of Oriented Gradients for Human Detection](), CVPR 2005

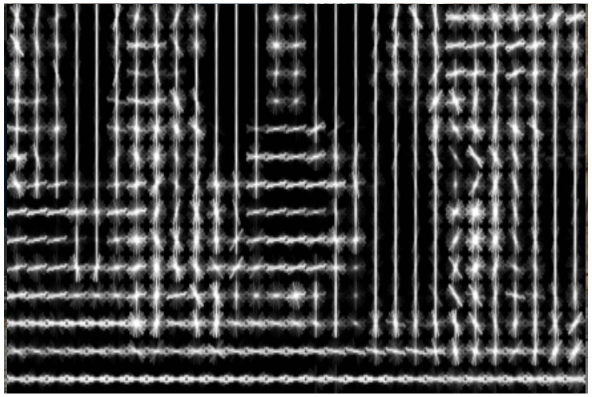
Effect of bin-size

- **Smaller bin-size:** better spatial resolution
- **Larger bin-size:** better invariance to deformations
- Optimal value depends on the object category being modeled
  - e.g. rigid vs. deformable objects

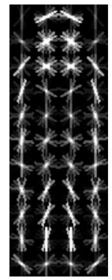


Works well for template matching

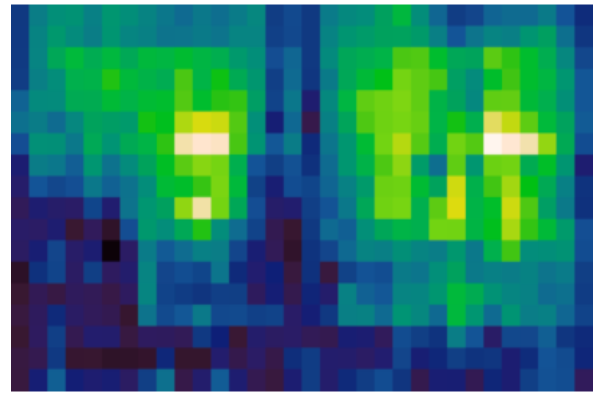
HOG feature map



Template

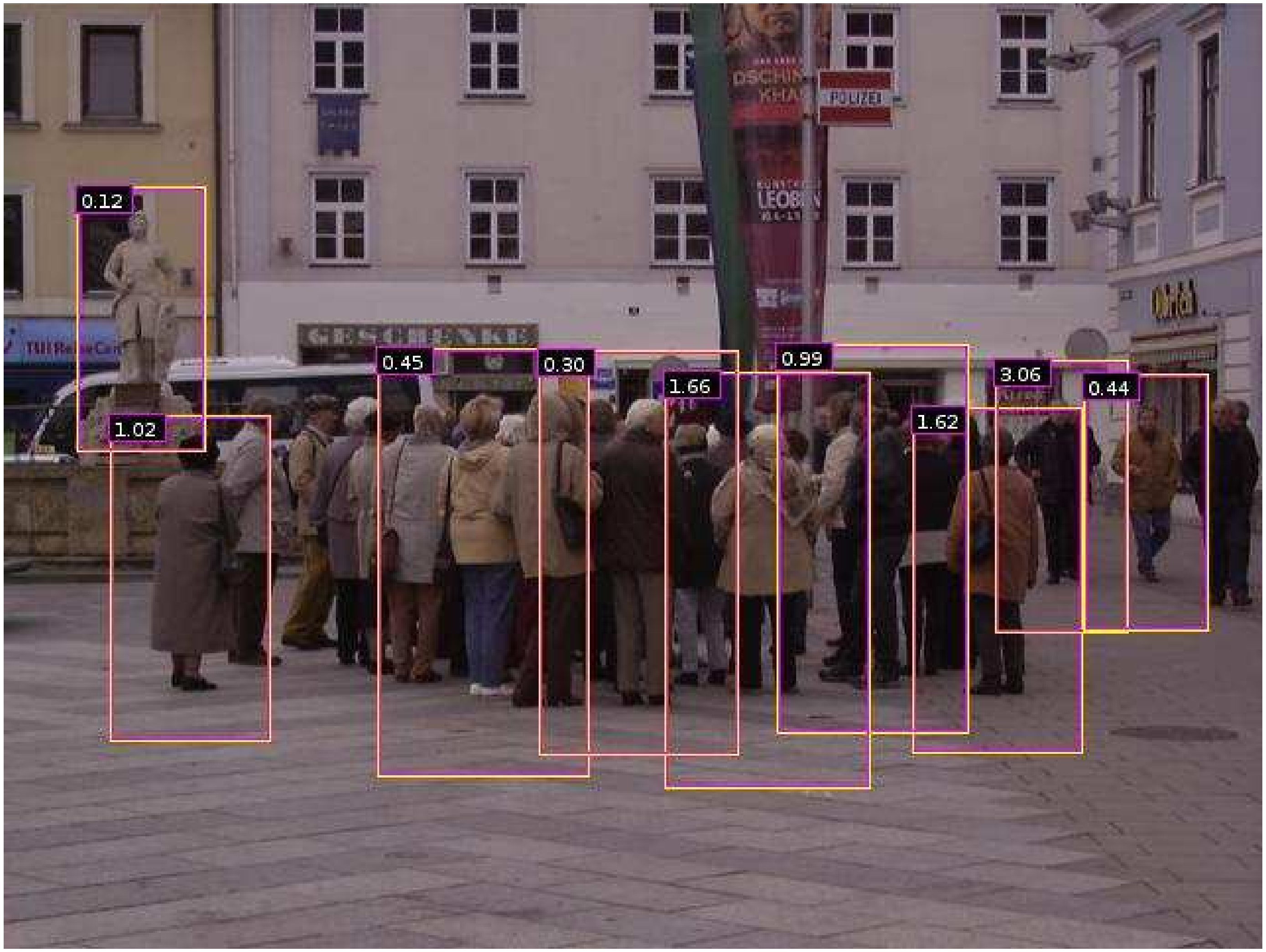


Detector response map



- Compute the HOG feature map for the image
- Convolve the template with the feature map to get score
  - Do this across scales (since we don't know the size of the person)
- Find peaks of the response map (non-max suppression)

Example pedestrian detections



We will discuss object detection in detail later

[Dalal06]

Hand-crafting features

- Two families of features that work well with simple classifiers
  - Histogram of oriented gradients — captures overall shape
  - Bag of visual words — captures local shape and texture

shape

texture

Bag of visual words

- Origin and motivation of the "bag of words" model
- Algorithm pipeline
  - Extracting local features
  - Learning a dictionary — clustering using k-means
  - Encoding methods — hard vs. soft assignment
  - Spatial pooling — pyramid representations

Figure from *Chatfield et al.,2011*

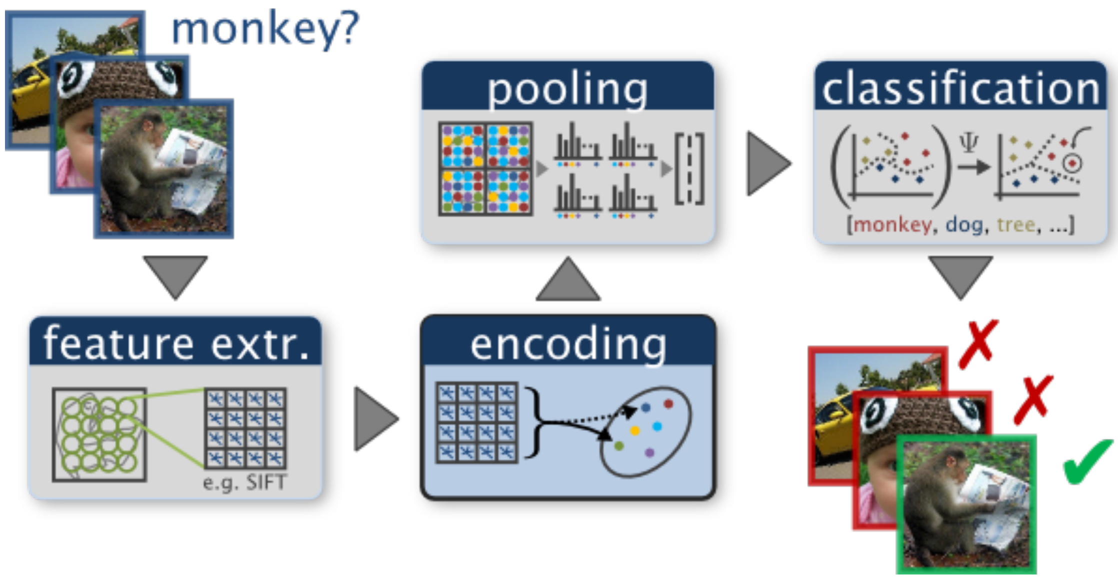
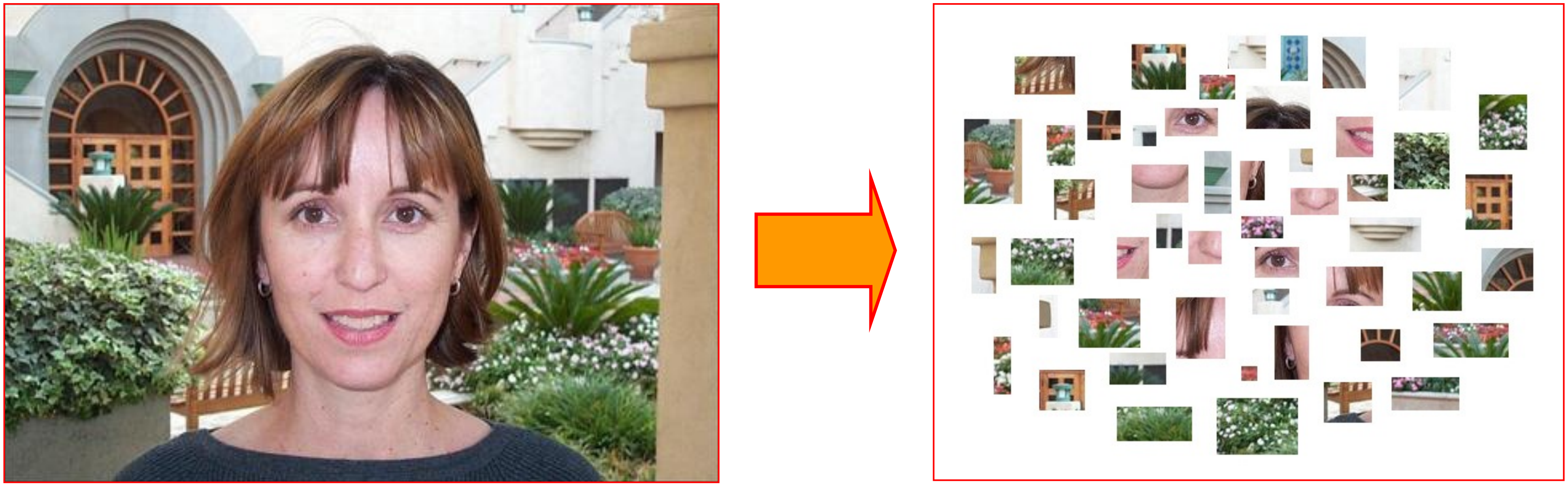


Image as a “bag of patches”



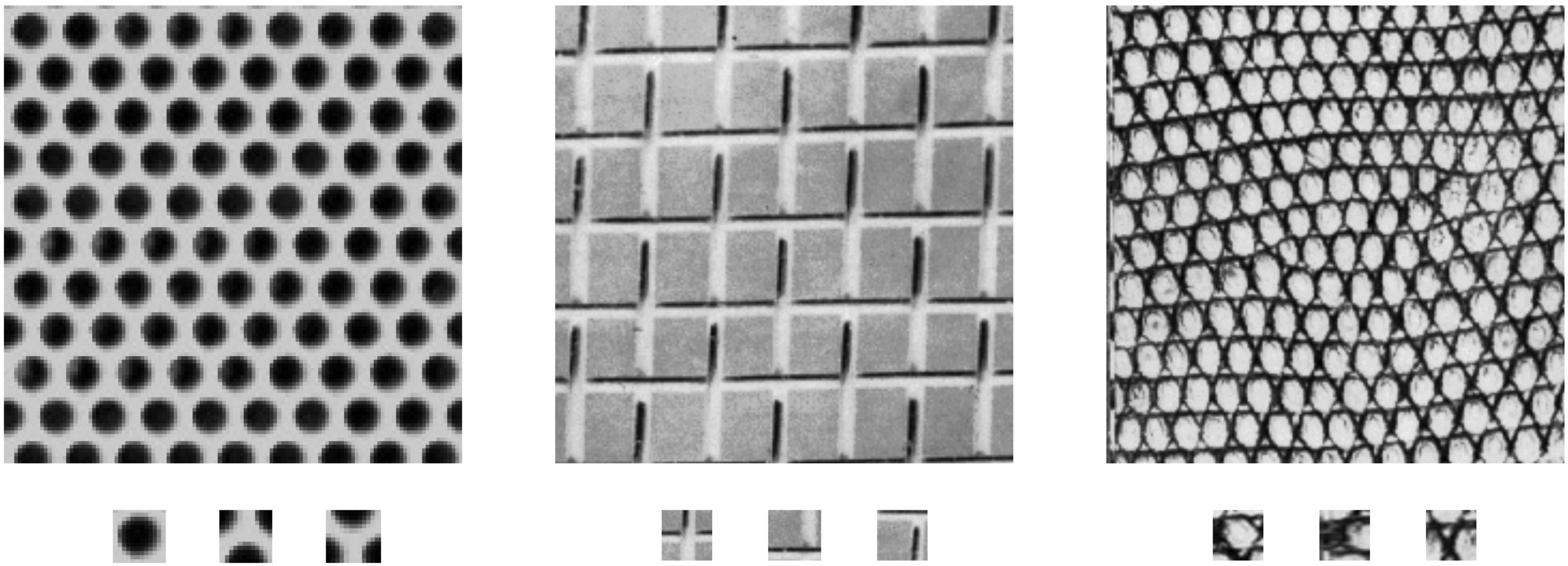
Properties:

- Spatial structure is not preserved
- Invariance to large translations

Compare this to the HOG feature

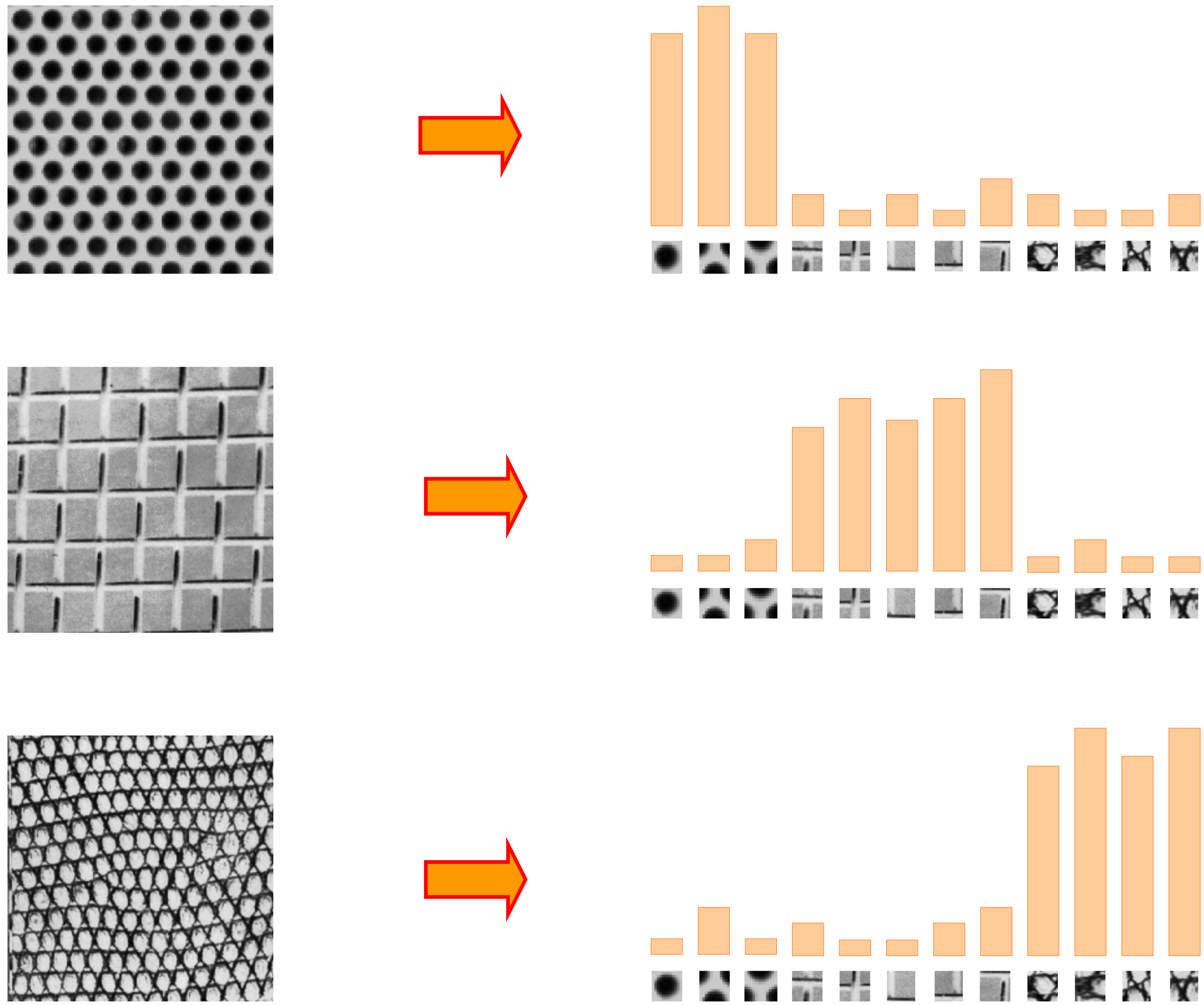
Origin 1: Texture recognition

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



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

Origin 1: Texture recognition



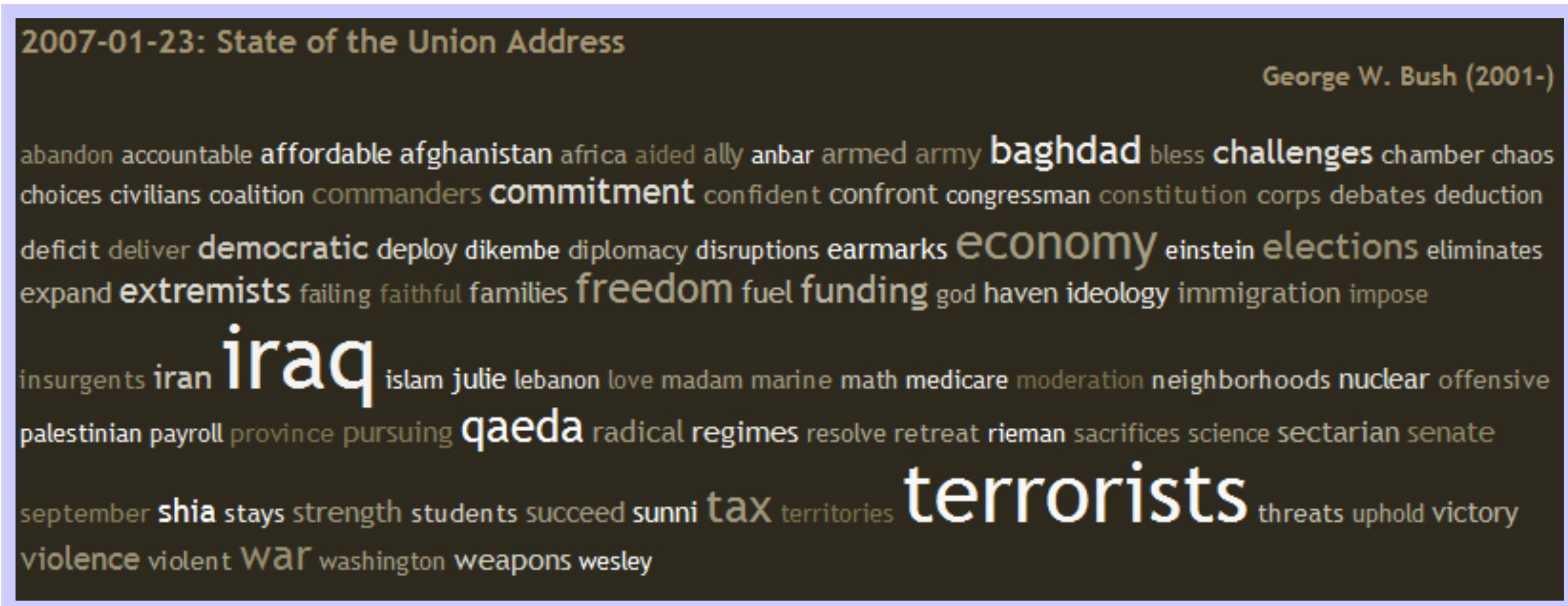
Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

Origin 2: Bag-of-words models

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

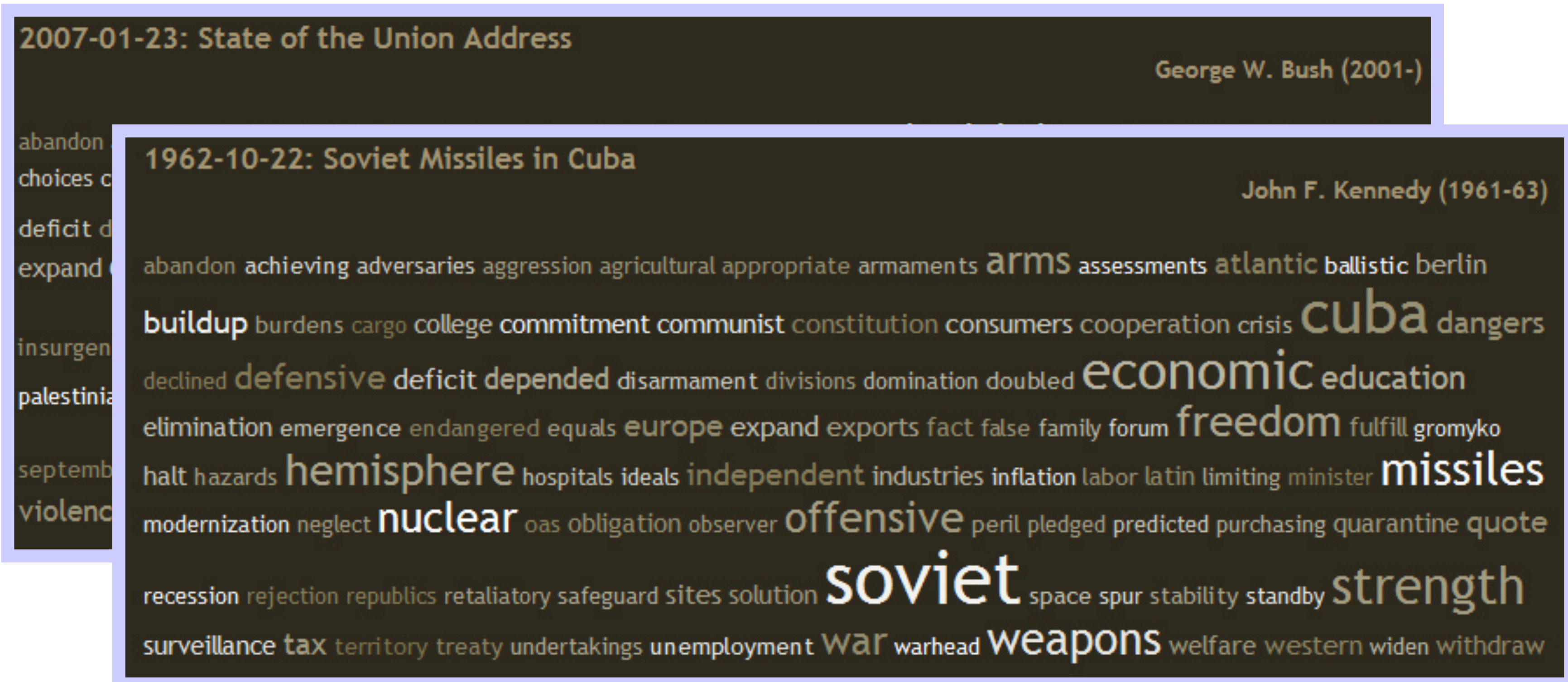
Origin 2: Bag-of-words models

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



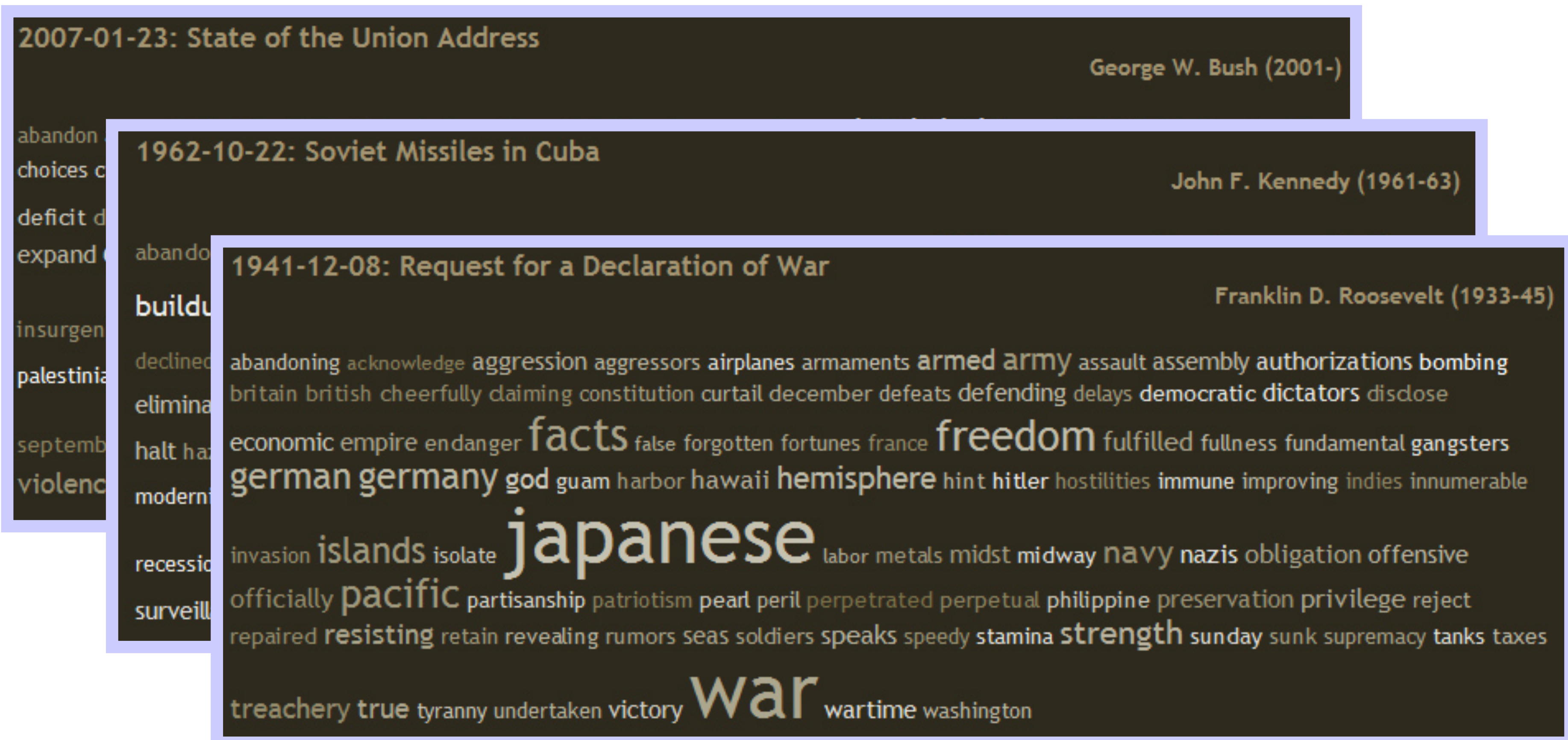
Origin 2: Bag-of-words models

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



Origin 2: Bag-of-words models

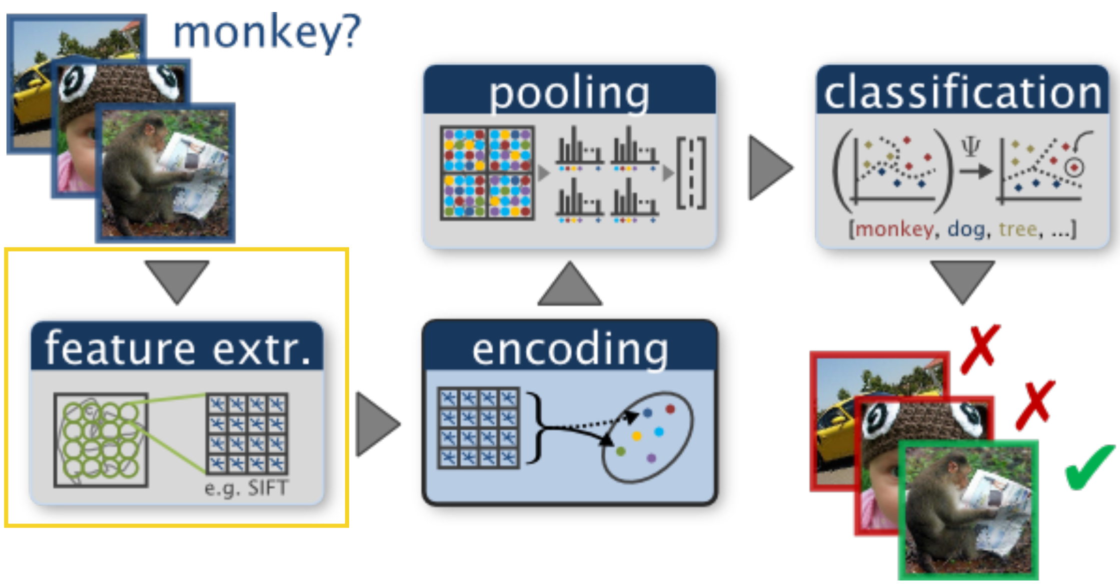
- Orderless document representation: frequencies of words from a dictionary   Salton & McGill (1983)



Lecture outline

- Origin and motivation of the "bag of words" model
- Algorithm pipeline
  - Extracting local features
  - Learning a dictionary — clustering using k-means
  - Encoding methods — hard vs. soft assignment
  - Spatial pooling — pyramid representations

Figure from *Chatfield et al.,2011*



Local feature extraction

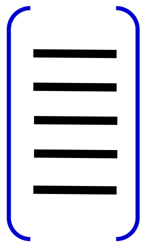
- Regular grid or interest regions

grid

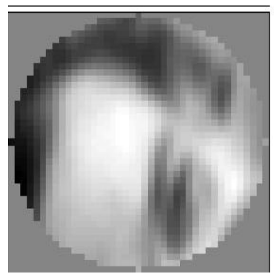
corner or blobs

Slide credit: Josef Sivic

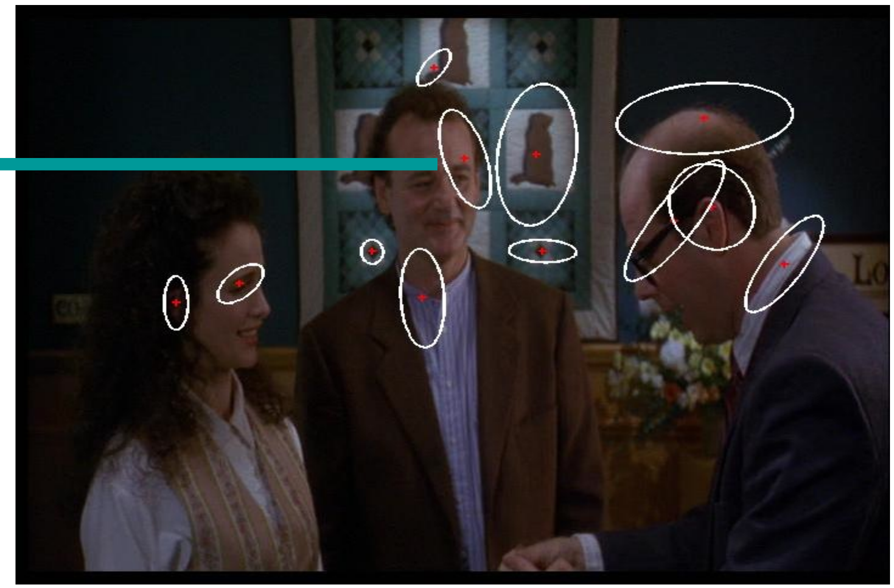
Local feature extraction



Compute descriptor



Normalize patch

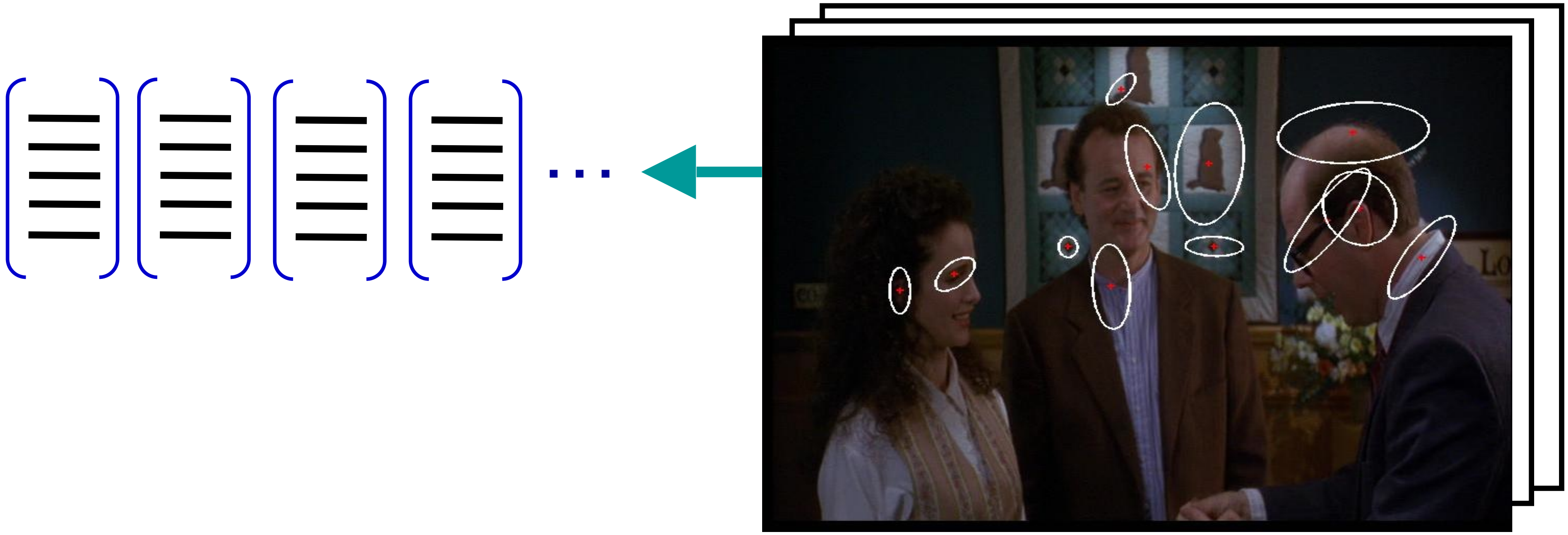


Detect patches

Choices of descriptor:
- SIFT
- The patch itself
- …

Slide credit: Josef Sivic

Local feature extraction



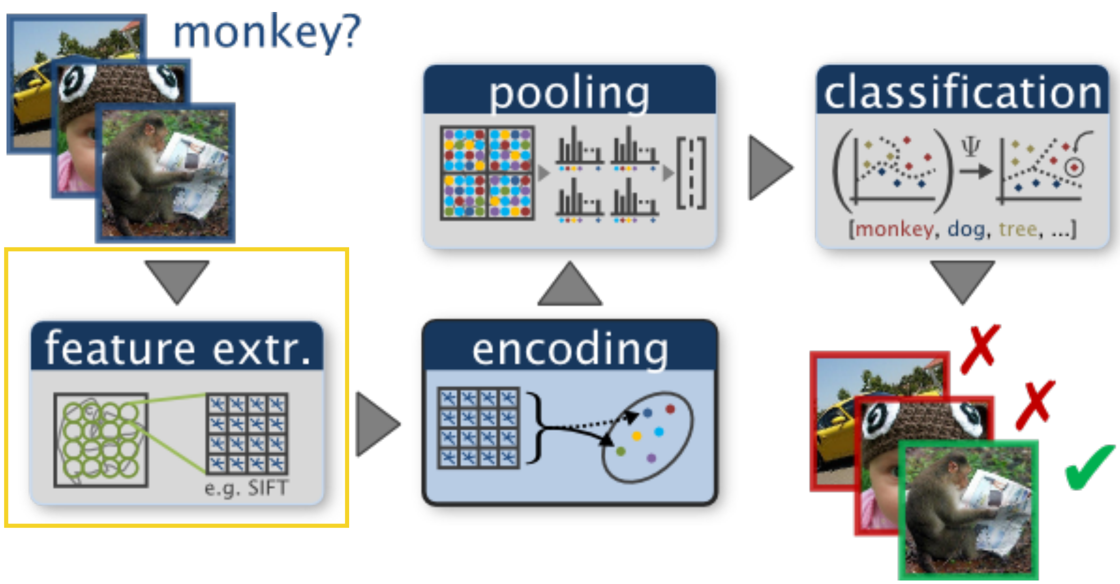
Extract features from many images

Slide credit: Josef Sivic

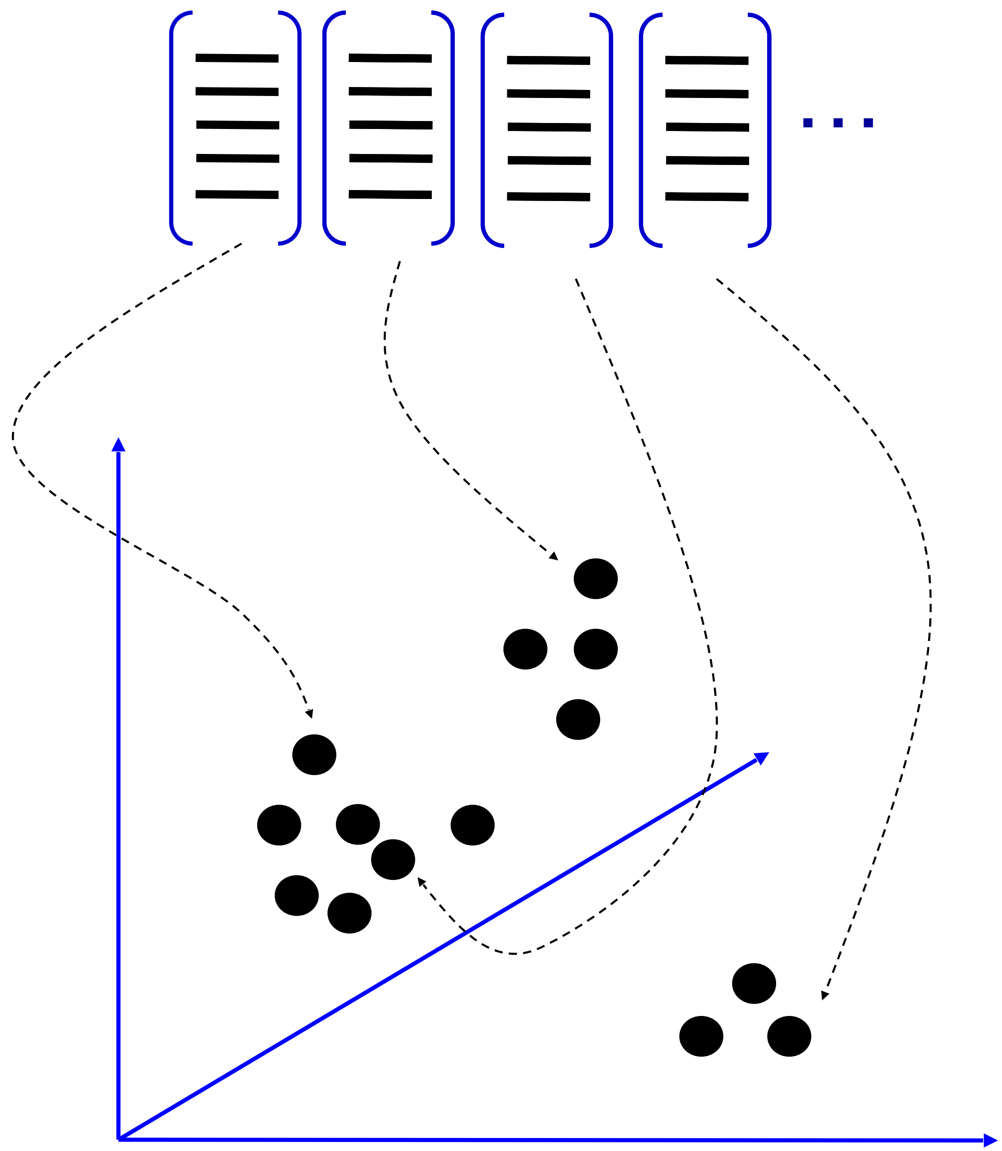
Lecture outline

- Origin and motivation of the "bag of words" model
- Algorithm pipeline
  - Extracting local features
  - Learning a dictionary — clustering using k-means
  - Encoding methods — hard vs. soft assignment
  - Spatial pooling — pyramid representations

Figure from *Chatfield et al.,2011*

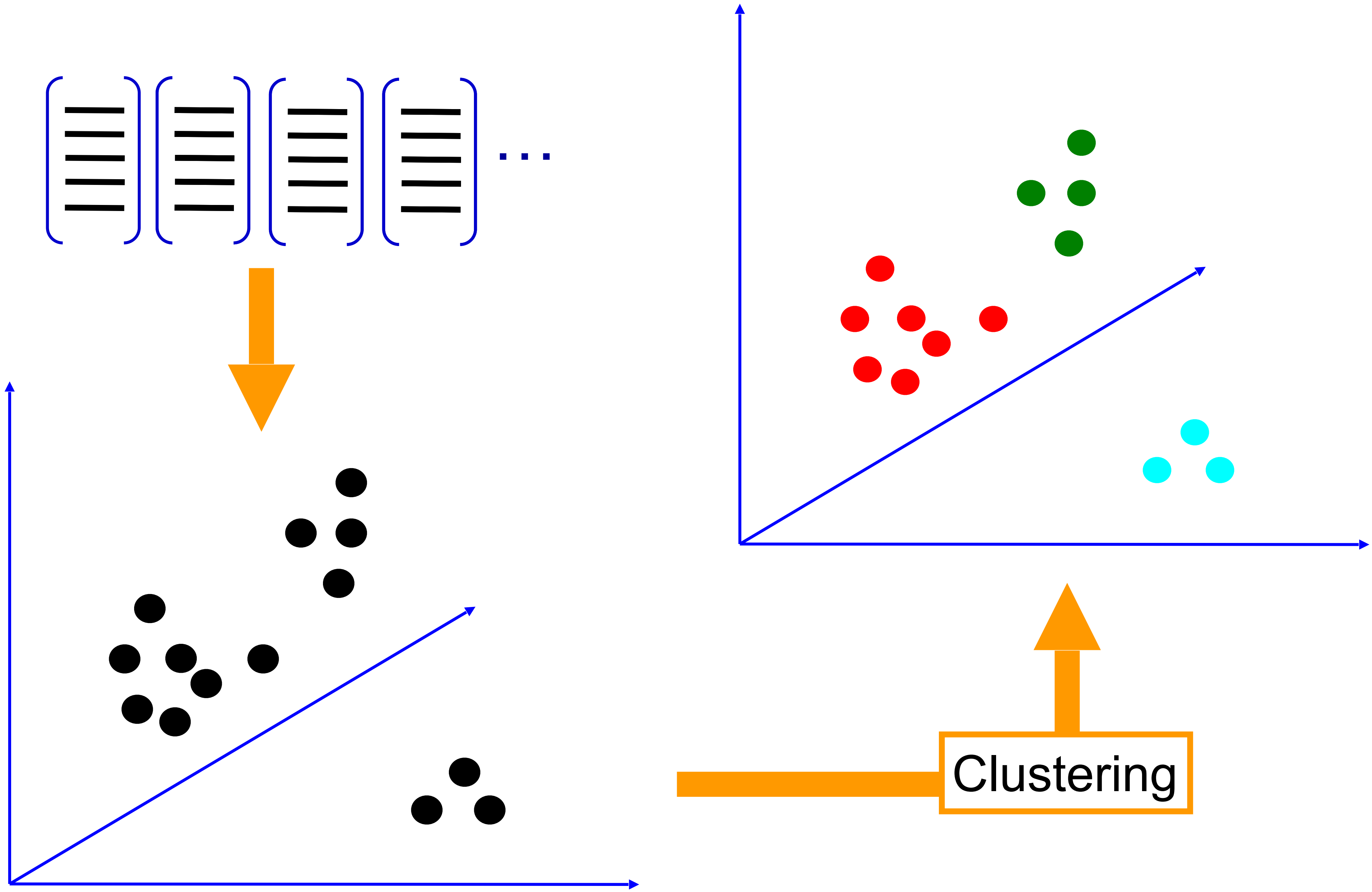


Learning a dictionary



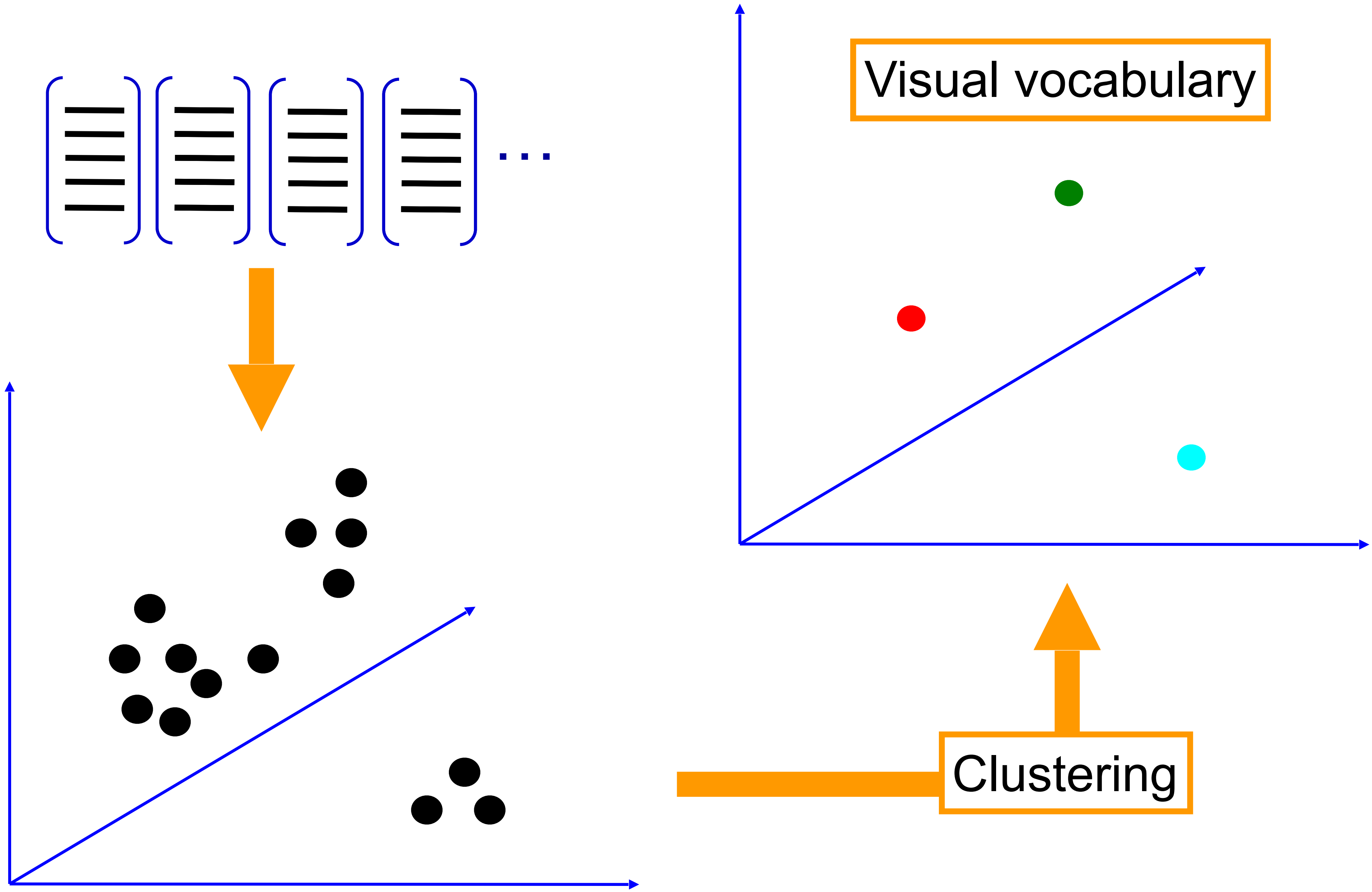
Slide credit: Josef Sivic

Learning a dictionary



Slide credit: Josef Sivic

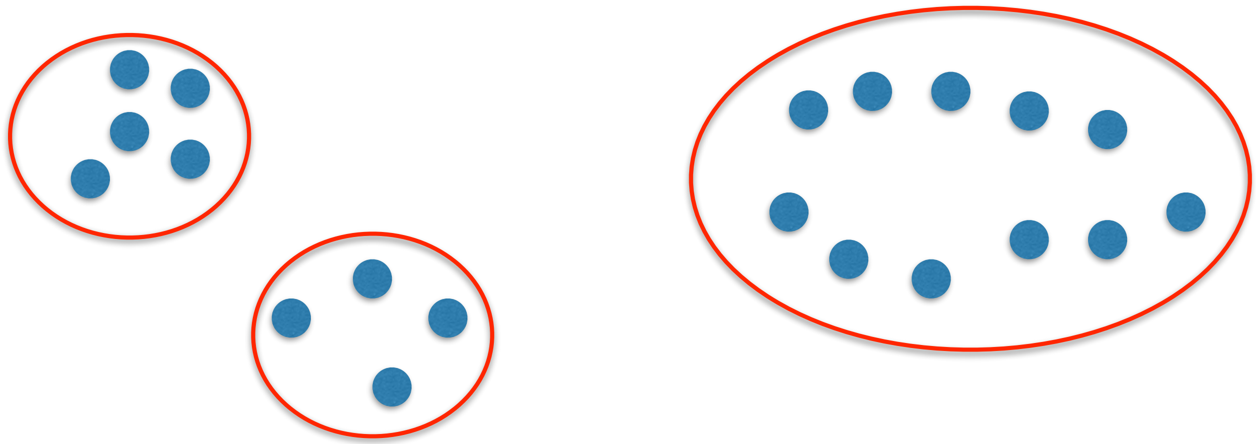
Learning a dictionary



Slide credit: Josef Sivic

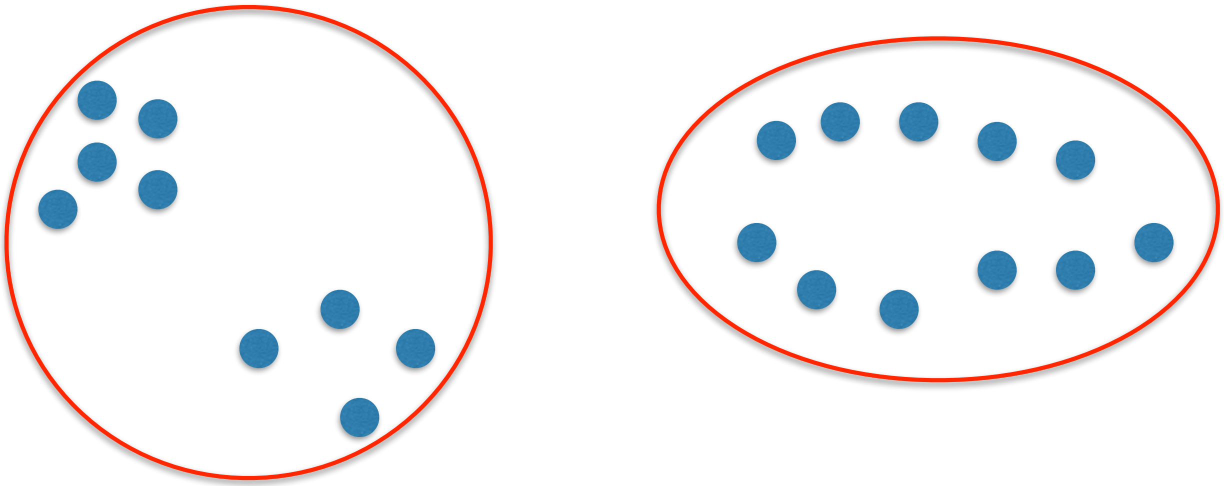
Clustering

- Basic idea: group together similar instances
- Example: 2D points



Clustering

- Basic idea: group together similar instances
- Example: 2D points



- What could similar mean?
  - One option: small Euclidean distance (squared)

$$\text{dist}(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_2^2$$

  - Clustering results are crucially dependent on the measure of similarity (or distance) between points to be clustered

Clustering using k-means

- Given $(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})$ partition the **n** observations into **k (≤ n)** sets **S =** $\{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squared distances

- The objective is to minimize:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} ||\mathbf{x} - \underbrace{\mu_i}_{\text{cluster center}}||^2$$

Lloyd's algorithm for k-means

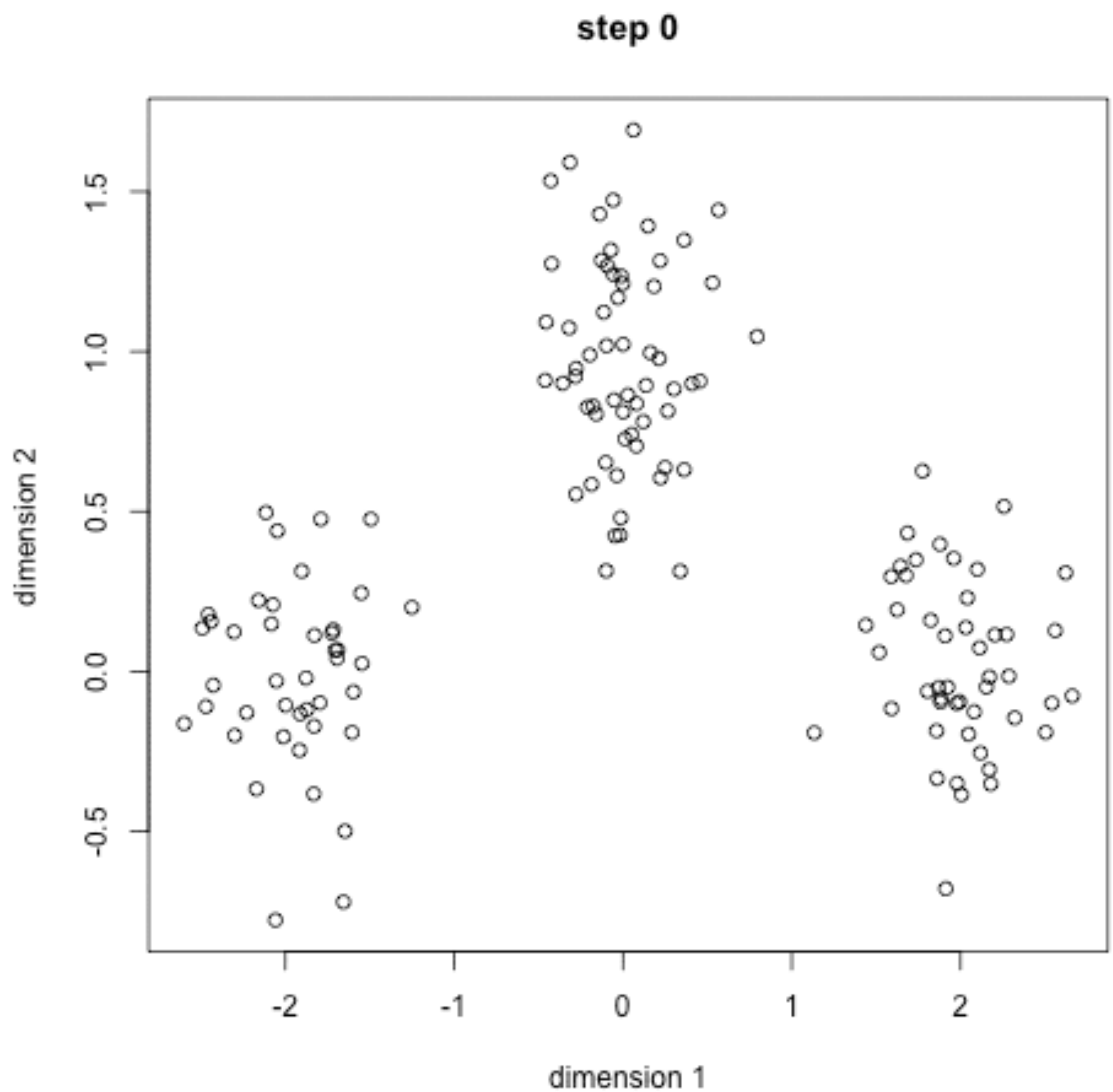
- Initialize k centers by picking k points randomly among all the points
- Repeat till convergence (or max iterations)
  - Assign each point to the nearest center (assignment step)

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} ||\mathbf{x} - \mu_i||^2$$

  - Estimate the mean of each group (update step)

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} ||\mathbf{x} - \mu_i||^2$$

k-means in action

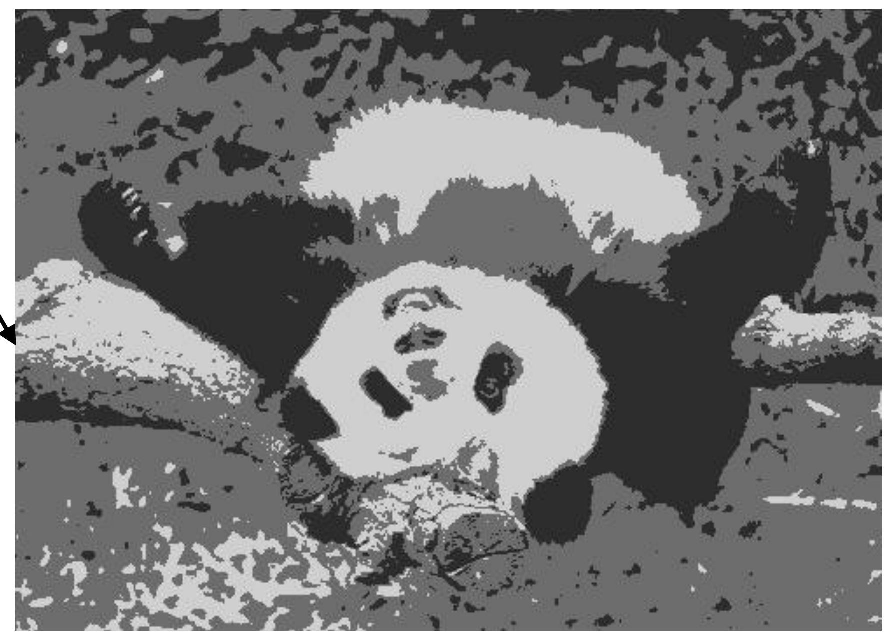


http://simplystatistics.org/2014/02/18/k-means-clustering-in-a-gif/

k-means for image segmentation

K=2

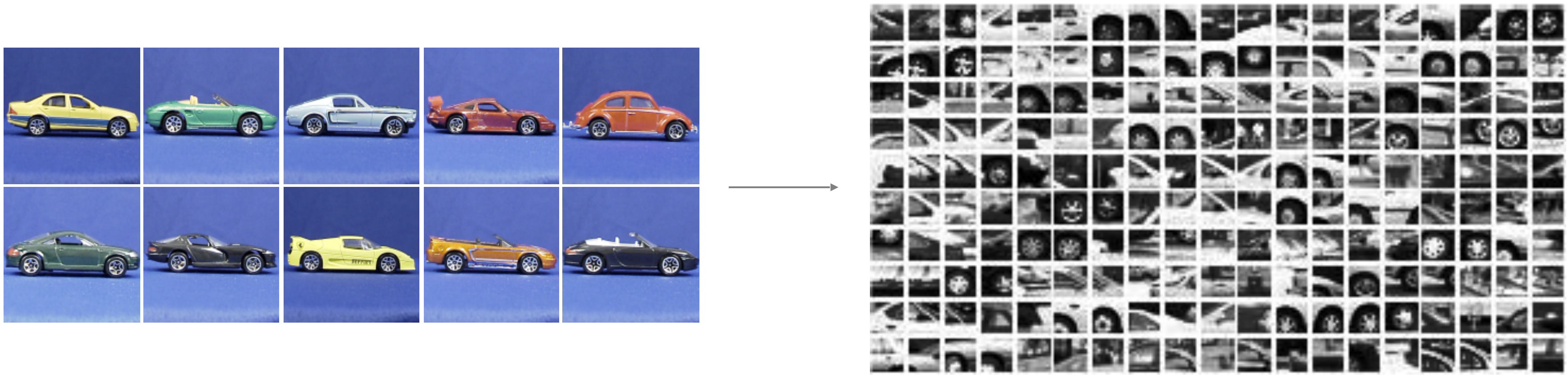
K=3

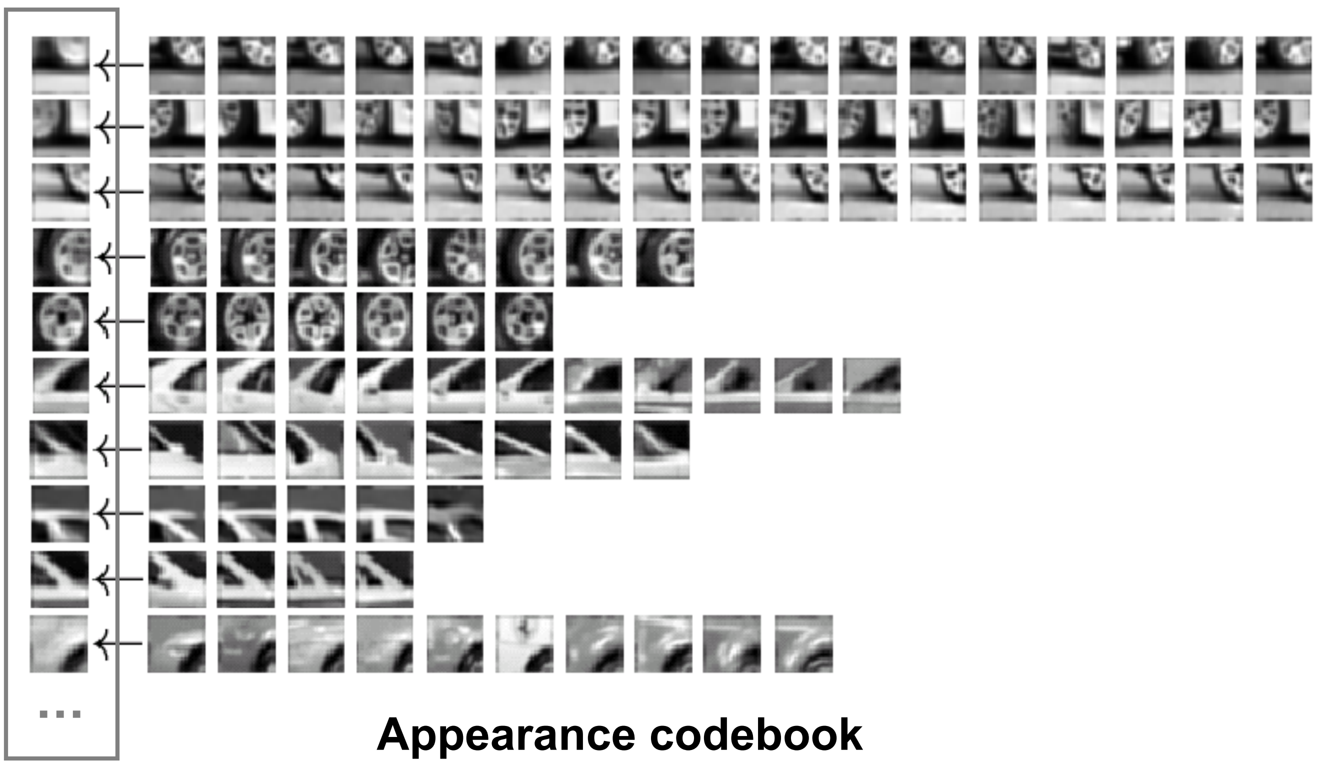


Grouping pixels based on **intensity** similarity

feature space: intensity value (1D)

Example codebook

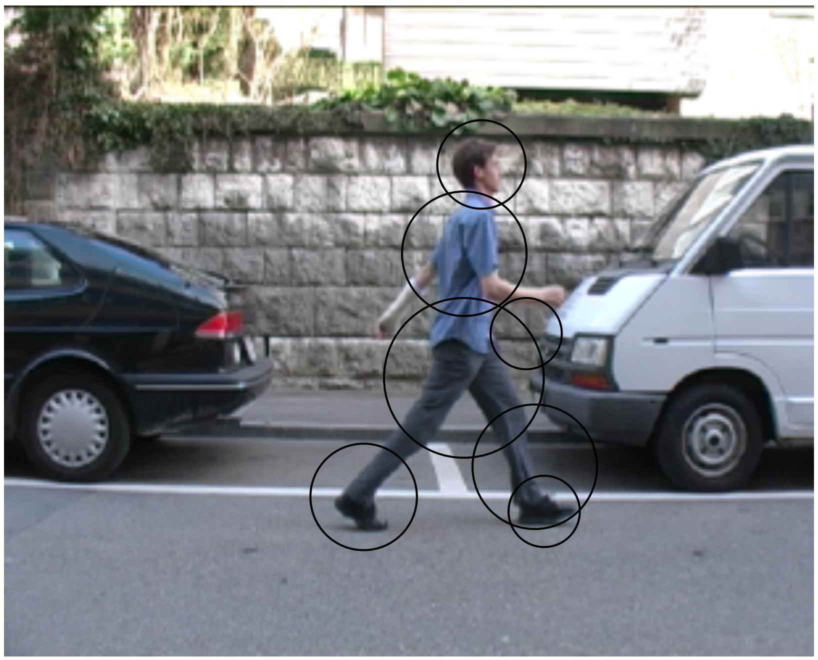




Appearance codebook

Source: B. Leibe

Another codebook



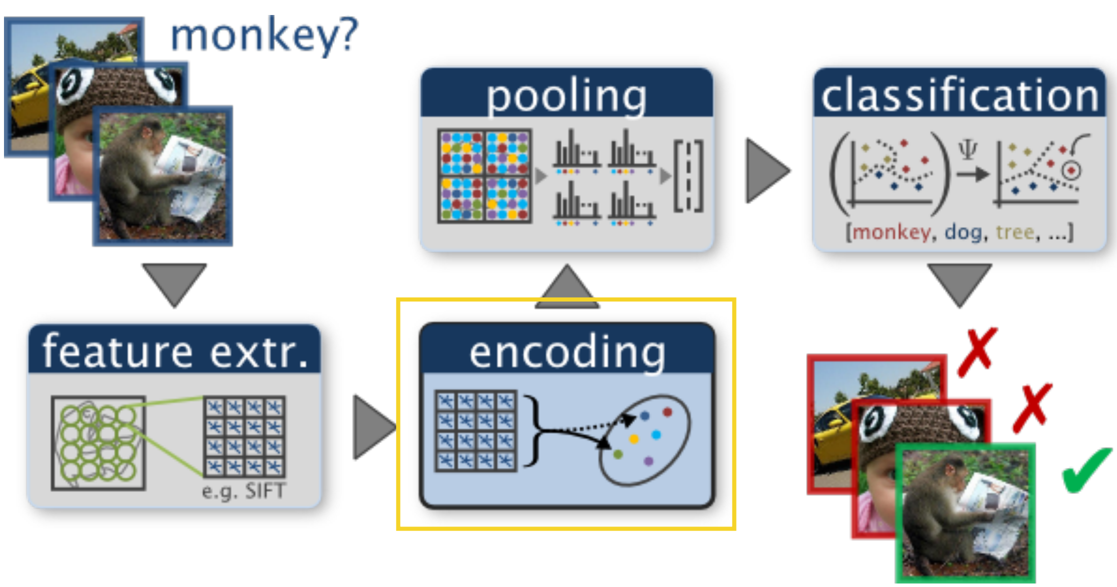
Appearance codebook

Source: B. Leibe

Lecture outline

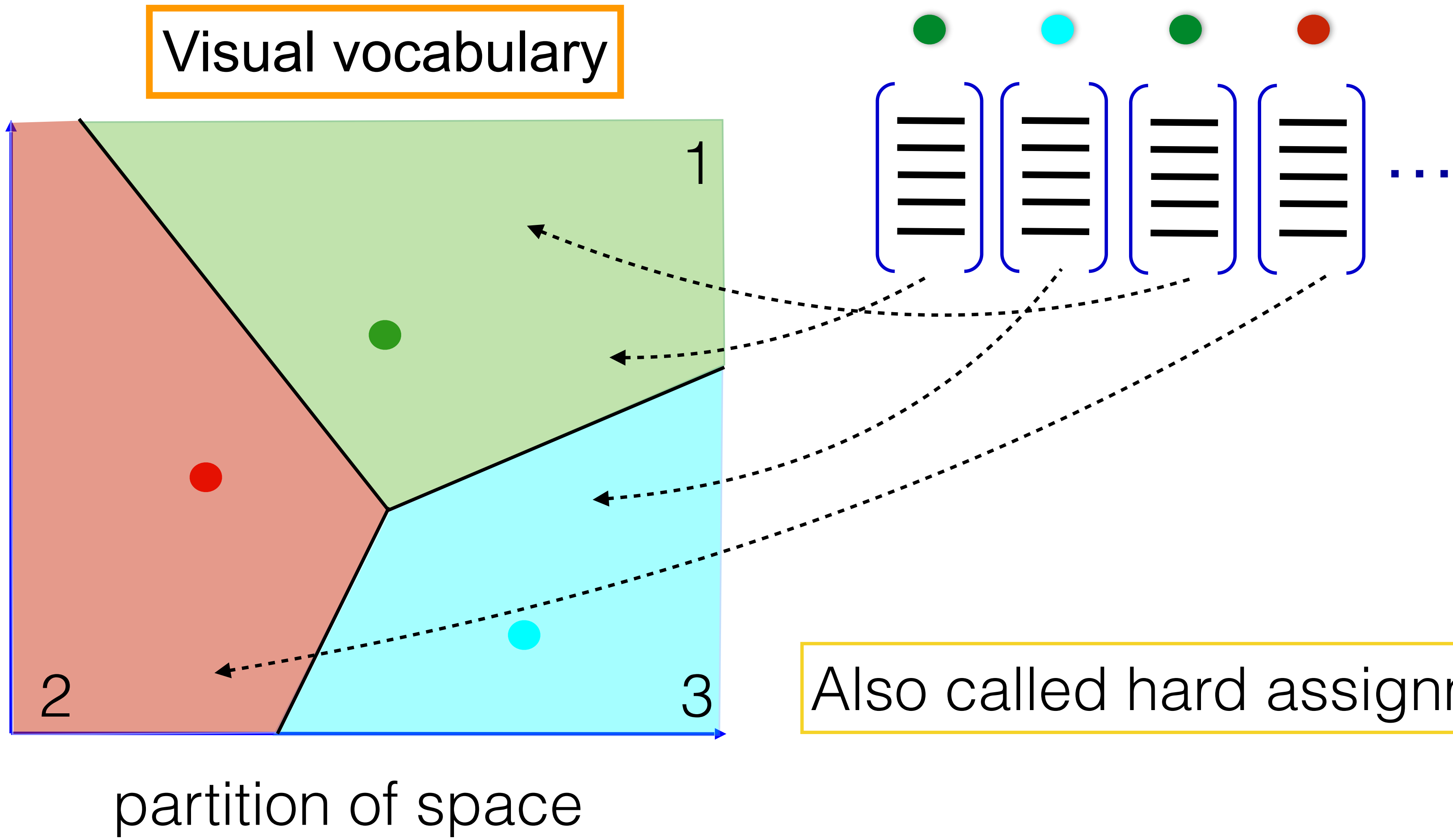
- Origin and motivation of the "bag of words" model
- Algorithm pipeline
  - Extracting local features
  - Learning a dictionary — clustering using k-means
  - Encoding methods — hard vs. soft assignment
  - Spatial pooling — pyramid representations

Figure from *Chatfield et al.,2011*



Encoding methods

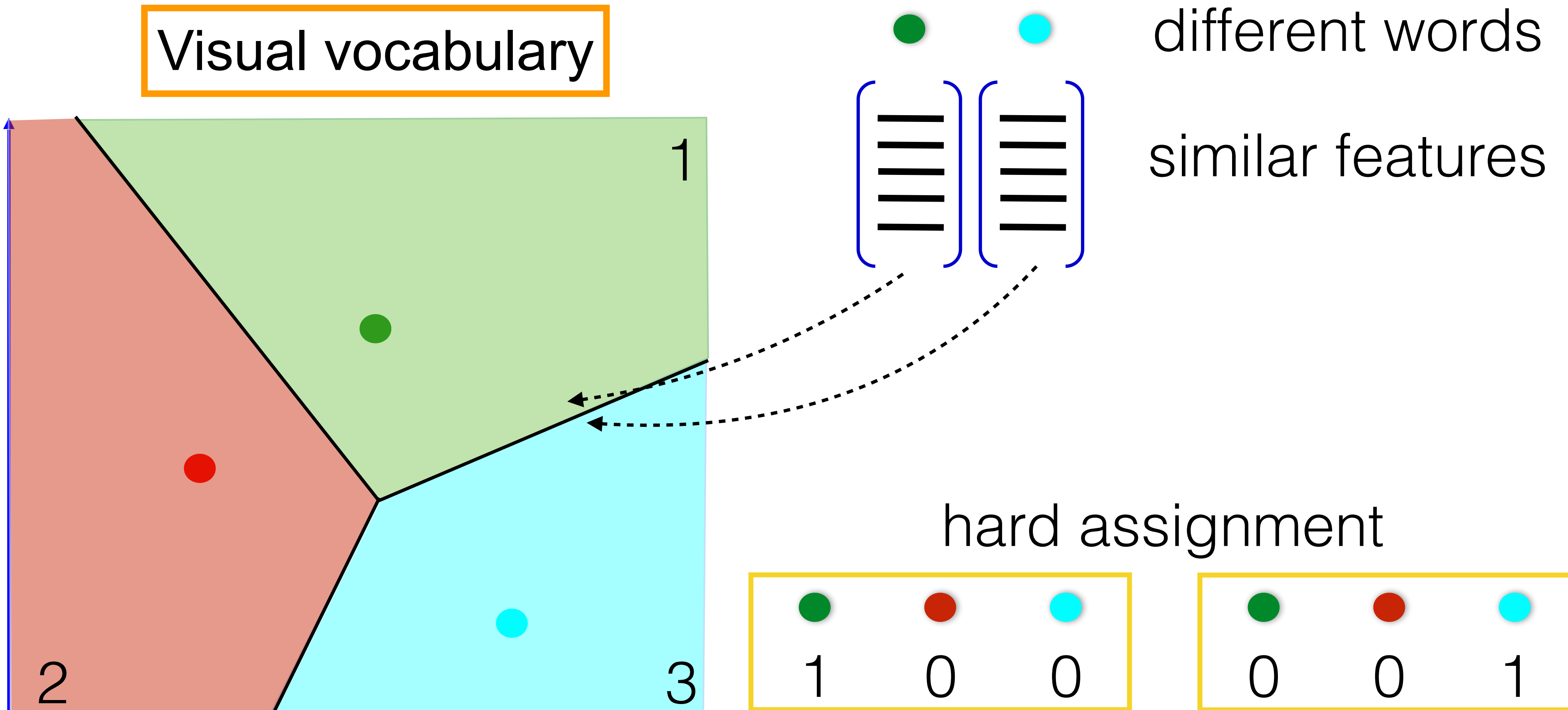
- Assigning words to features



Also called hard assignment

Encoding methods

- Assigning words to features



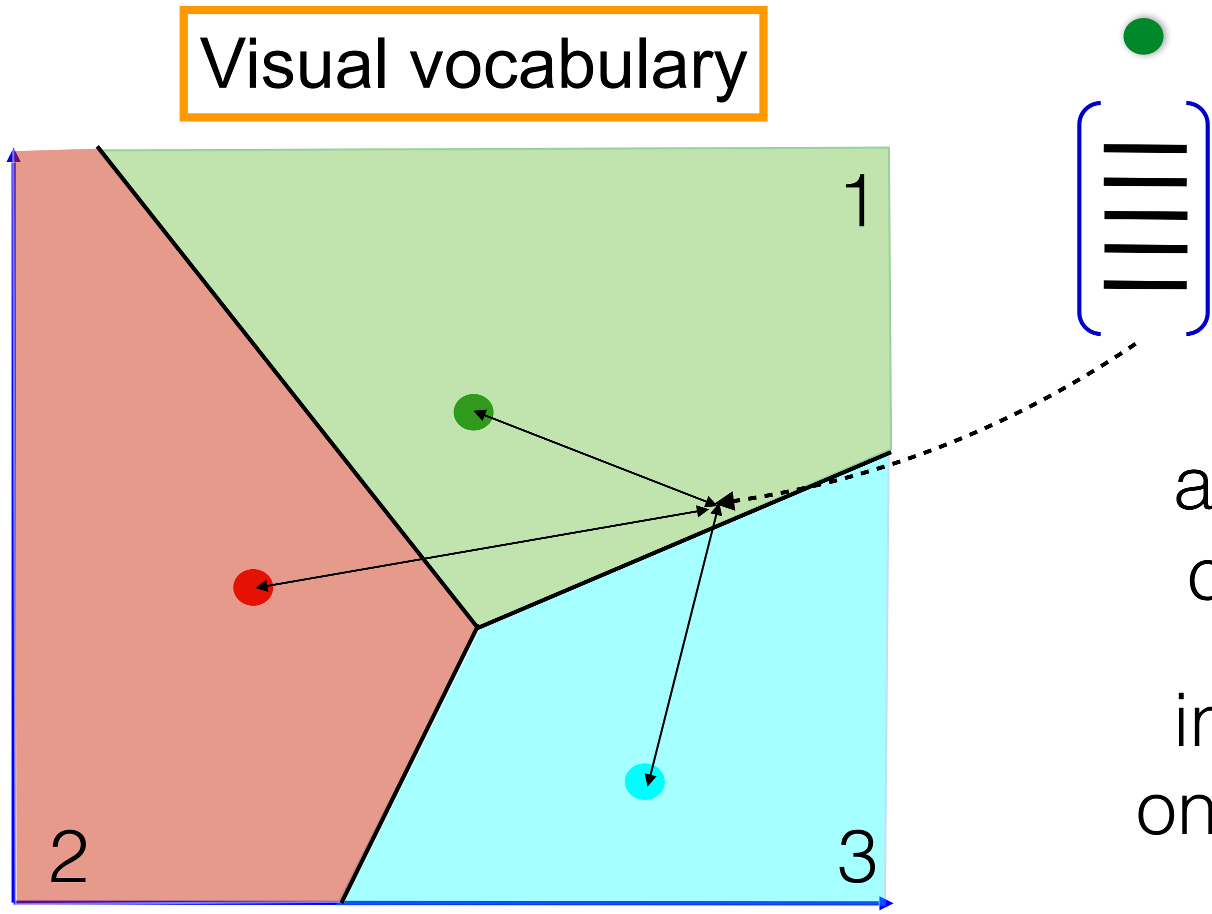
partition of space

large quantization error

Encoding methods

- Assigning words to features

soft assignment



partition of space

$$\alpha_i \propto e^{-f(d(\mathbf{x},\mathbf{c_i}))}$$

assign high weights to centers that are close

in practice non-zero to only k-nearest neighbors

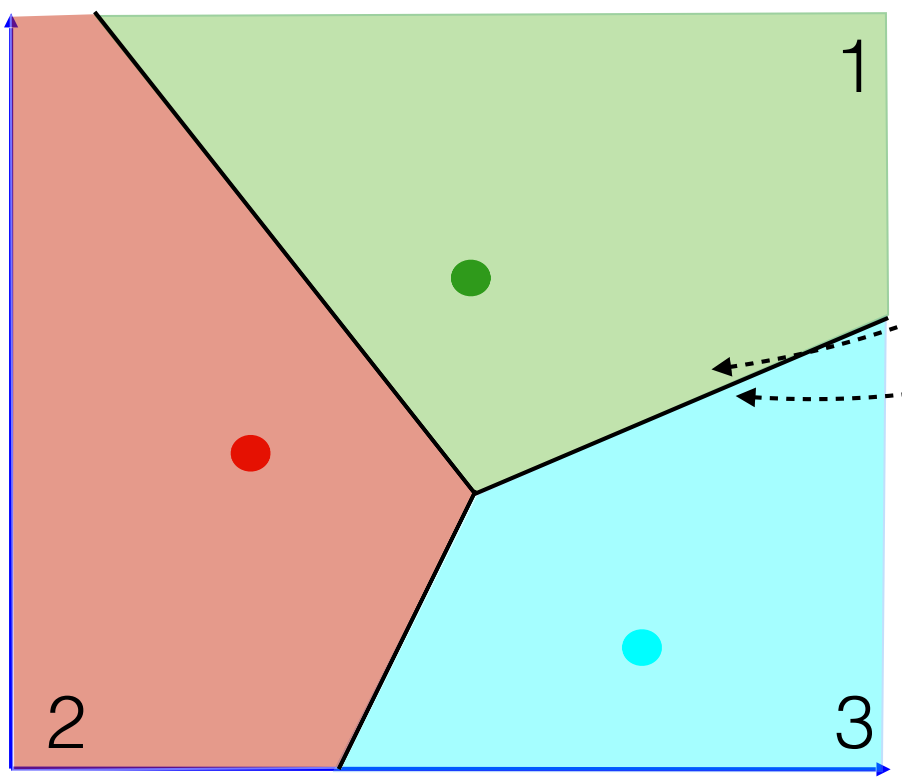
Encoding methods

- Assigning words to features

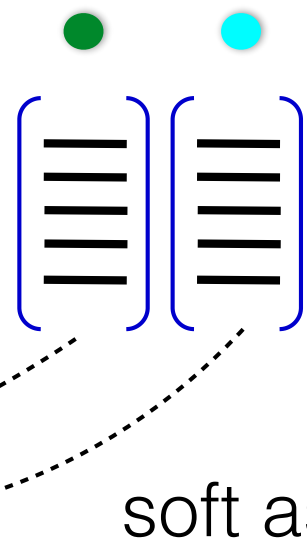
soft assignment

$$\alpha_i \propto e^{-f(d(\mathbf{x},\mathbf{c_i}))}$$

Visual vocabulary



partition of space



similar features

soft assignment

0.6	0	0.4

0.4	0	0.6

hard assignment

1	0	0

0	0	1

Encoding considerations

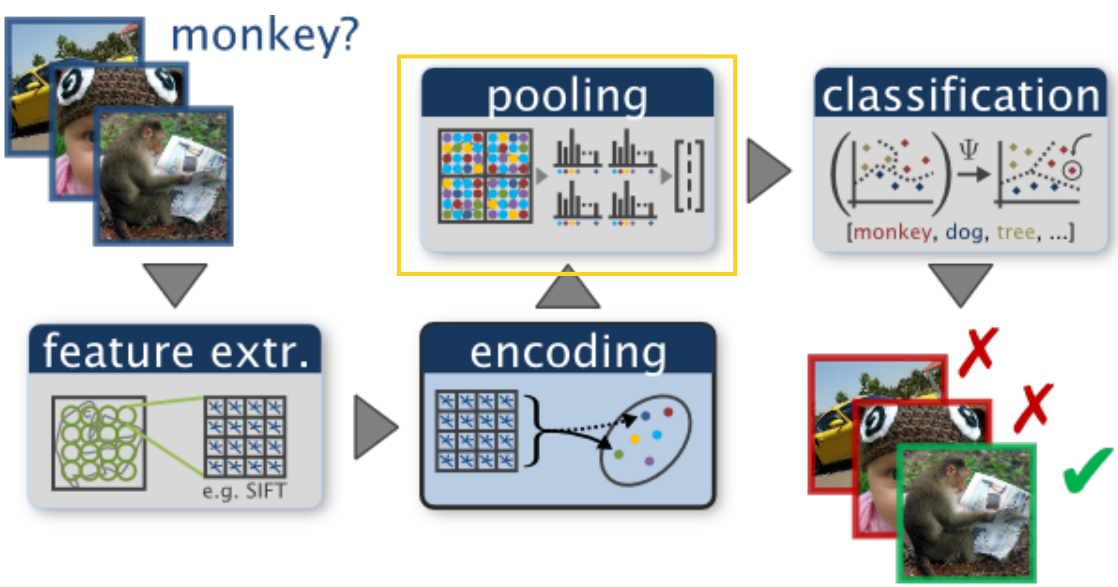
- What should be the size of the dictionary?
  - Too small: doesn't capture the variability of the data (underfitting)
  - Too large: too few points per cluster (overfitting)

- Speed of embedding
  - Exact nearest neighbor is slow if the dictionary is large
  - Approximate nearest neighbor techniques
    - Search trees — organize data in a tree
    - Hashing — create buckets in the feature space

Lecture outline

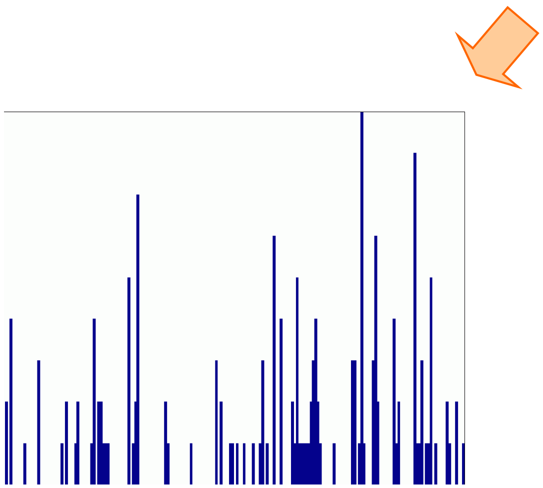
- Origin and motivation of the "bag of words" model
- Algorithm pipeline
  - Extracting local features
  - Learning a dictionary — clustering using k-means
  - Encoding methods — hard vs. soft assignment
  - Spatial pooling — pyramid representations

Figure from *Chatfield et al.,2011*



Spatial pyramids

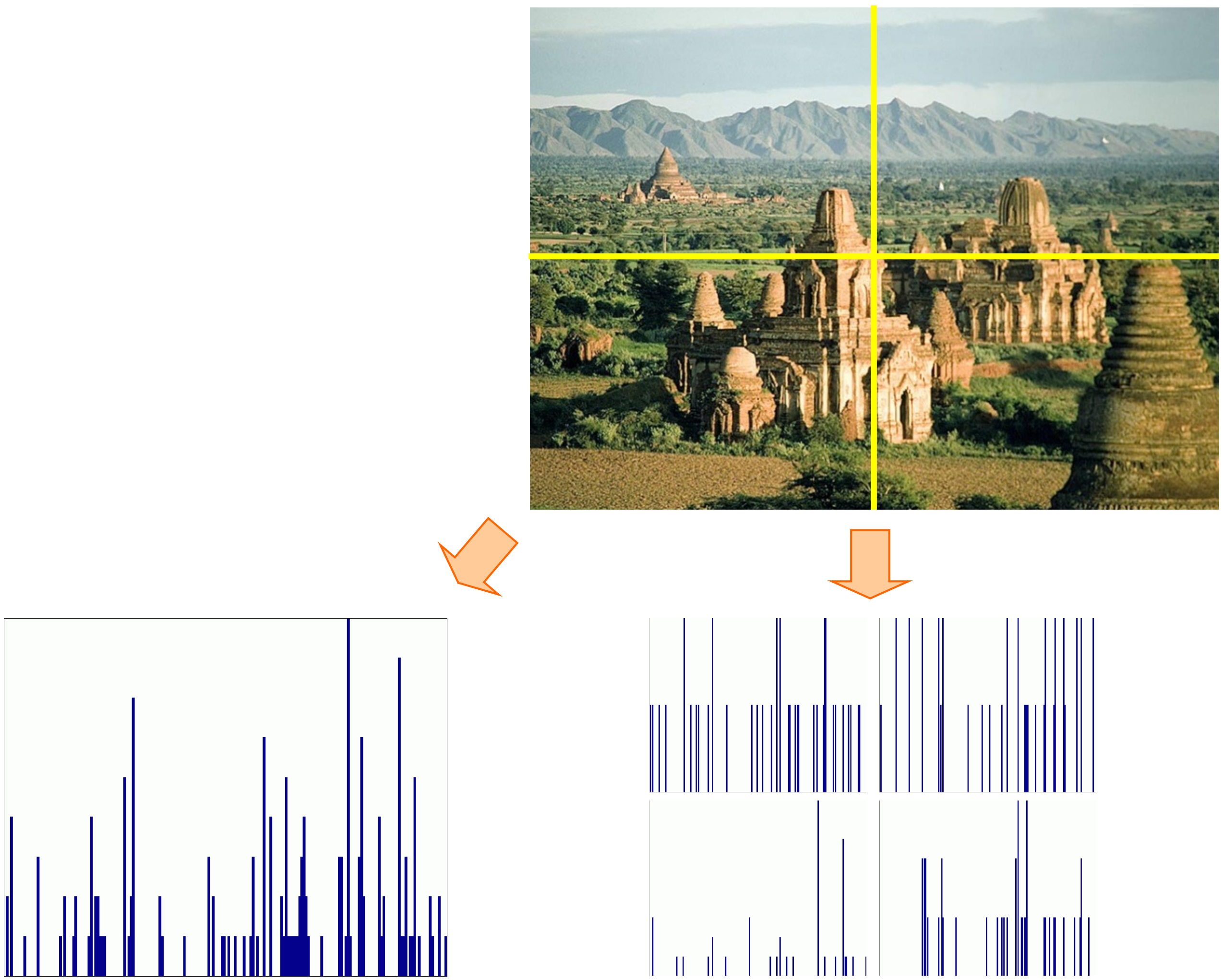
pooling: sum embeddings of local features within a region



Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramids

pooling: sum embeddings of local features within a region

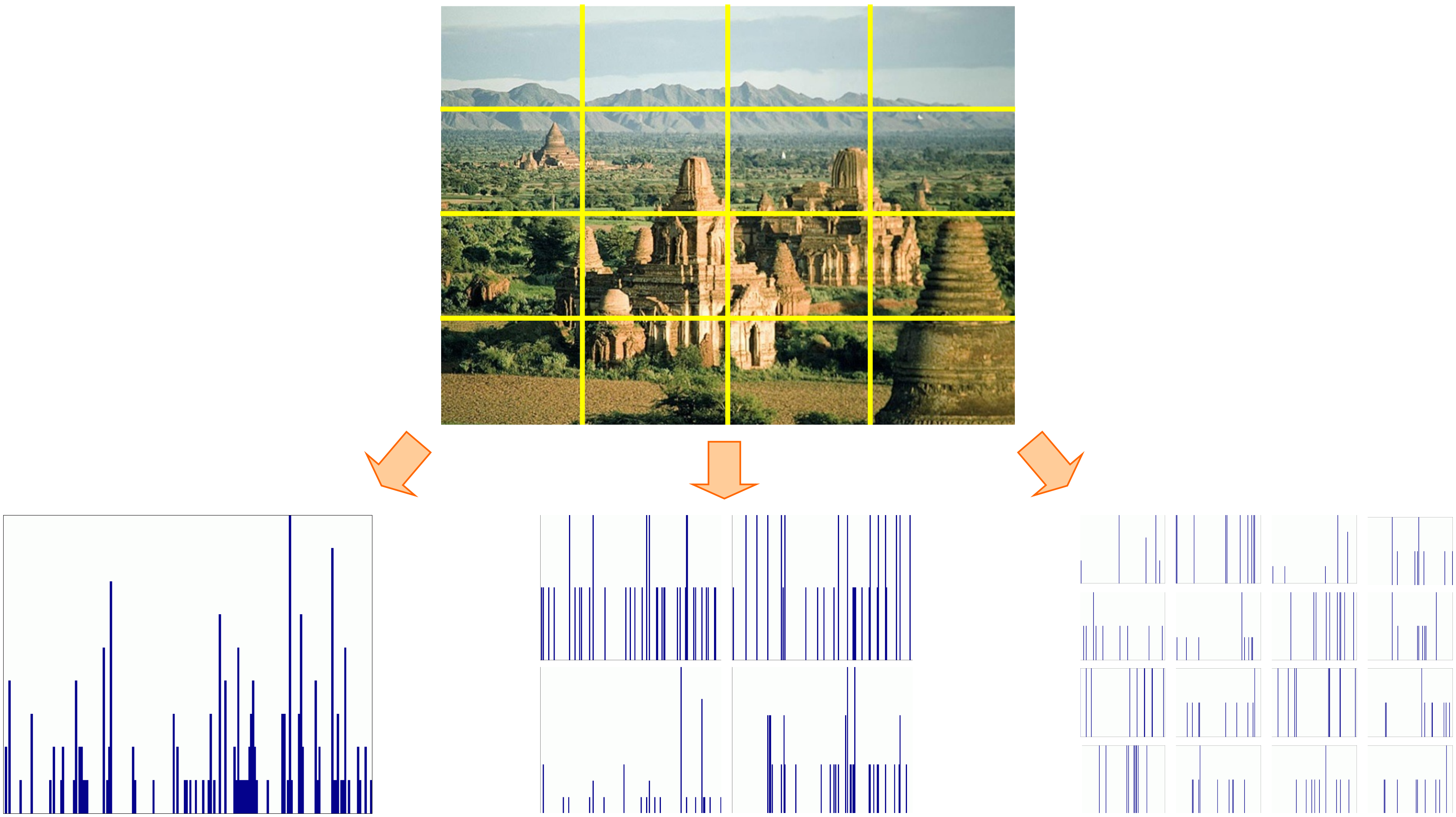


Same motivation as **SIFT** — keep coarse layout information

Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramids

pooling: sum embeddings of local features within a region



Same motivation as **SIFT** — keep coarse layout information

Lazebnik, Schmid & Ponce (CVPR 2006)

Summary of hand-crafted features

- Two families of features that work well with simple classifiers
  - Histogram of oriented gradients — captures overall shape
  - Bag of visual words — captures local shape and texture

shape

texture