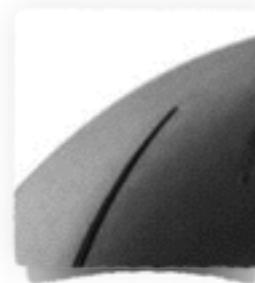
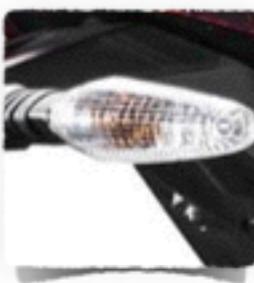


# Bag-of-Visual-Words

16-385 Computer Vision (Kris Kitani)  
**Carnegie Mellon University**

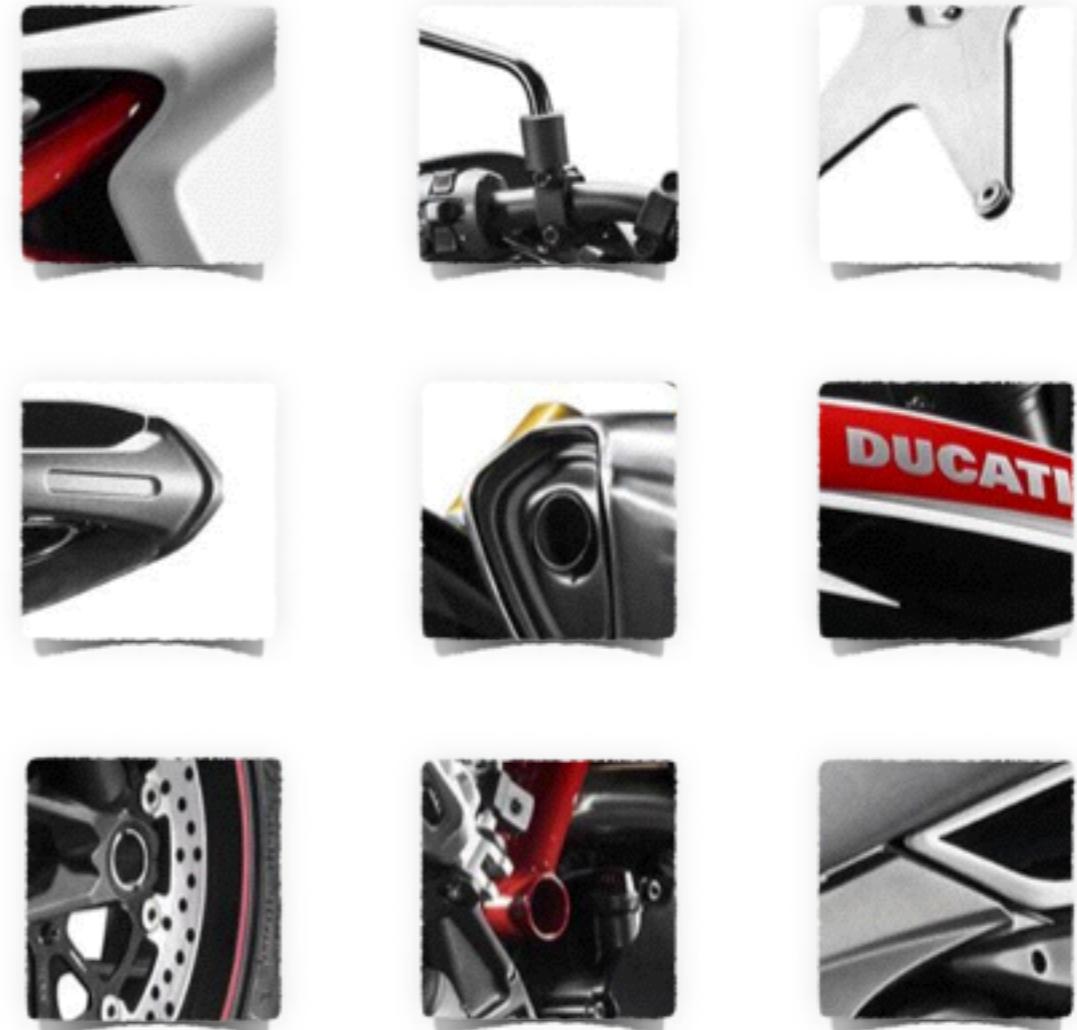
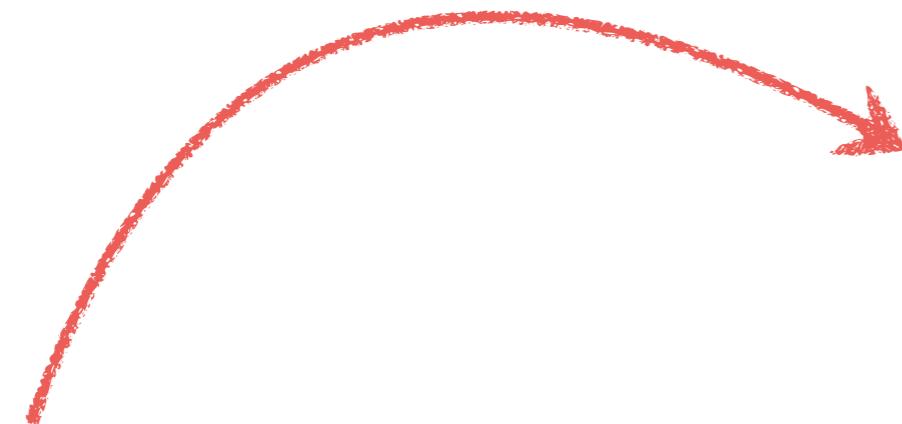
What object do these parts belong to?



Some local feature are very informative



An object as



a collection of local features  
(bag-of-features)

- deals well with occlusion
- scale invariant
- rotation invariant

# (not so) crazy assumption



spatial information of local features  
can be ignored for object recognition (i.e., verification)

# CalTech6 dataset



class	bag of features	bag of features	Parts-and-shape model
	Zhang et al. (2005)	Willamowski et al. (2004)	Fergus et al. (2003)
airplanes	<b>98.8</b>	97.1	90.2
cars (rear)	98.3	<b>98.6</b>	90.3
cars (side)	<b>95.0</b>	87.3	88.5
faces	<b>100</b>	99.3	96.4
motorbikes	<b>98.5</b>	98.0	92.5
spotted cats	<b>97.0</b>	—	90.0

Works pretty well for image-level classification

# Bag-of-features

represent a data item (document, texture, image)  
as a histogram over features

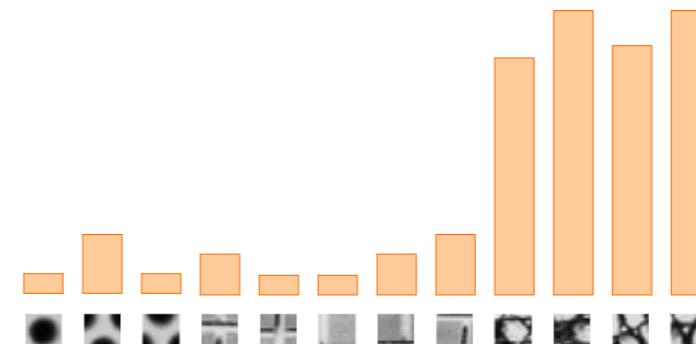
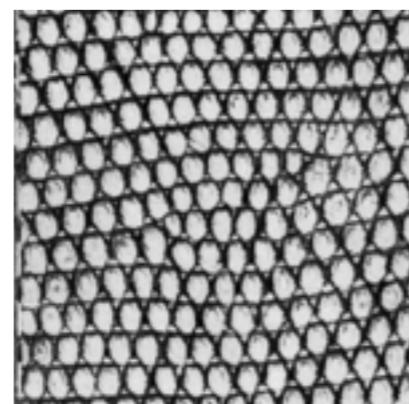
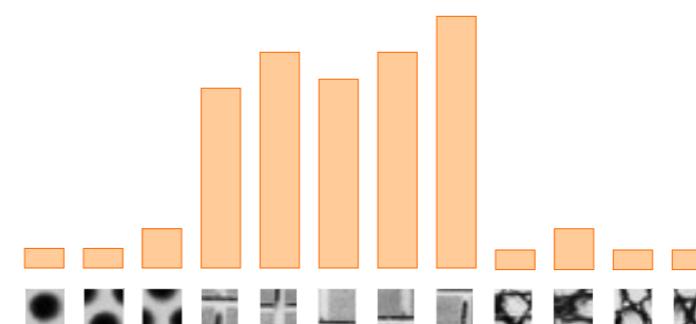
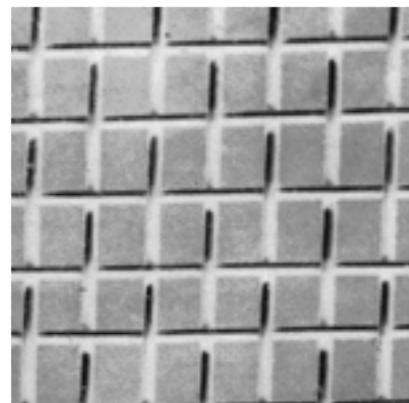
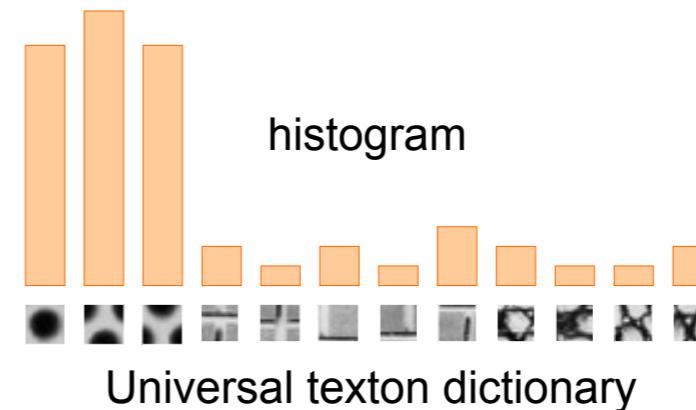
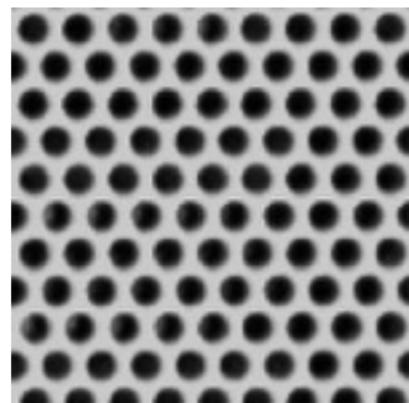
# Bag-of-features

represent a data item (document, texture, image)  
as a histogram over features

an old idea

(e.g., texture recognition and information retrieval)

# Texture recognition

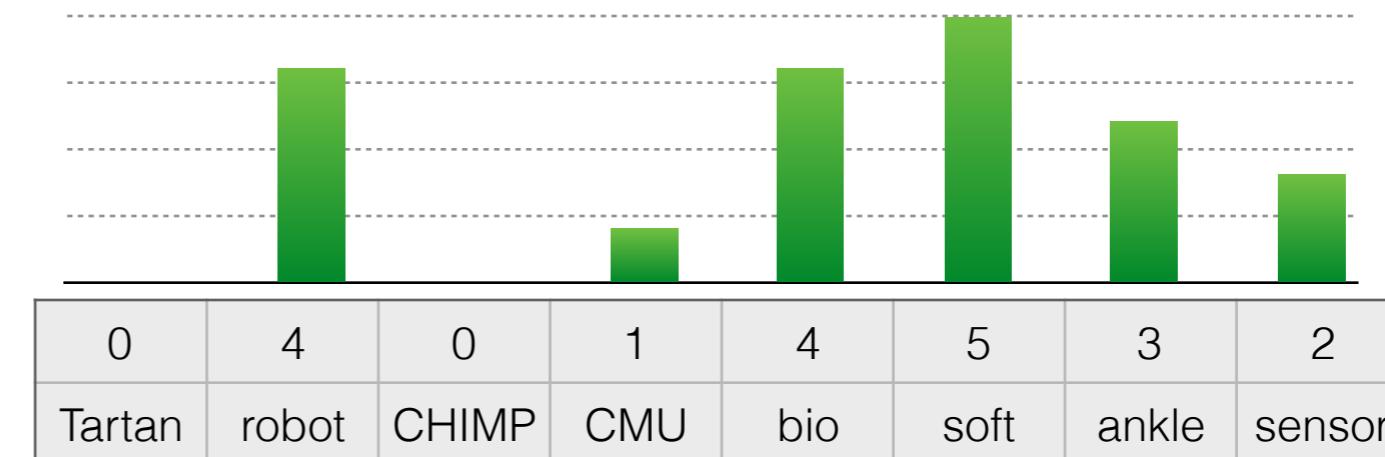
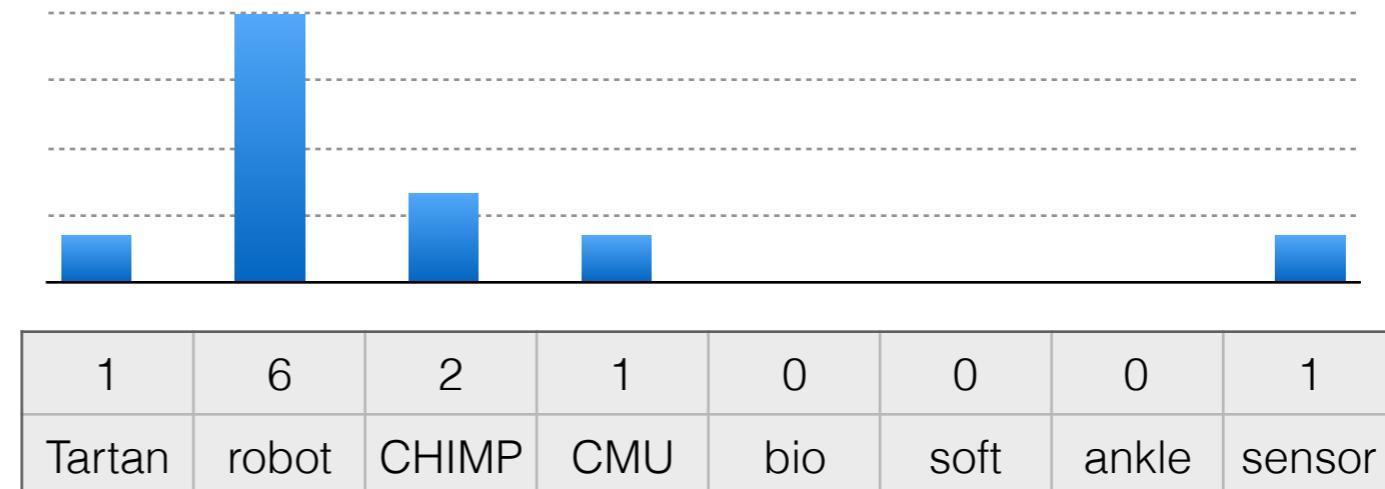


Julesz, 1981

Mori, Belongie and Malik, 2001

# Vector Space Model

G. Salton. 'Mathematics and Information Retrieval' Journal of Documentation, 1979



A document (datapoint) is a vector of counts over each word (feature)

$$\mathbf{v}_d = [n(w_{1,d}) \ n(w_{2,d}) \ \cdots \ n(w_{T,d})]$$

$n(\cdot)$  counts the number of occurrences

just a histogram over words

What is the similarity between two documents?



A document (datapoint) is a vector of counts over each word (feature)

$$\mathbf{v}_d = [n(w_{1,d}) \ n(w_{2,d}) \ \cdots \ n(w_{T,d})]$$

$n(\cdot)$  counts the number of occurrences

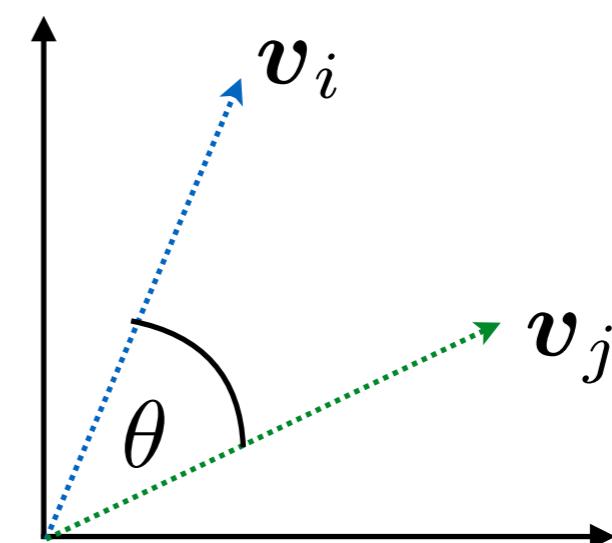
just a histogram over words

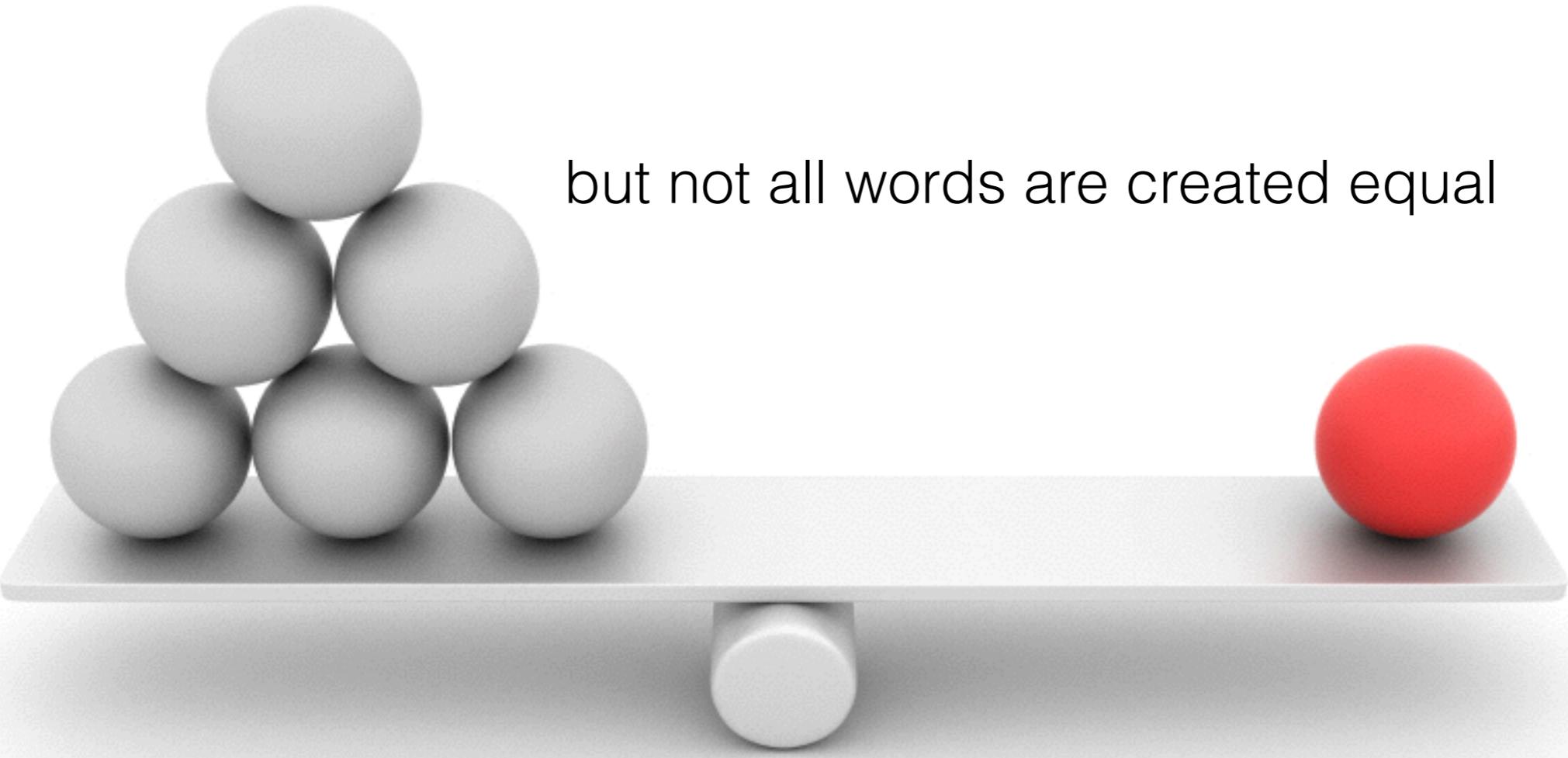
What is the similarity between two documents?



Use any distance you want but the cosine distance is fast.

$$\begin{aligned} d(\mathbf{v}_i, \mathbf{v}_j) &= \cos \theta \\ &= \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\| \|\mathbf{v}_j\|} \end{aligned}$$





but not all words are created equal

# TF-IDF

Term Frequency Inverse Document Frequency

$$\mathbf{v}_d = [n(w_{1,d}) \ n(w_{2,d}) \ \cdots \ n(w_{T,d})]$$

weigh each word by a heuristic

$$\mathbf{v}_d = [n(w_{1,d})\alpha_1 \ n(w_{2,d})\alpha_2 \ \cdots \ n(w_{T,d})\alpha_T]$$

term  
frequency                          inverse document  
    frequency

$$n(w_{i,d})\alpha_i = n(w_{i,d}) \log \left\{ \frac{D}{\sum_{d'} 1[w_i \in d']} \right\}$$

(down-weights **common** terms)

# Standard BOW pipeline

(for image classification)

## **Dictionary Learning:**

Learn Visual Words using clustering

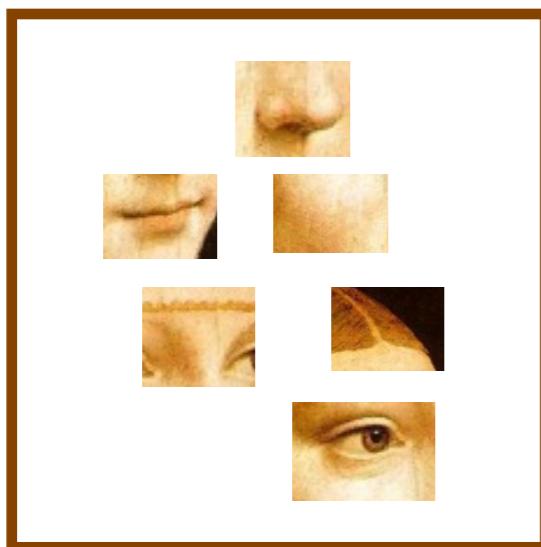
**Encode:**  
build Bags-of-Words (BOW) vectors  
for each image

**Classify:**  
Train and test data using BOWs

# Dictionary Learning:

## Learn Visual Words using clustering

1. extract features (e.g., SIFT) from images



# **Dictionary Learning:**

## Learn Visual Words using clustering

2. Learn visual dictionary (e.g., K-means clustering)

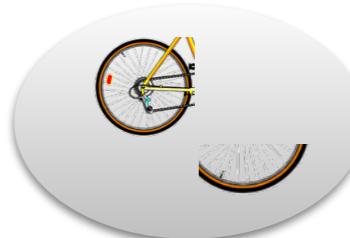


## **Dictionary Learning:**

Learn Visual Words using clustering

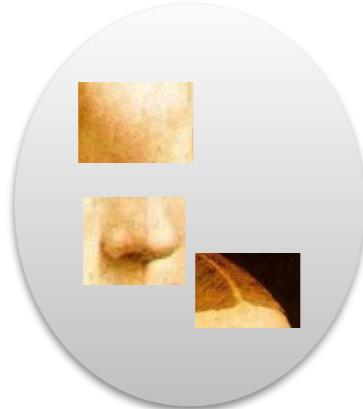
**Encode:**  
build Bags-of-Words (BOW) vectors  
for each image

**Classify:**  
Train and test data using BOWs



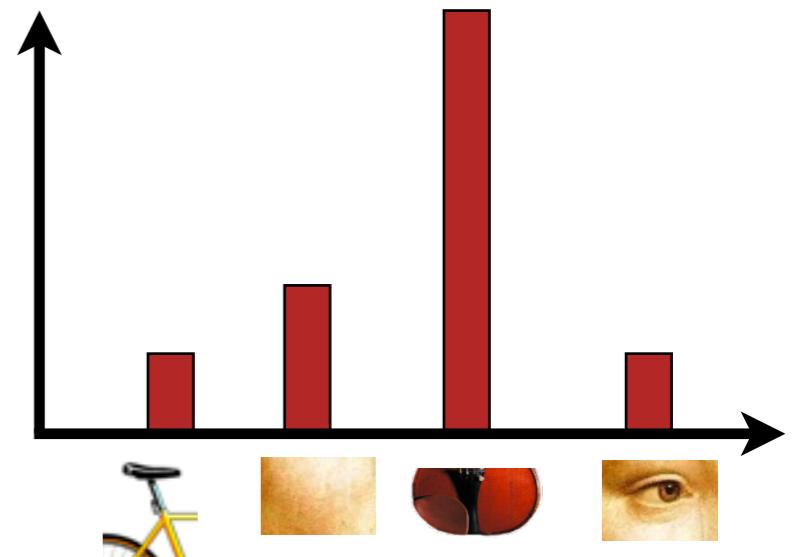
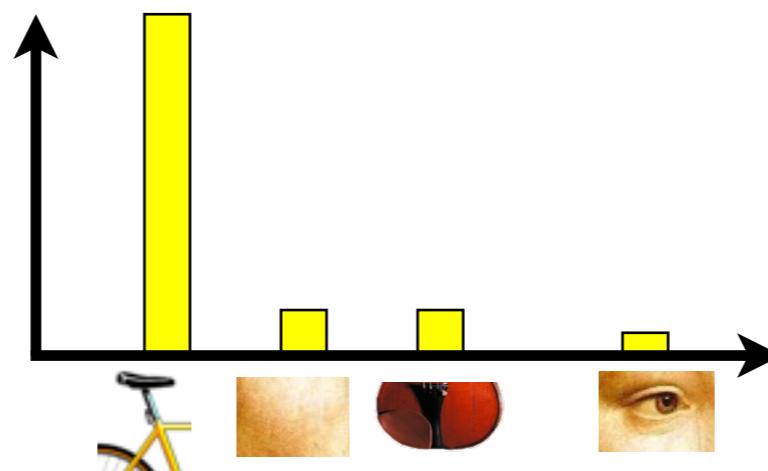
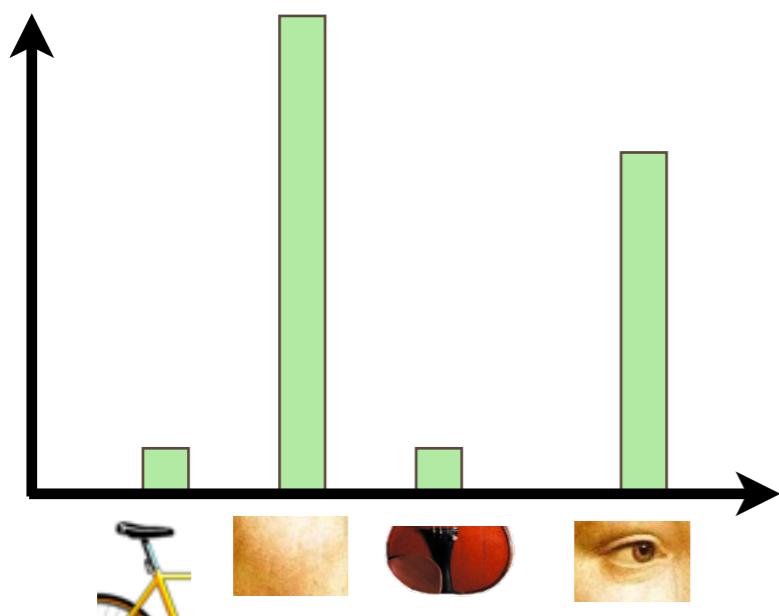
1. Quantization: image features gets associated to a visual word (nearest cluster center)

**Encode:**  
build Bags-of-Words (BOW) vectors  
for each image



**Encode:**  
build Bags-of-Words (BOW) vectors  
for each image

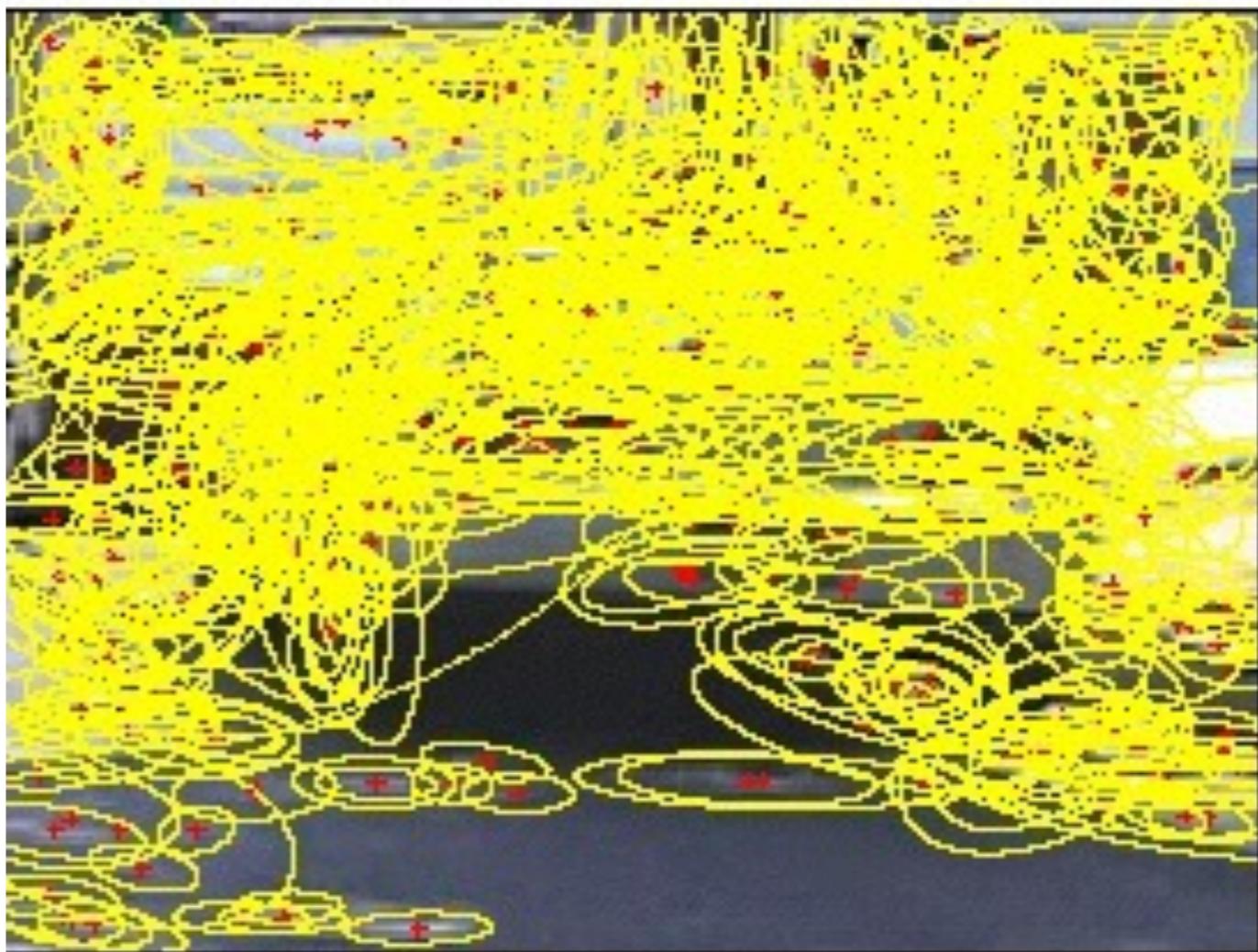
2. Histogram: count the  
number of visual word  
occurrences

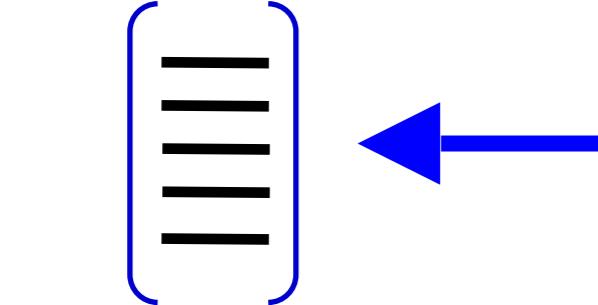


# Feature Extraction

*What kinds of features can we extract?*

- 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)

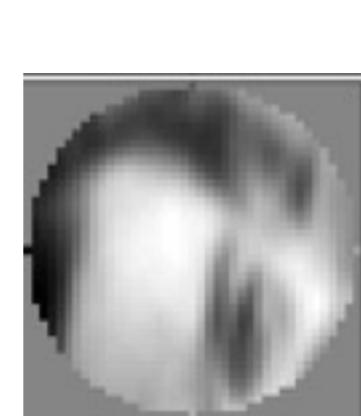




**Compute SIFT  
descriptor**

[Lowe'99]

**Normalize patch**

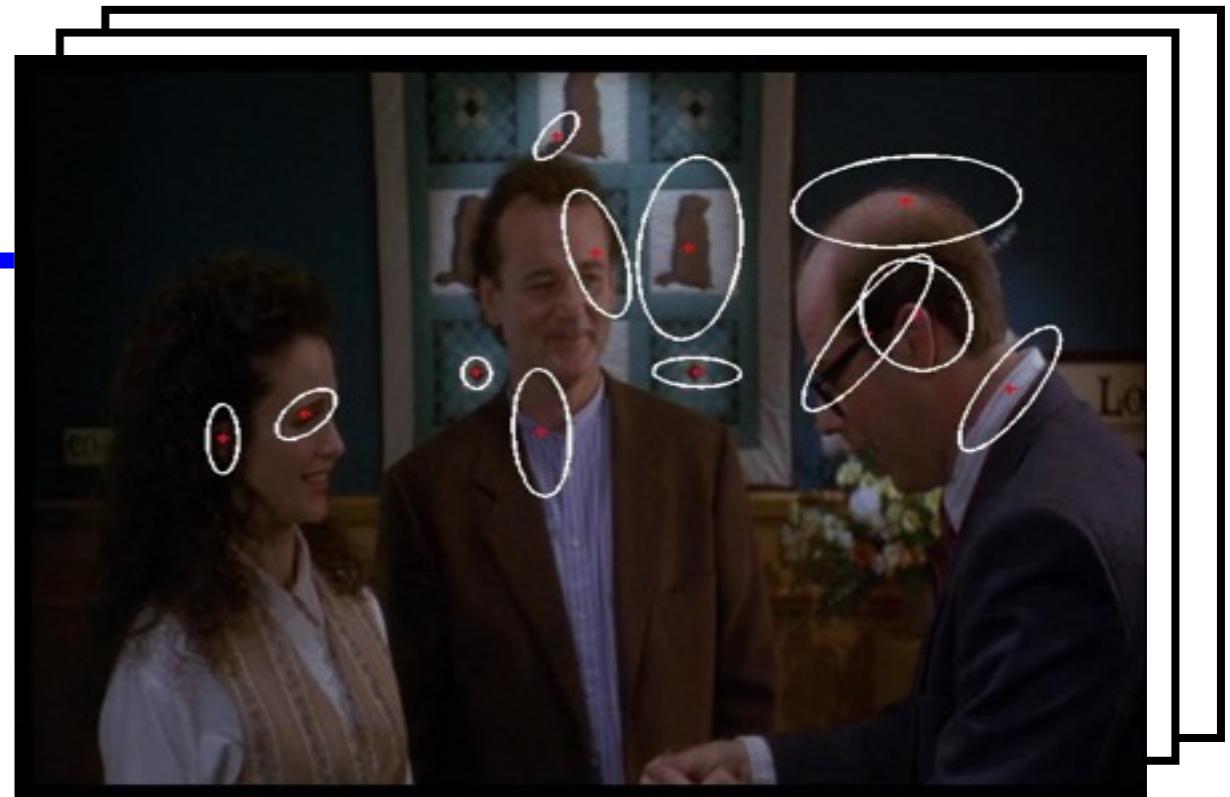
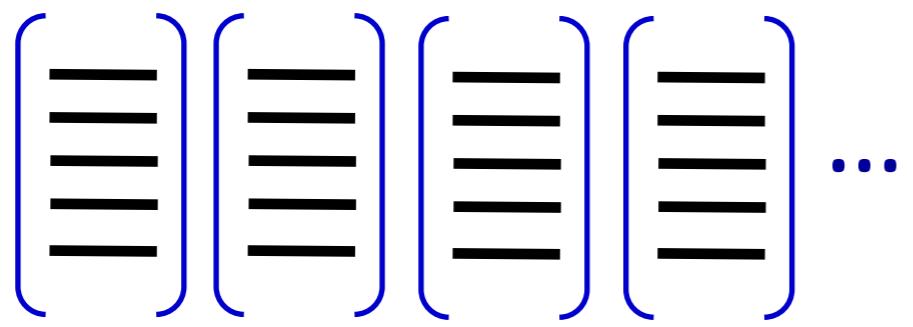


**Detect patches**

[Mikojaczyk and Schmid '02]

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

[Sivic & Zisserman, '03]



# Visual Vocabulary

(coding and vector quantization)

# Alternative perspective...

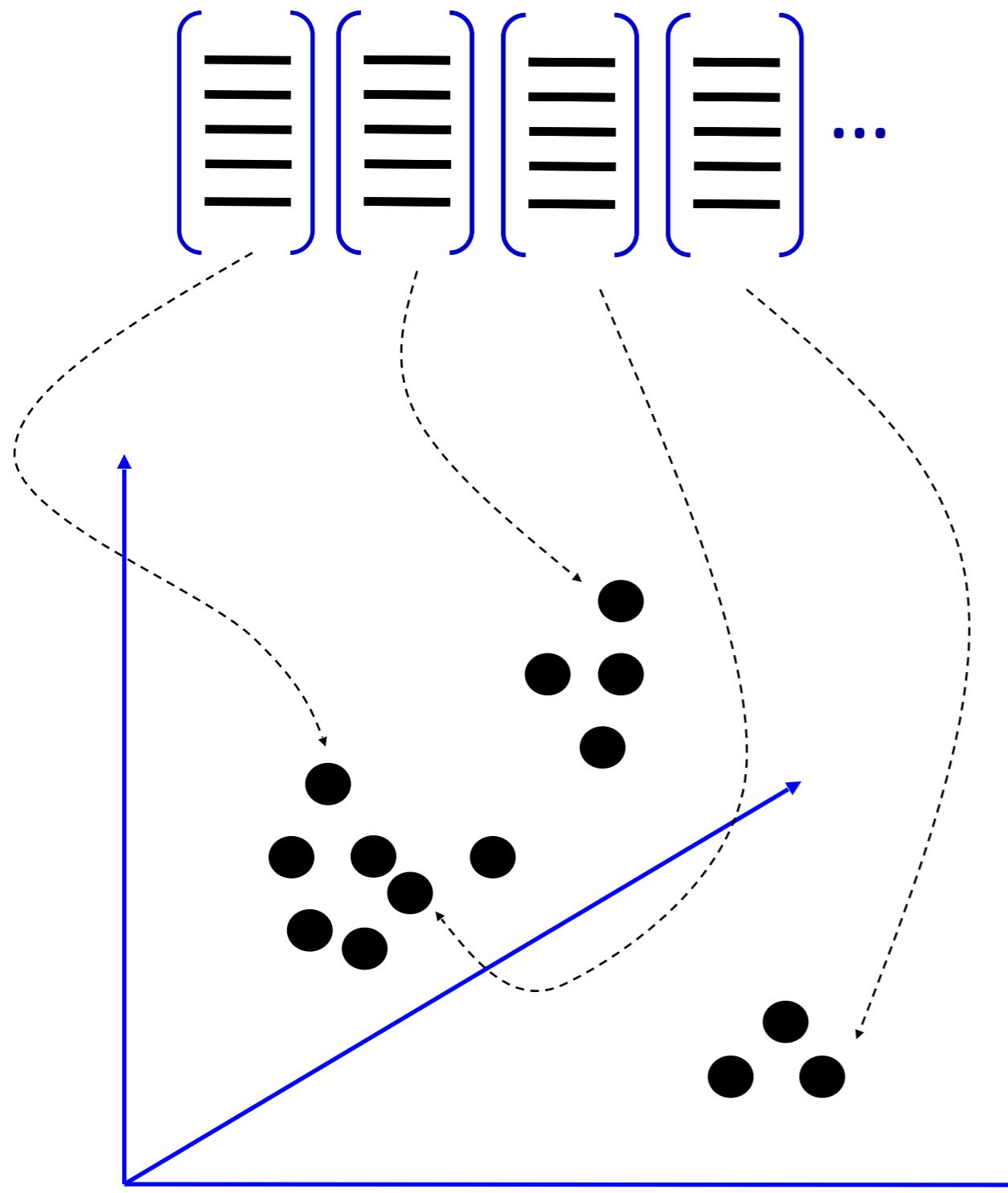
visual vocabulary = code book

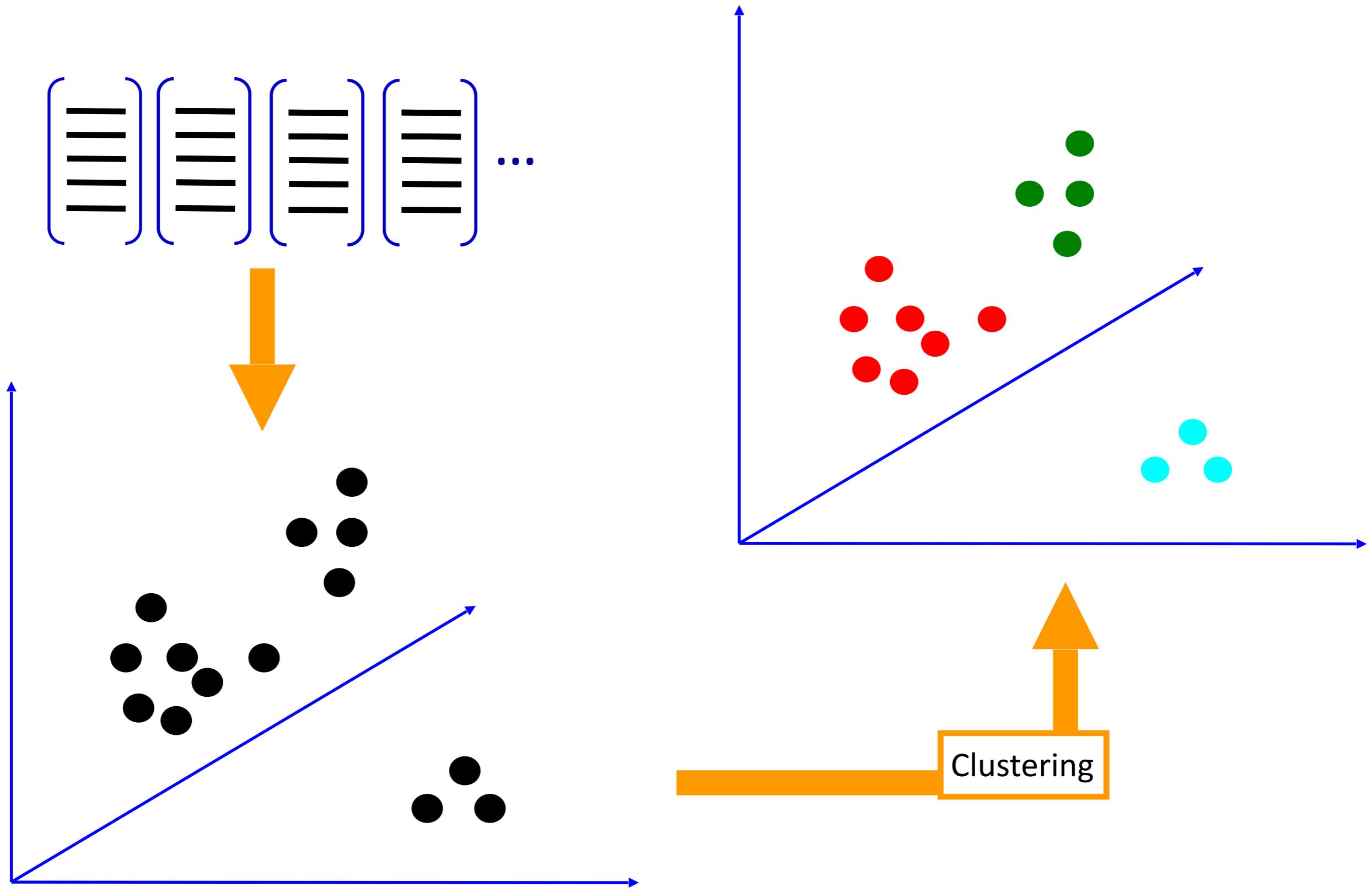
visual word = code vector

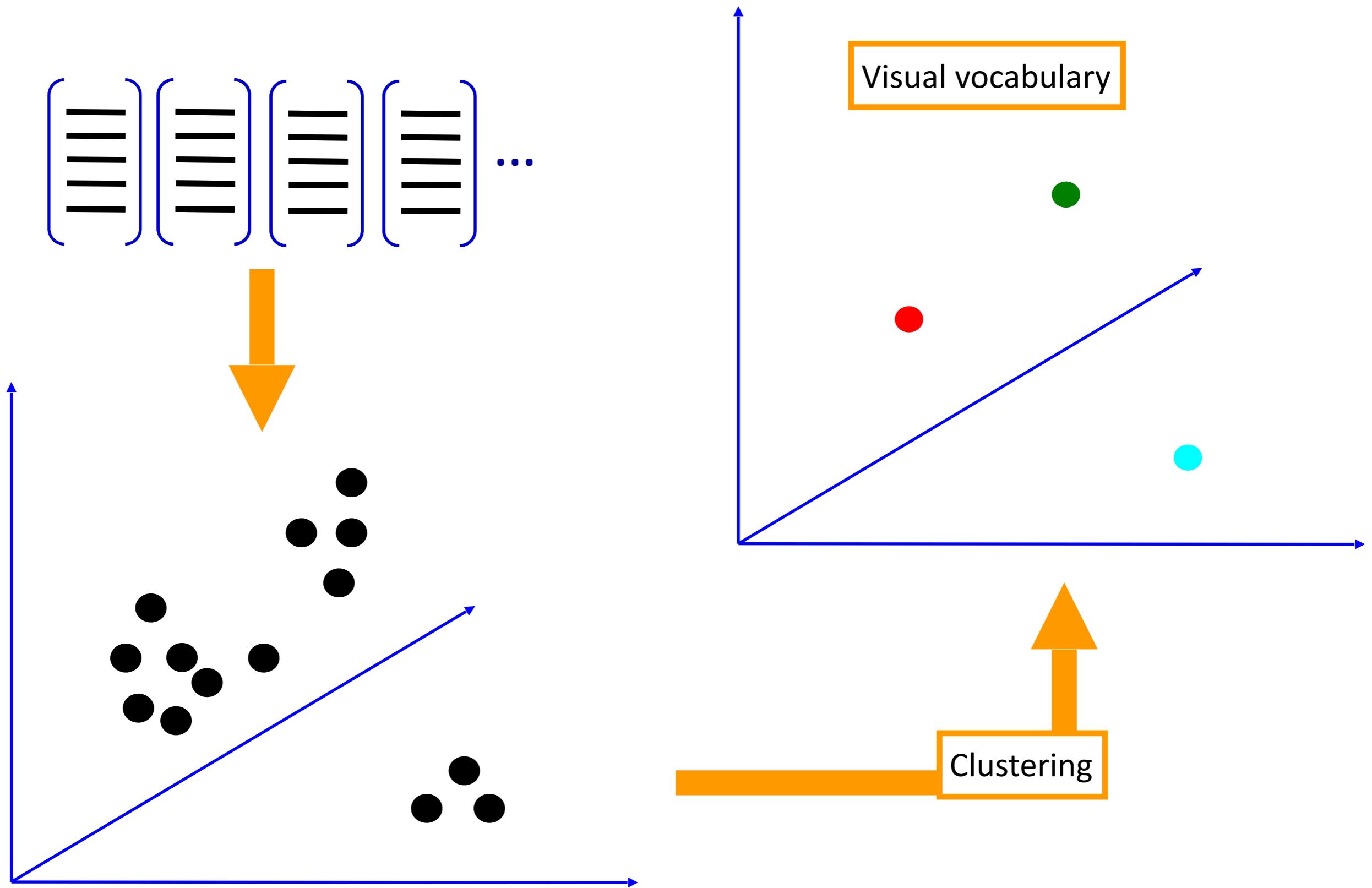
This is part of the secret code used by General Washington, Benjamin Tallmadge, Robert Townsend, and Abraham Woodhull, during the Revolutionary War.			
USE OF MEANS	MEANS	USE OF	MEANS
e	a	711	General Washington
f	b	712	Clinton
g	c	713	Tryon
h	d	721	Major Tallmadge
i	e	722	alias John Bolton
j	f	723	Abraham Woodhull
k	g	724	alias Samuel Culper
l	h	725	alias Samuel Culper, Jr.
m	i	726	Robert Townsend
n	j	727	alias Samuel Culper, Jr.
p	k	728	Aupin Roe
q	l	729	Caleb Brewster
r	m	730	Rivington
s	n	731	New York
t	o	732	Long Island
w	p	733	Setauket
x	q	745	England
y	r	341	January
z	s	345	February
		374	March
		22	April
		373	May
		336	June
		337	July
		29	August
		616	September
		462	October
		427	November
		154	December
		15	advice
		28	appointment
		60	better
		121	day
		156	deliver
		151	disorder
		178	enemy
		174	express
		230	guineas
		286	ink
		309	infantry
		317	importance
		322	inquiry
		345	knowledge
		347	land
		349	low
		355	lady
		356	letter
		371	man
		476	parts
		585	refugees
		592	ships
		660	vigilant
		680	war
		691	written
		708	year
		73	camp

The **codebook** is used for quantizing features

A **vector quantizer** takes a feature vector and maps it to the index of the nearest code vector in a codebook



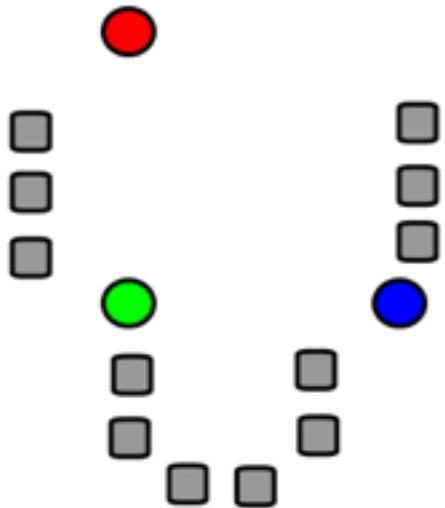




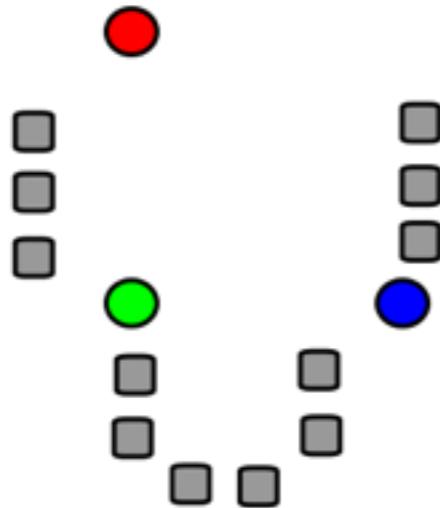
# K-means Clustering

Given k:

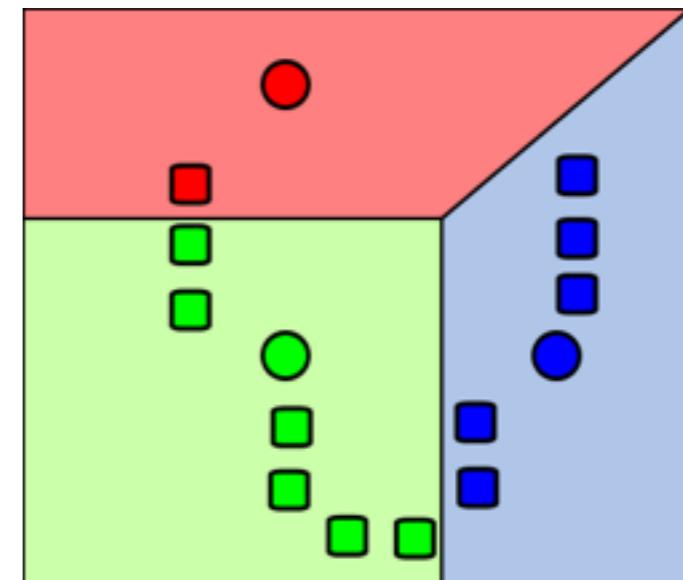
1. Select initial centroids at random.
2. Assign each object to the cluster with the nearest centroid.
3. Compute each centroid as the mean of the objects assigned to it.
4. Repeat previous 2 steps until no change.



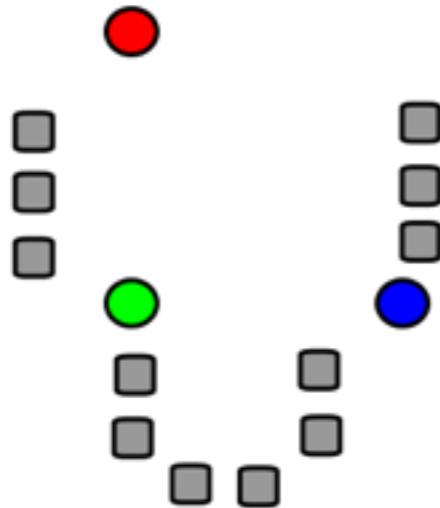
1. Select initial  
centroids at random



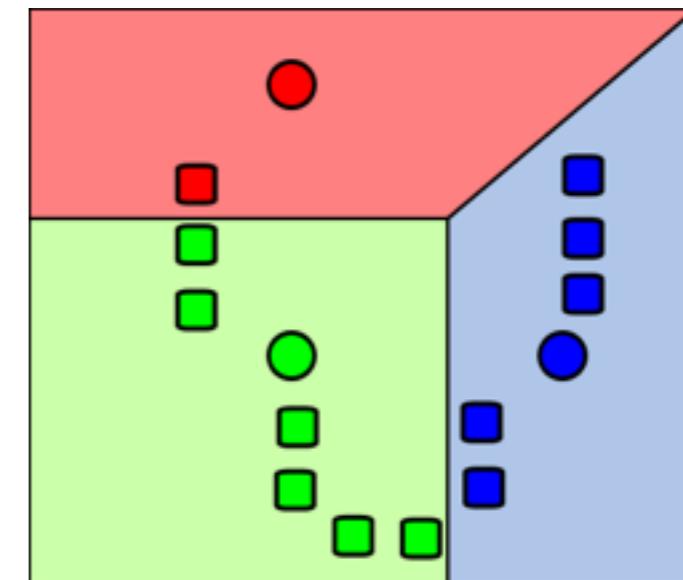
1. Select initial  
centroids at random



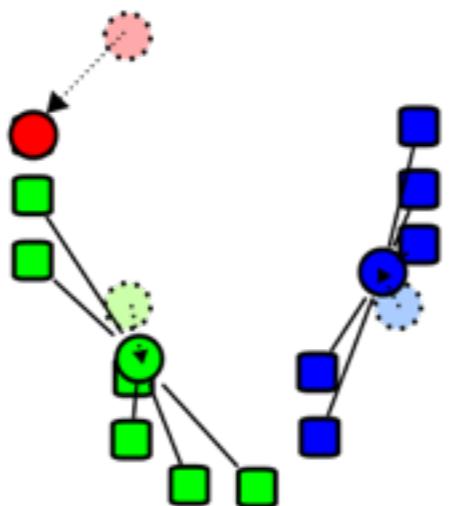
2. Assign each object to  
the cluster with the  
nearest centroid.



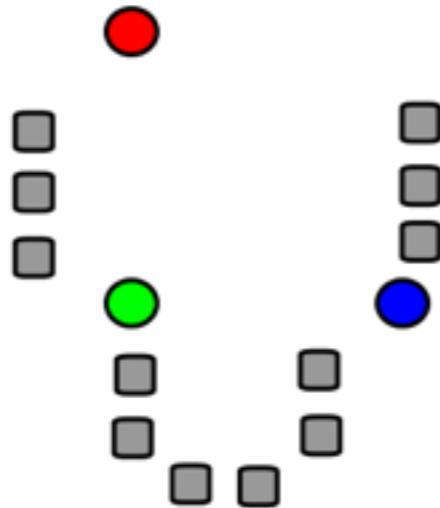
1. Select initial centroids at random



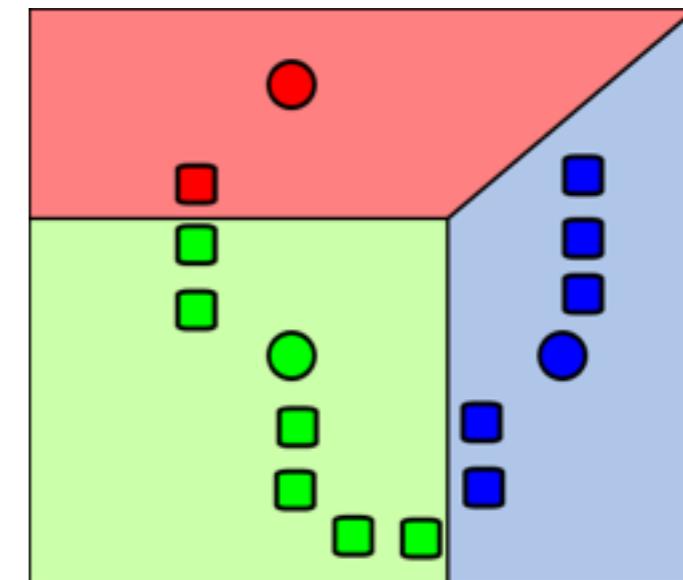
2. Assign each object to the cluster with the nearest centroid.



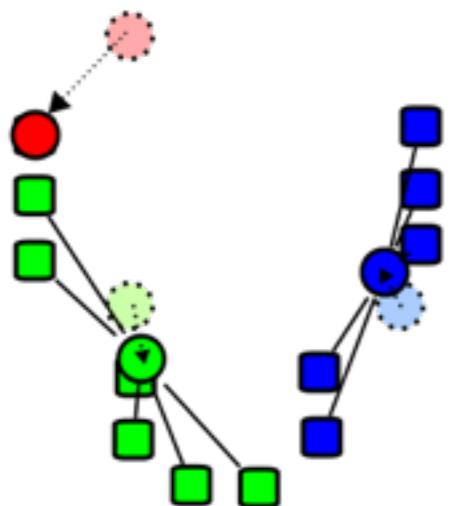
3. Compute each centroid as the mean of the objects assigned to it (go to 2)



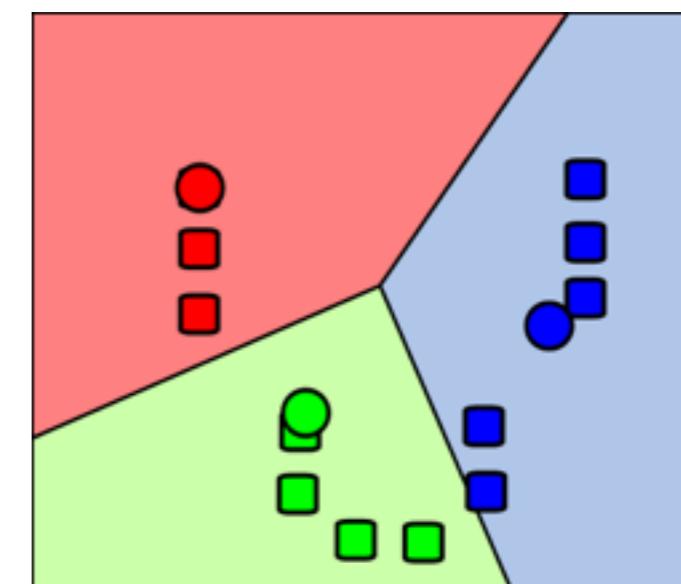
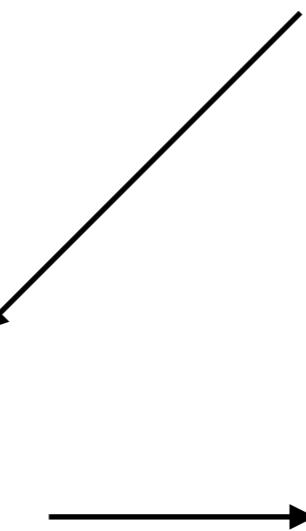
1. Select initial  
centroids at random



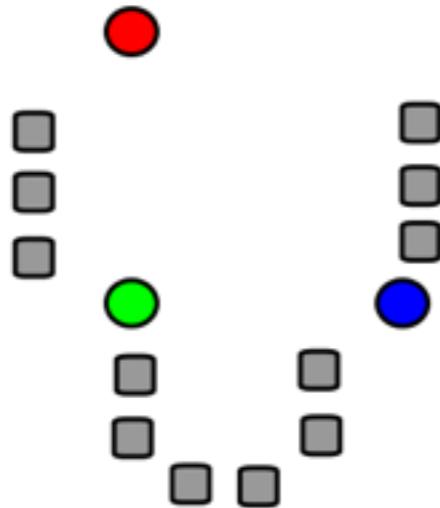
2. Assign each object to  
the cluster with the  
nearest centroid.



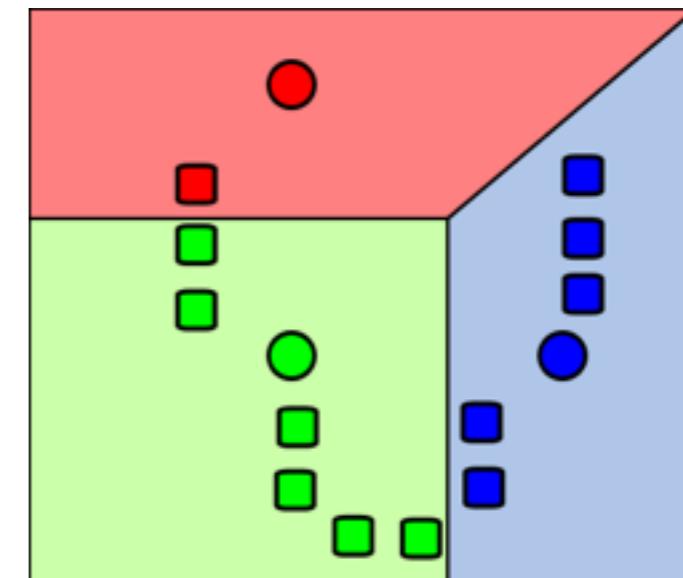
3. Compute each centroid as the  
mean of the objects assigned to  
it (go to 2)



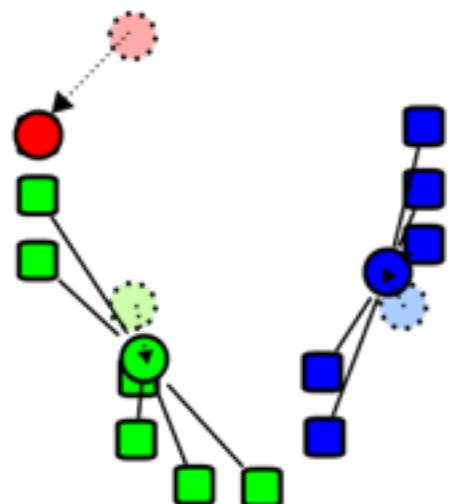
2. Assign each object to  
the cluster with the  
nearest centroid.



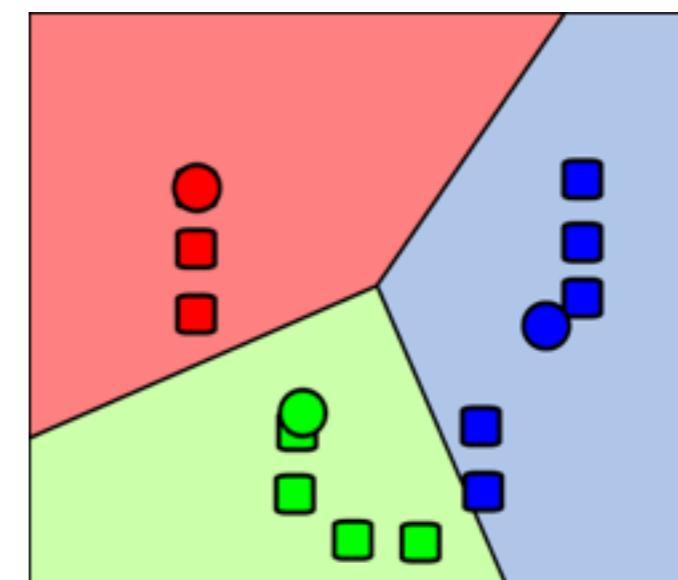
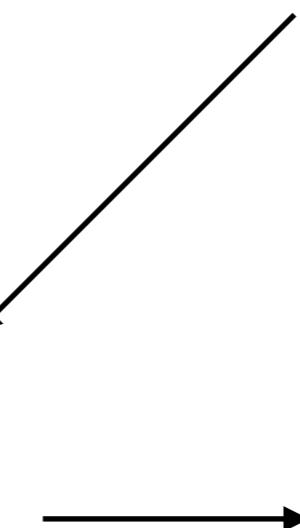
1. Select initial  
centroids at random



2. Assign each object to  
the cluster with the  
nearest centroid.



3. Compute each centroid as the  
mean of the objects assigned to  
it (go to 2)



2. Assign each object to  
the cluster with the  
nearest centroid.

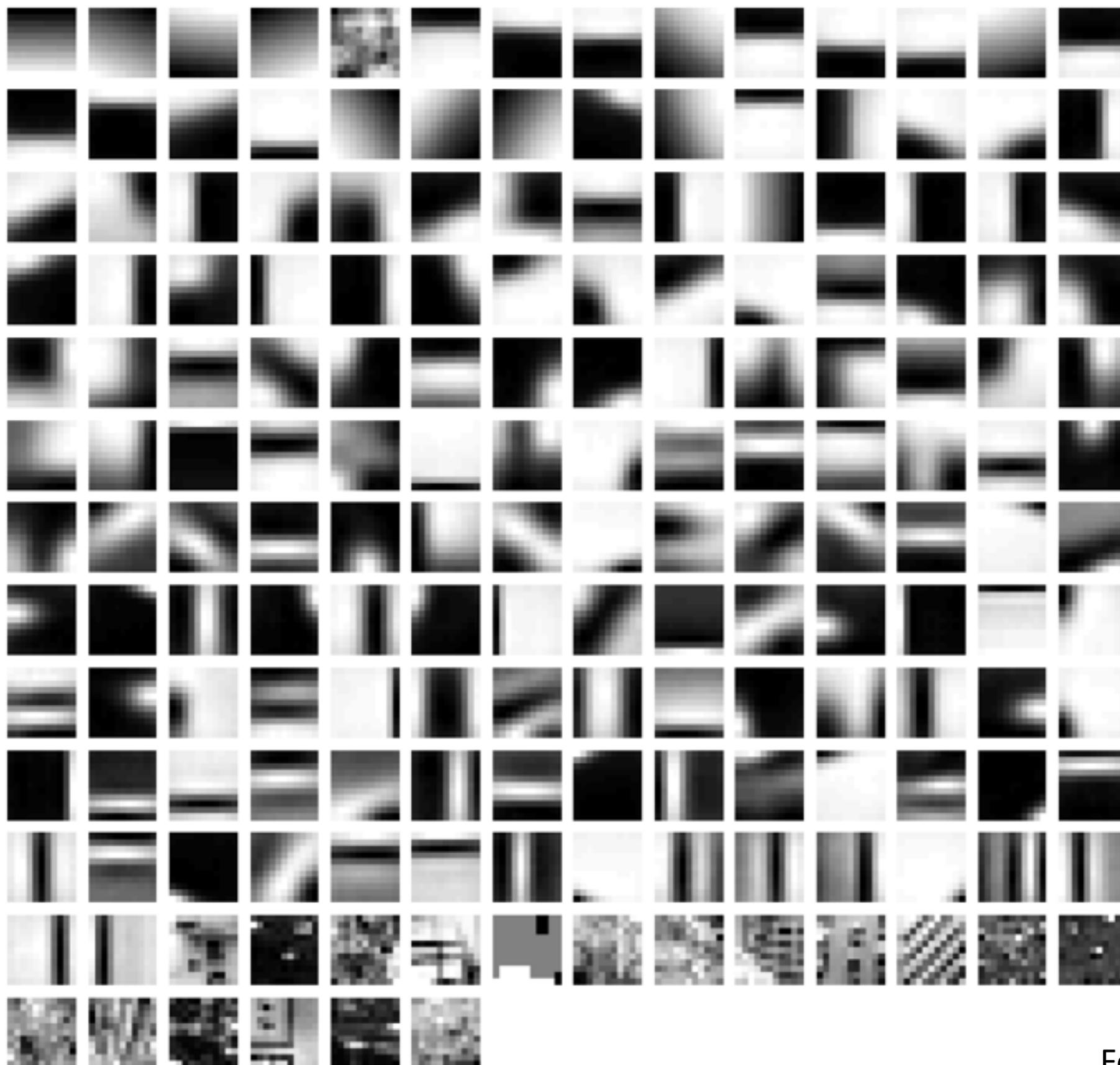
Repeat previous 2 steps until no change

*From what **data** should I learn the code book?*

- Codebook can be learned on separate training set
- Provided the training set is sufficiently representative, the codebook will be “universal”

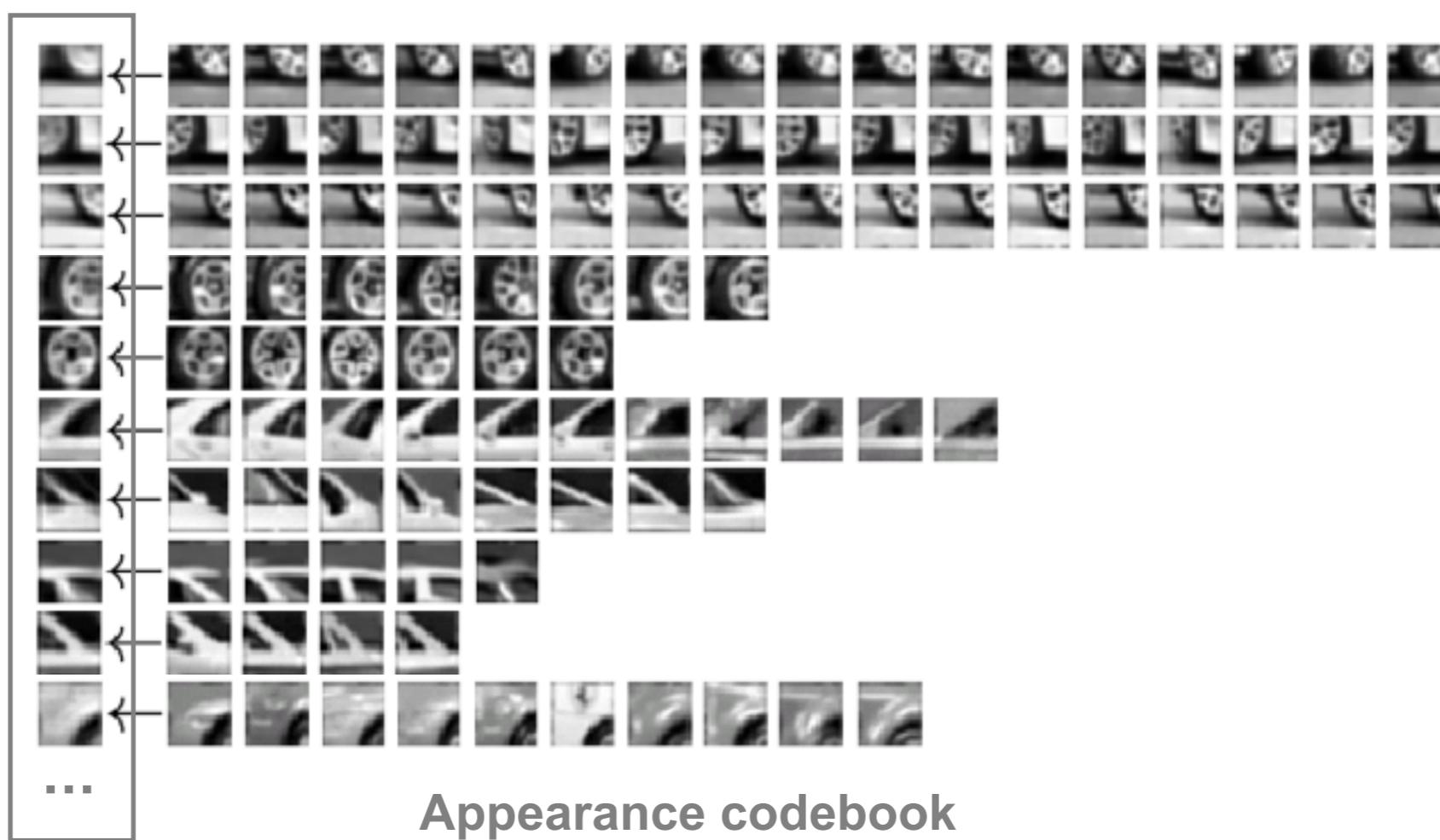
# Example visual vocabulary

---



# Example codebook

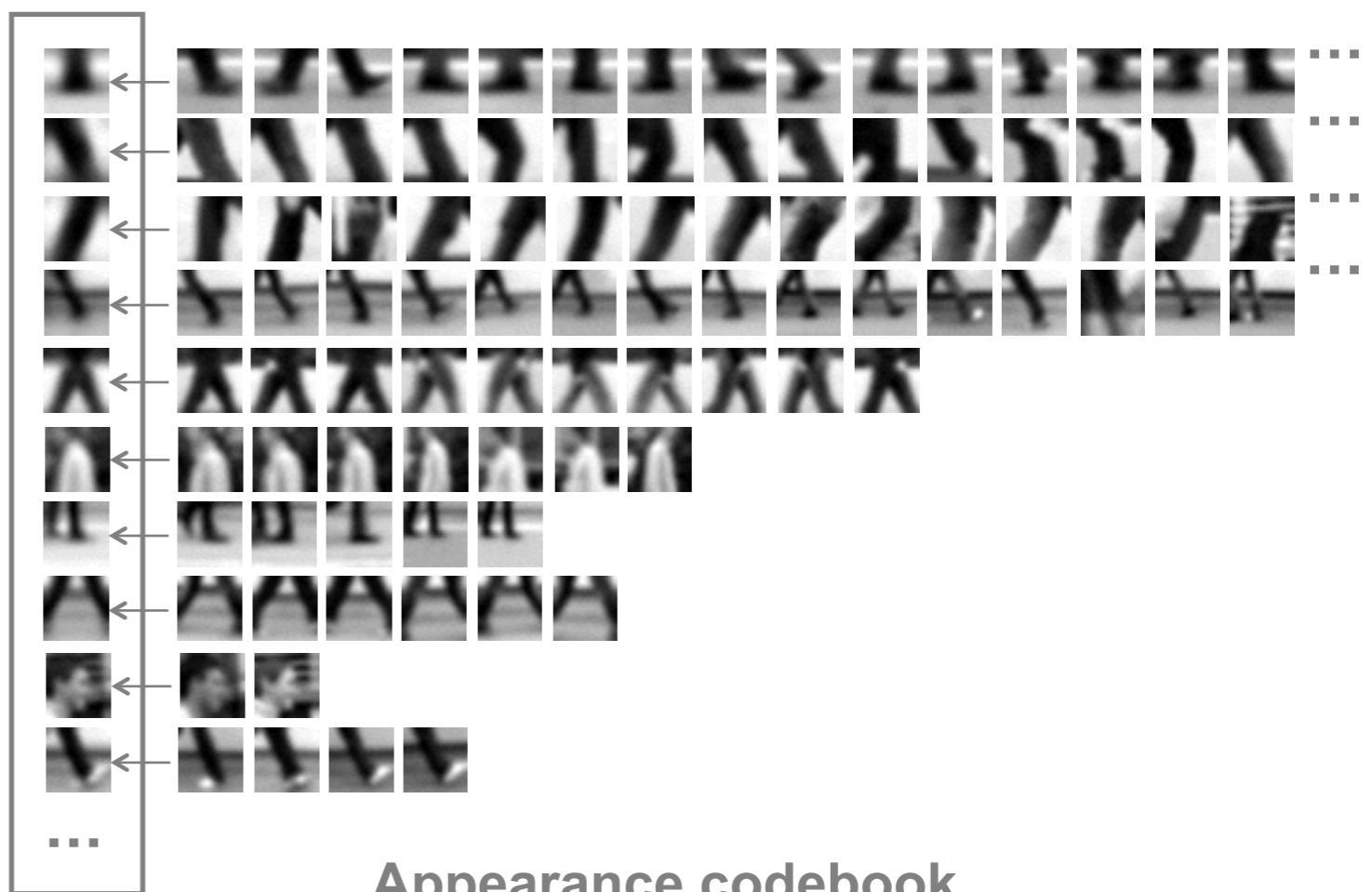
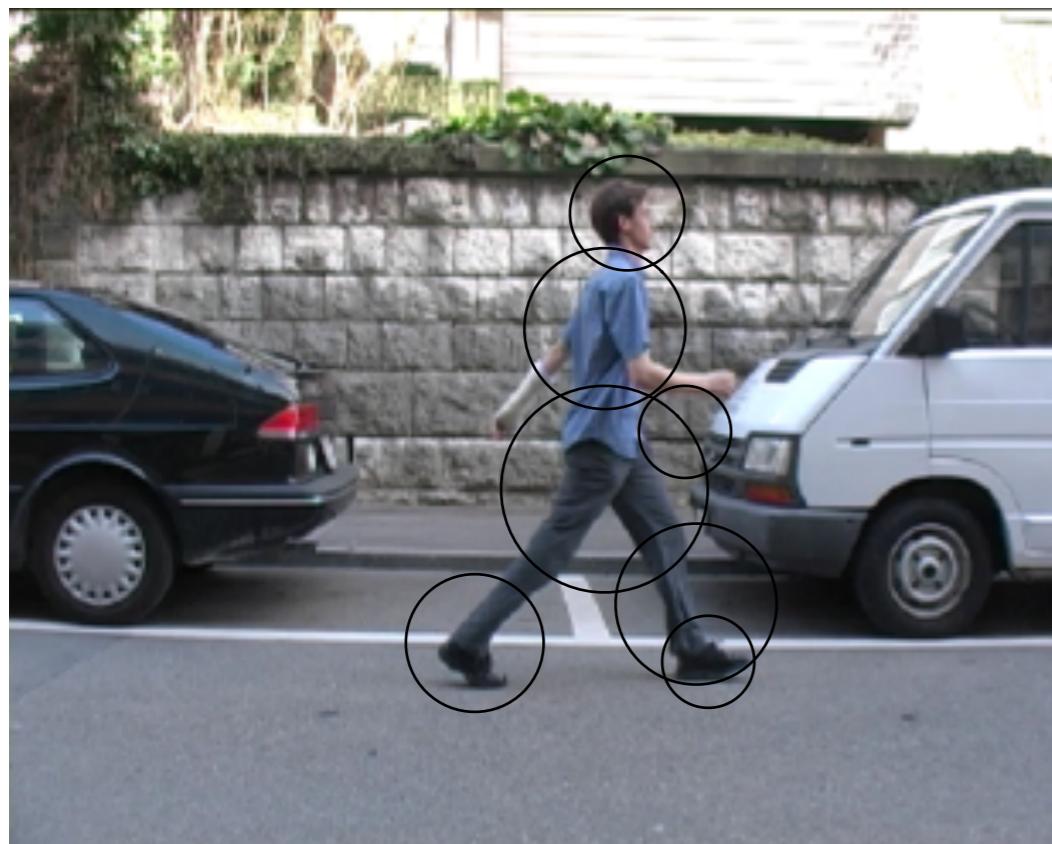
---



Appearance codebook

# Another codebook

---

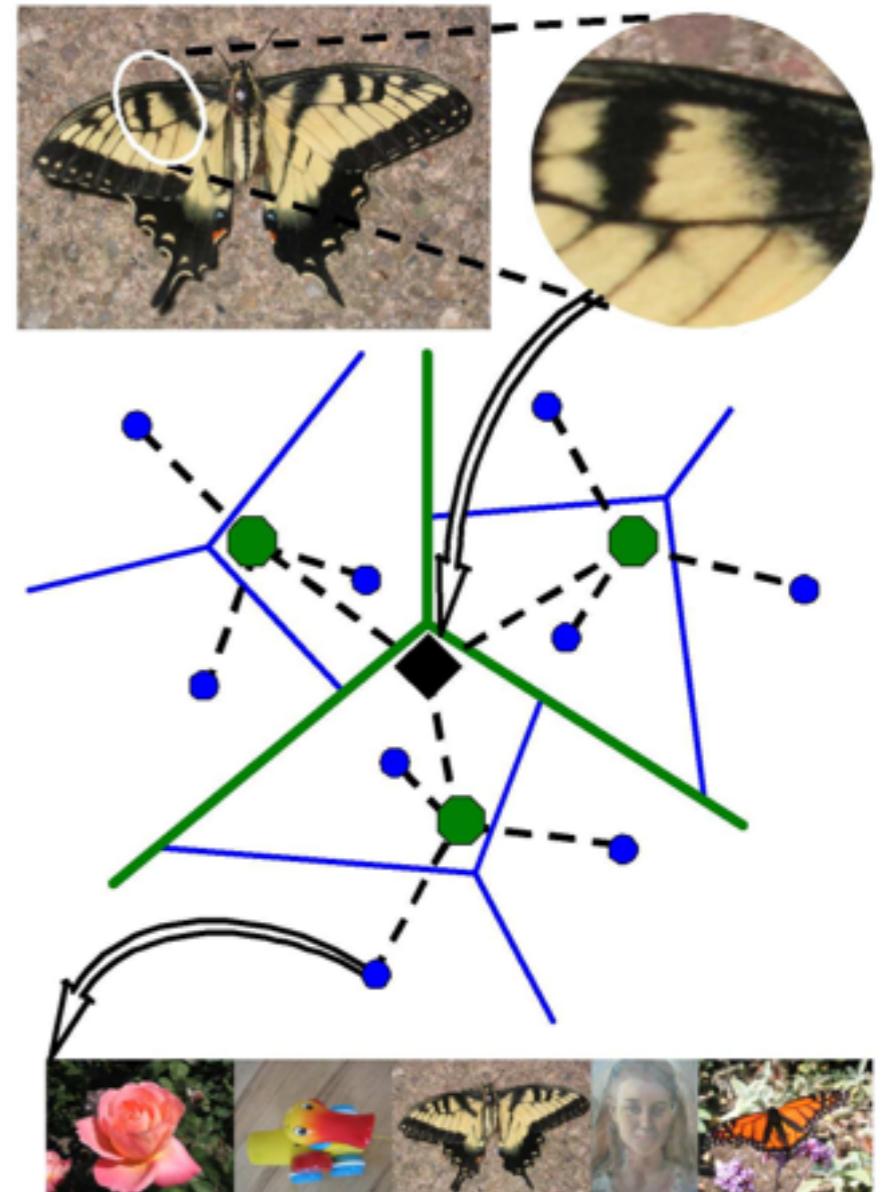


Appearance codebook

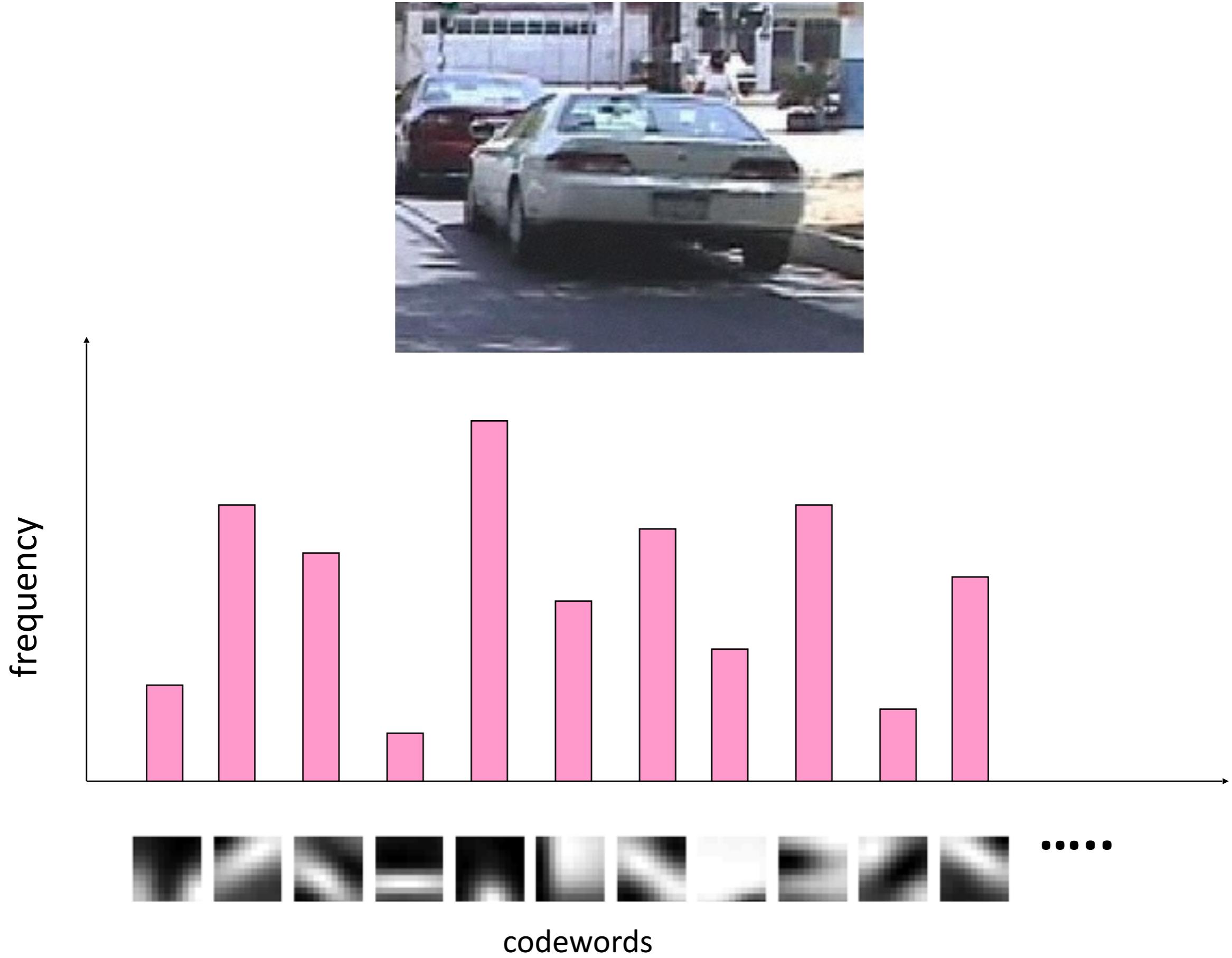
# Visual vocabularies: Issues

---

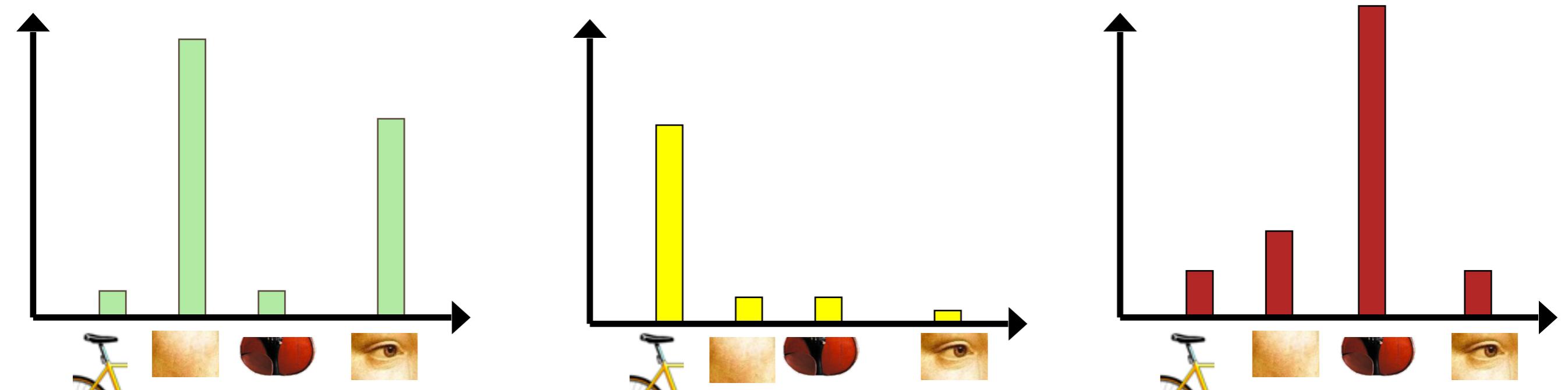
- How to choose vocabulary size?
  - Too small: visual words not representative of all patches
  - Too large: quantization artifacts, overfitting
- Computational efficiency
  - Vocabulary trees  
(Nister & Stewenius, 2006)



# Histogram



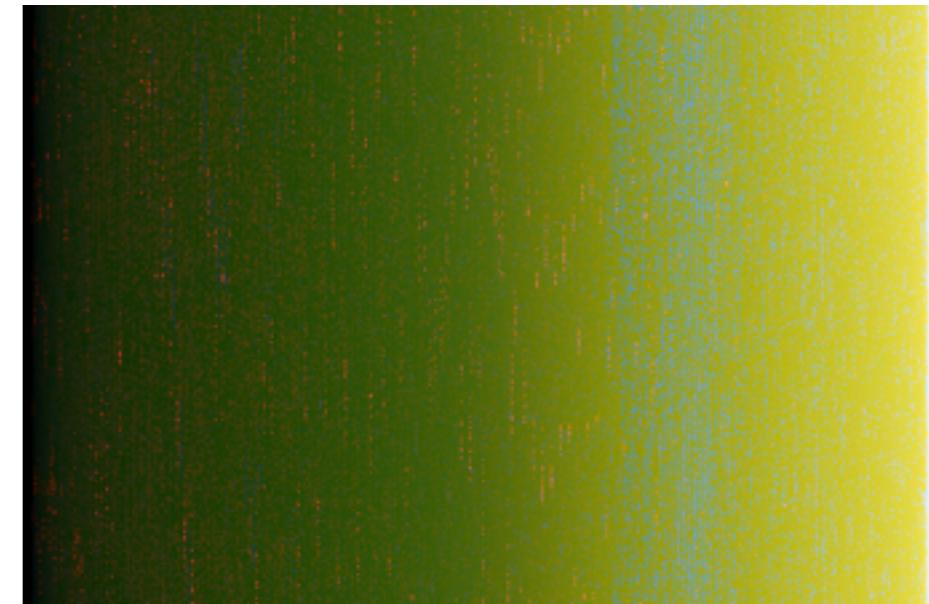
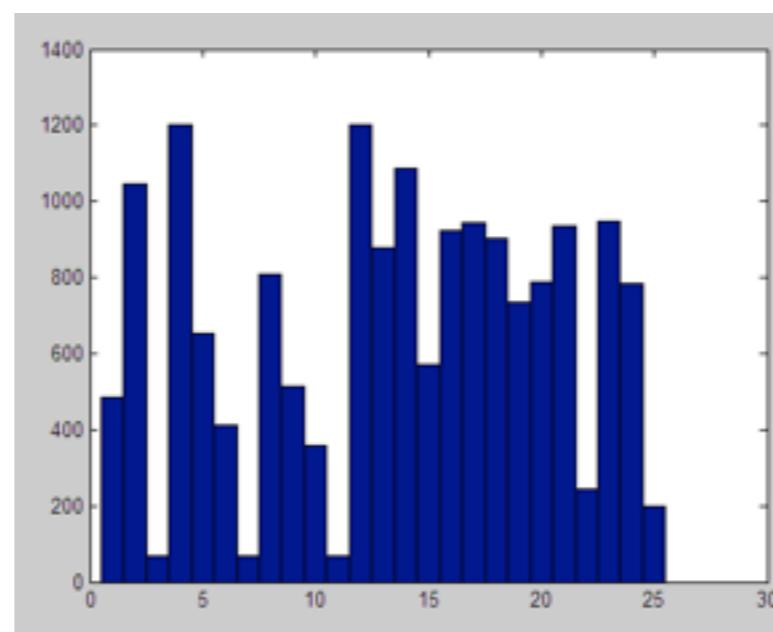
# Classification



Given the bag-of-features representations of images from different classes, learn a classifier using machine learning  
(more on this soon)

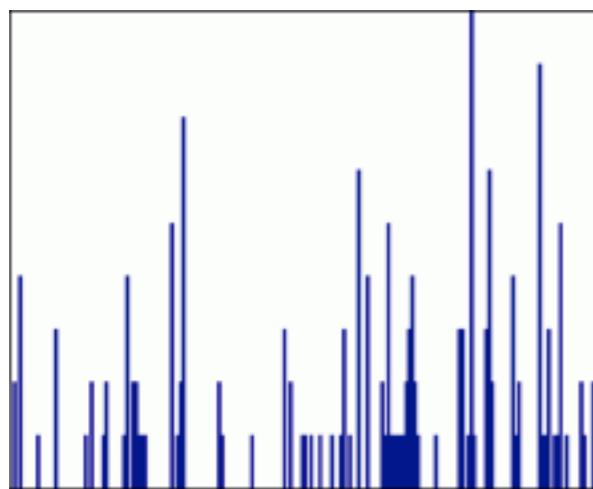
Extension to bag-of-  
words models

All of these images have the same color histogram!

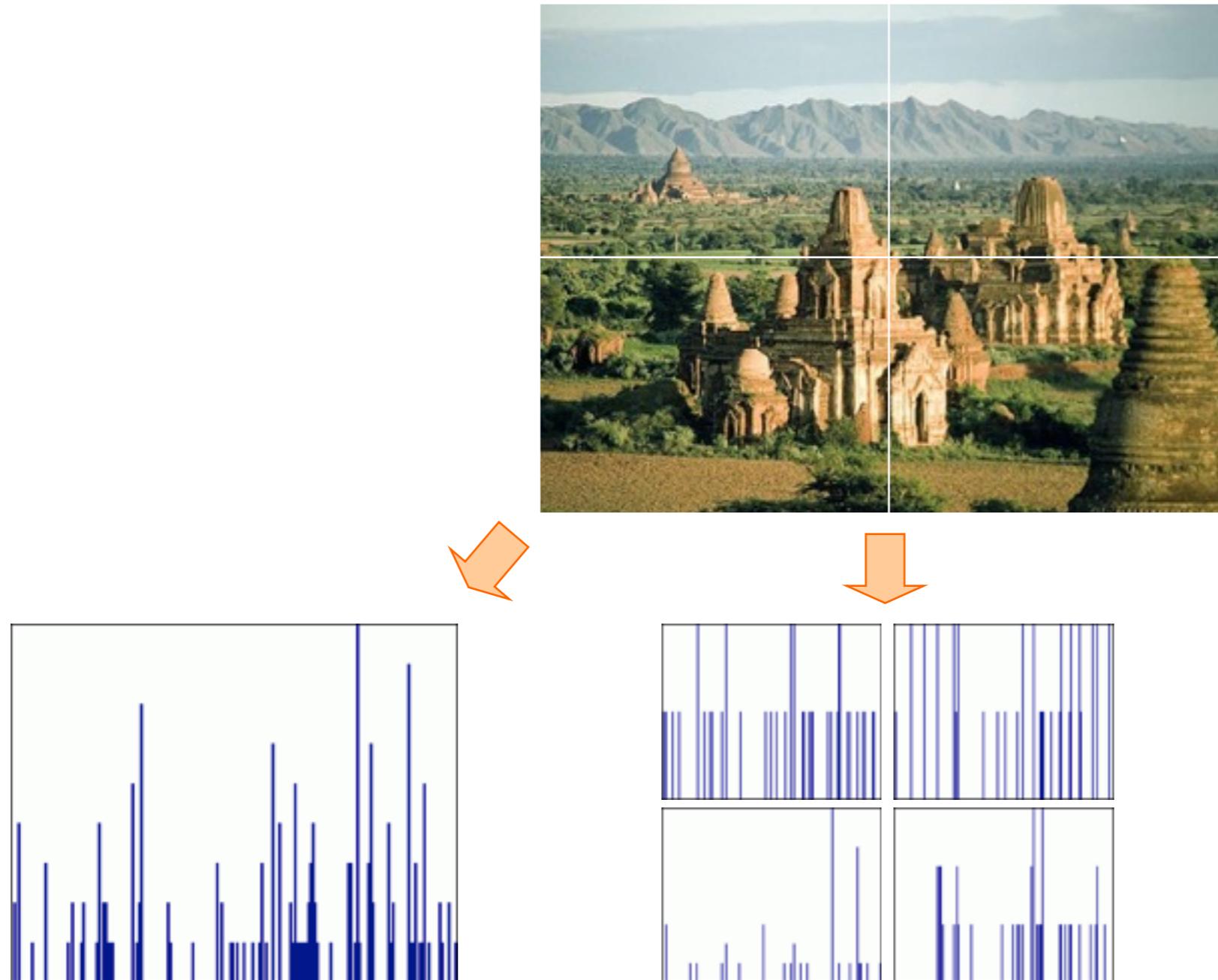


*How can we encode the spatial layout?*

# Spatial Pyramid representation



# Spatial Pyramid representation



# Spatial Pyramid representation

