CSE 152: Computer Vision

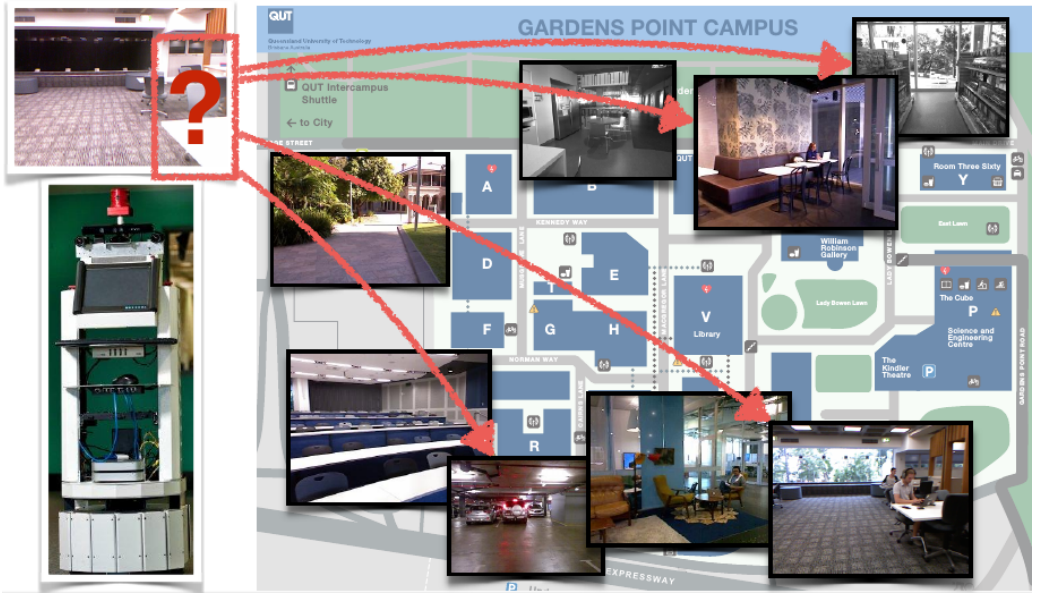
Hao Su

Lecture 2: Bag of Words

Credit: Manmohan Chandraker

Recognizing or retrieving specific objects

Example 1: Place recognition for self-driving or robot navigation



Recognizing or retrieving specific objects

Example 2: Visual search in movies

Visually defined query

"Find this clock"

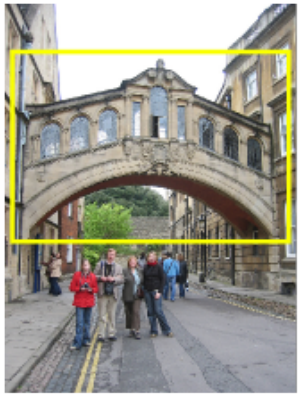
"Find this place"

"Groundhog Day" [Rammis, 1993]

Slide credit: J. Sivic

Recognizing or retrieving specific objects

Example 3: Search photos for particular places



Find these landmarks

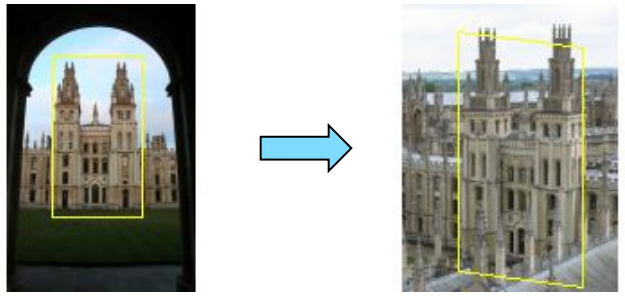
...in these images and 1M more

Slide credit: J. Sivic

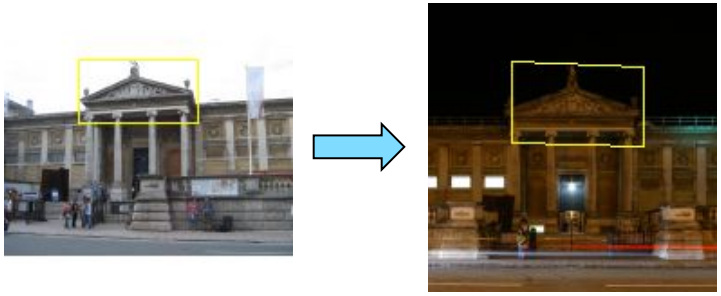
Why is it difficult?

Want to find the object despite possibly large changes in scale, viewpoint, lighting and partial occlusion

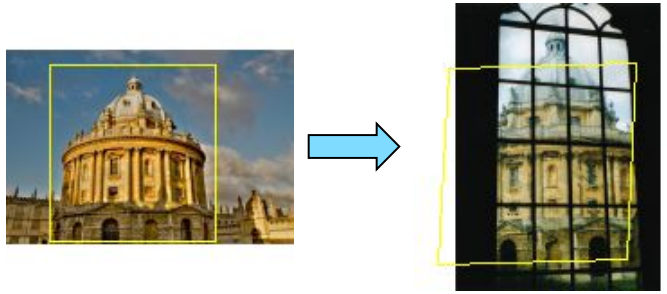
Scale



Viewpoint



Lighting



Occlusion

Slide credit: J. Sivic

Object → Bag of ‘words’

Origin: Bag-of-words models

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

US Presidential Speeches Tag Cloud
http://chir.ag/phernalia/preztags/

Origin: Bag-of-words models

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

US Presidential Speeches Tag Cloud
http://chir.ag/phernalia/preztags/

Origin: Bag-of-words models

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

US Presidential Speeches Tag Cloud
http://chir.ag/phernalia/preztags/

Origin: Bag-of-words models

face, flowers, building

- Works quite well for image-level classification and for recognizing object *instances*

Csurka et al. (2004), Willamowski et al. (2005), Grauman & Darrell (2005), Sivic et al. (2003, 2005)

Origin: Bag-of-words models

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	**98.8**	97.1	90.2
cars (rear)	98.3	**98.6**	90.3
cars (side)	**95.0**	87.3	88.5
faces	**100**	99.3	96.4
motorbikes	**98.5**	98.0	92.5
spotted cats	**97.0**	—	90.0

Recall: matching local features

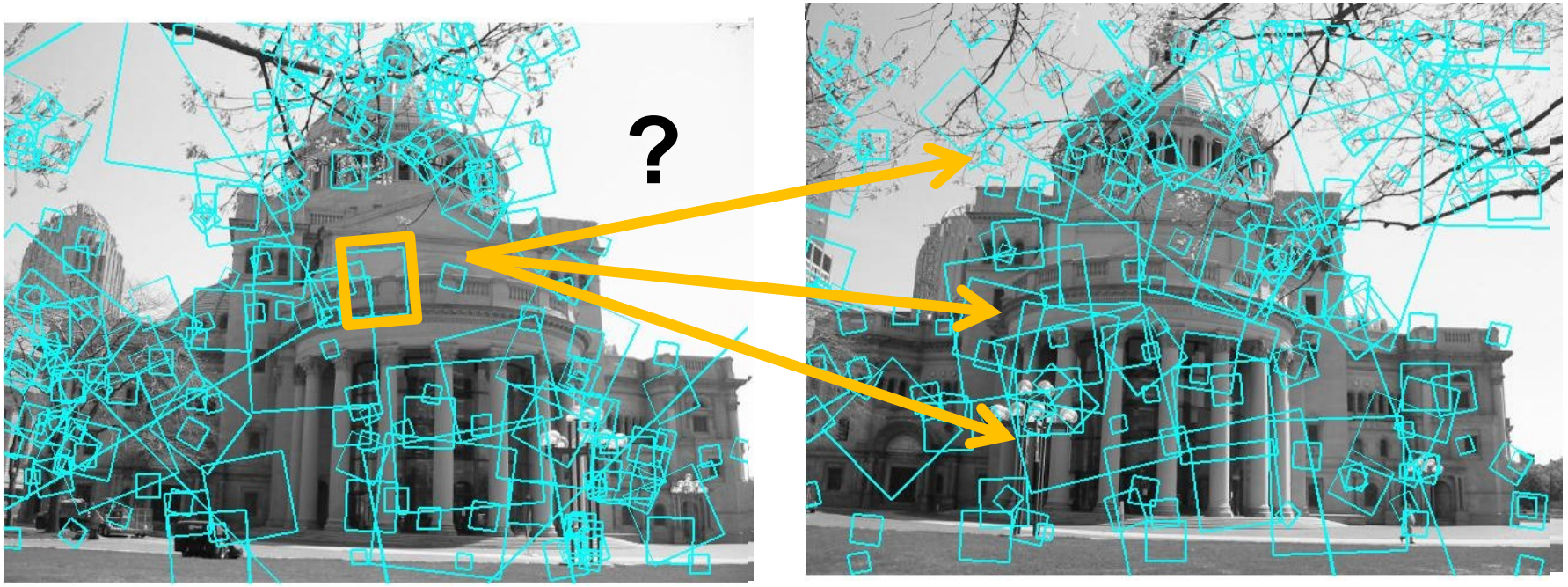


Image 1

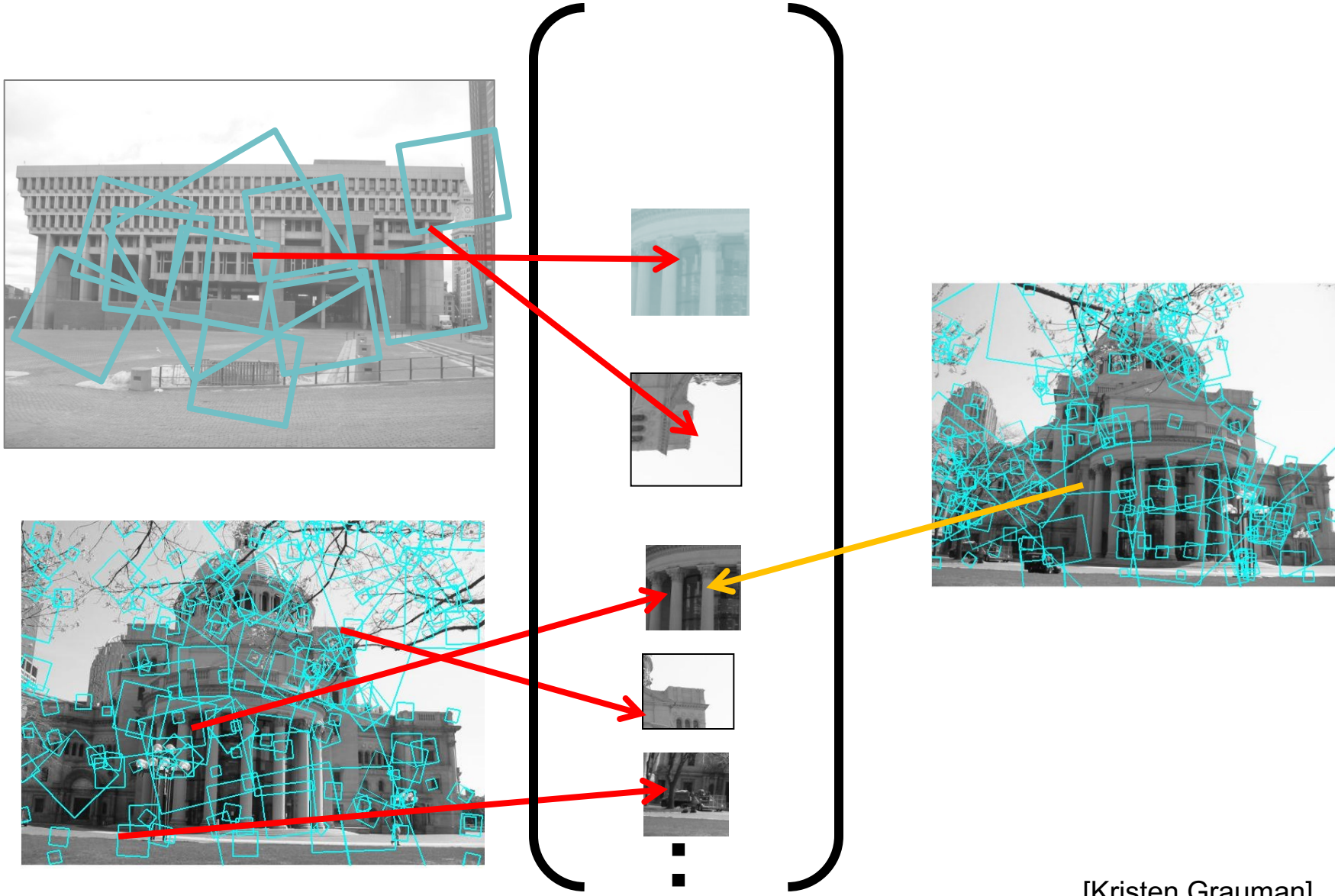
Image 2

To generate **candidate matches**, find patches that have the most similar appearance (e.g., lowest SSD)

Simplest approach: compare them all, take the closest (or closest k, or within a thresholded distance)

[Kristen Grauman]

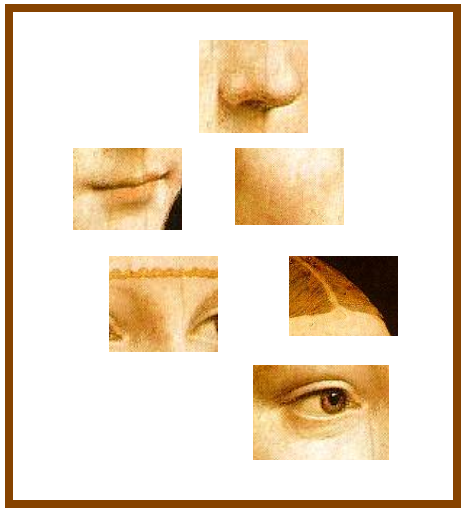
Indexing local features

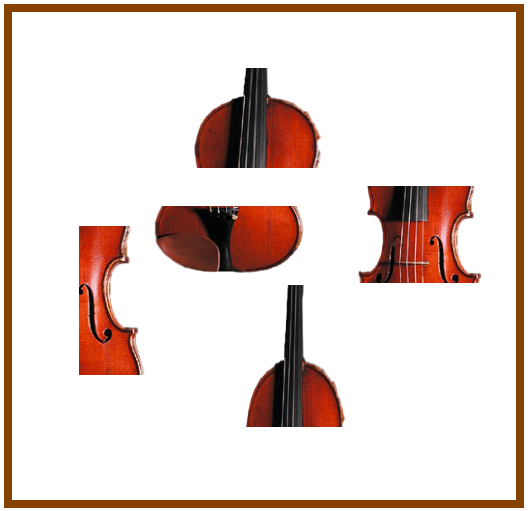


[Kristen Grauman]

Bag of features: outline

1. Extract features





Bag of features: outline

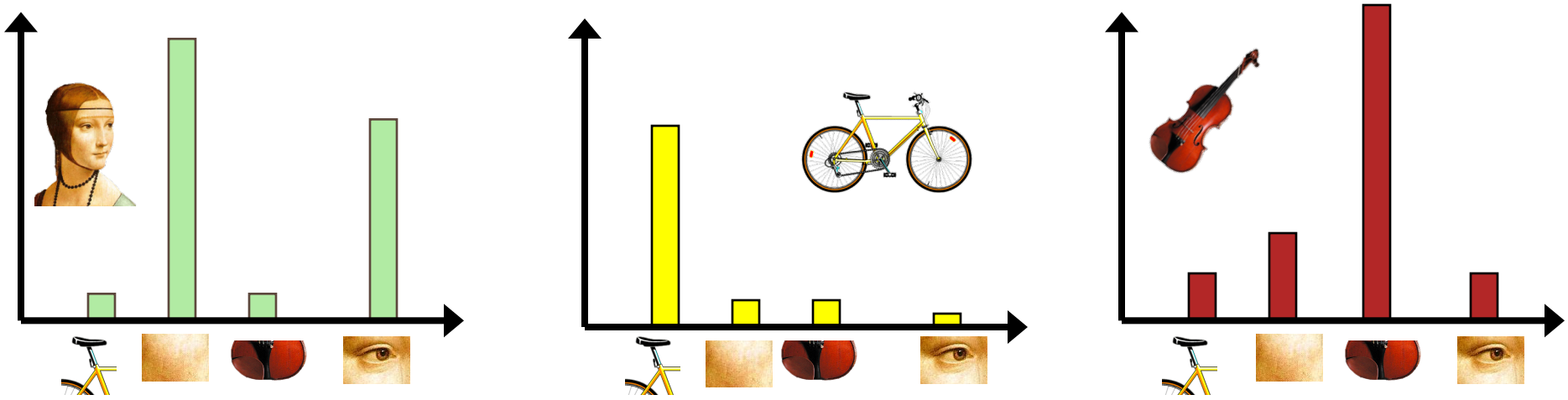
1. Extract features
2. Learn “visual vocabulary”

Bag of features: outline

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

Bag of features: outline

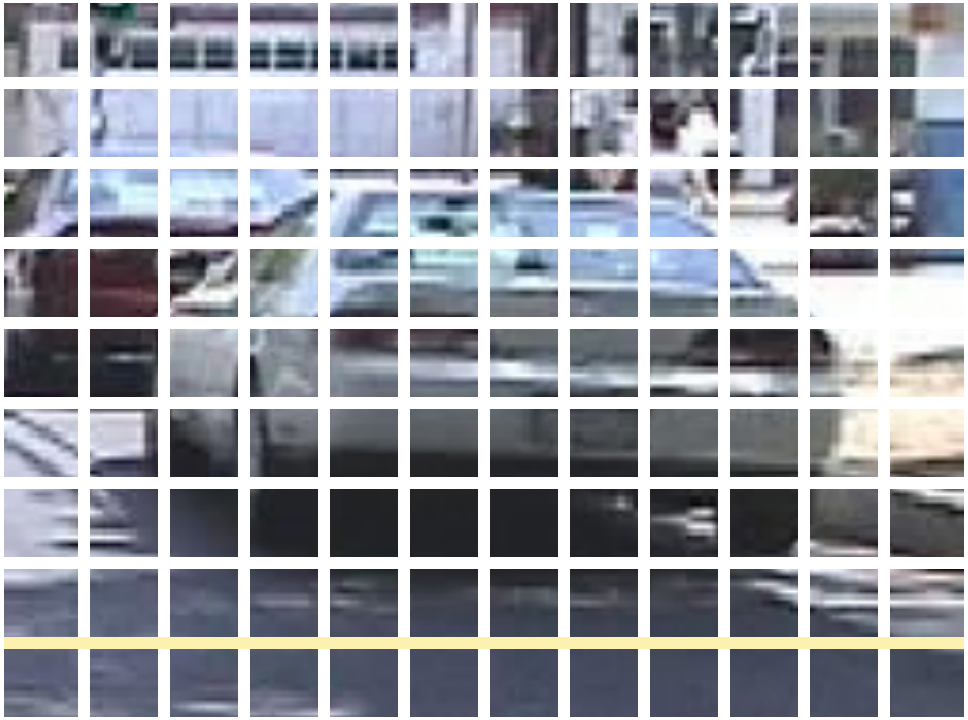
1. Extract features
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary
4. Represent images by frequencies of “visual words”



1. Feature extraction

Regular grid

- Vogel & Schiele, 2003
- Fei-Fei & Perona, 2005



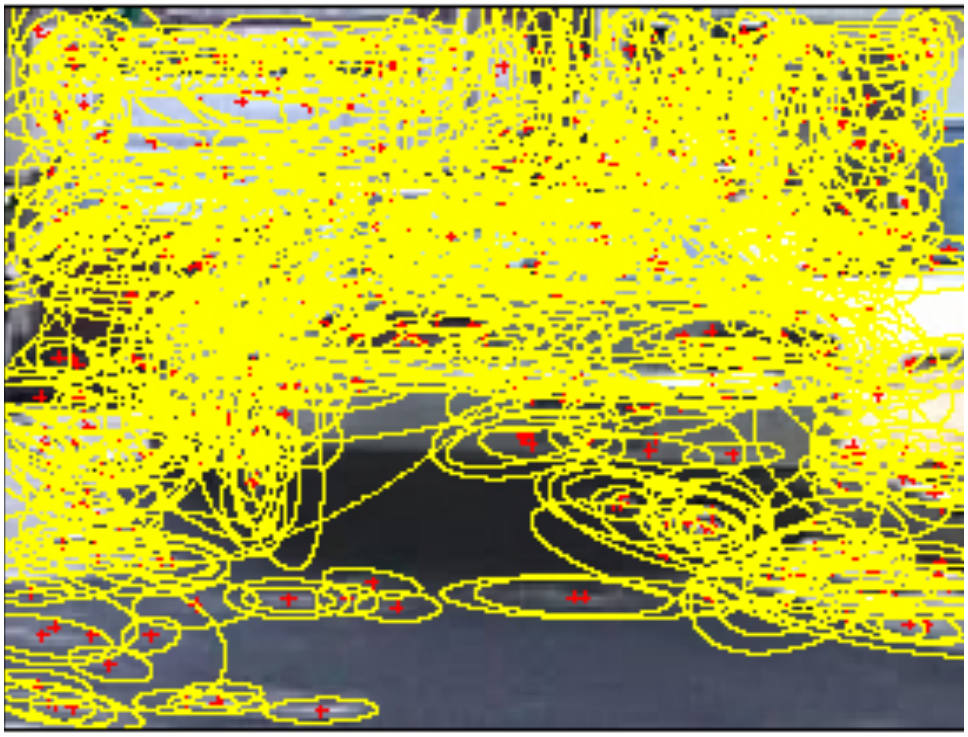
1. Feature extraction

Regular grid

- Vogel & Schiele, 2003
- Fei-Fei & Perona, 2005

Interest point detector

- Csurka et al. 2004
- Fei-Fei & Perona, 2005
- Sivic et al. 2005



1. Feature extraction

Regular grid

- Vogel & Schiele, 2003
- Fei-Fei & Perona, 2005

Interest point detector

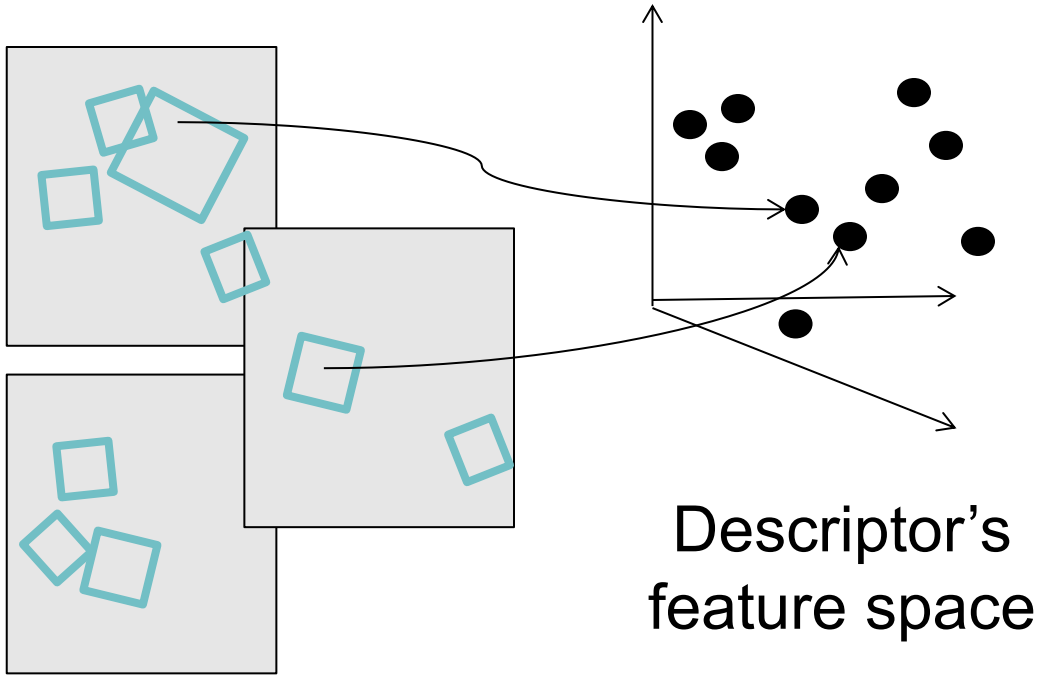
- Csurka et al. 2004
- Fei-Fei & Perona, 2005
- Sivic et al. 2005

Other methods

- Random sampling (Vidal-Naquet & Ullman, 2002)
- Segmentation-based patches (Barnard et al. 2003)

Indexing local features

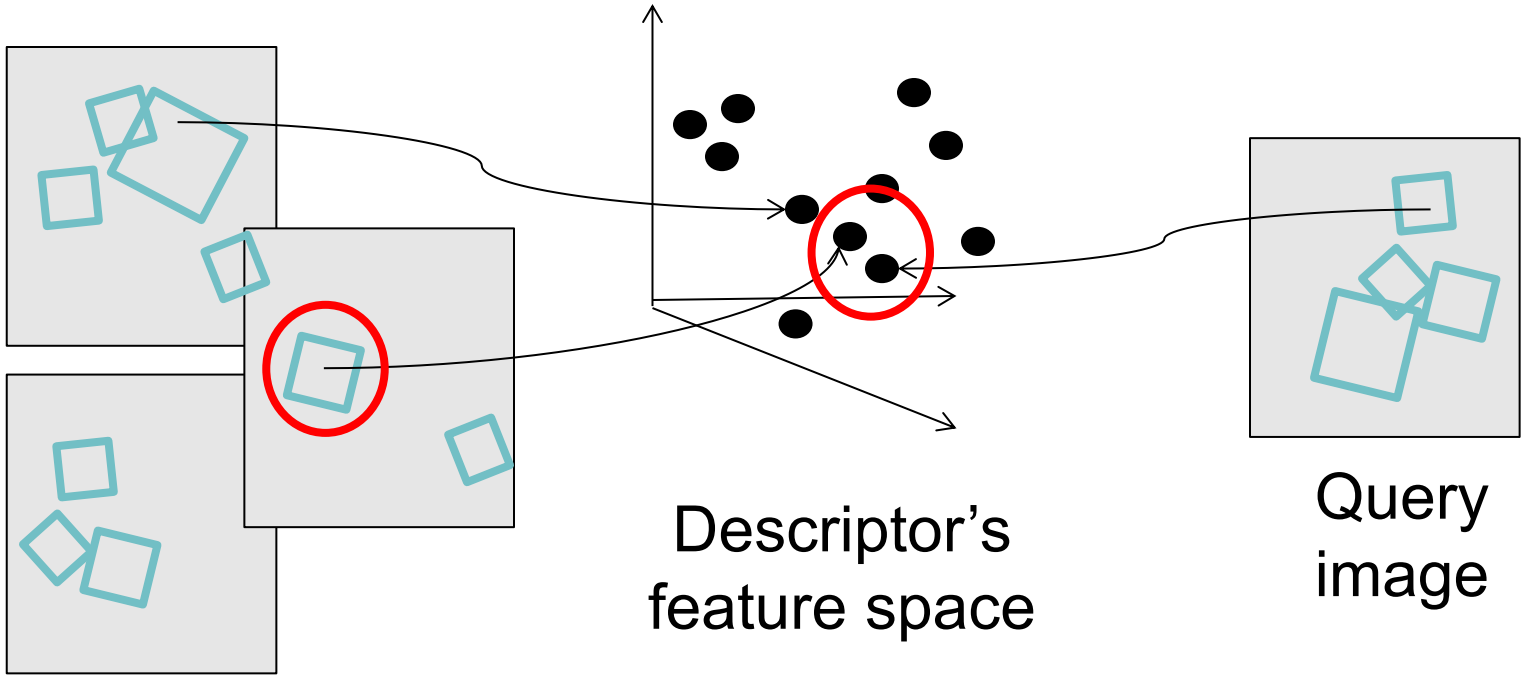
- Each patch or region has a descriptor, which is a point in some high-dimensional feature space (for example, SIFT)



[Kristen Grauman]

Indexing local features

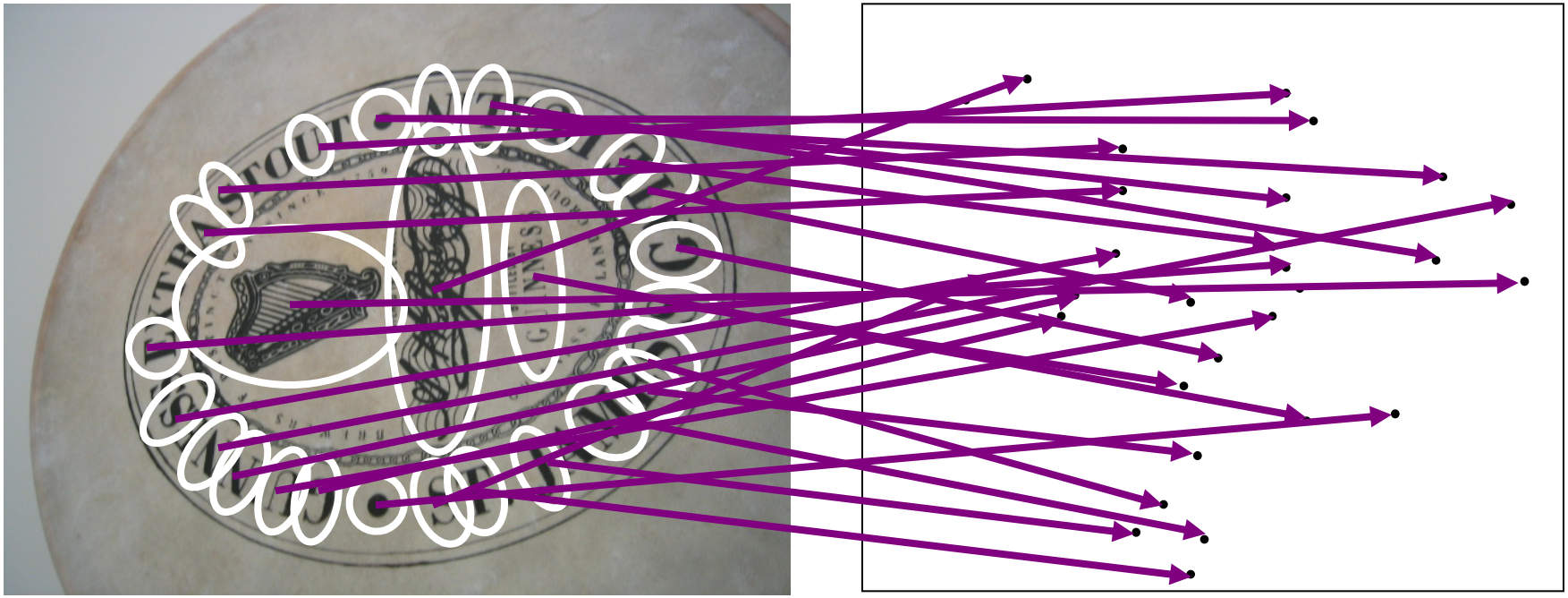
- When we see close points in feature space, we have similar descriptors, which indicates similar local content.



[Kristen Grauman]

Visual words: main idea

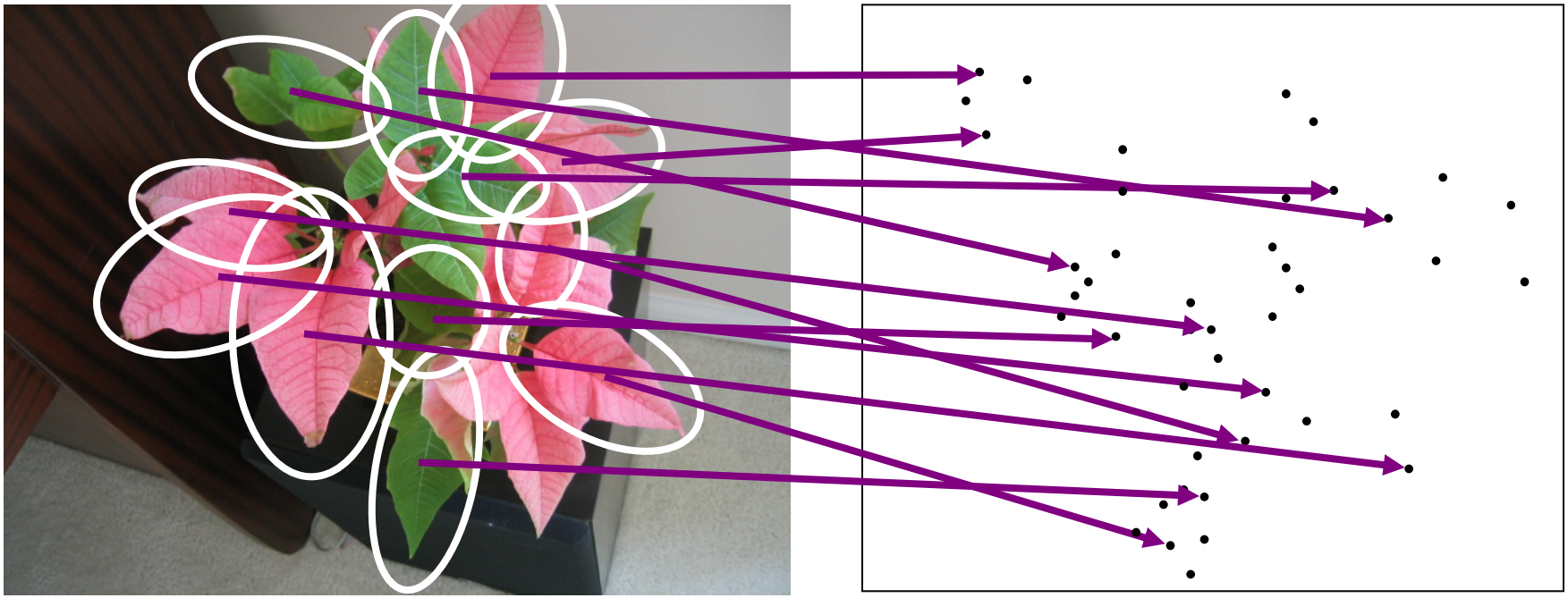
- Extract some local features from a number of images …



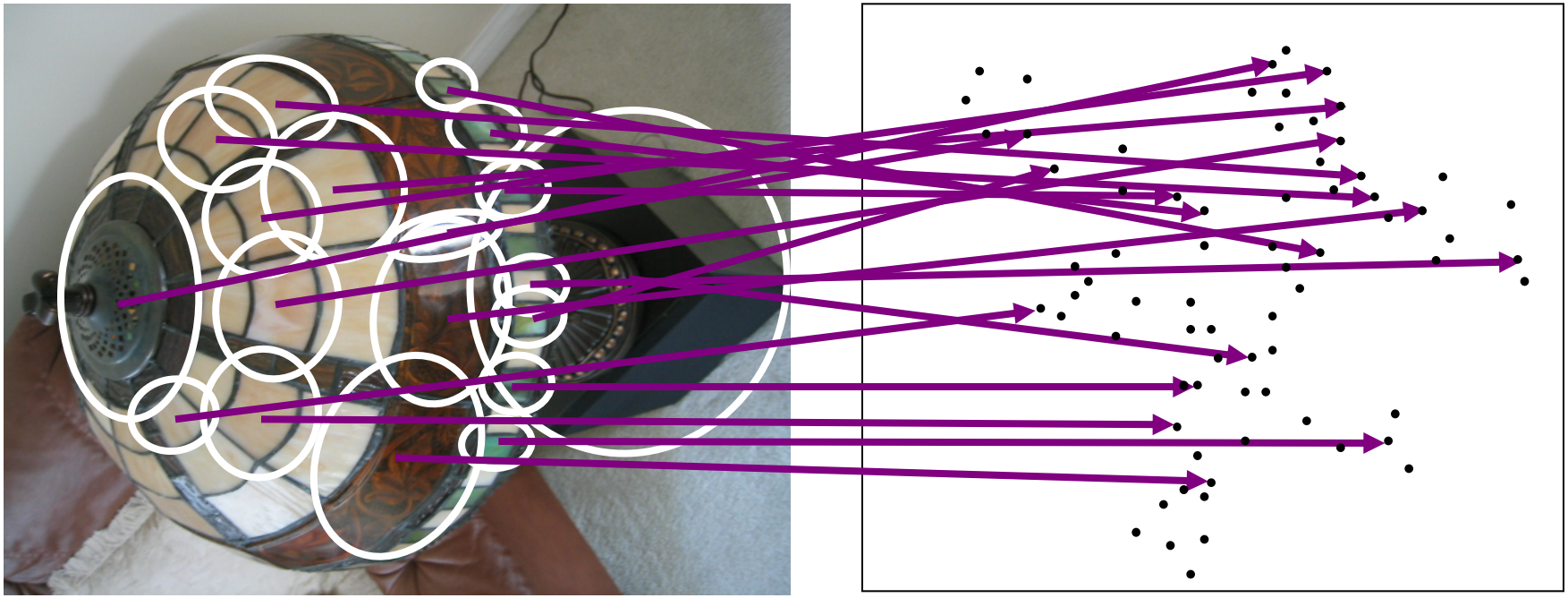
Example: For SIFT descriptor space, each point is 128-dimensional

D. Nister, CVPR 2006

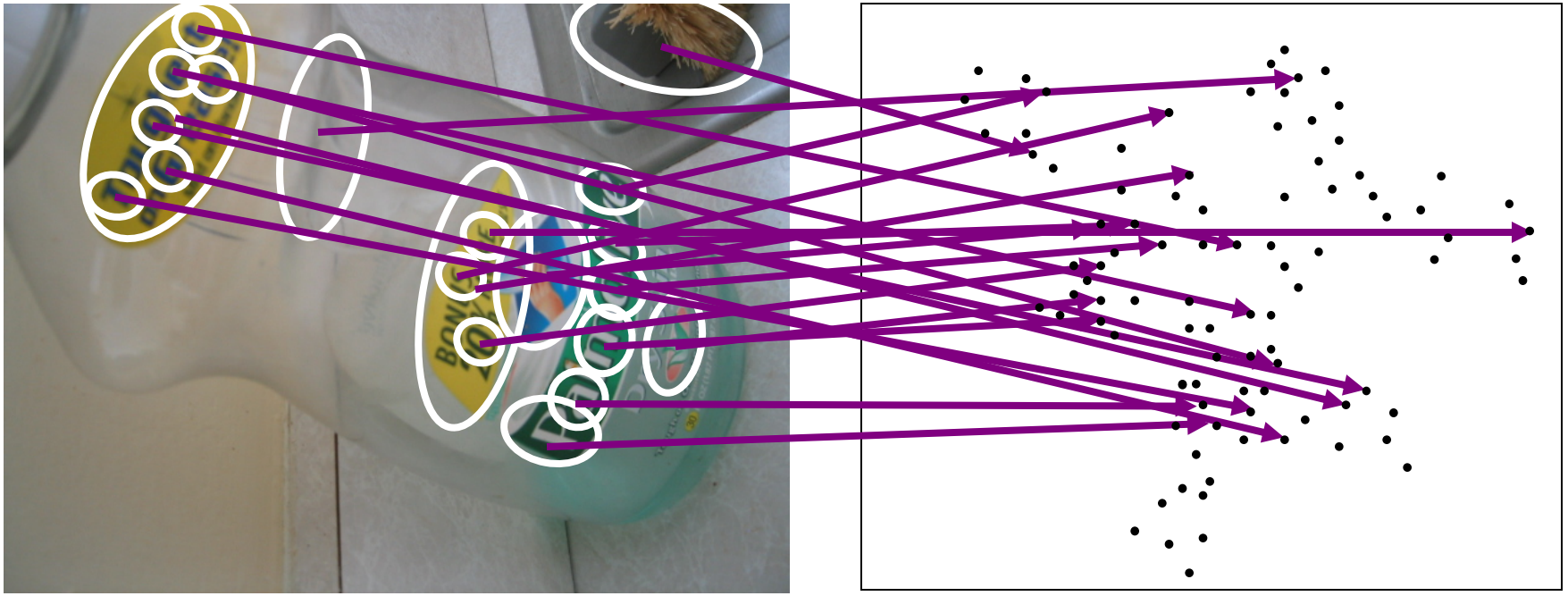
Visual words: main idea

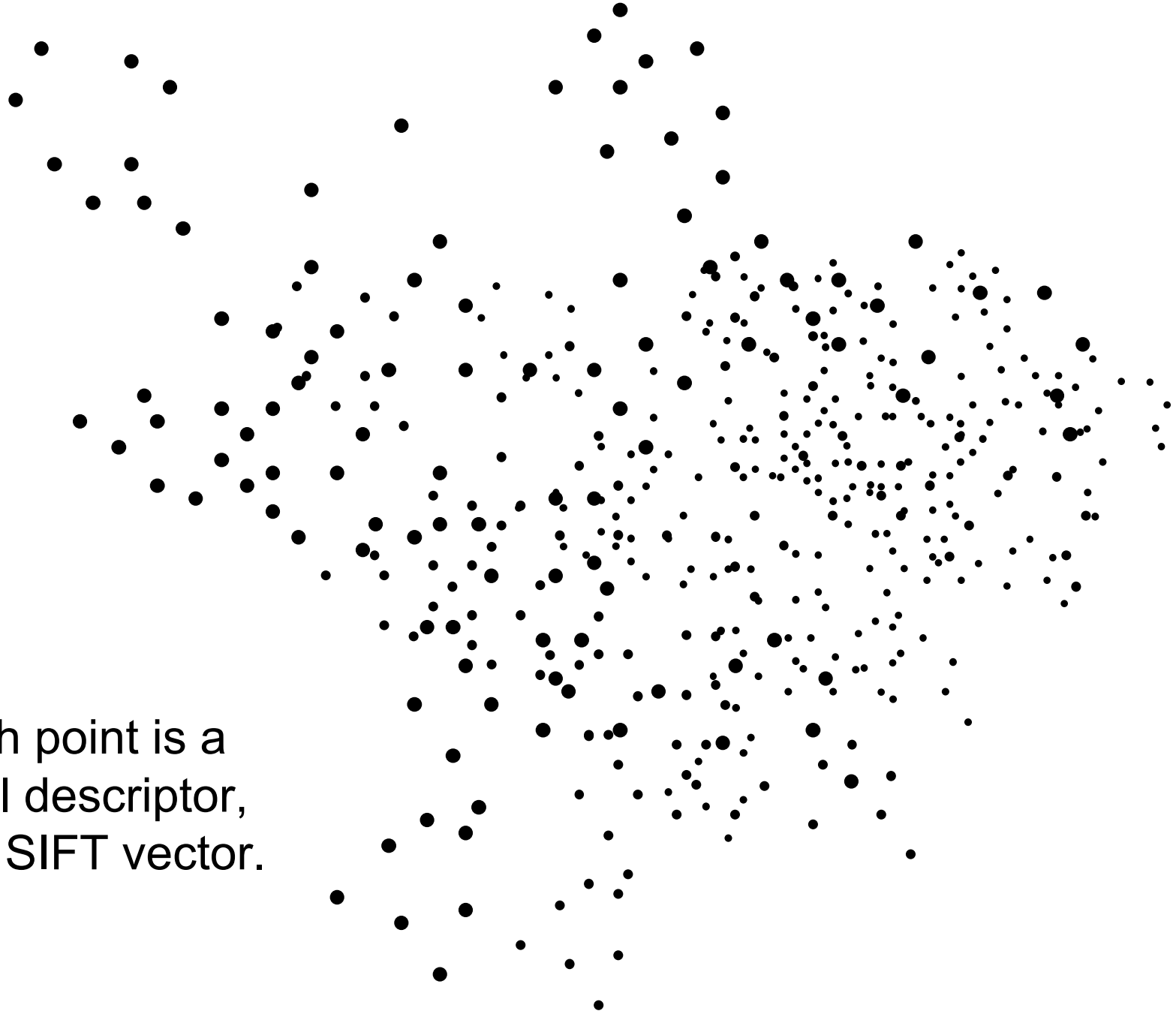


Visual words: main idea



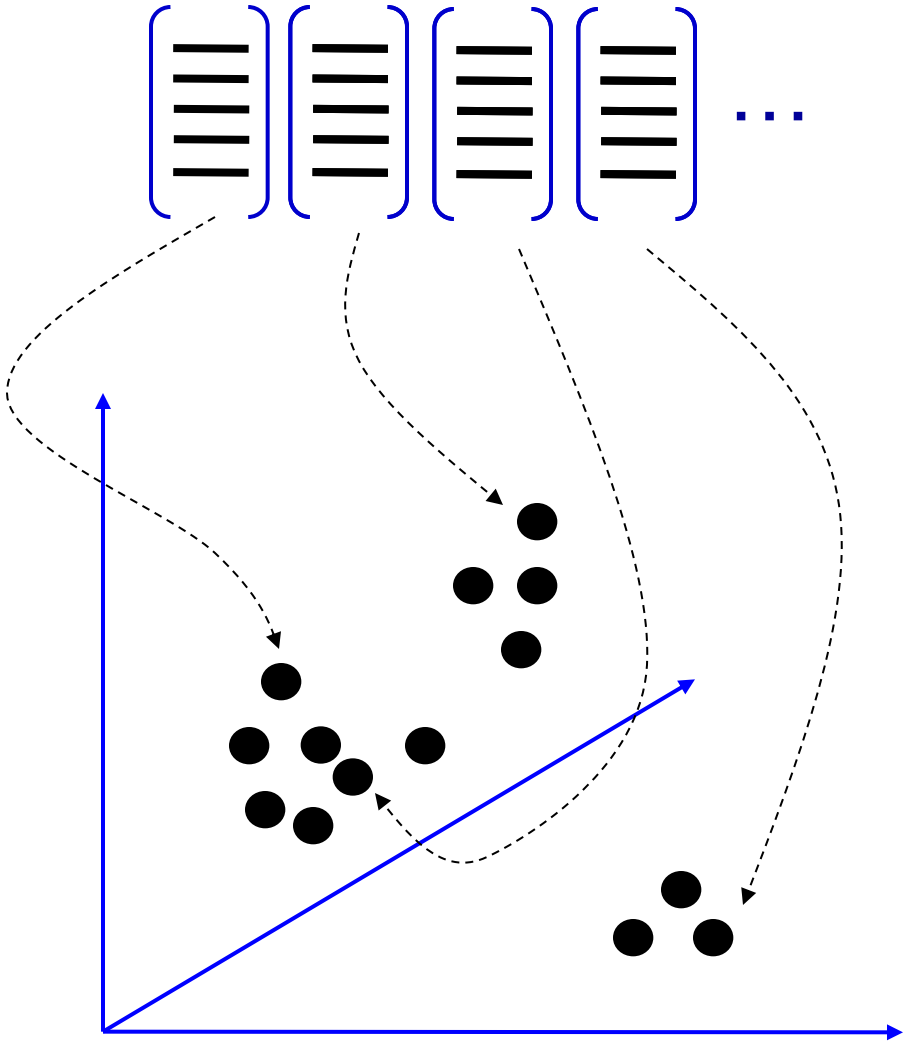
Visual words: main idea



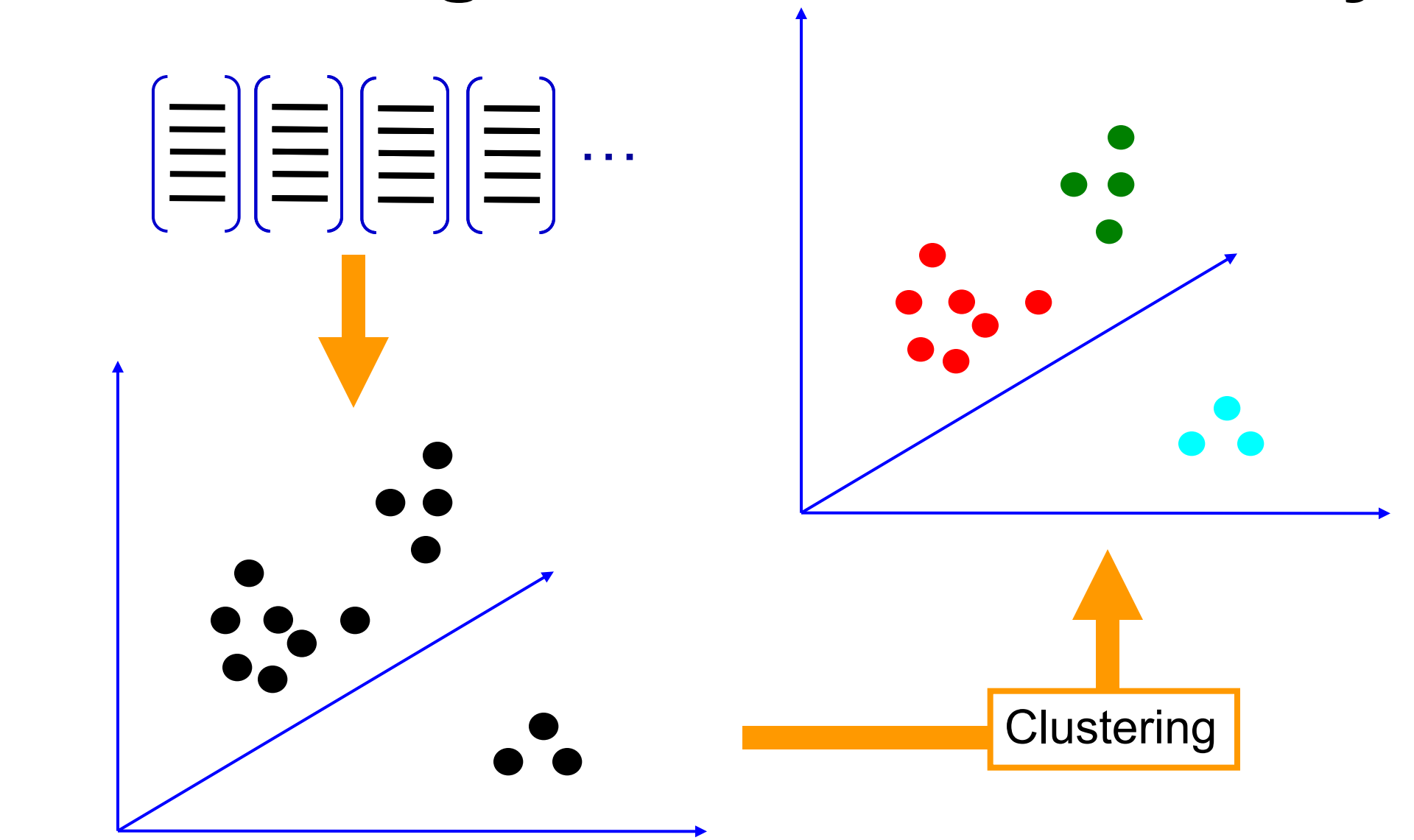


Each point is a
local descriptor,
e.g. SIFT vector.

2. Learning the visual vocabulary

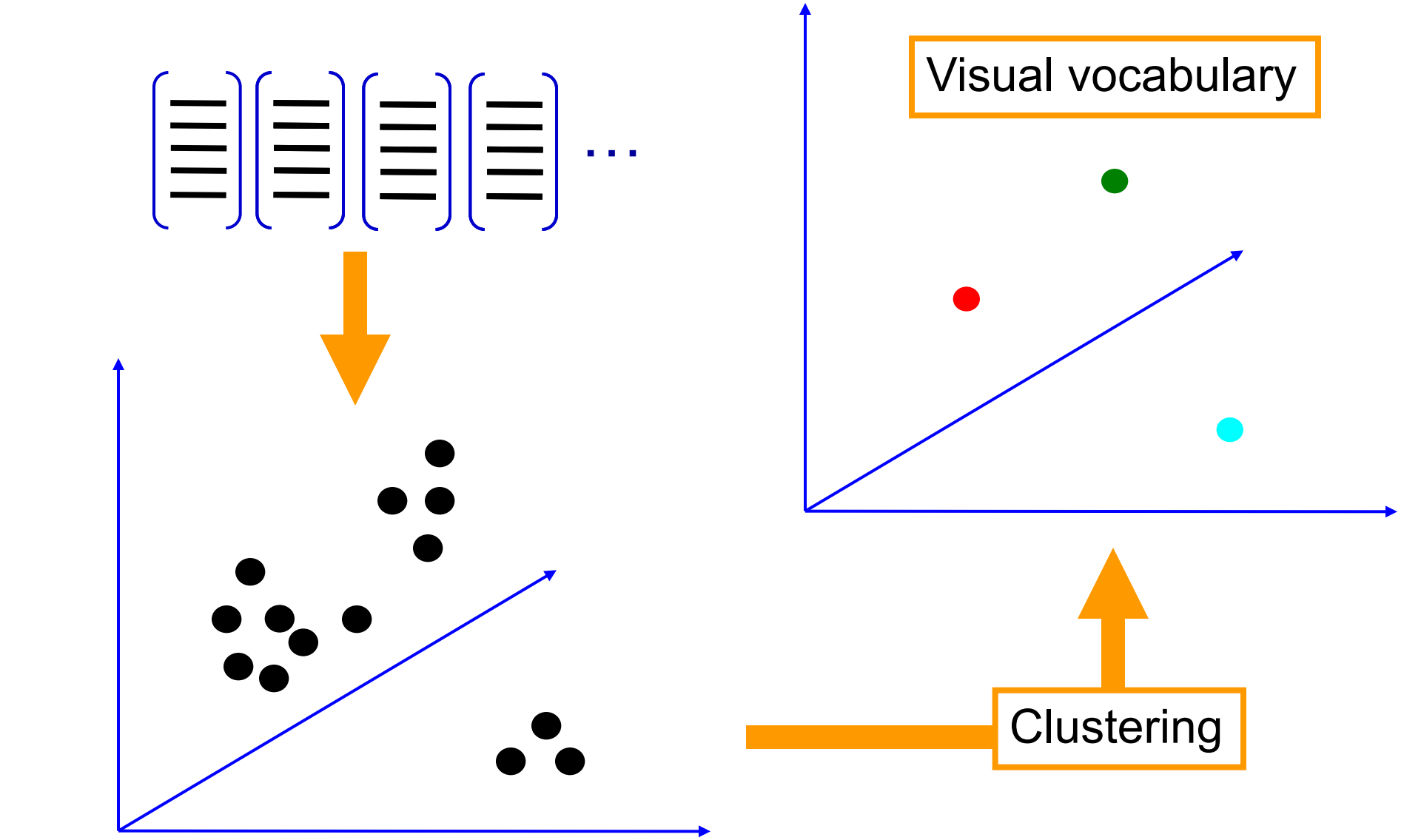


2. Learning the visual vocabulary



Slide credit: Josef Sivic

2. Learning the visual vocabulary



Slide credit: Josef Sivic

K-means clustering

- Want to minimize sum of squared Euclidean distances between points x_i and their nearest cluster centers $\bar{c}_k$

$$D(X, C) = \sum_{\text{cluster } C_k} \sum_{\text{point } i \in C_k} (x_i - \bar{c}_k)^2$$

Algorithm:

- Randomly initialize K cluster centers
- Iterate until convergence:
  - Assign each data point to the nearest center
  - Recompute each cluster center as the mean of all points assigned to it

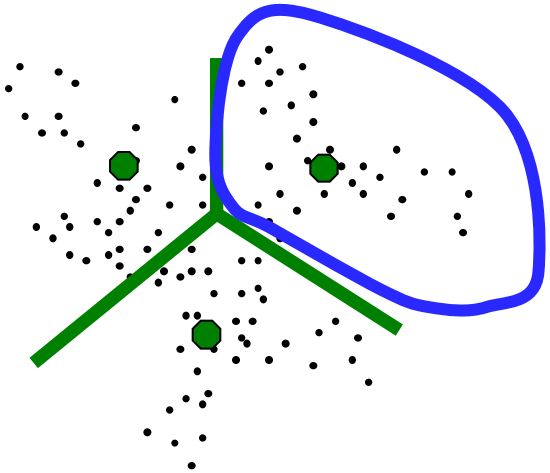
From clustering to vector quantization

- 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

Visual words

- Example: each group of patches belongs to the same visual word



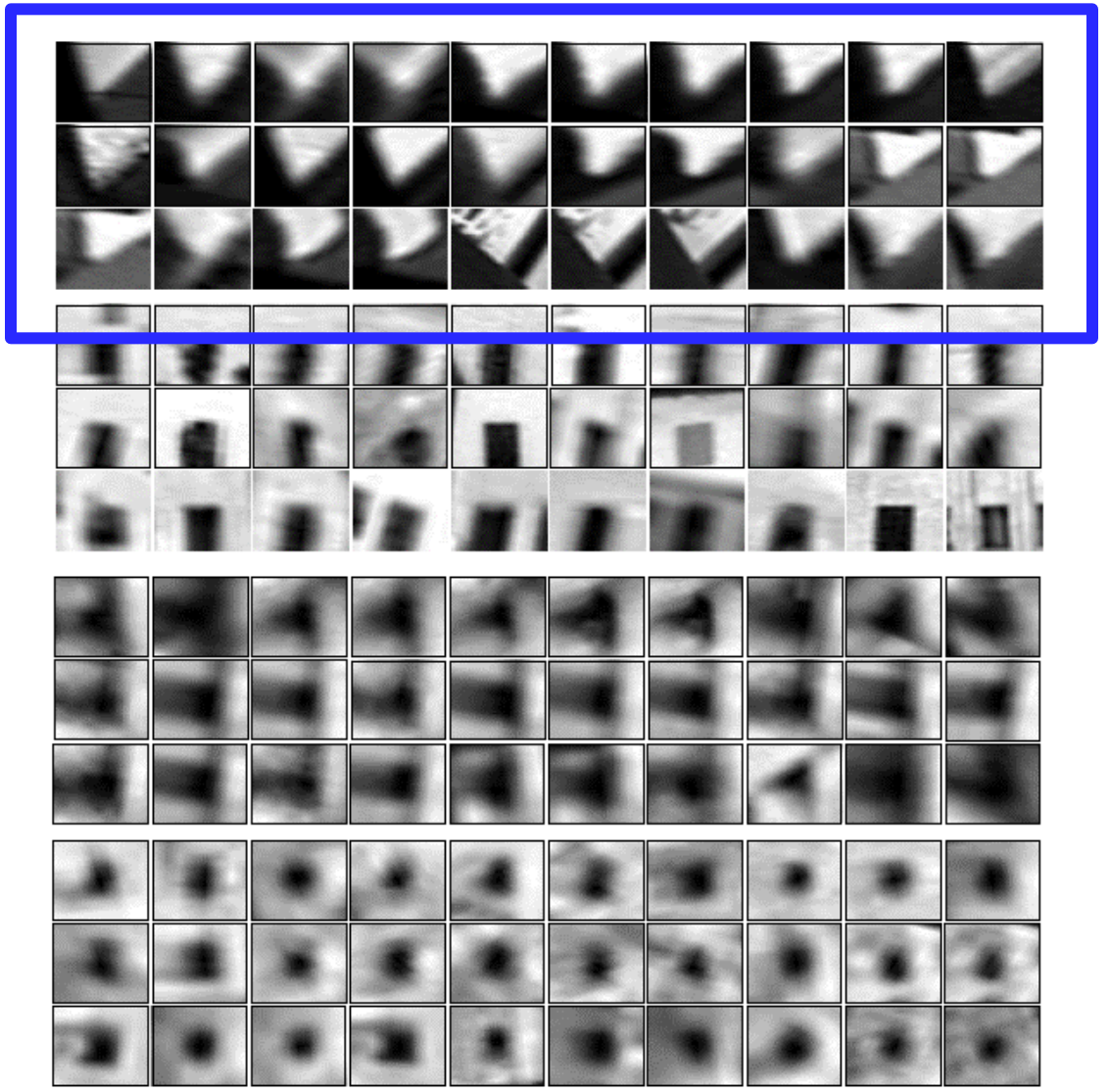
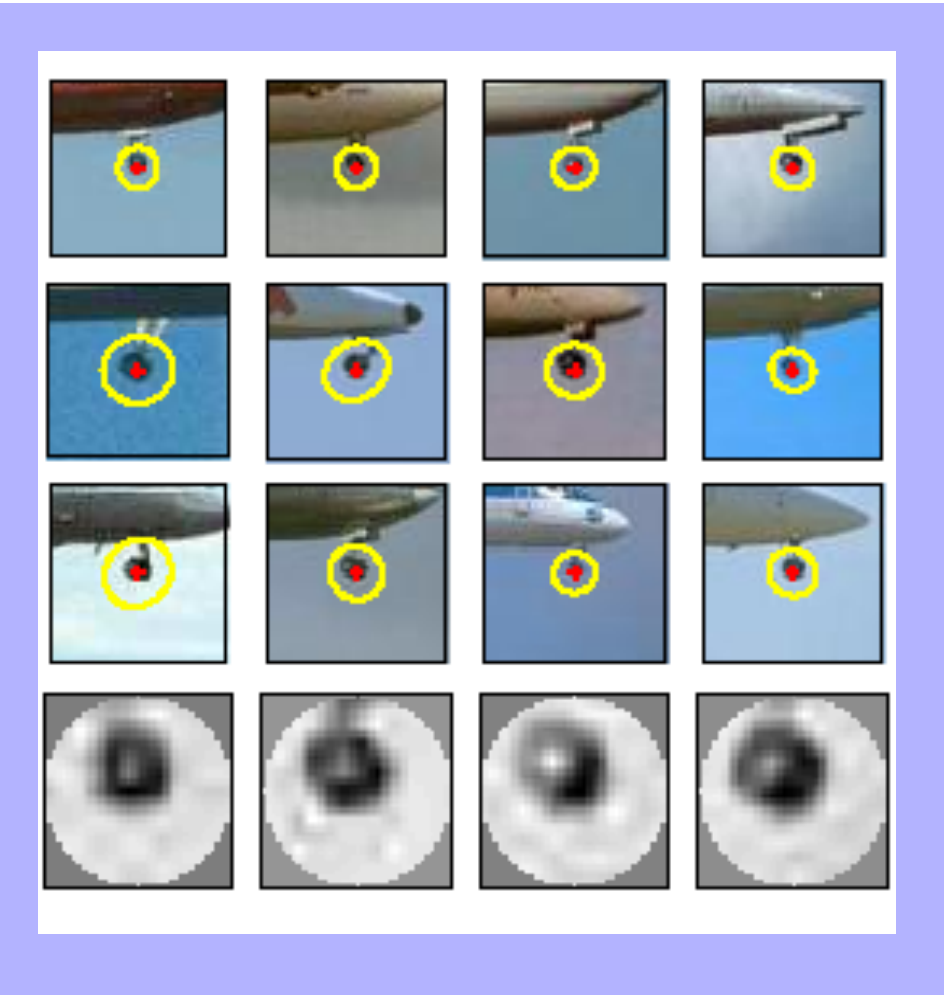
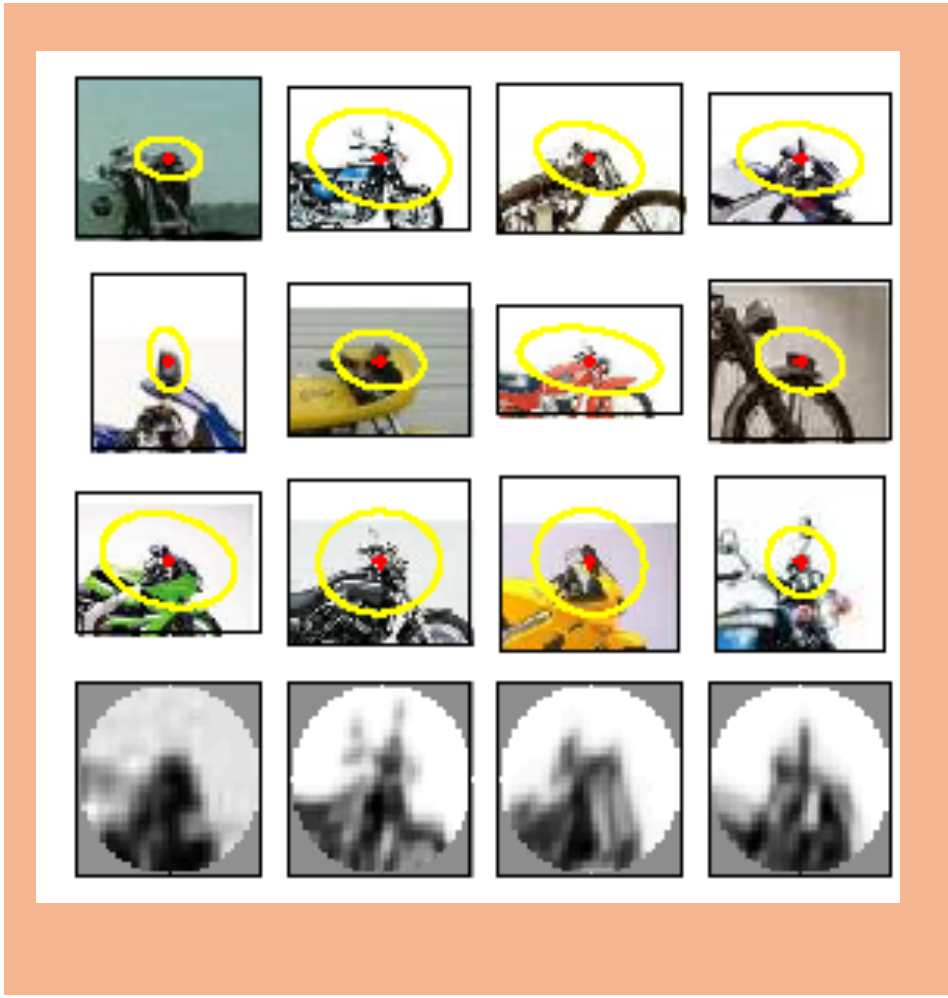


Figure from Sivic & Zisserman, ICCV 2003

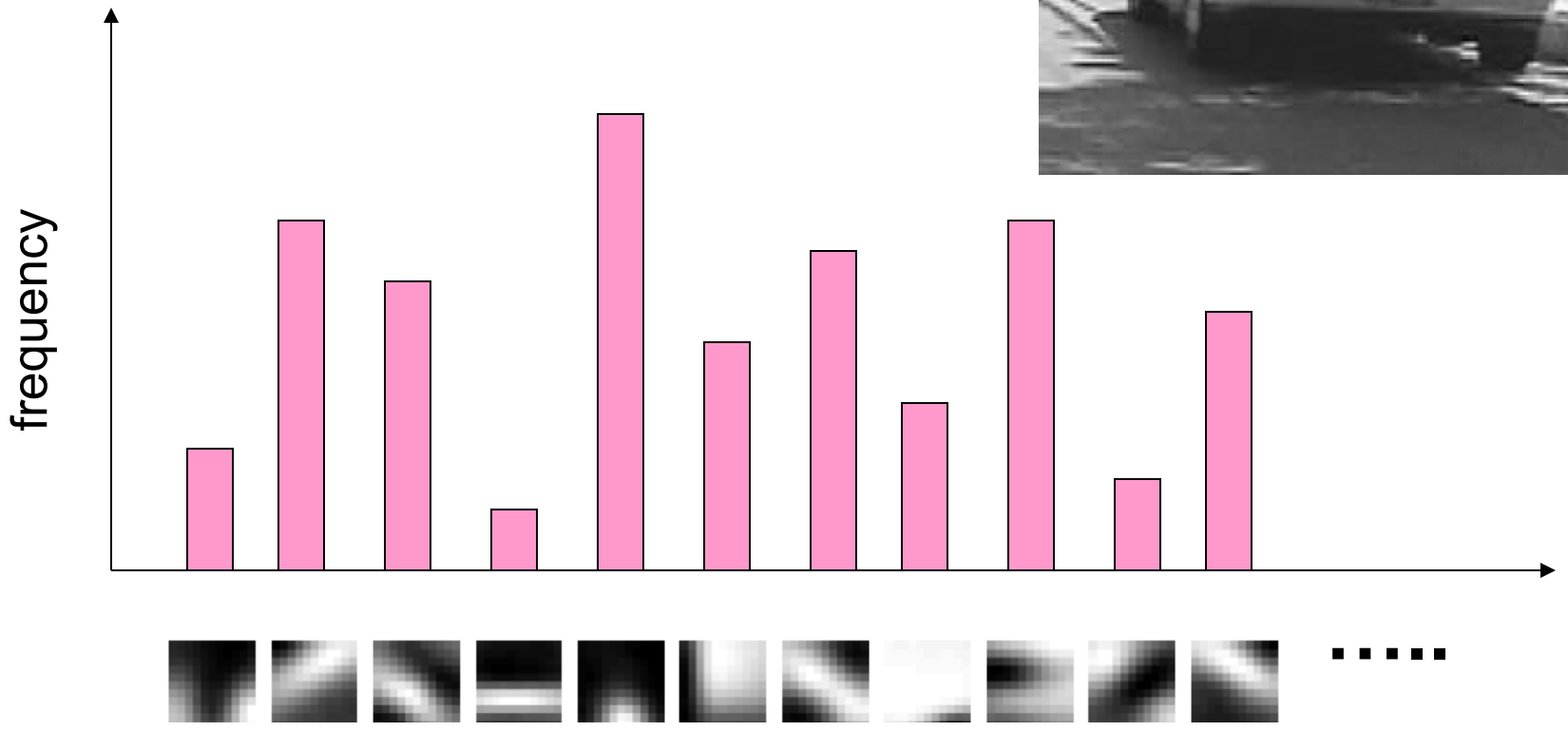
Visual words





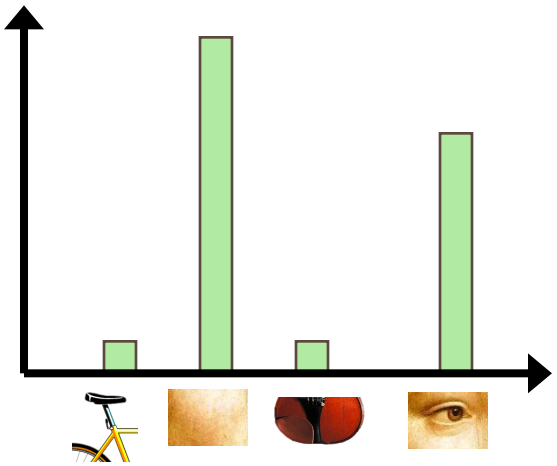
Sivic et al. 2005

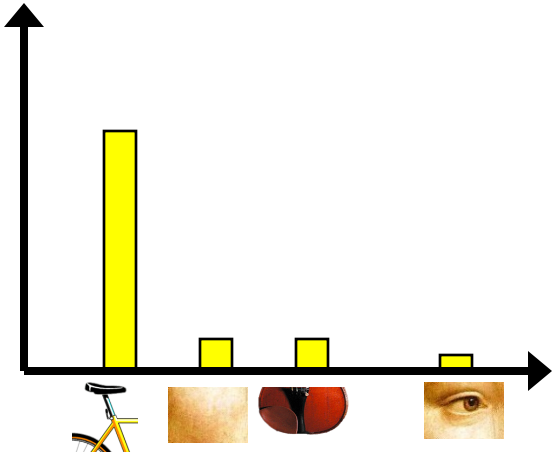
3. Image representation

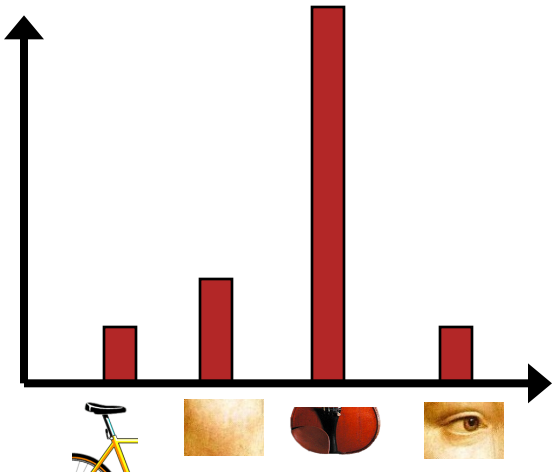


4. Image classification

- Given the bag-of-features representations of images from different classes, how do we learn a model for distinguishing them?



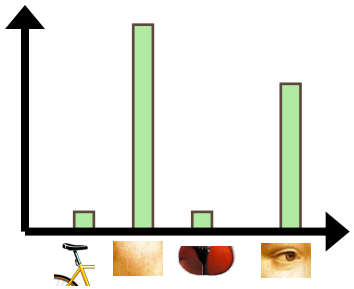




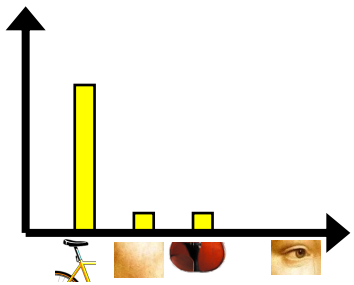
Comparing bags of words

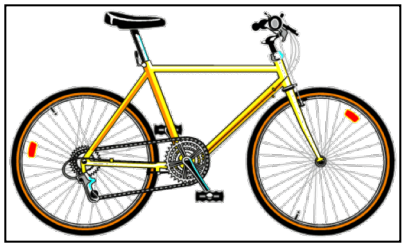
- Rank frames by normalized scalar product between their (possibly weighted) occurrence counts---*nearest neighbor* search for similar images.

[1 8 1 4]



[5 1 1 0]





$\vec{d}_j$ $\vec{q}$

$$sim(d_j, q) = \frac{\langle d_j, q \rangle}{\|d_j\| \|q\|}$$

$$= \frac{\sum_{i=1}^{V} d_j(i) * q(i)}{\sqrt{\sum_{i=1}^{V} d_j(i)^2} * \sqrt{\sum_{i=1}^{V} q(i)^2}}$$

for vocabulary of *V* words

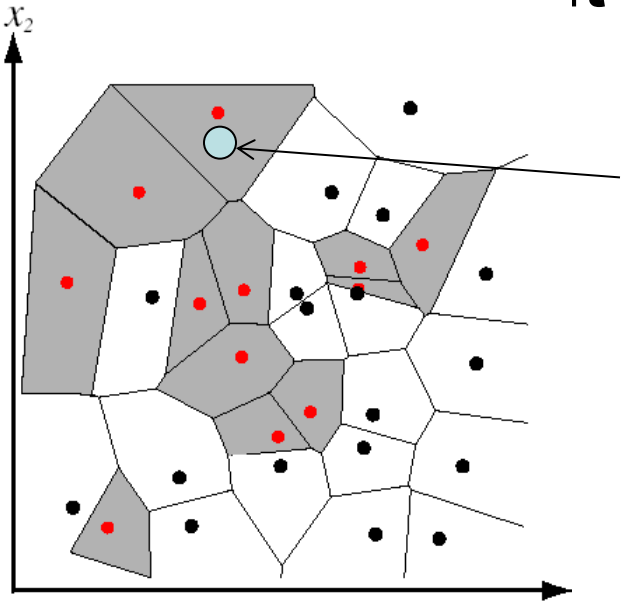
[Kristen Grauman]

Nearest Neighbors

Nearest Neighbor classification

- Assign label of nearest training data point to each test data point

Black = negative
Red = positive



from Duda *et al.*

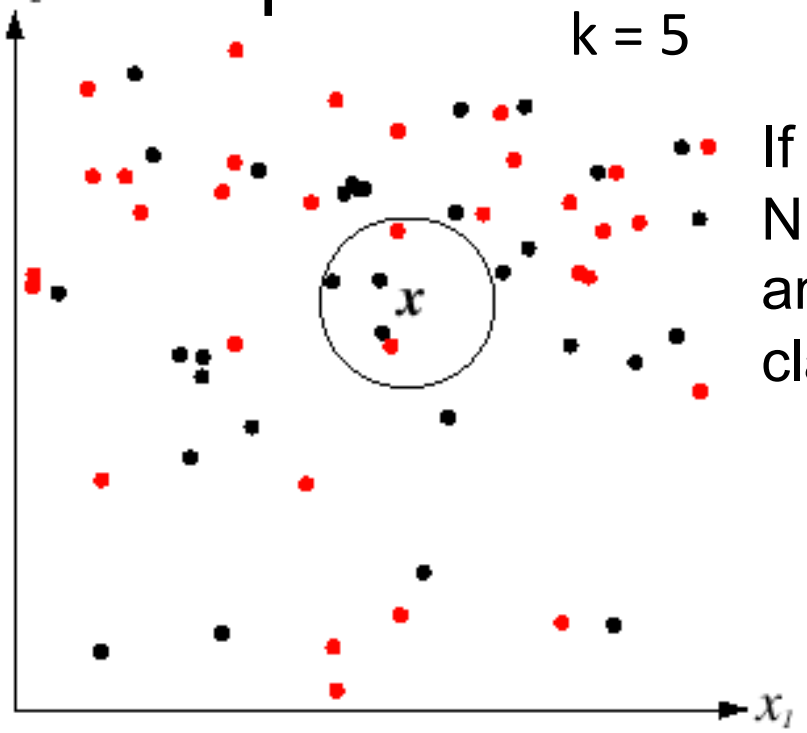
Novel test example

Closest to a positive example from the training set, so classify it as positive.

K-Nearest Neighbors classification

- For a new point, find the k closest points from training data
- Labels of the k points "vote" to classify

Black = negative
Red = positive



If query lands here, the 5 NN consist of 3 negatives and 2 positives, so we classify it as negative.

Source: D. Lowe

Nearest neighbors

- **Advantages**:
  - Simple to implement
  - Flexible to feature or distance choices
  - Can do well in practice with enough representative data
- **Limitations:**
  - Large search problem to find nearest neighbors
  - Storage of data
  - Must know we have a meaningful distance function