ACV – Applied Computer Vision

Bachelor Medientechnik & Creative Computing

Matthias Zeppelzauer
matthias.zeppelzauer@fhstp.ac.at

Djordje Slijepcevic
djordje.slijepcevic@fhstp.ac.at

Outlook

- So far:
  - Pixel-based operations on images
  - Filter (smooting, sharpening)
  - Edge detection
  - Corner detection
  - Local features and descriptors
  - Feature matching
- Today:
  - Bag of Words models

Roadmap

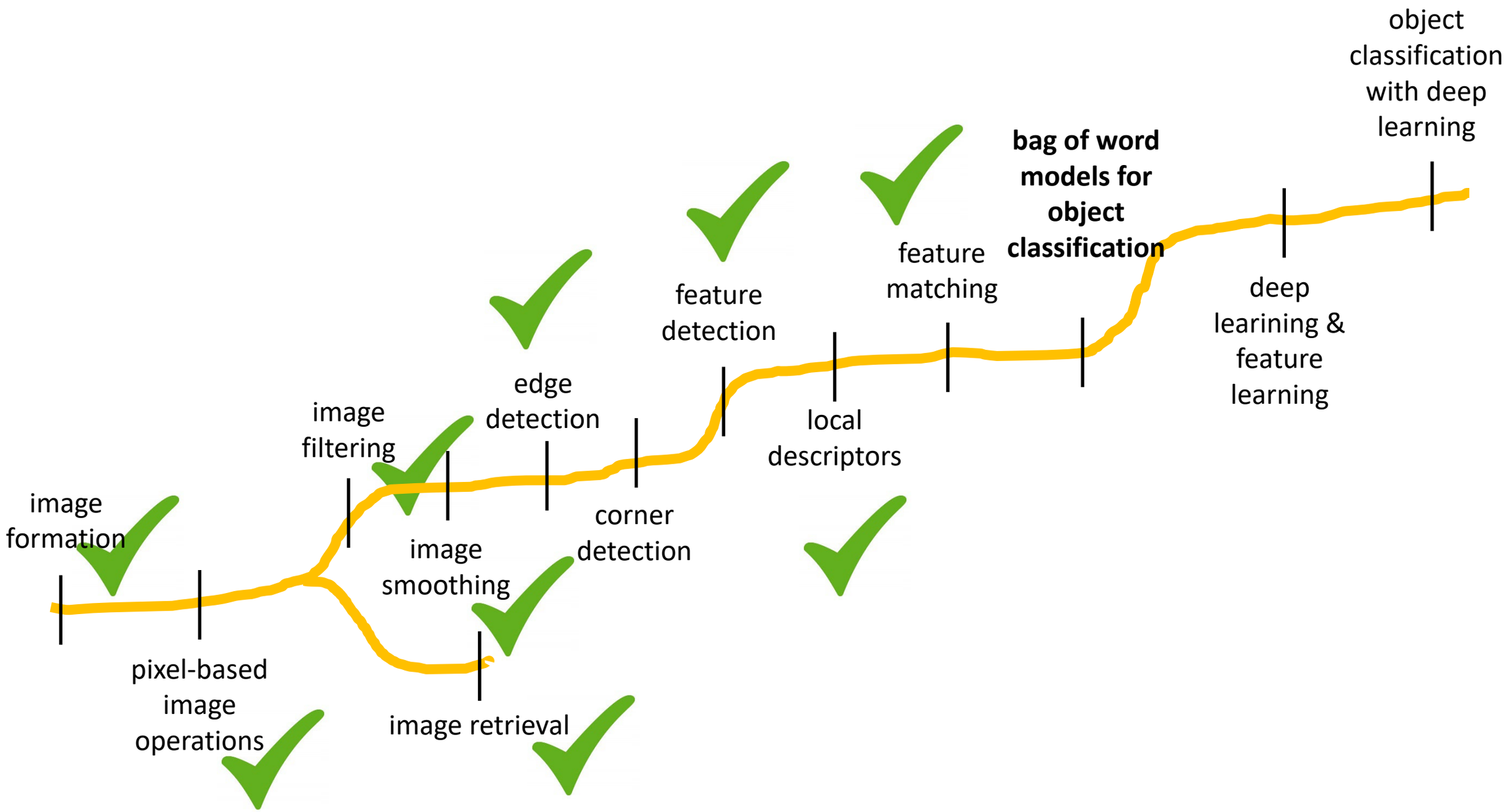


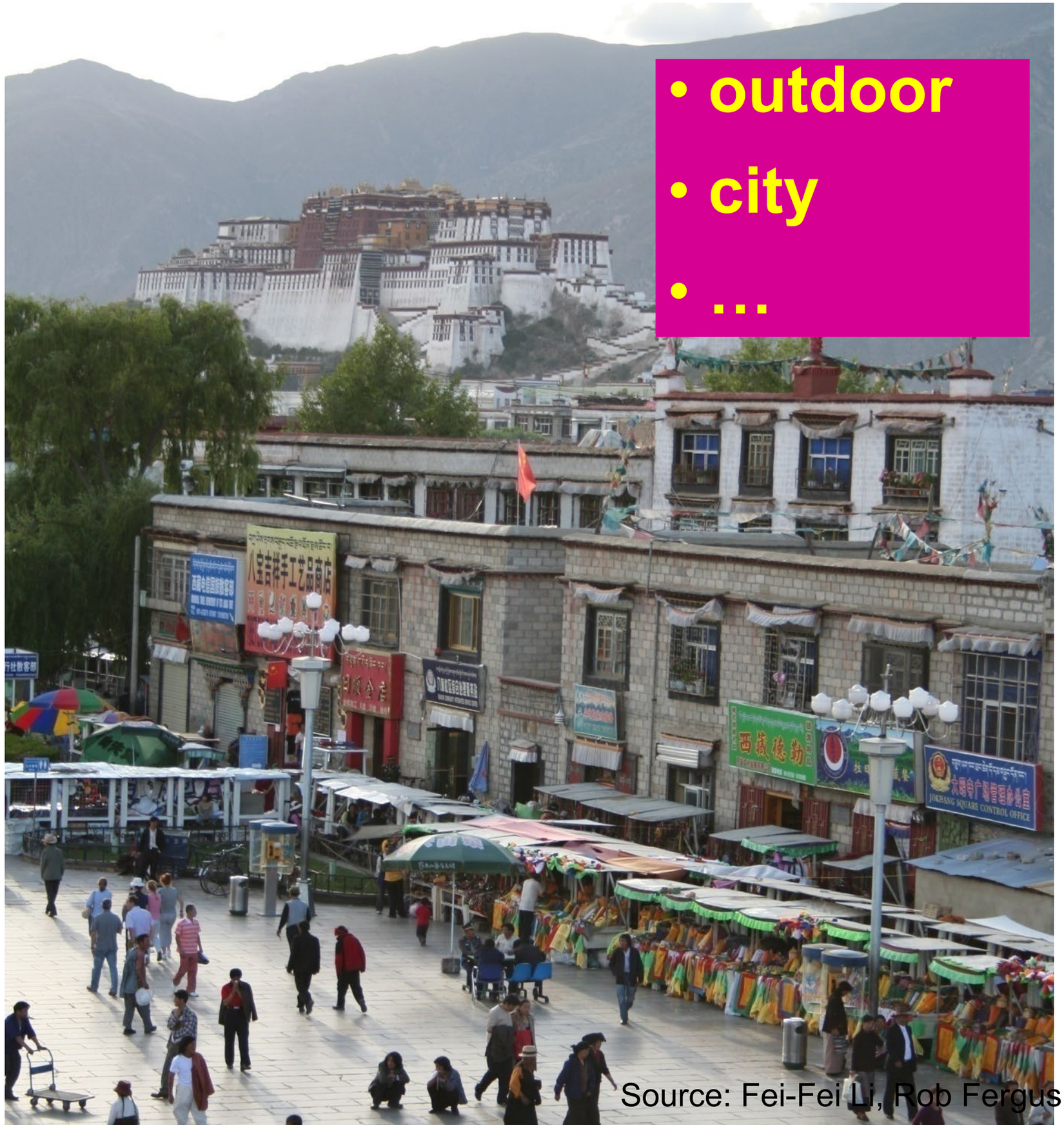
image formation
pixel-based image operations
image filtering
image smoothing
edge detection
corner detection
feature detection
image retrieval
feature matching
local descriptors
bag of word models for object classification
deep learining & feature learning
object classification with deep learning

Digitale Medientechnologien

Object and Scene Recognition

Getting higher-level representations
with pixels, edges and corners…

Scene recognition



Source: Fei-Fei Li, Rob Fergus, Antonio Torralba.

Object Recognition: distinguish between

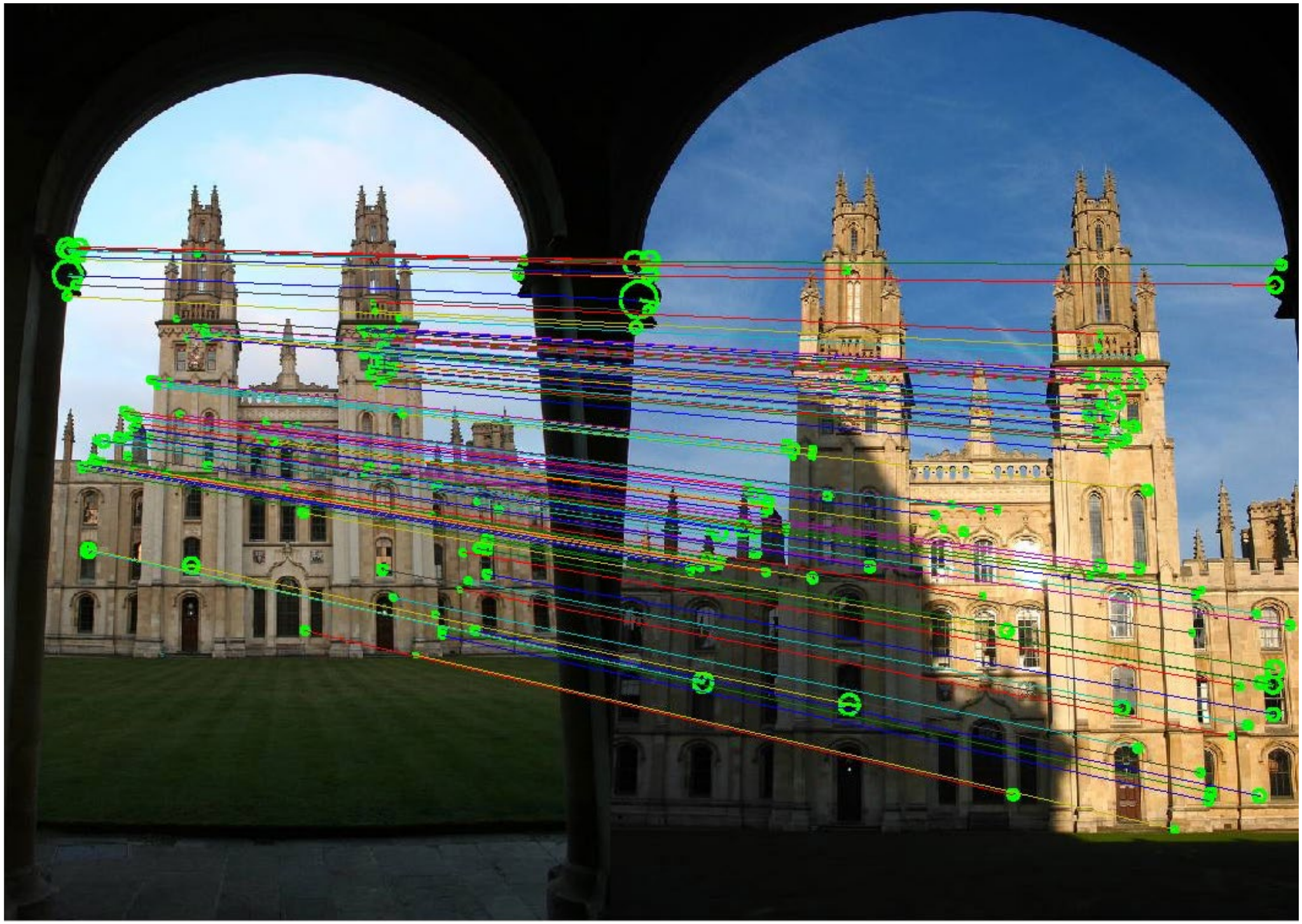
- *Instance-level* recognition problem

- *Generic* categorization problem




K. Grauman

Remember: what does this corresponds to?



Generic object Categorization

- Object detection for specific object types

Challenges: robustness to…

Illumination

Object pose

Clutter

Occlusions

Intra-class appearance

Viewpoint

Object Categorization

- Task Description
  - “Given a small number of training images of a category, recognize a-priori unknown instances of that category and assign the correct category label.”

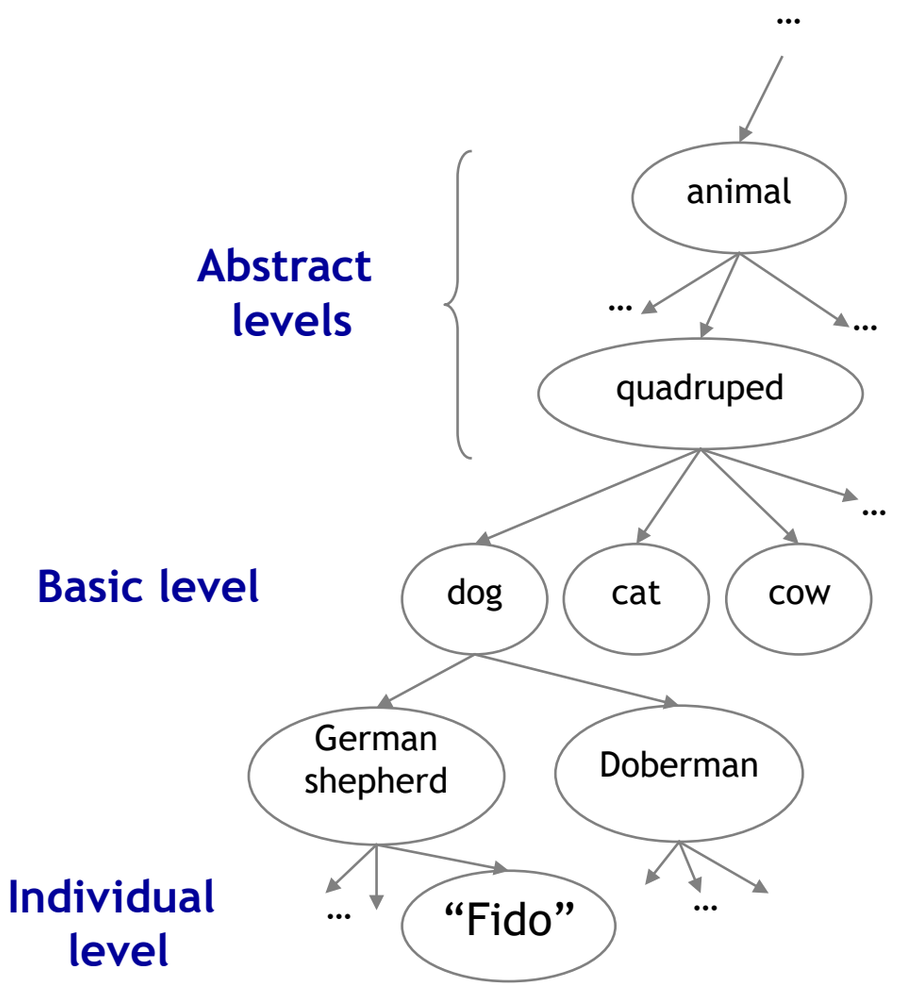
- Which categories are feasible visually?

K. Grauman, B. Leibe

Visual Object Categories

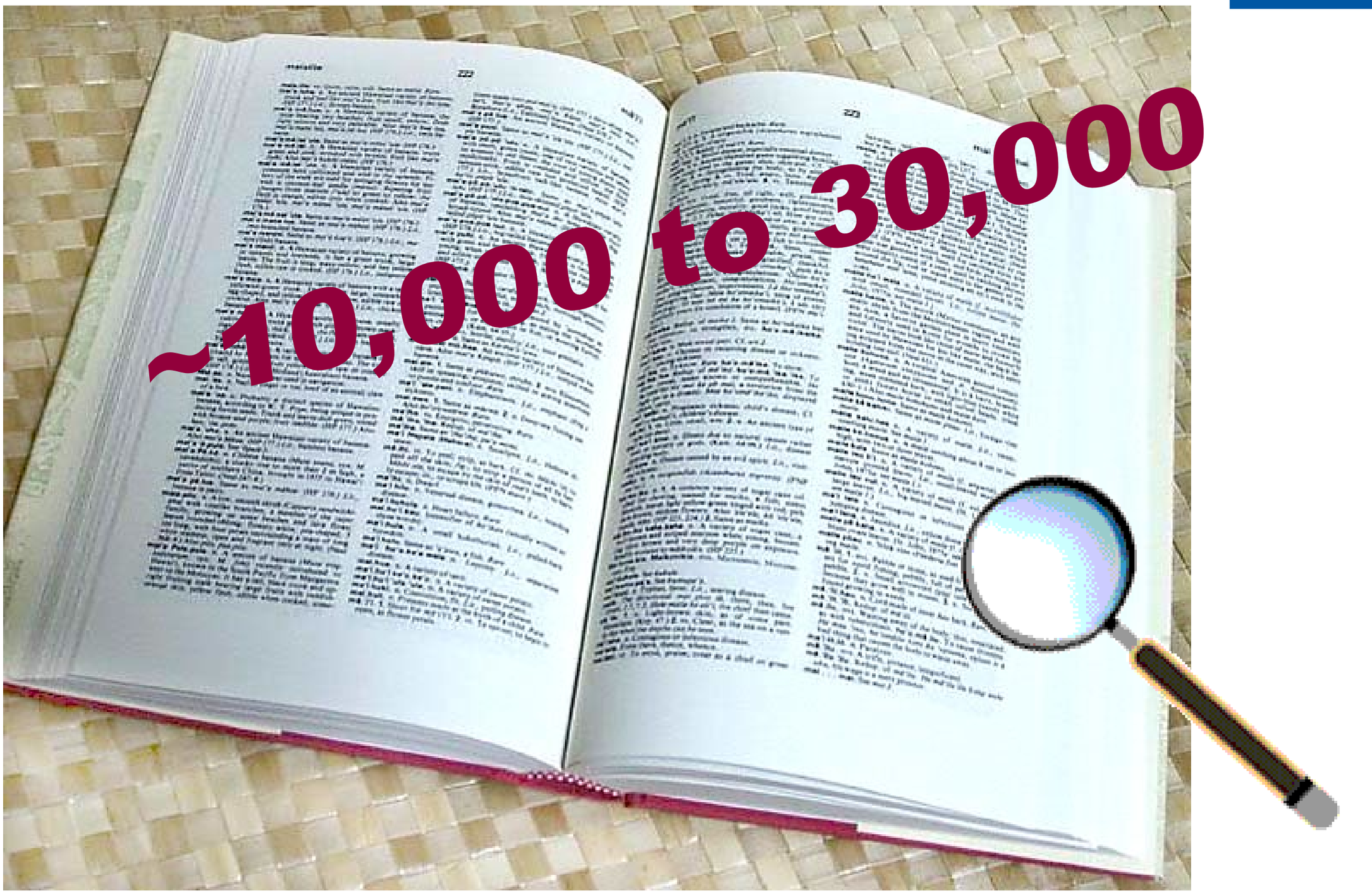
- Basic-level categories in humans seem to be defined predominantly visually.
- There is evidence that humans (usually) start with basic-level categorization *before* doing identification.

⇒ Basic-level categorization is easier and faster for humans than object identification!



K. Grauman, B. Leibe

How many object categories are there?



Slide: Fei-Fei Li, Rob Fergus, Antonio Torralba.

[Biederman 1987]

~10,000 to 30,000

How to define an object category?

- Define **explicit** rules
  - e.g. “A chair has 4 legs”
  - what’s the problem?





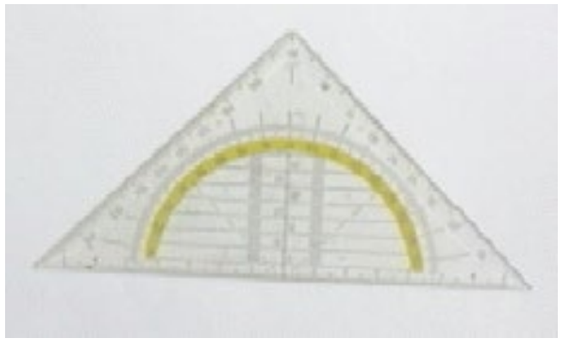

How to define an object category?

- Functional Categories
  - e.g. chairs = *“something you can sit on”*

K. Grauman, B. Leibe

How to define an object category?

- Ad-hoc categories
  - e.g. *“something you can find in an office environment”*



A first data-driven approach to object recognition..

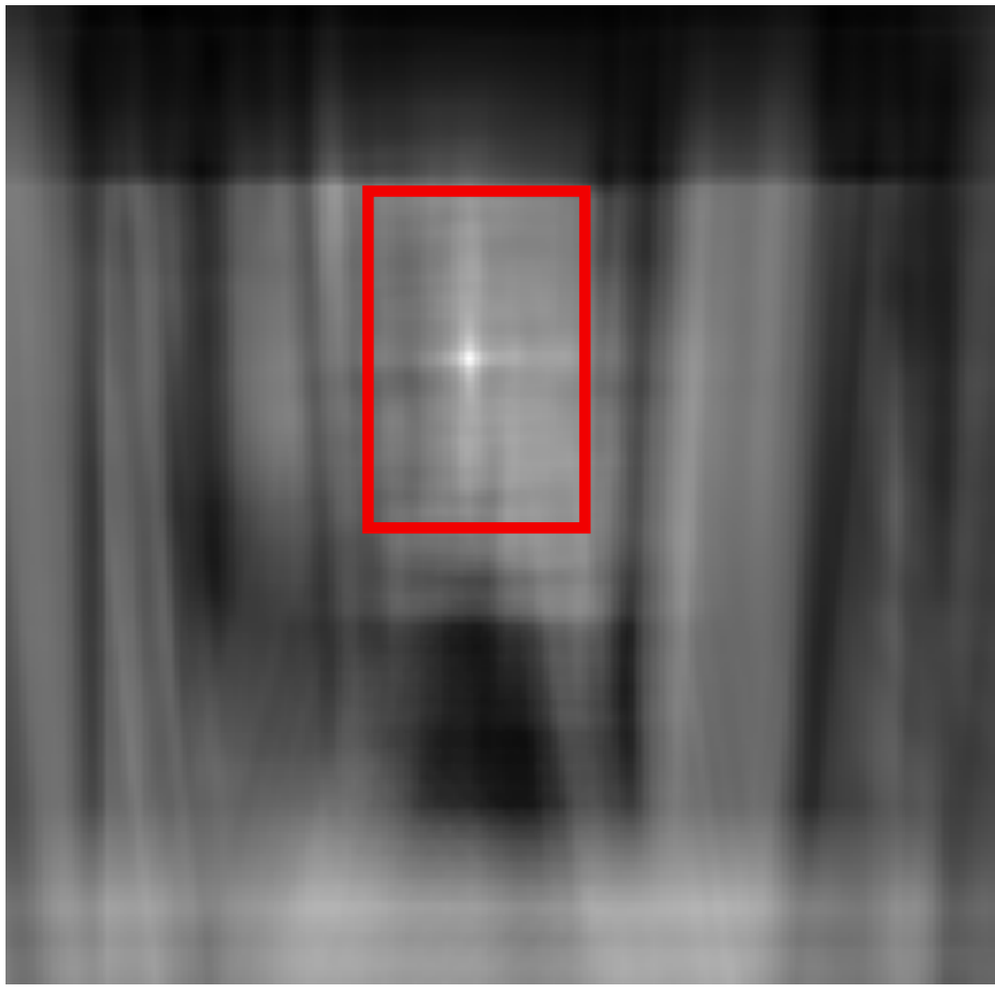
- Let’s try out template matching
  - Define a template, slide it over the image and try to find similar objects as the template

Template Matching – Is Object recognition really so hard?

This is a chair

Find the chair in this image

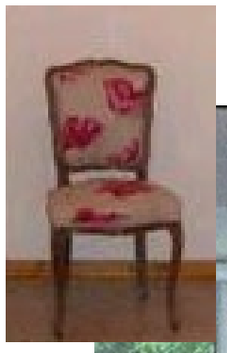
Output of normalized correlation



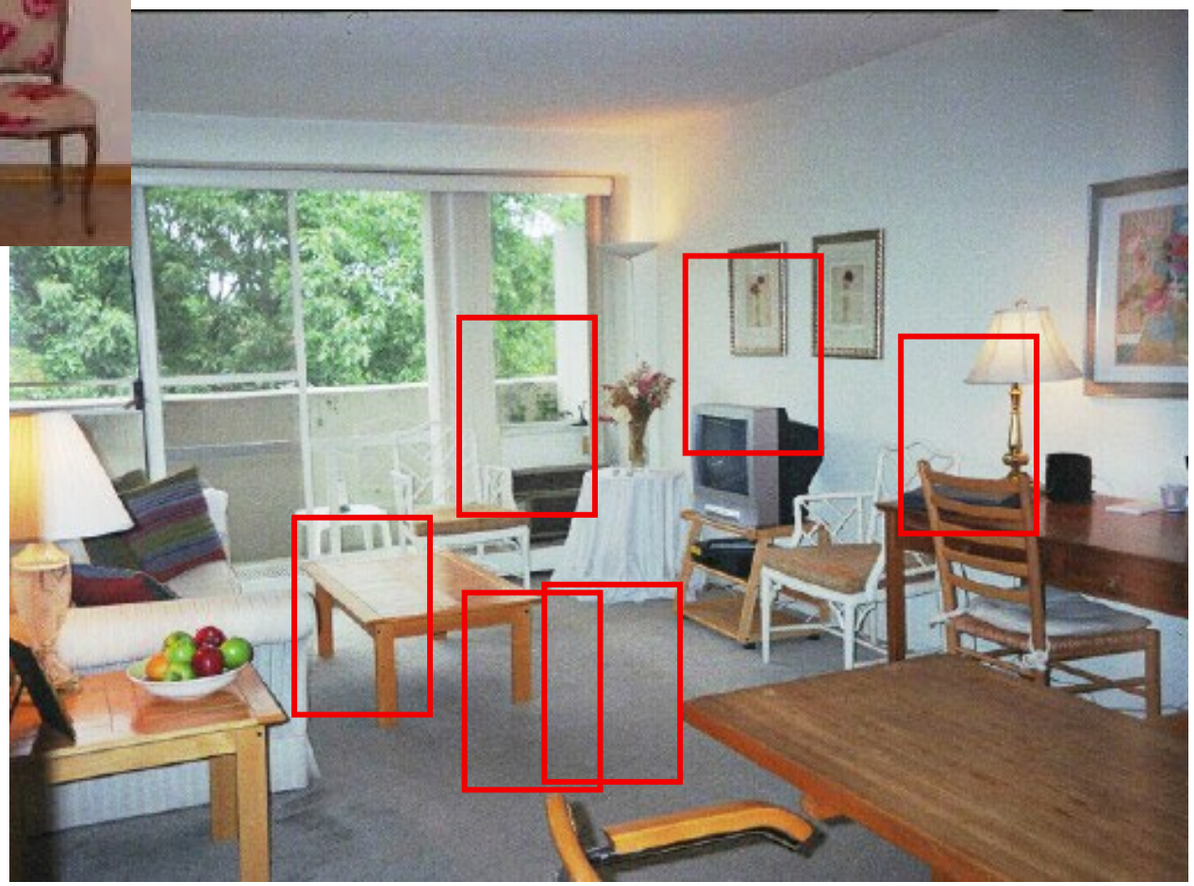
Cross-correlation!

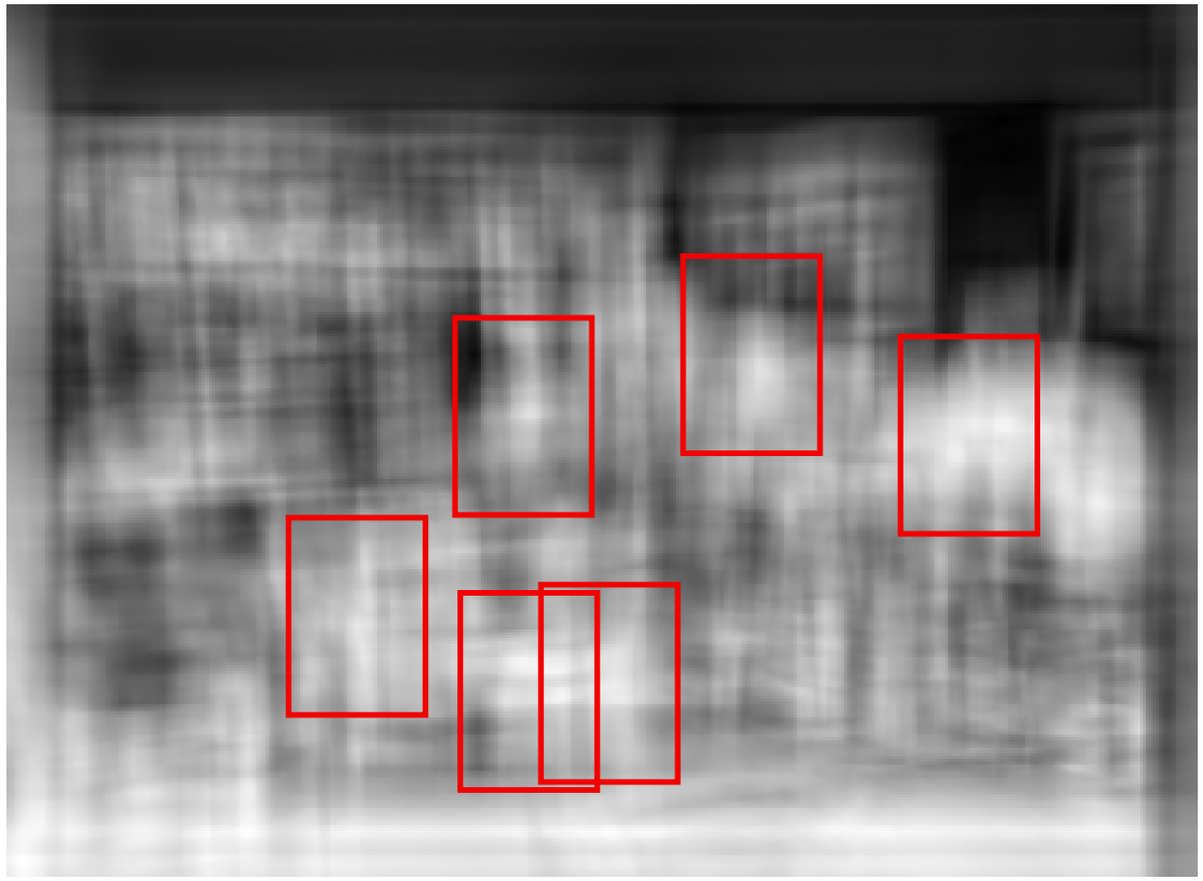
Slide by Antonio Torralba

Template Matching – Is Object recognition really so hard?



Find the chair in this image





Pretty much garbage
Simple template matching is not going to make it

Antonio's biggest concern:
how do I justify 50 years of research if this experiment did work?

Slide by Antonio Torralba

We need a cleverer approach..

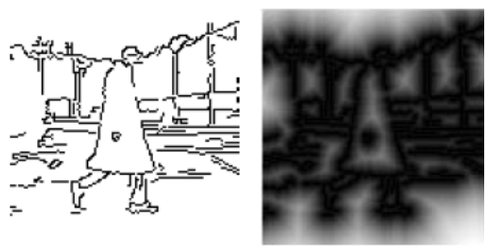
- Explicit definition of rules / templates is not flexible enough
- We need more general models
- ***Implicit*** object definitions!

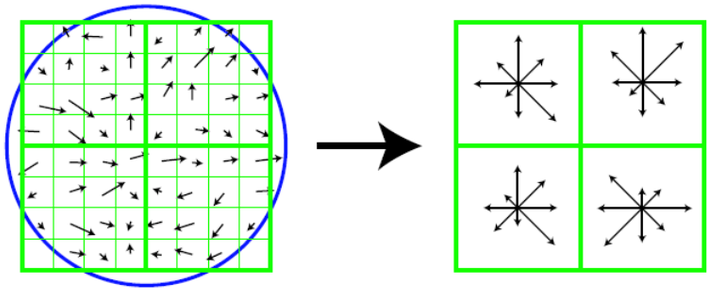
- Learn models for objects that implicitly capture the characteristics of the object class...

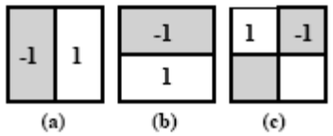
We need a cleverer approach..

- What “features” could we use to match our template to the image?
- Obviously using the pixels directly and SSD is not working → too strict

- Why not use
  - Edges,
  - Corners,
  - Feature Points,







- and build more robust representations from them?

Credit: Deva Ramanan, A. Efros

Generic category recognition: basic framework

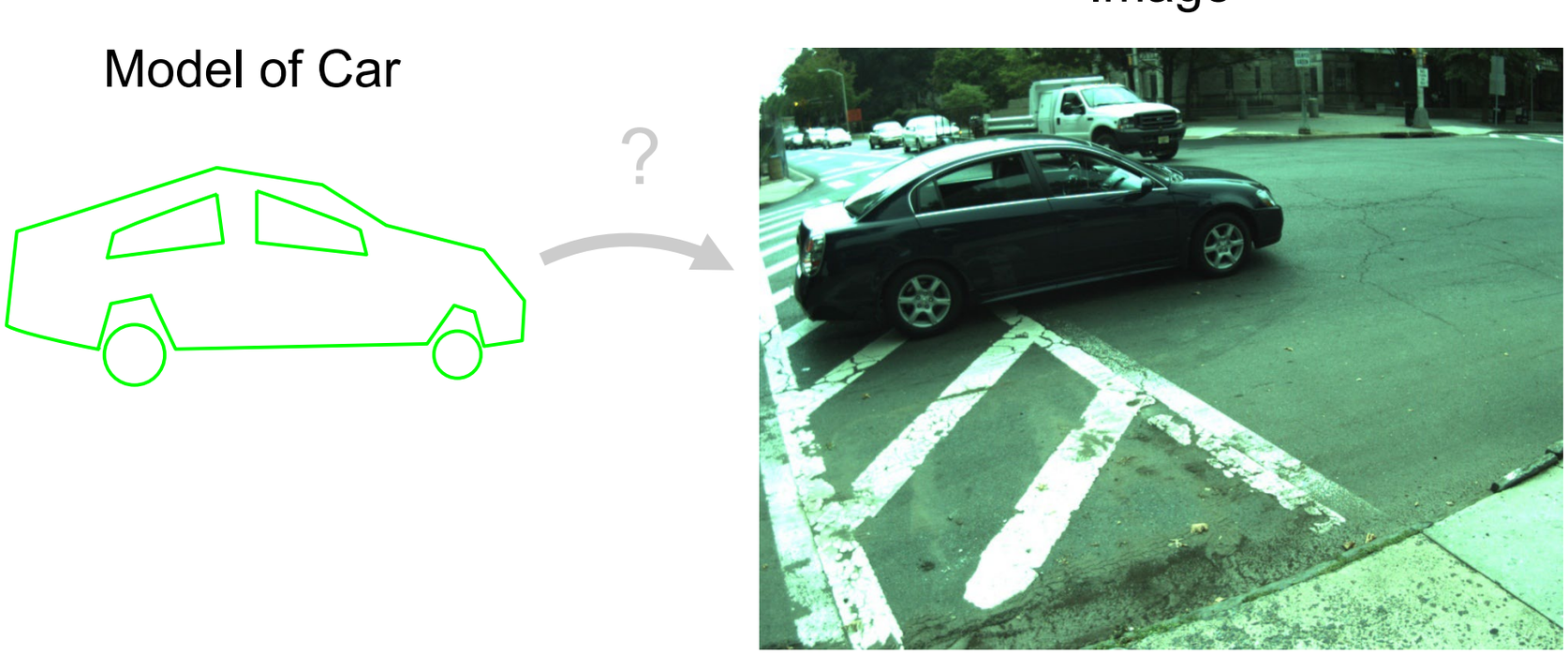
Machine Learning!

- Build/train object model
  - Choose a representation (e.g. local features)
  - Learn or fit parameters of model / classifier
- Generate candidates in new image
- Score the candidates

Find a good representation for objects

- Objects are well characterized by their individual *parts*
- E.g. Faces:
  - Eyes, nose, mouth
- Or Cars:
  - wheels, window, lights, etc.

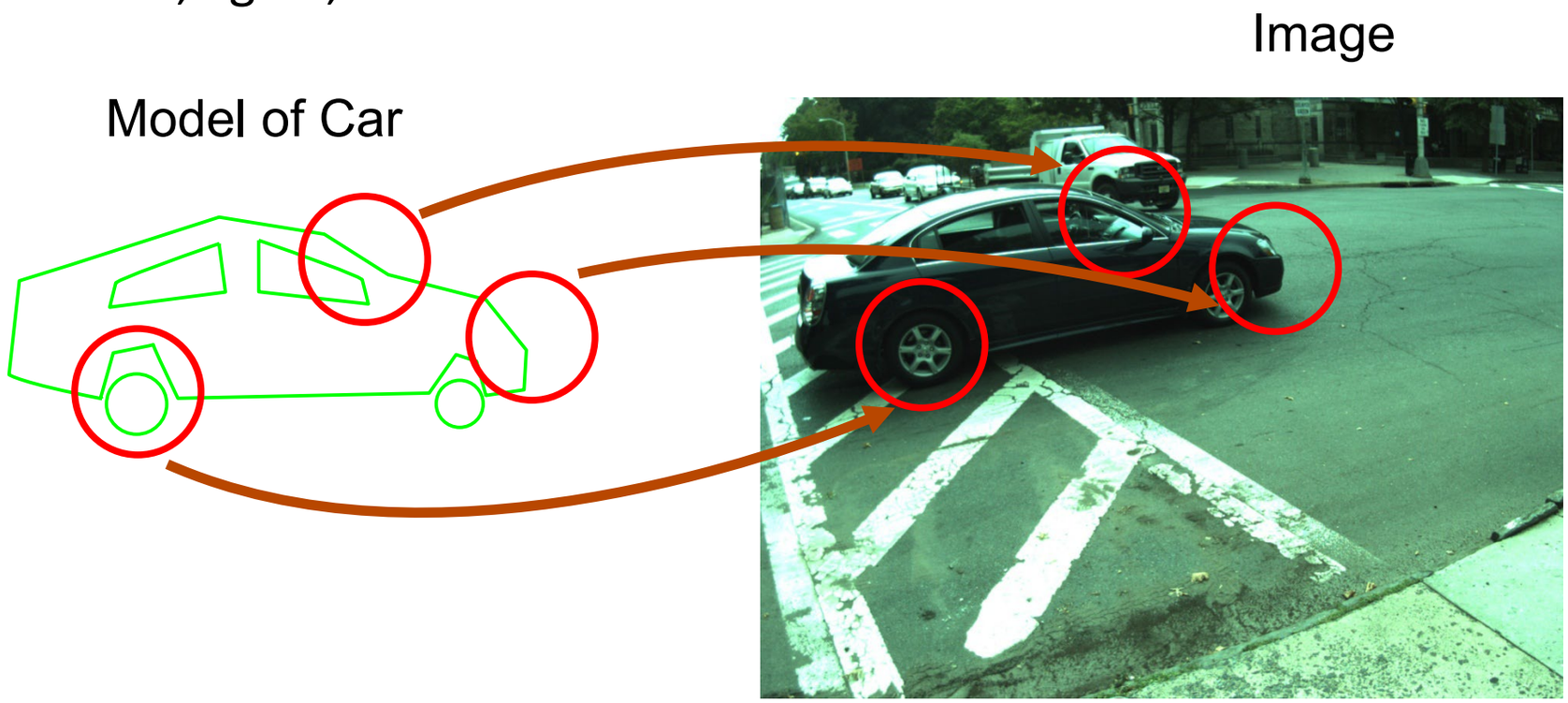
Btw: this is David Lowe, the inventor of SIFT!



Credit: Alex Berg

Find a good representation for objects

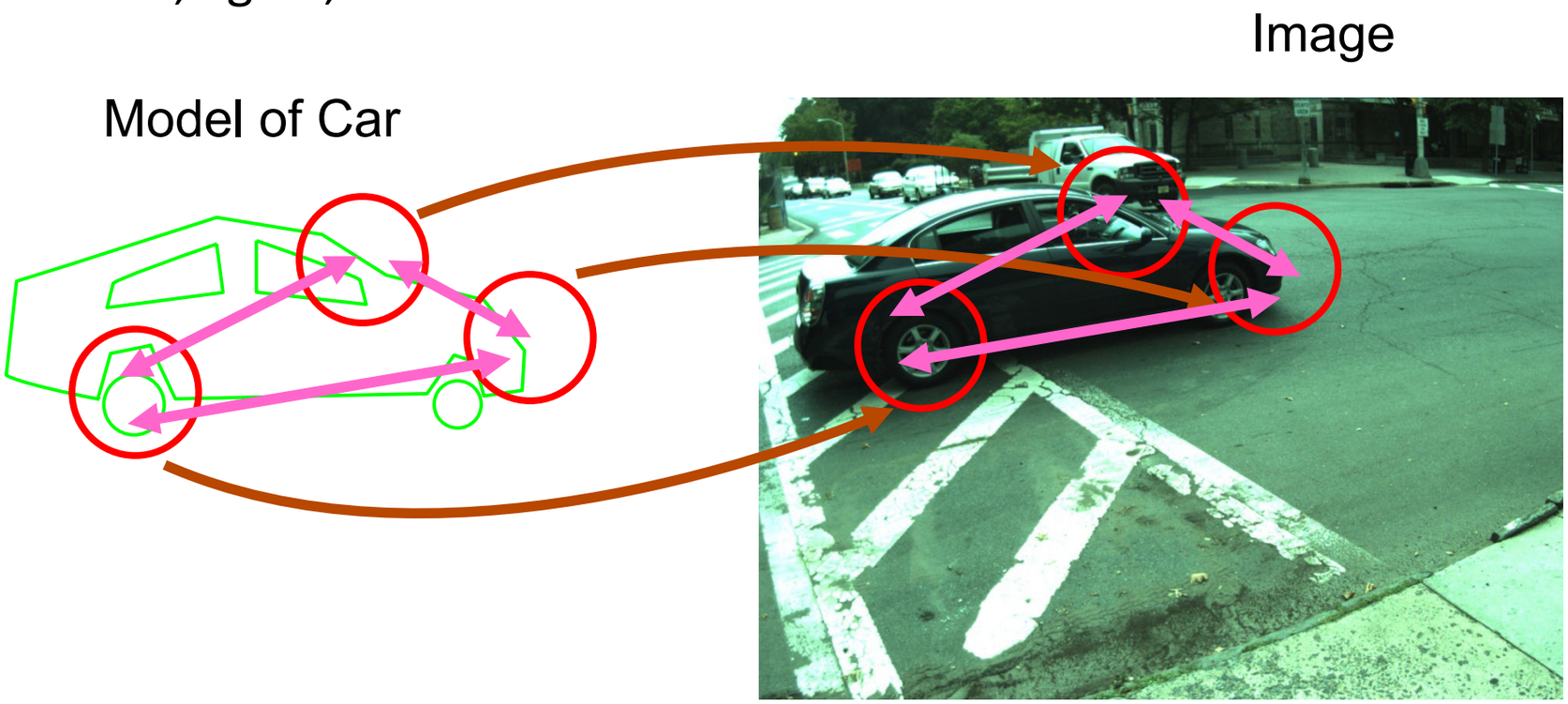
- Objects are well characterized by their *object parts*
- E.g. Faces:
  - Eyes, nose, Mouth
- Or Cars:
  - wheels, window, lights, etc.



Credit: Alex Berg

Find a good representation for objects

- Objects are well characterized by their *object parts*
- E.g. Faces:
  - Eyes, nose, Mouth
- Or Cars:
  - wheels, window, lights, etc.



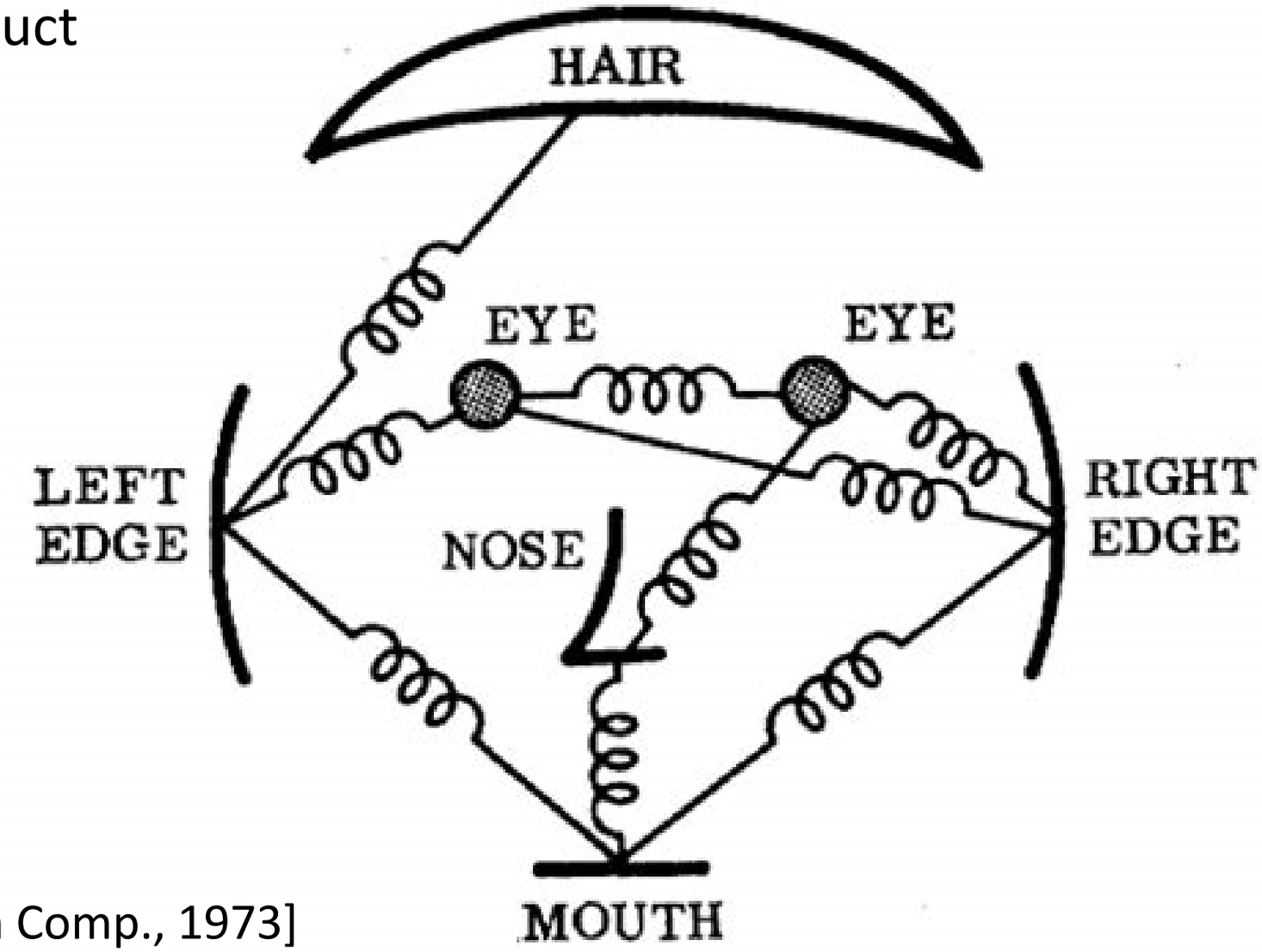
Credit: Alex Berg

How to describe Objects and Object Parts

- How do we describe objects and object parts?
  - Example: wheel of a car
  - Description: circular, symmetric, usually silver near the center and black outside, there are holes…
  - Relations: the car has four wheels, two at the front and two at the back
  - → we describe parts and relations useing **words**

Part-based models

- Early idea for object detection
- Main components: parts with relations (springs in the figure)
- In practice complex to construct
- Next, we consider a very simplified model based only on components, neglecting their relations...



[Fischler and Elschlager, IEEE Trans. on Comp., 1973]

The Bag of Words (BoW) model

- Origin: Text-Retrieval (e.g. searching for similar websites)
- Input = textual query, e.g. “Hotel in Paris”
- Output = related websites to the query
- How to find related websites?

- Idea: represent documents (websites) as “Bags of Words”
  - “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

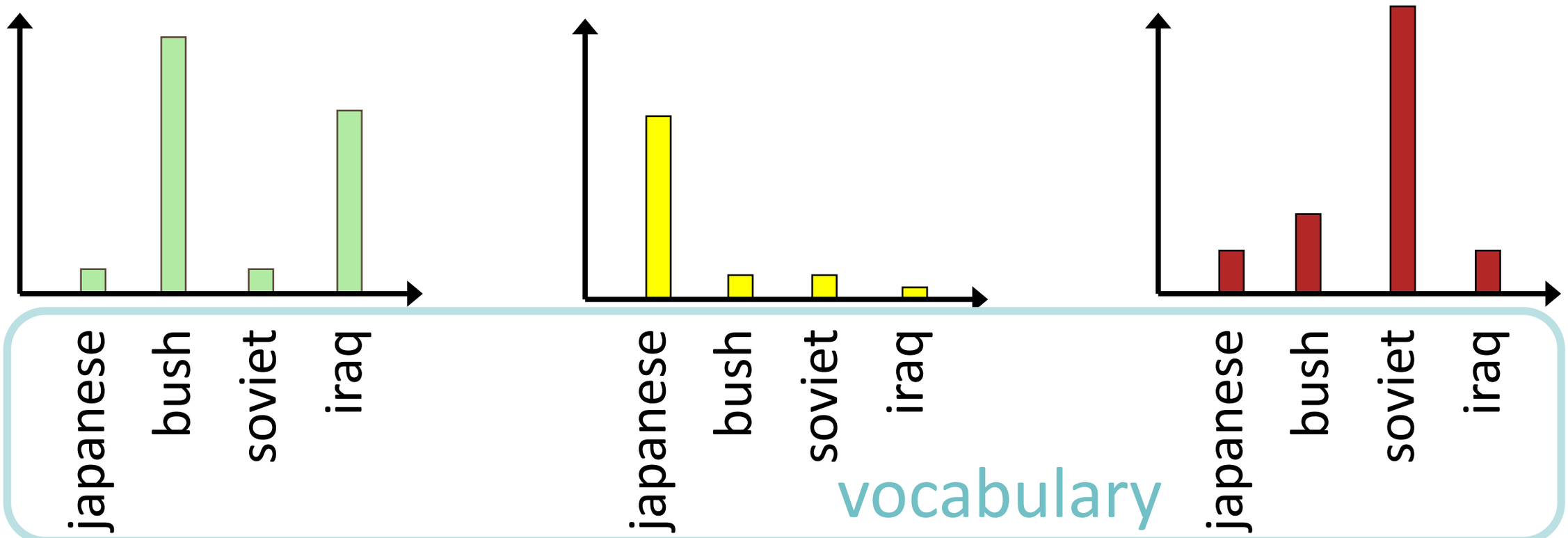
- Orderless document representation: frequencies of words from a dictionary
  Salton & McGill (1983)

Bag-of-words models

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

Representation: Bag of Words Histogram

- Find frequent words → build a vocabulary of words
- Count number of occurrences per word in each document
- Assemble in a histogram:



Visual Words…

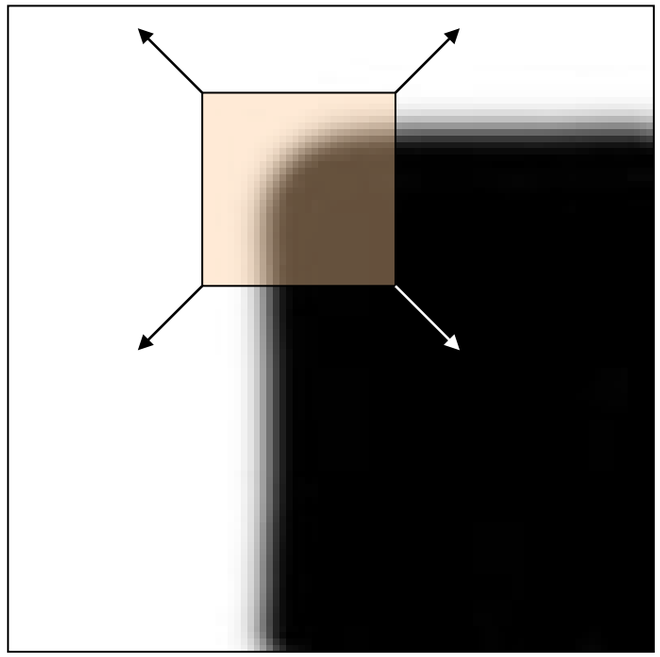
Build a Visual Vocabulary:
The Bag of Visual Words (BoVW) Model

Image → Bag of ‘words’

What is a good visual word?

- How to find good visual words?

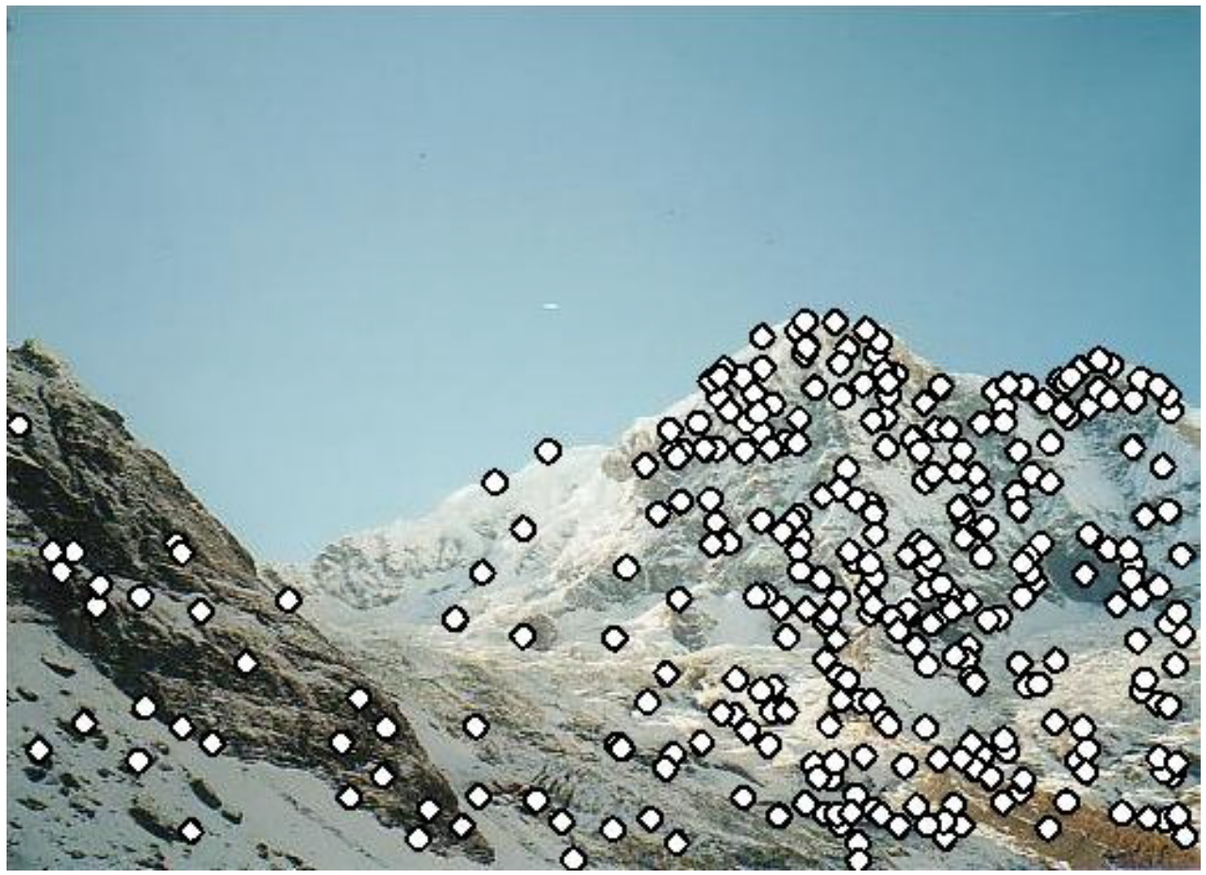
Edges?

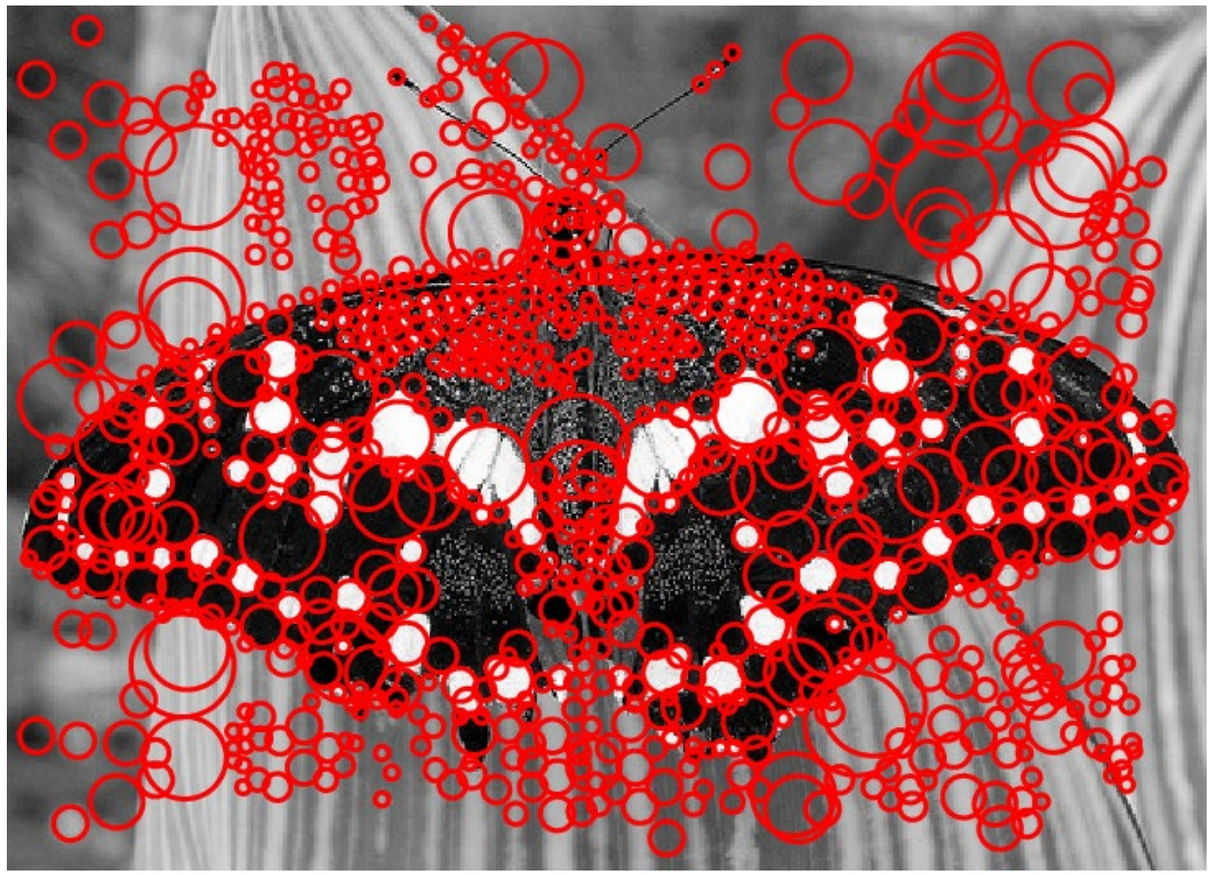


Corners?

What is a good visual word?

- How to find good visual words?



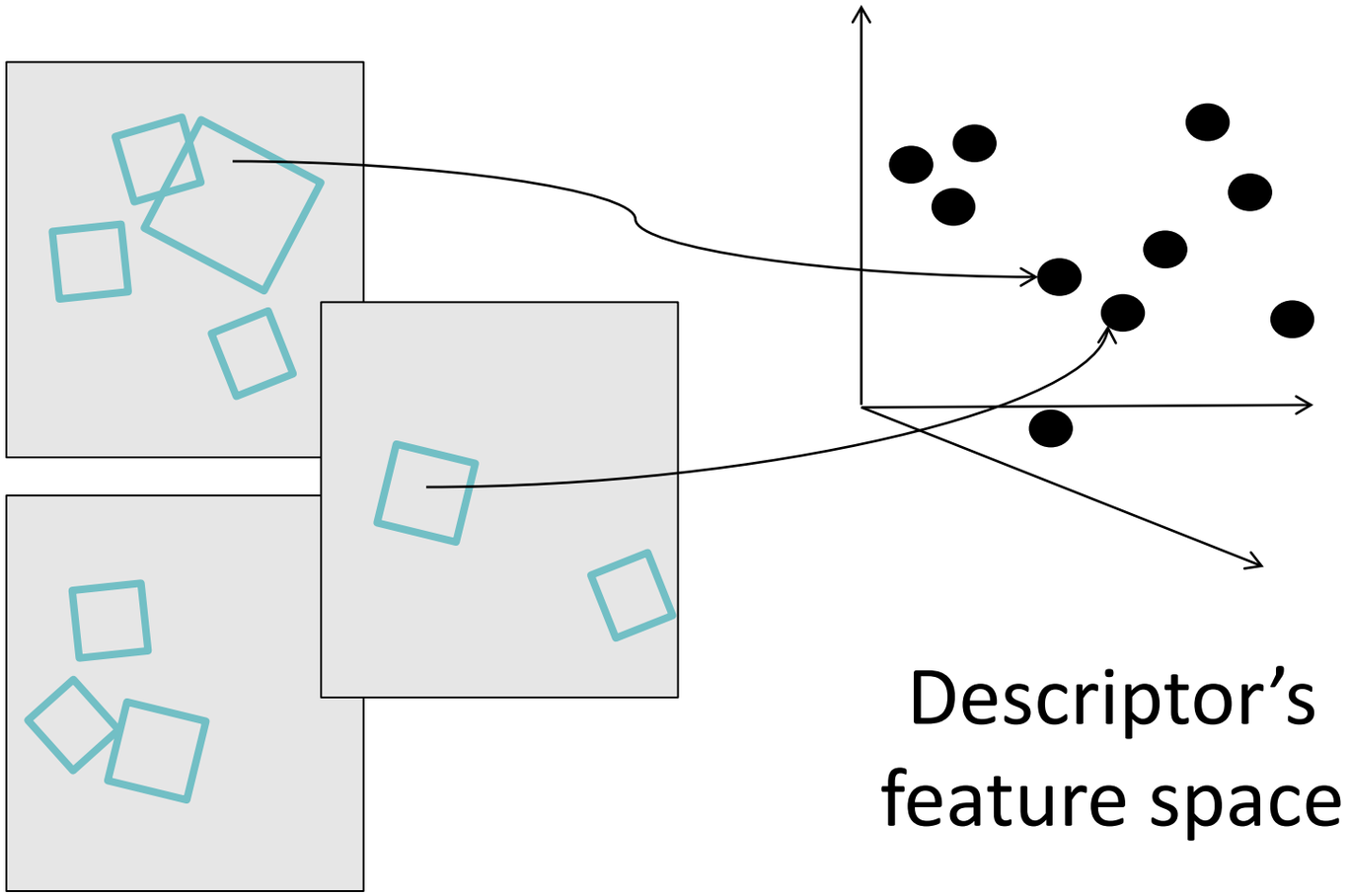


- Use local feature locations (at multiple scales) to extract characteristic image patches

- Usel local descriptors, e.g. SIFT descriptor to describe them numerically

Indexing local features

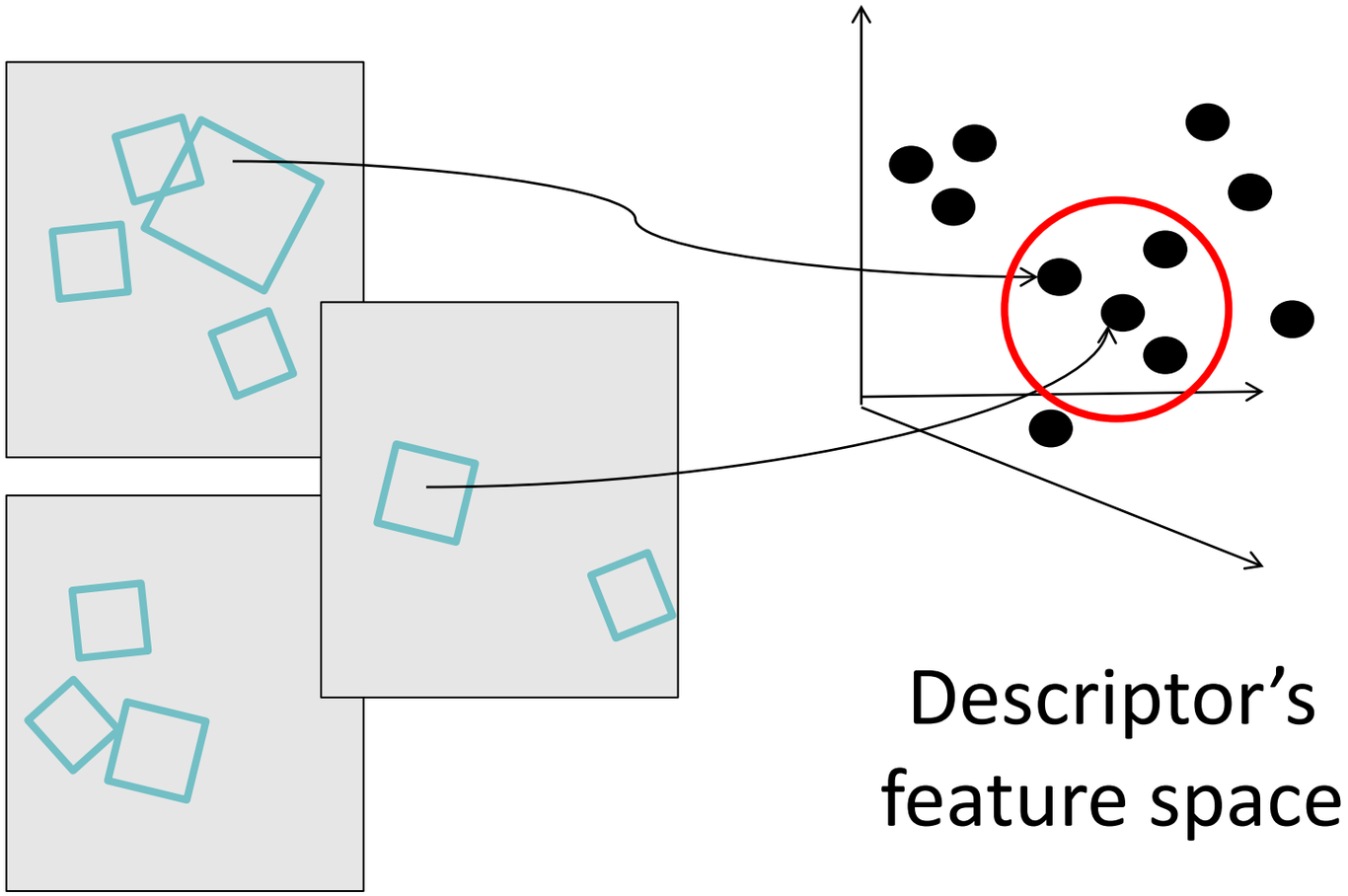
- Each patch / region has a descriptor, which is a point in some high-dimensional feature space (e.g., SIFT)



Kristen Grauman

Indexing local features

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

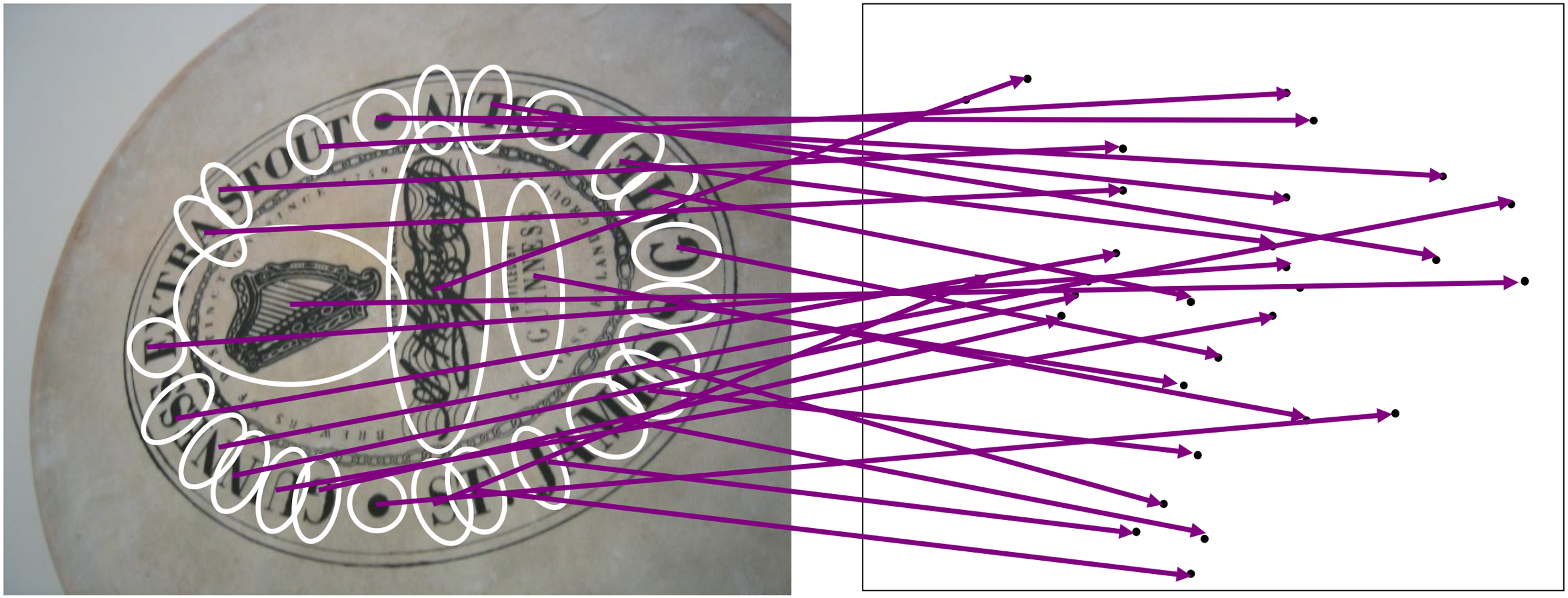


similar points form clusters
→ visual words

Kristen Grauman

Visual words: main idea

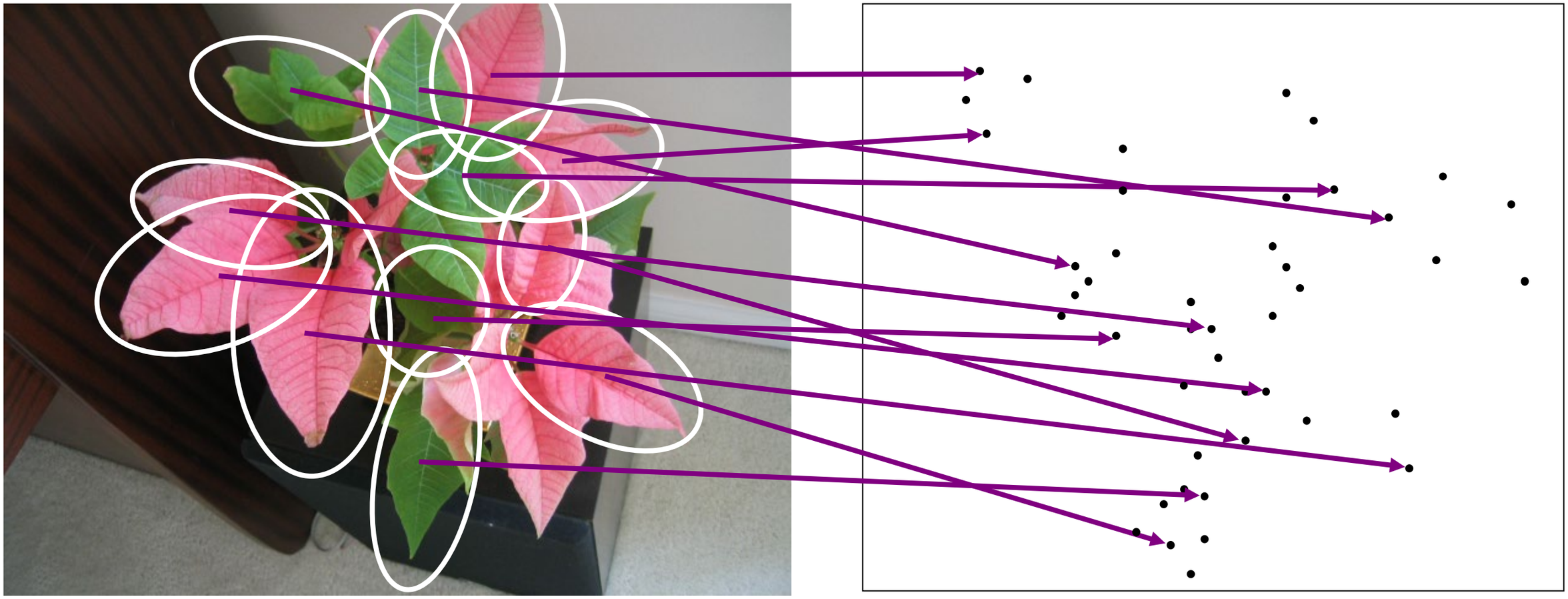
- Extract some local features from a number of images …



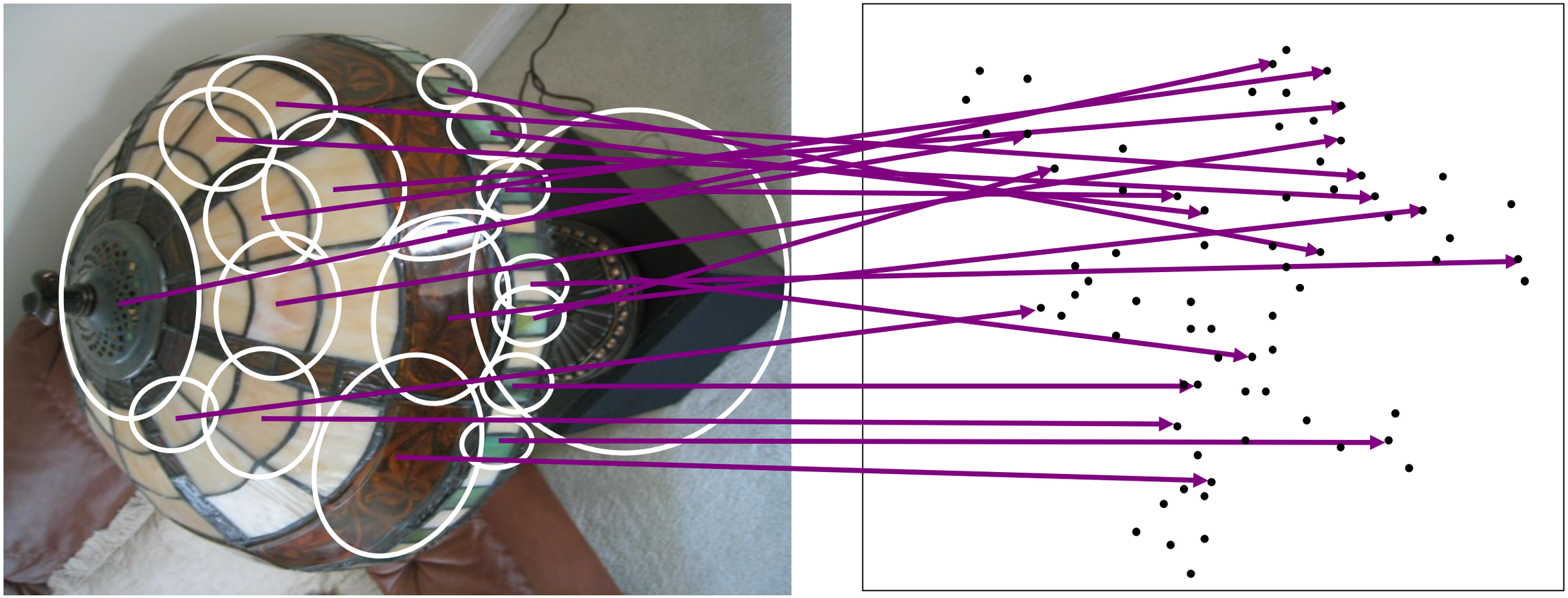
e.g., SIFT descriptor space: each point is 128-dimensional

Slide credit: D. Nister, CVPR 2006

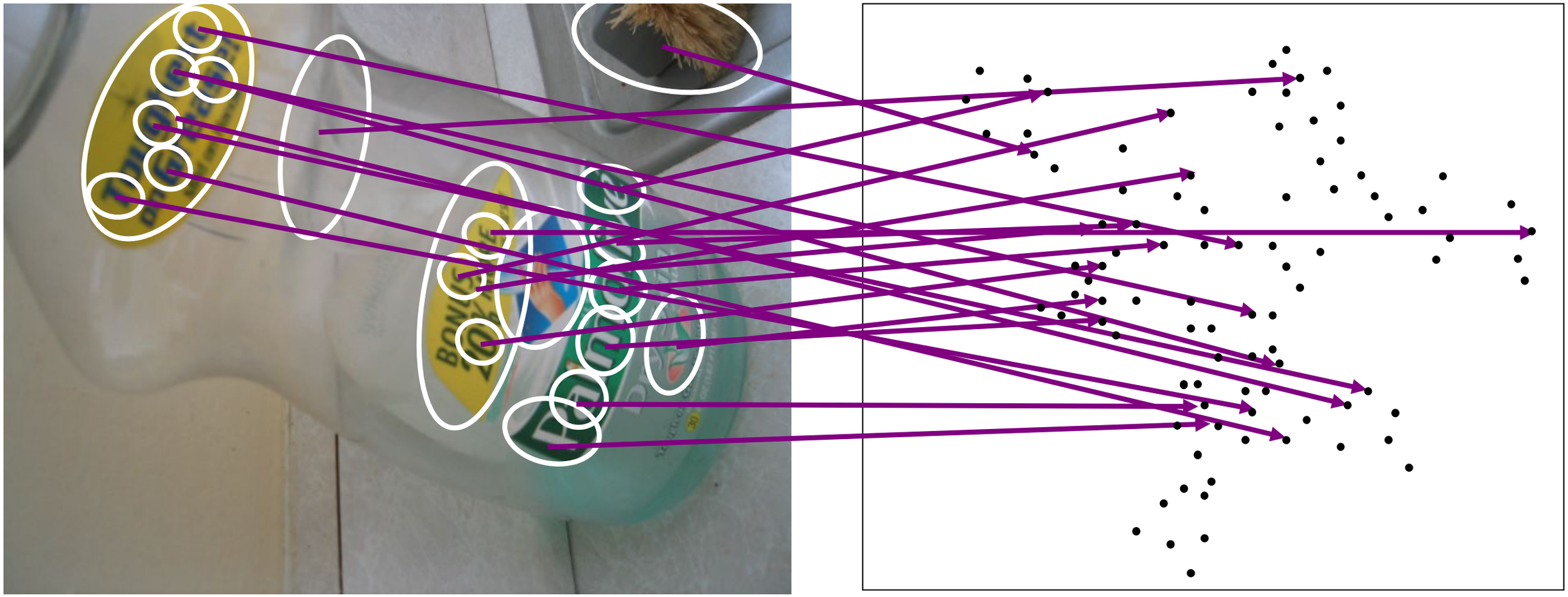
Visual words: main idea

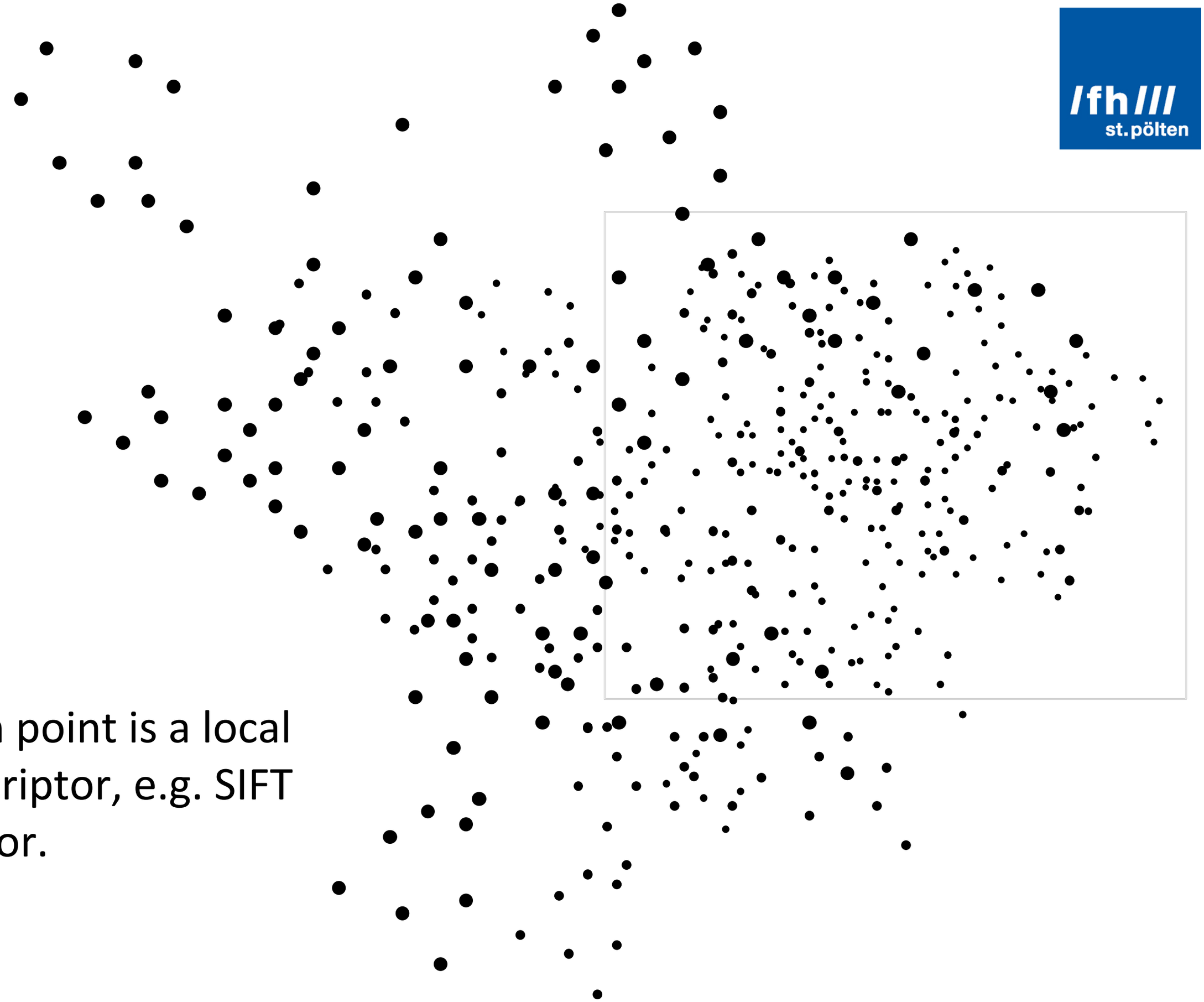


Visual words: main idea



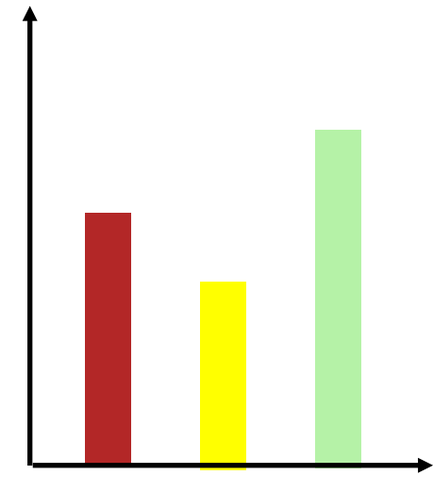
Visual words: main idea





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

Cluster → visual words



Example:

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

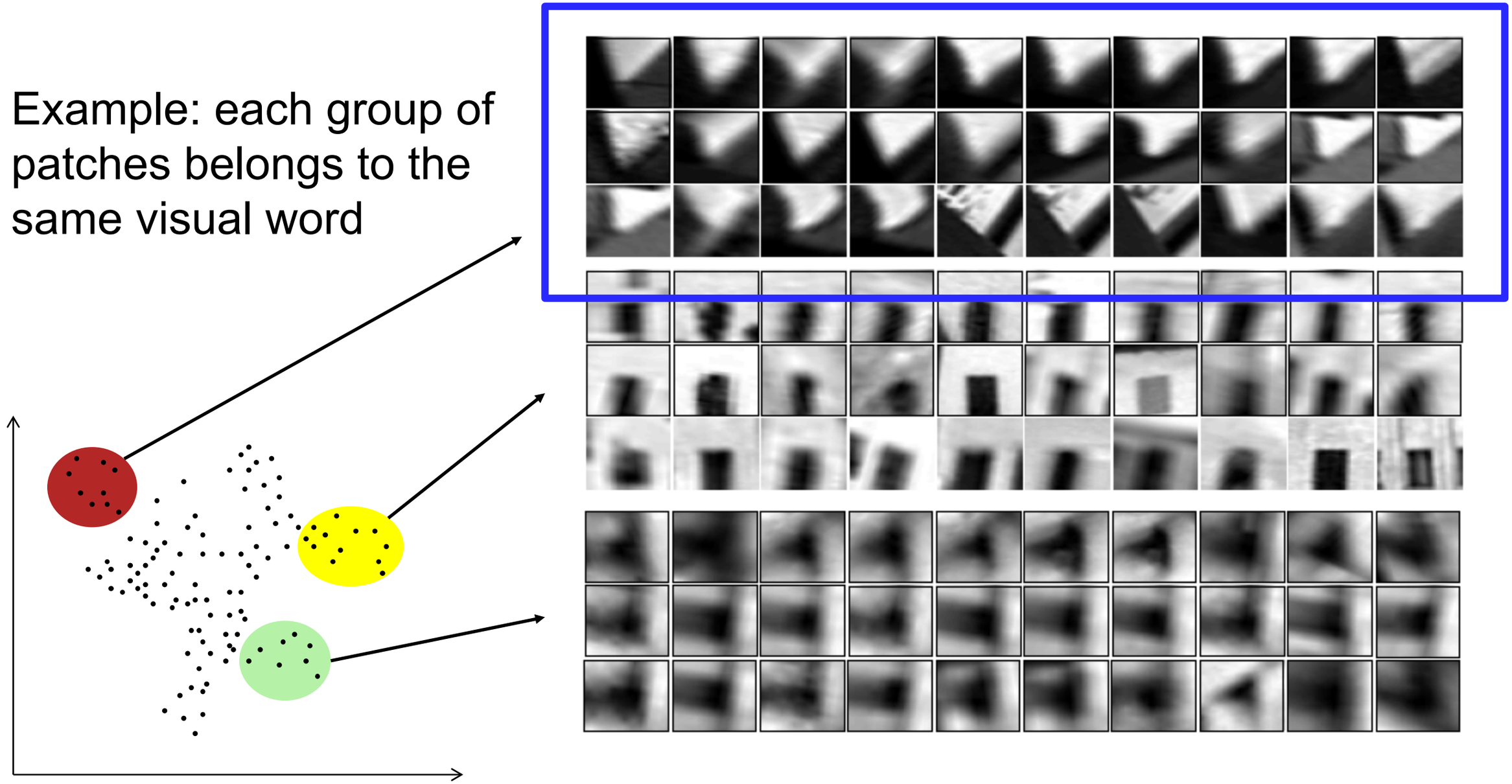


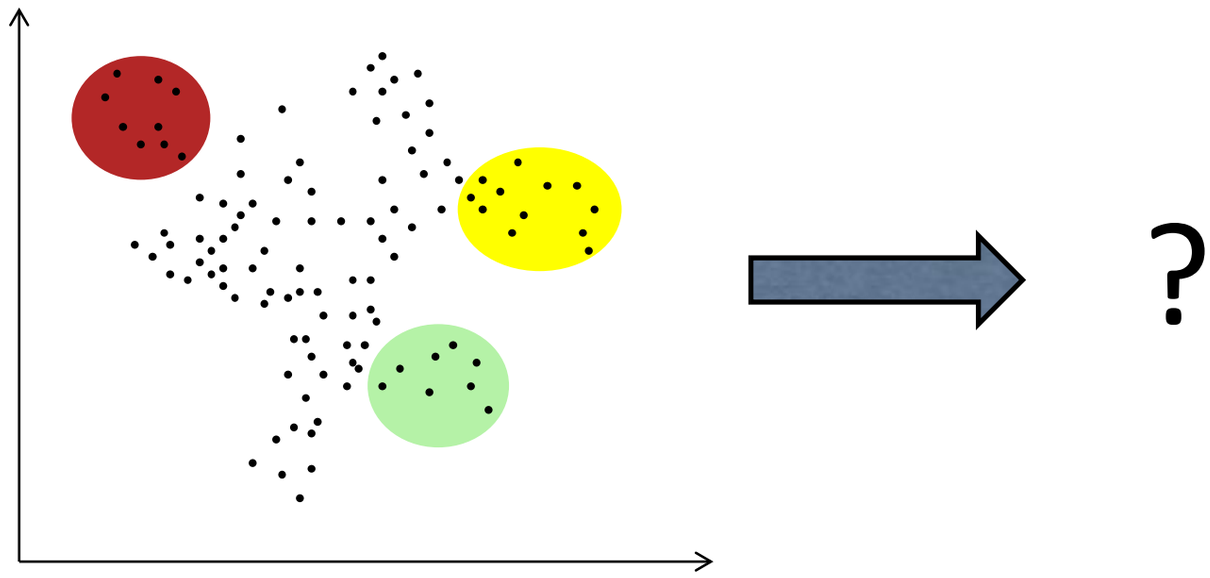
Figure from Sivic & Zisserman, ICCV 2003

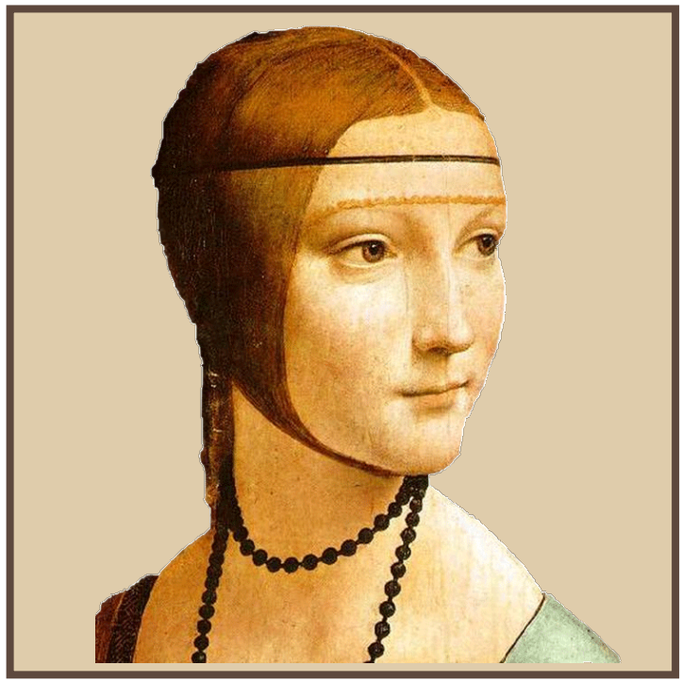
Kristen Grauman

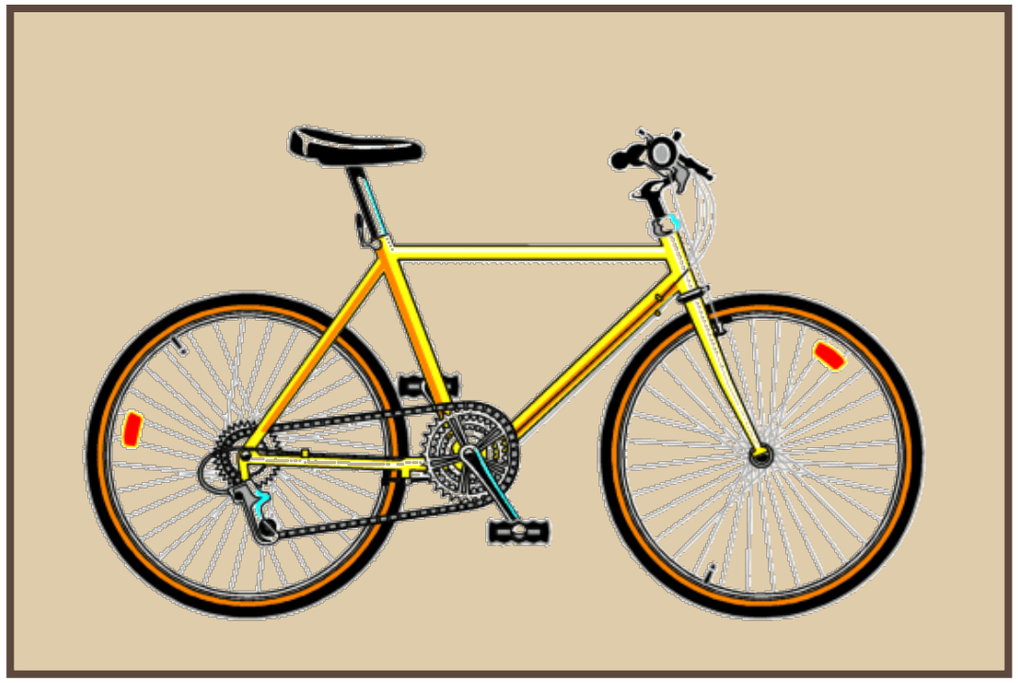
Representing images

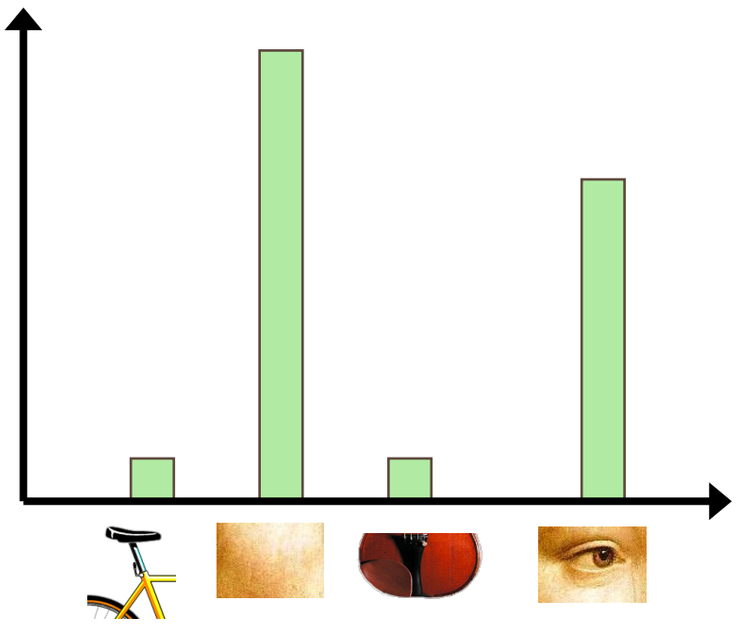
- If a local image region is a visual word, how can we summarize the content of an entire image (the “document” in our case)?

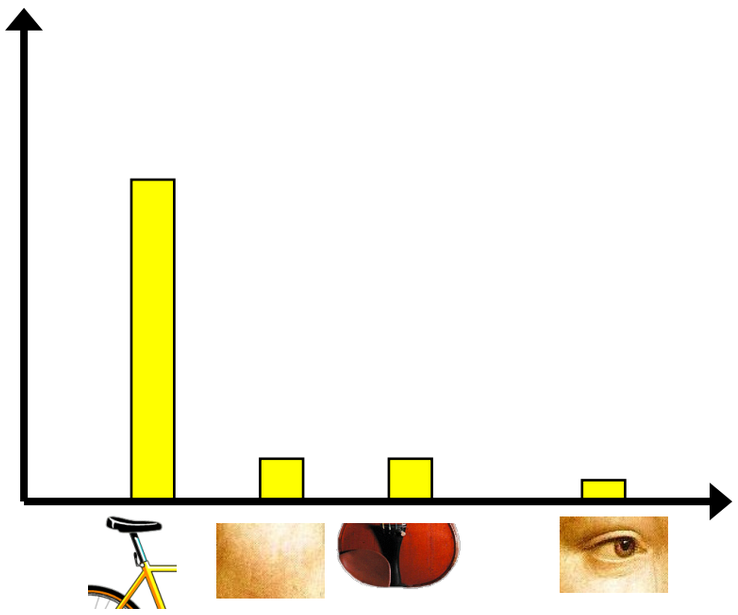
- Answer: build histograms of visual words, so called **codeword histograms!**

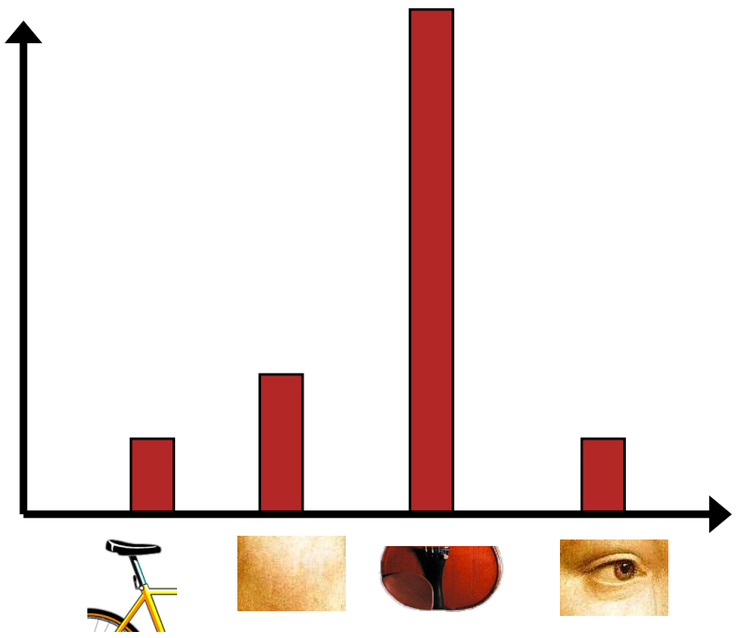






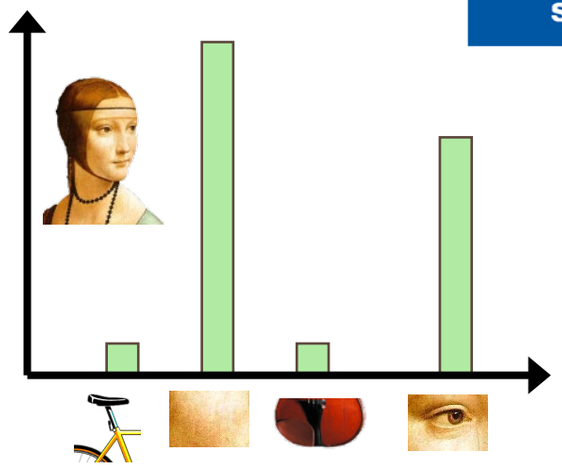


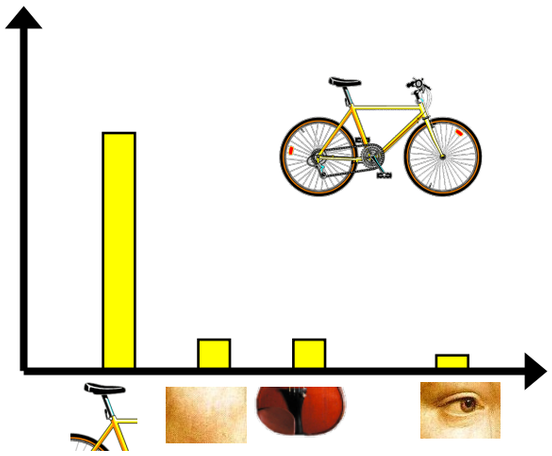


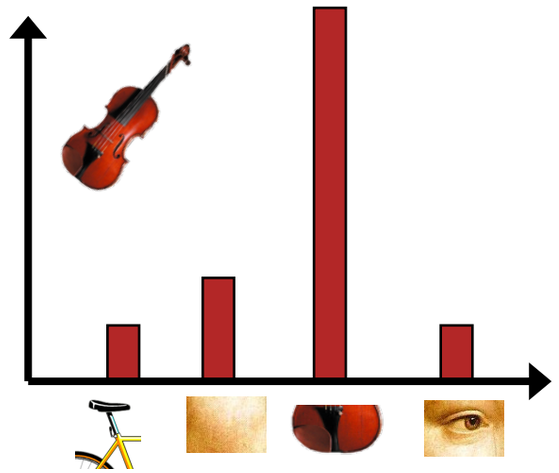


Bags of visual words

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





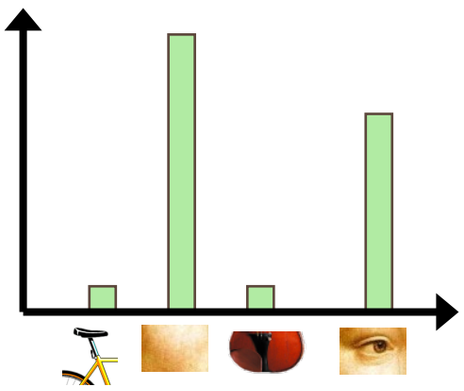


Kristen Grauman

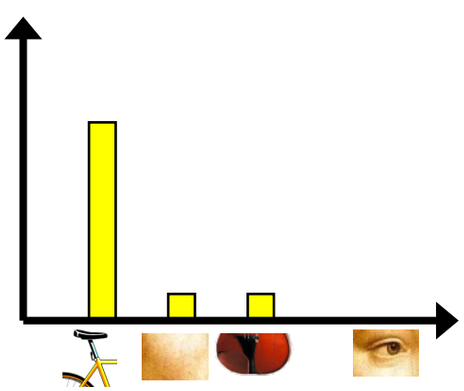
Comparing bags of words

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

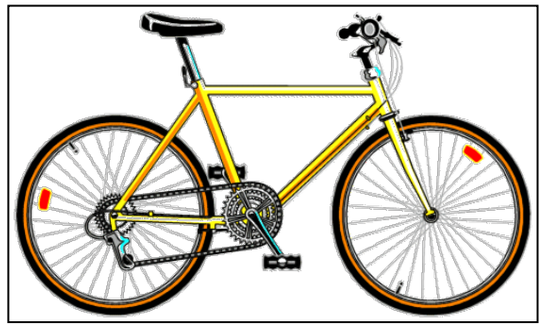
[1 8 1 4]



[5 1 1 0]



$\vec{d}_j$



$\vec{q}$

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

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

for vocabulary of *V* words

Kristen Grauman

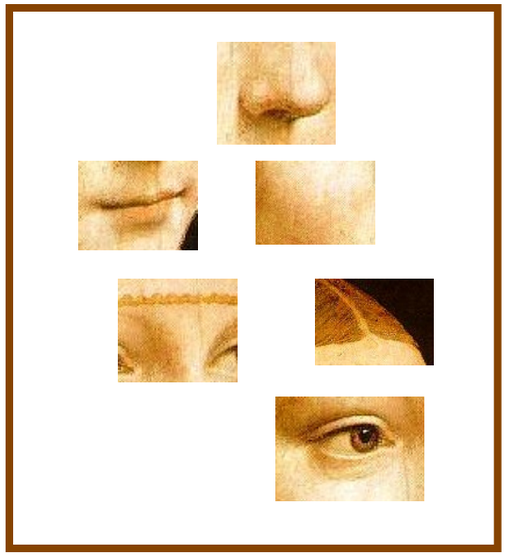
Bag of Visual Words (BoVW) Models

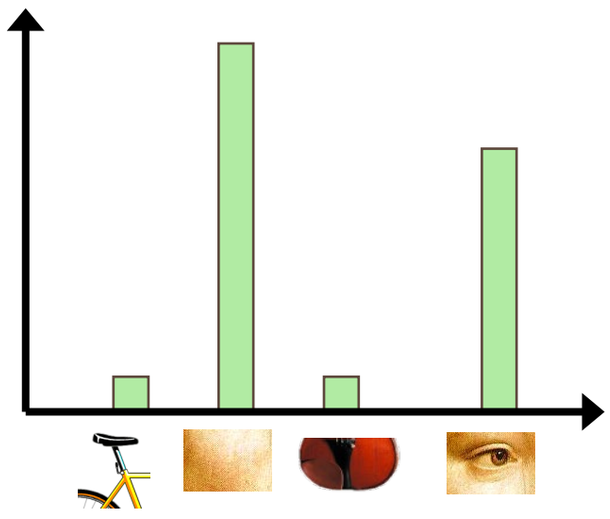
- All of these visual words are treated as being the same

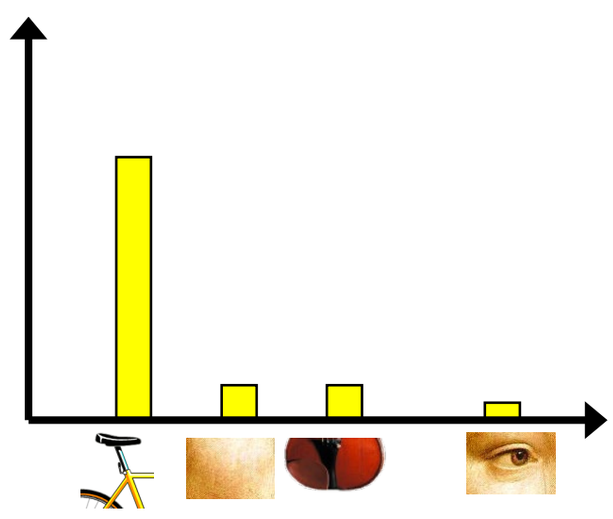
- No distinction between foreground and background
- No distinction of their location and their relations (it's really a bag)
- This is the most simple model!

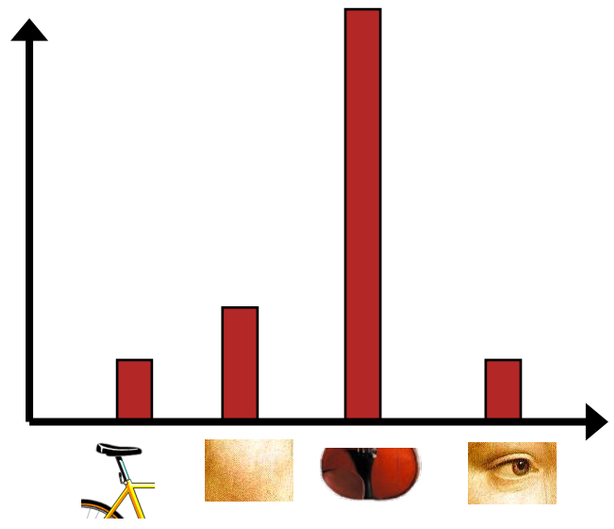
Summary: Bag of Visual Words Construction

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









1. Feature extraction – where to extract features?

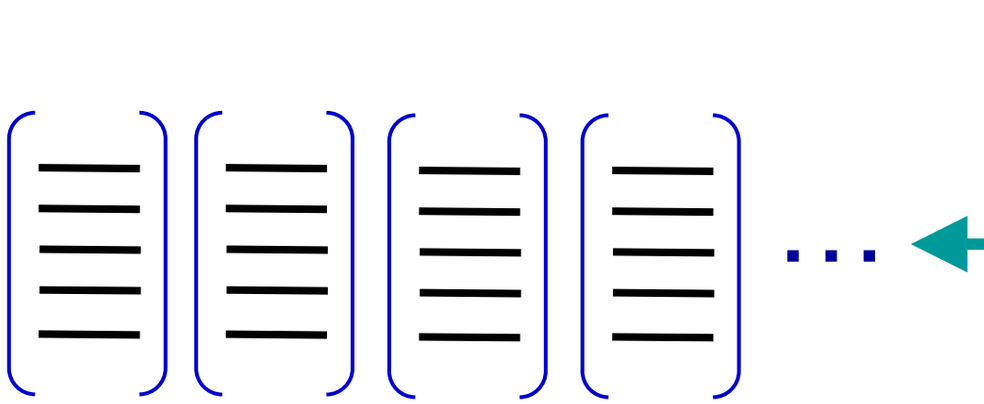
- Interest points (multi-scale) or regular grid (dense sampling)

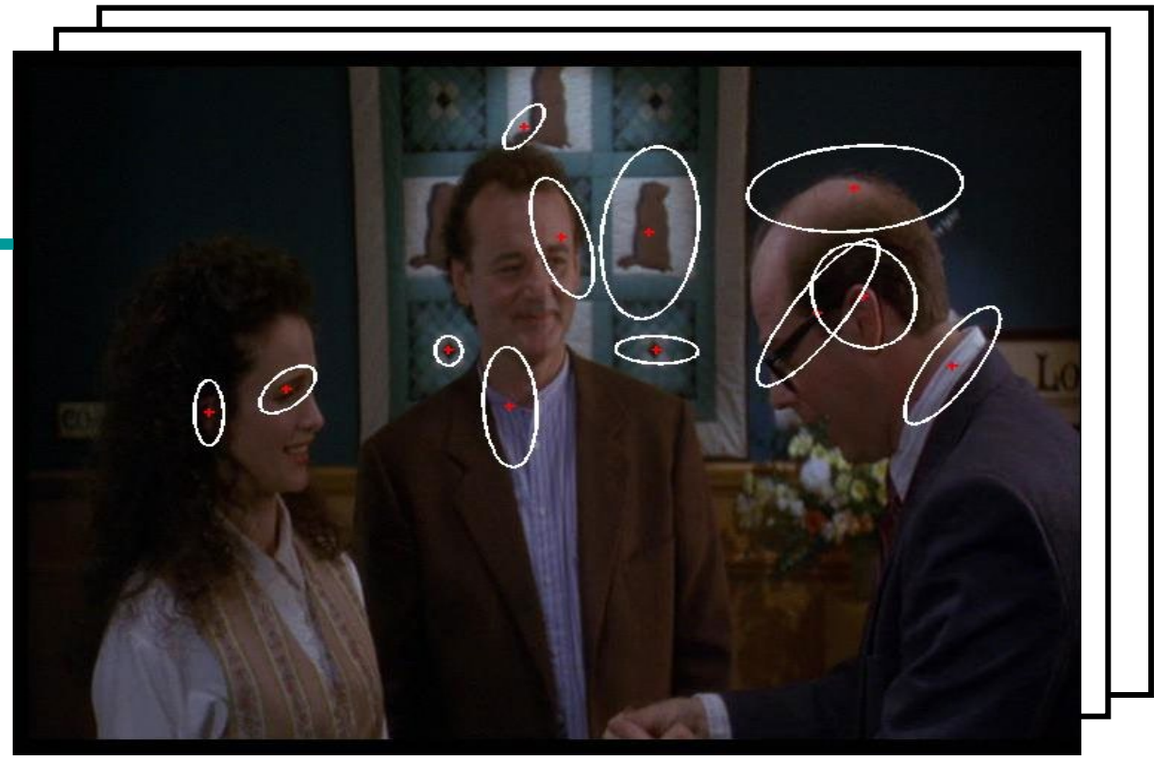
Dense sampling
- Usually only one scale
- Multiple dense samplings with different scales possible

Sparse sampling
- By nature multiscale
- Coverage problem: homo-geneous regions may be missed

Credit: James Hays

1. Feature extraction



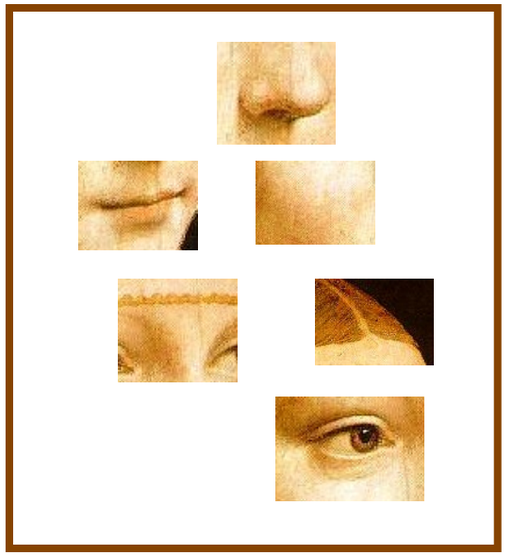


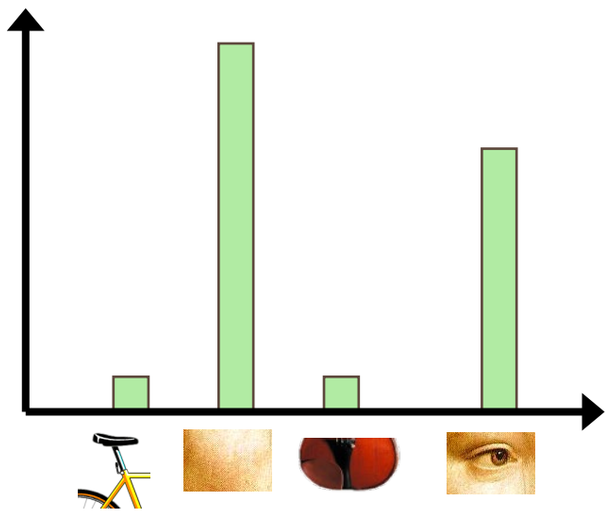
- For each image we get a series of descriptors
- The descriptors describe patches around keypoints in the image
- These patches are considered instances of different visual words
- Collect descriptors from all images and simply put them into our “bag”

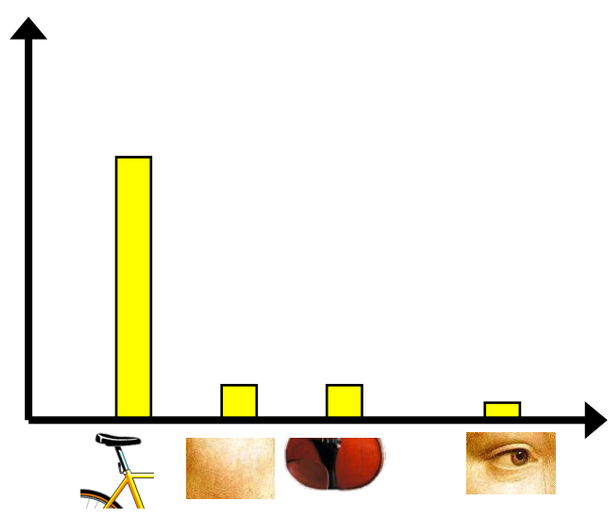
Slide credit: Josef Sivic

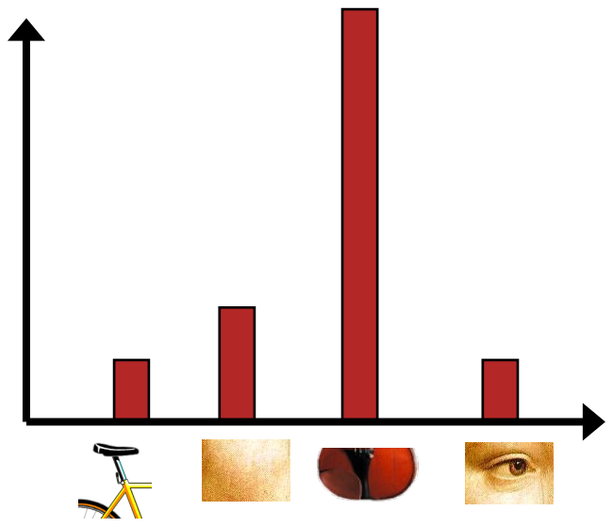
Bag of Visual Words Construction

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

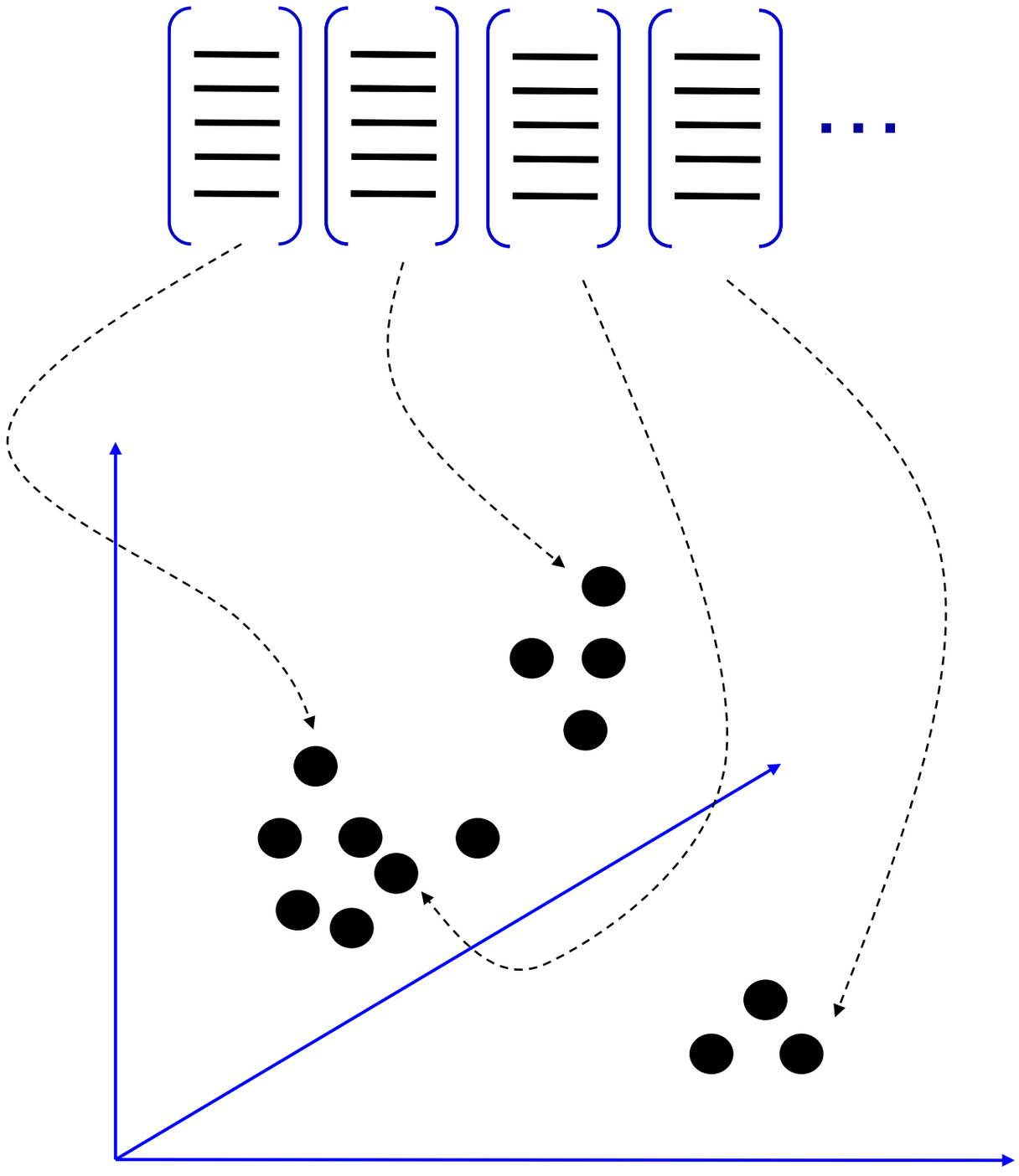








2. Learning the visual vocabulary

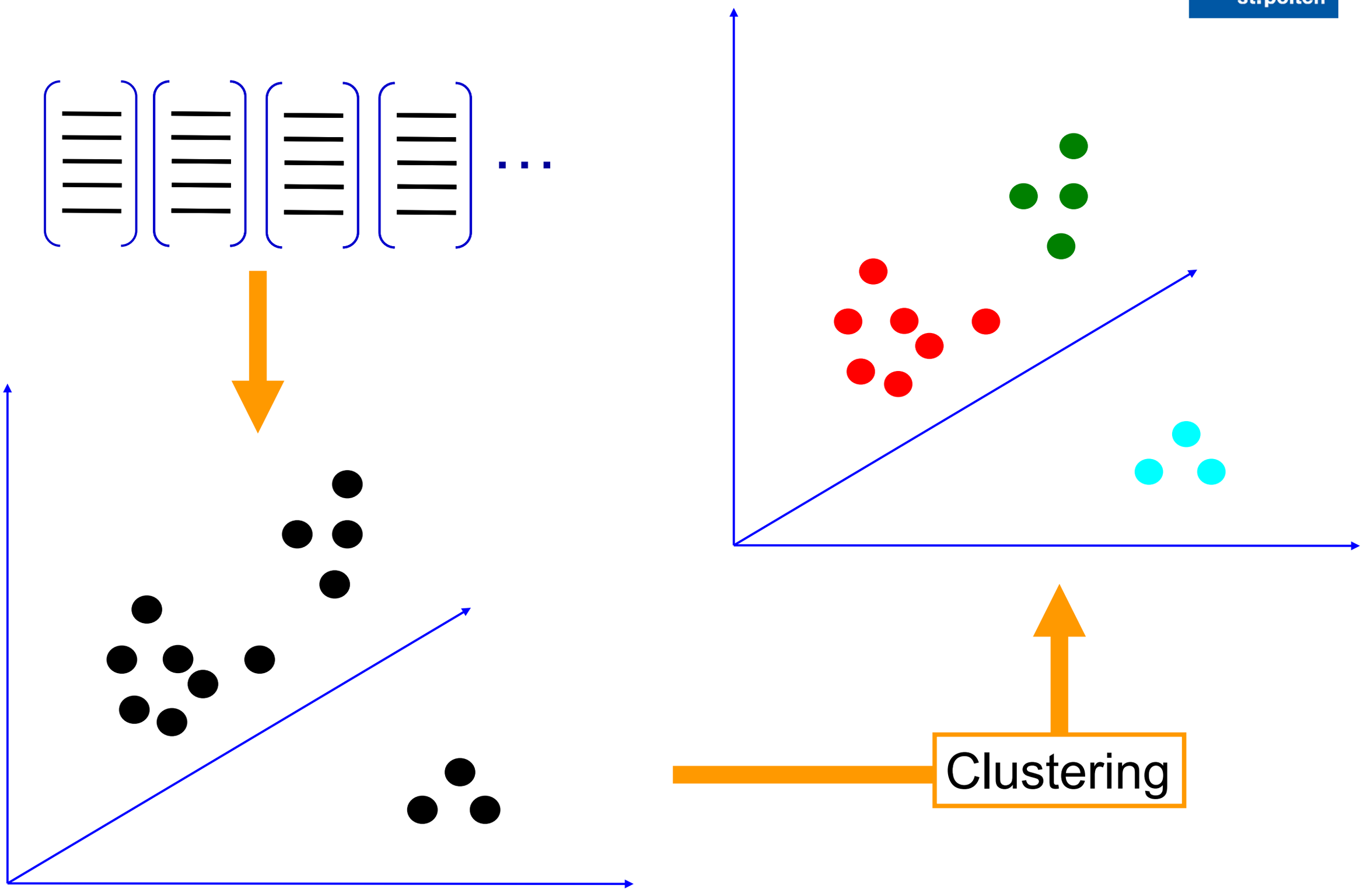


- Each descriptor represents a point in an N dimensional space
- Here: N=3 dimensions
- For SIFT: N=128 dimensions

- Result: a point distribution in an N-dimensional space
- Basic assumption: **similar visual patterns will be represented by nearby points!**

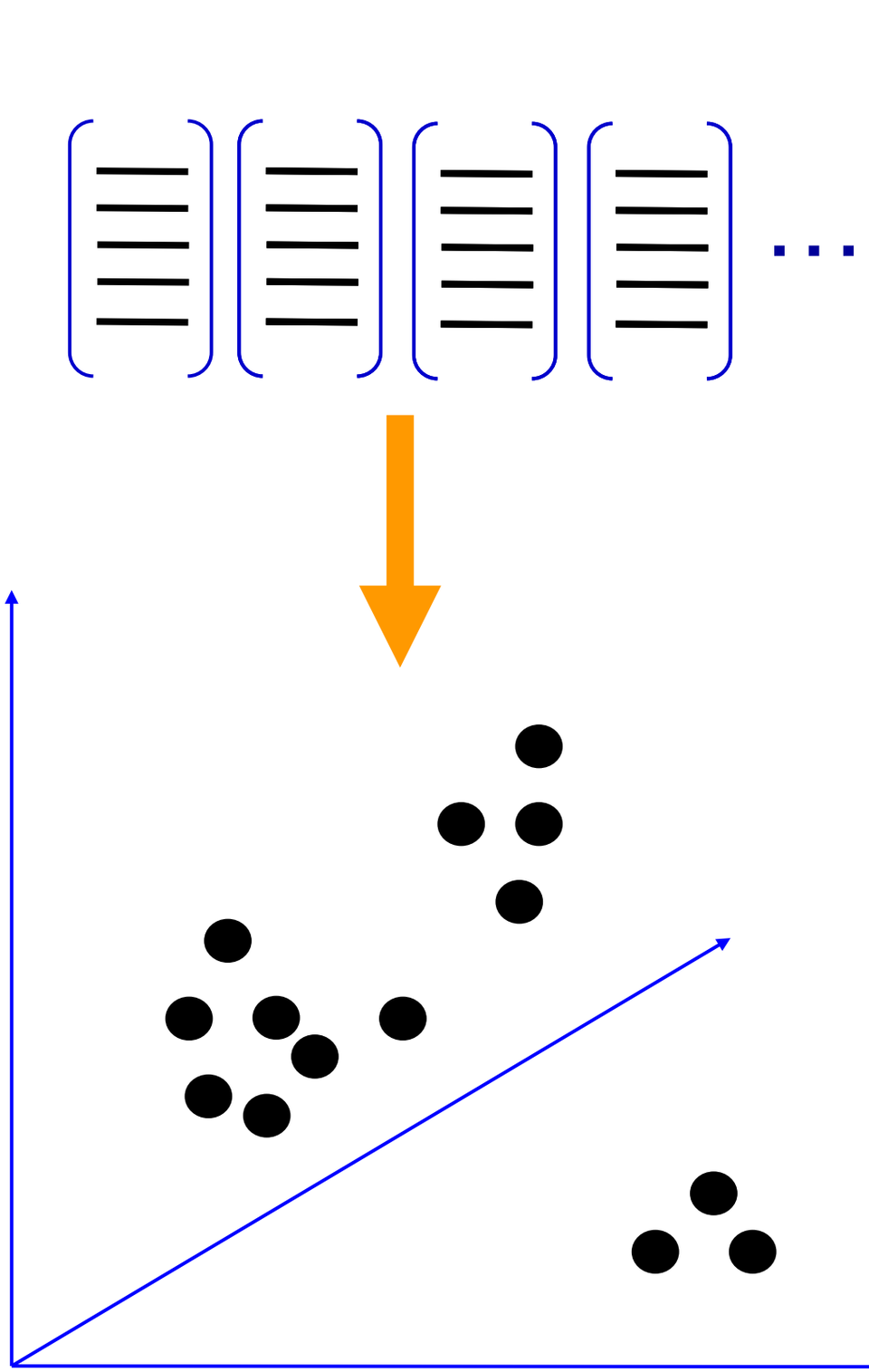
Slide credit: Josef Sivic

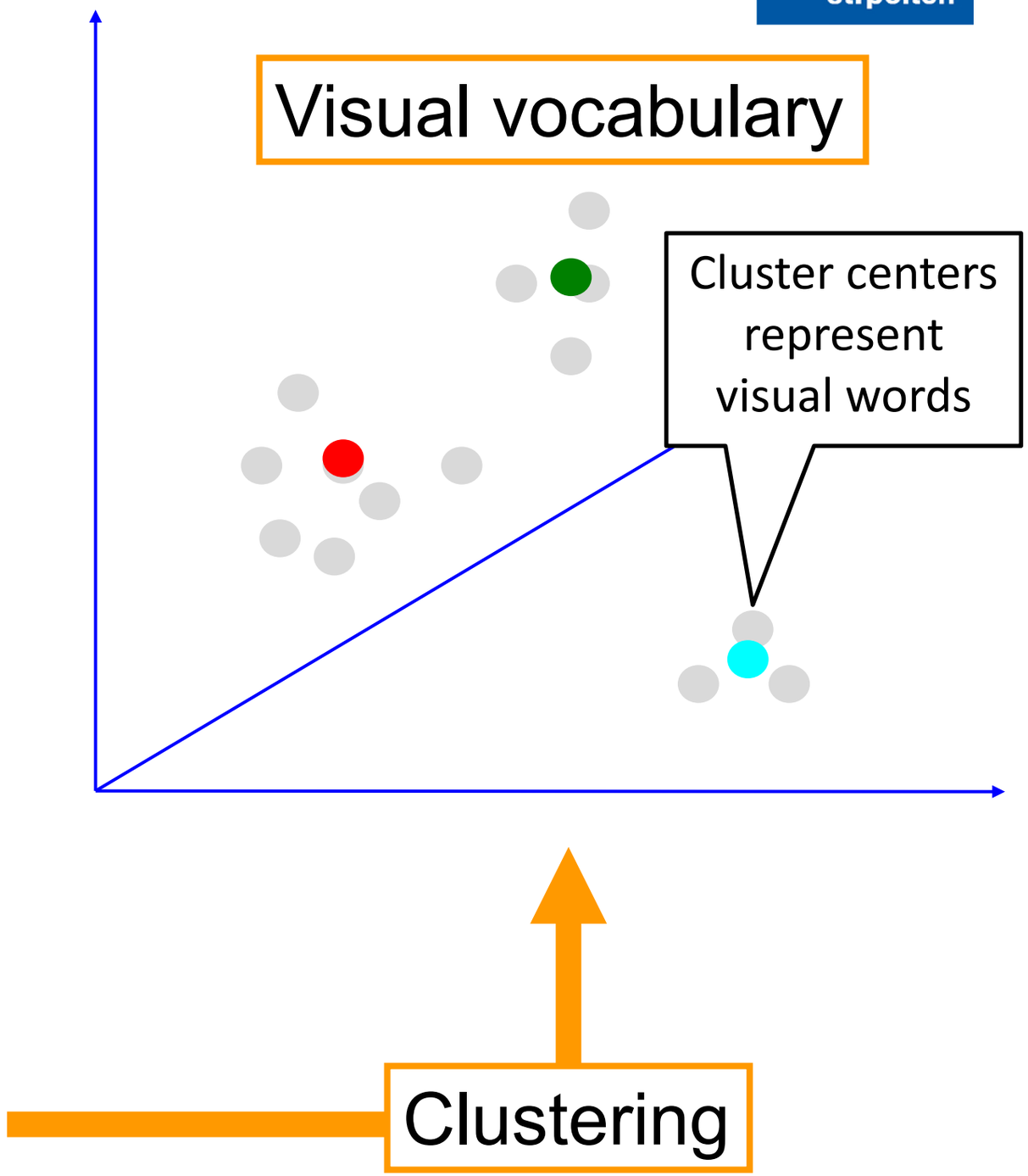
2. Learning the visual vocabulary



Slide credit: Josef Sivic

2. Learning the visual vocabulary

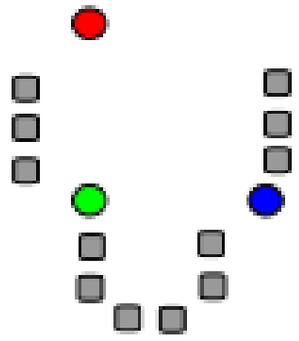




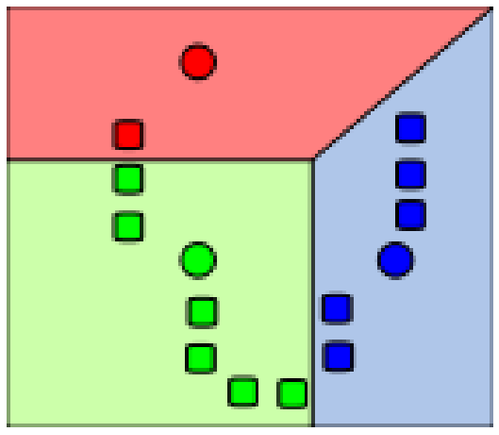
Slide credit: Josef Sivic

K-means algorithm

1. Randomly select K centers



2. Assign each point to nearest center



3. Compute new center (mean) for each cluster

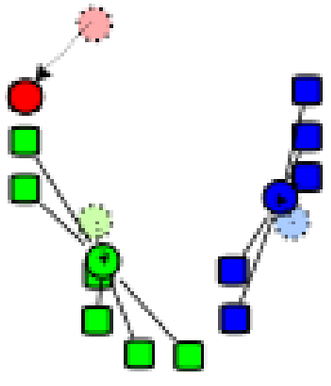
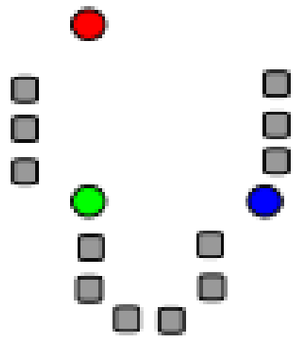


Illustration: http://en.wikipedia.org/wiki/K-means_clustering

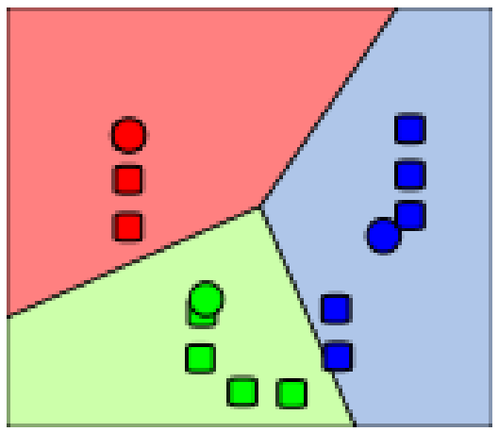
Credit: James Hays

K-means algorithm

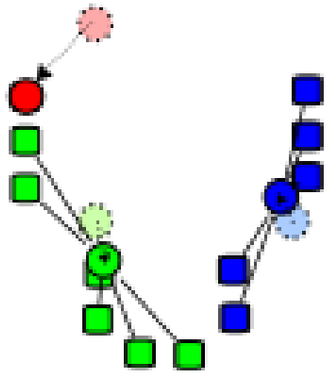
1. Randomly select K centers



2. Assign each point to nearest center



3. Compute new center (mean) for each cluster



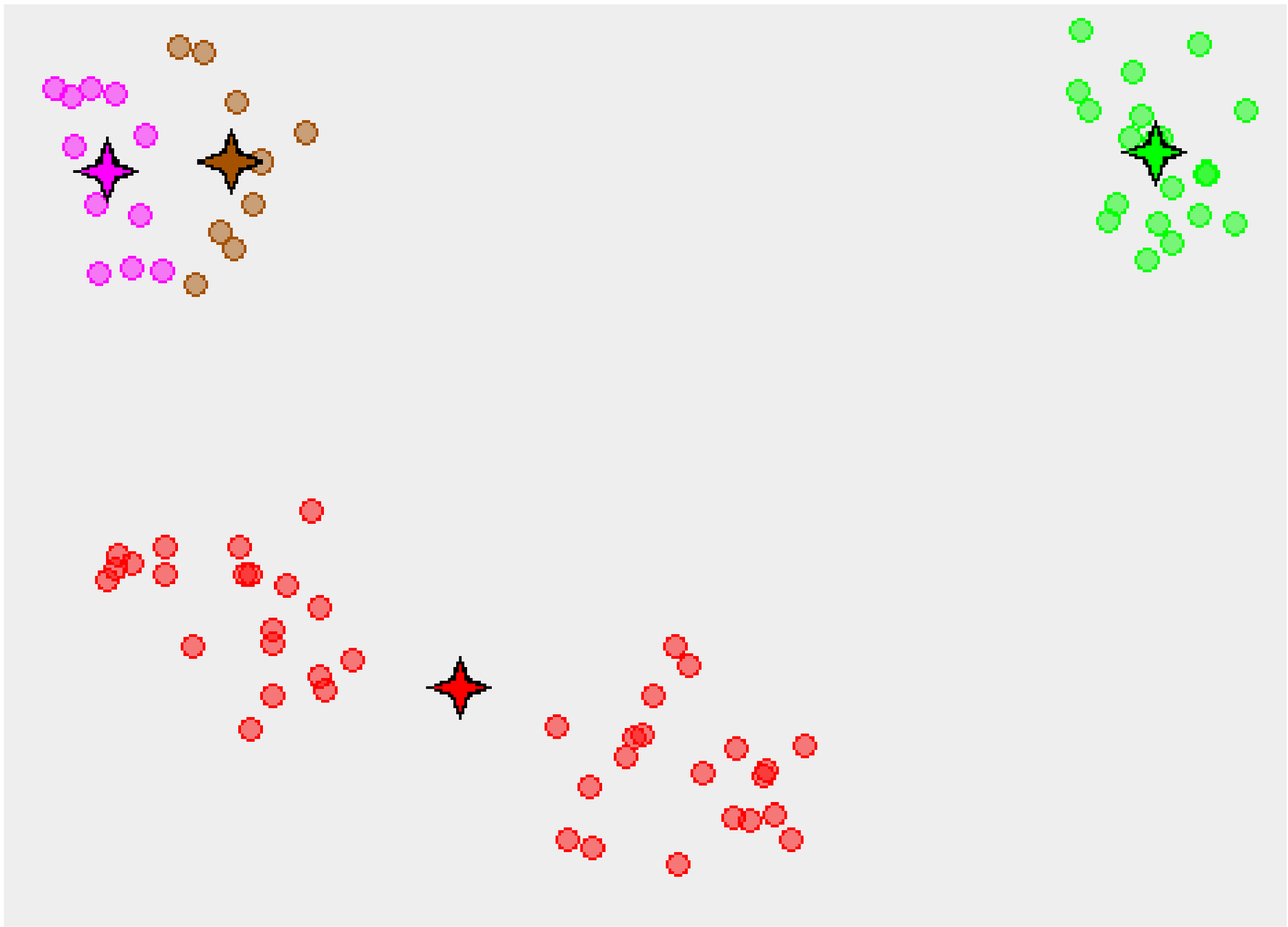
Back to 2

- Repeat until no points are re-assigned any more → clusters are stable

Illustration: http://en.wikipedia.org/wiki/K-means_clustering

Credit: James Hays

K-means converges to a local minimum



Credit: James Hays

K-means: design choices

- Initialization
  - Randomly select K points as initial cluster center
  - Or greedily choose K points to minimize residual
- Distance measures
  - Traditionally Euclidean, could be others
- Optimization
  - Will converge to a *local minimum*
  - May want to perform multiple restarts

Credit: James Hays

How to choose the number of clusters?

- Validation set
  - Try different numbers of clusters and look at performance
    - When building dictionaries (discussed later), more clusters typically work better

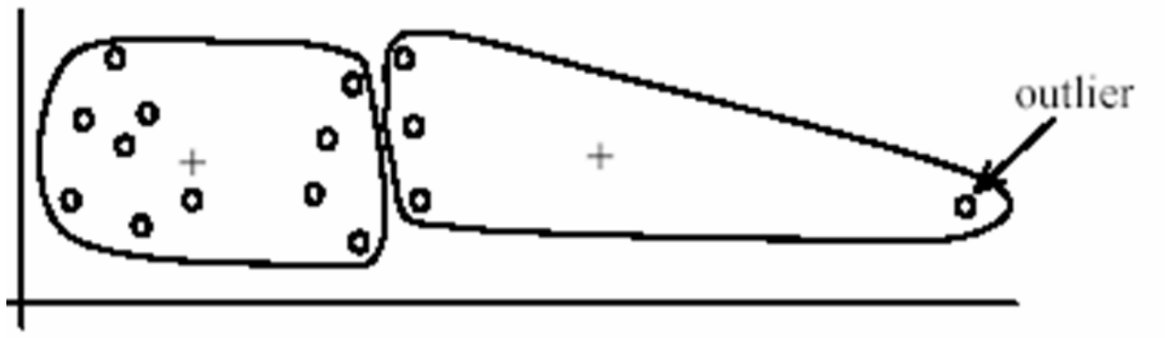
Slide: Derek Hoiem

K-Means pros and cons

- Pros
  - Finds cluster centers that minimize conditional variance (good representation of data)
  - Simple and fast*
  - Easy to implement

(B): Ideal clusters

- Cons
  - Need to choose K
  - Sensitive to outliers
  - Prone to local minima



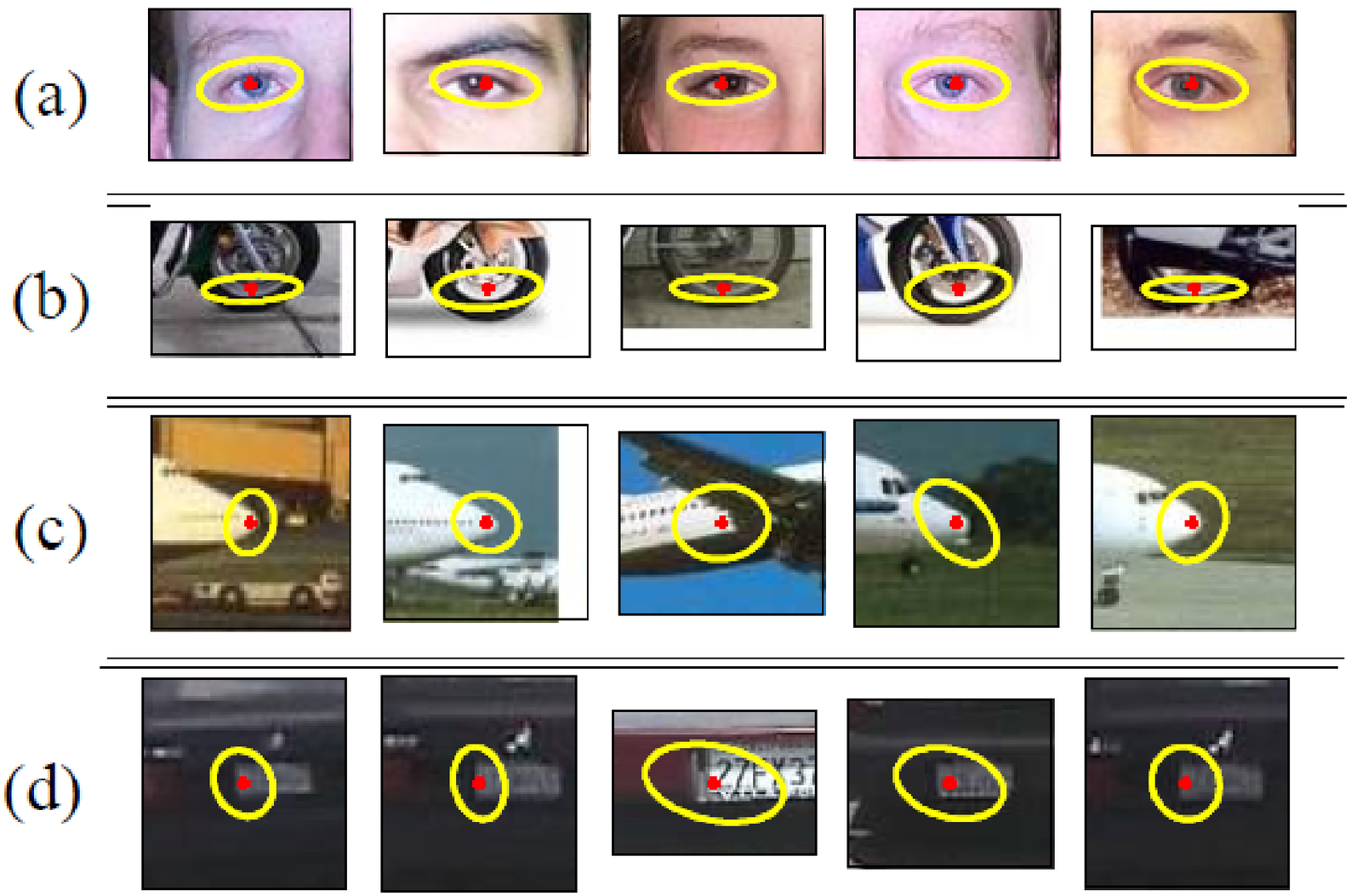
  - All clusters have the same parameters (e.g., distance measure is non-adaptive)
  - *Can be slow: each iteration is O(KNd) for N d-dimensional points

Clustering for Vocabulary Generation

- 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”

Credit: James Hays

Examples of learned codewords

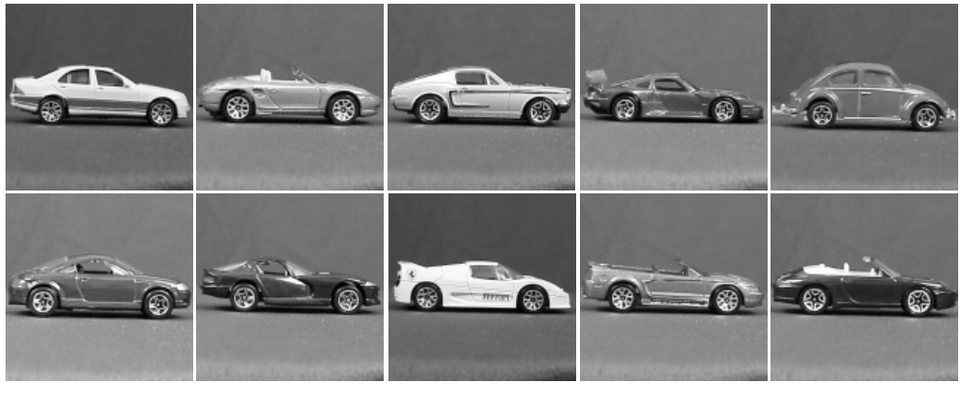


Most likely codewords for 4 learned “topics”

http://www.robots.ox.ac.uk/~vgg/publications/papers/sivic05b.pdf Sivic et al. ICCV 2005

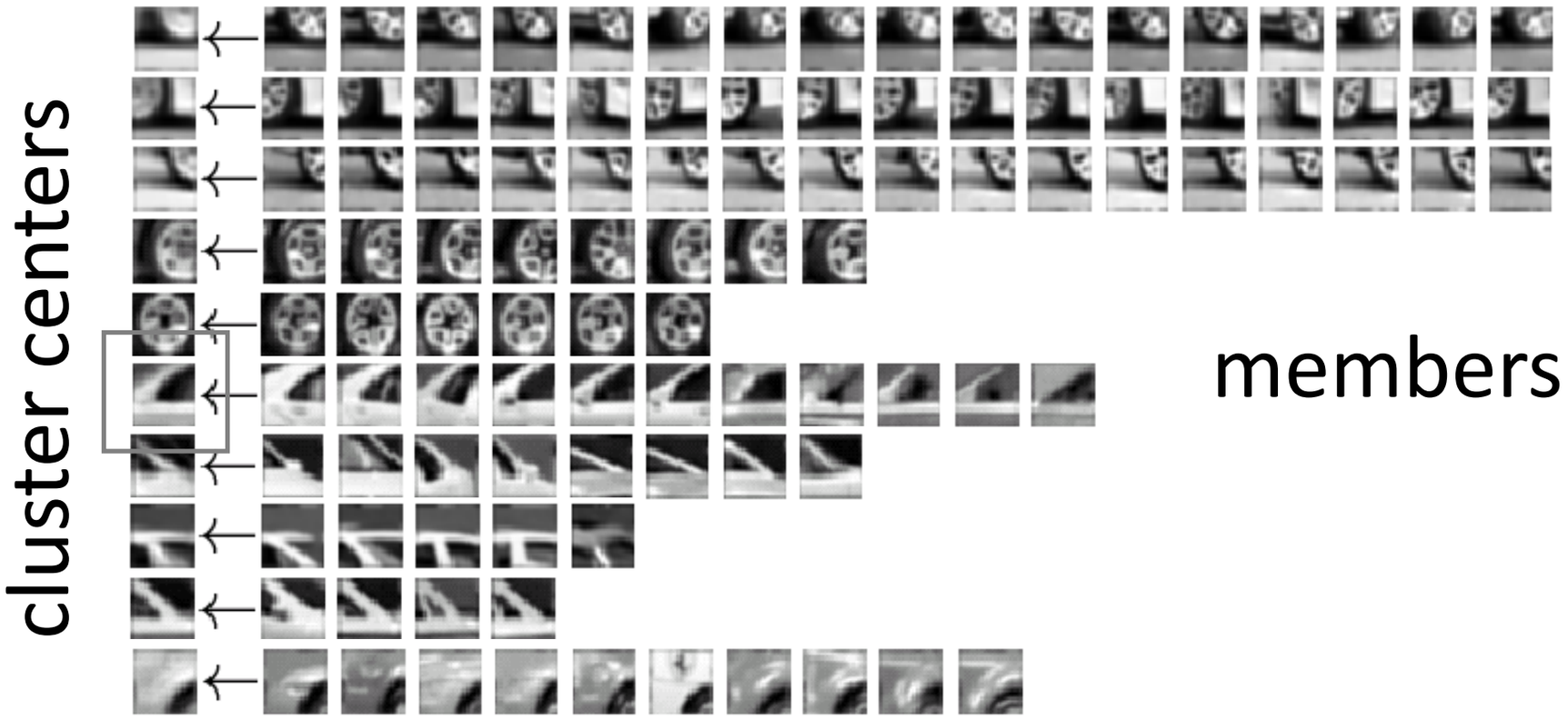
Example Visual Vocabulary

training data



patches around local features

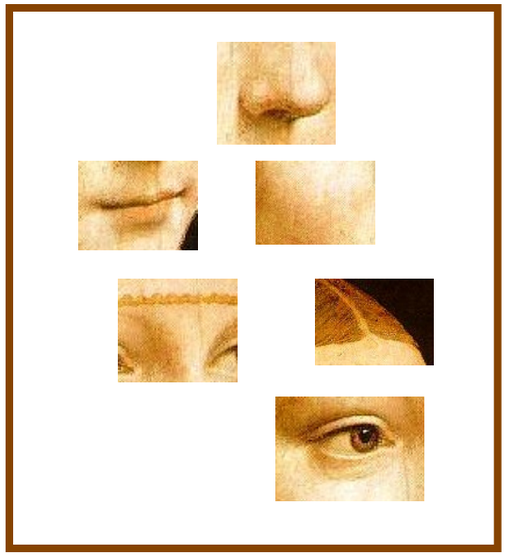
patch to visual word assignment (clustering)

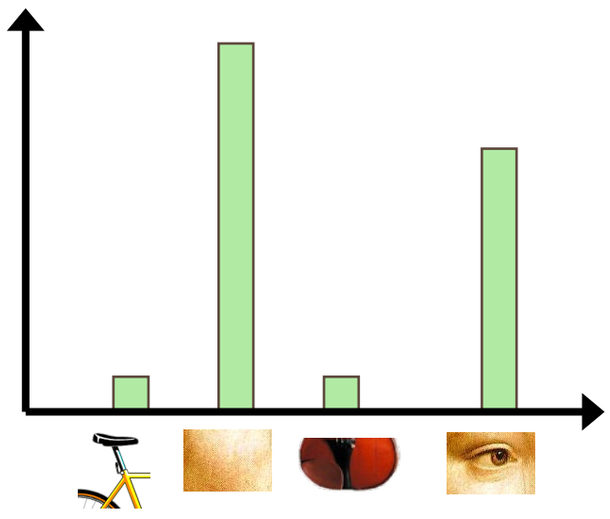


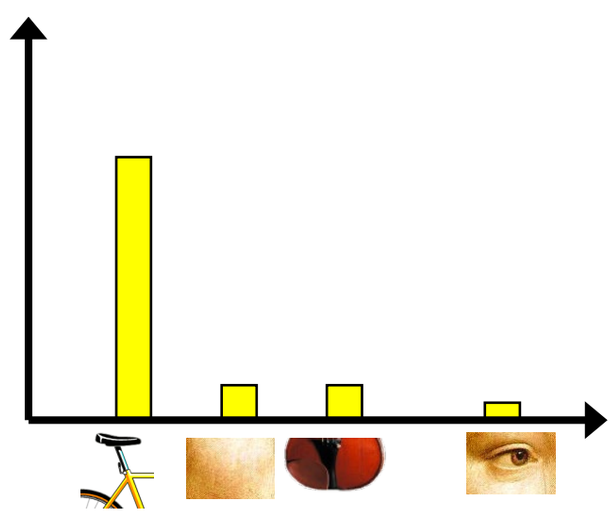
Source: B. Leibe

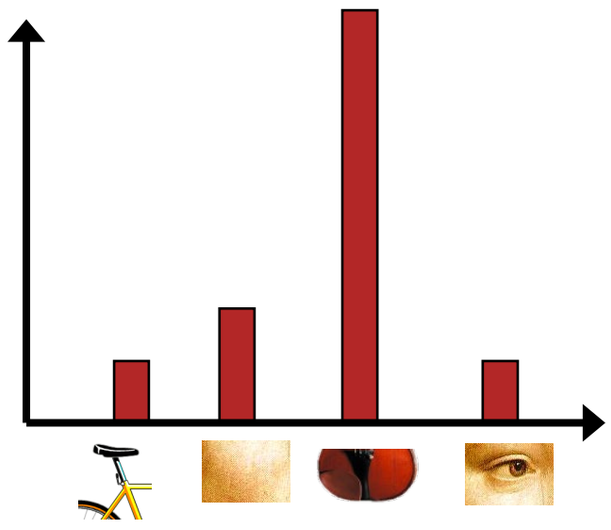
Bag of Visual Words Construction

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



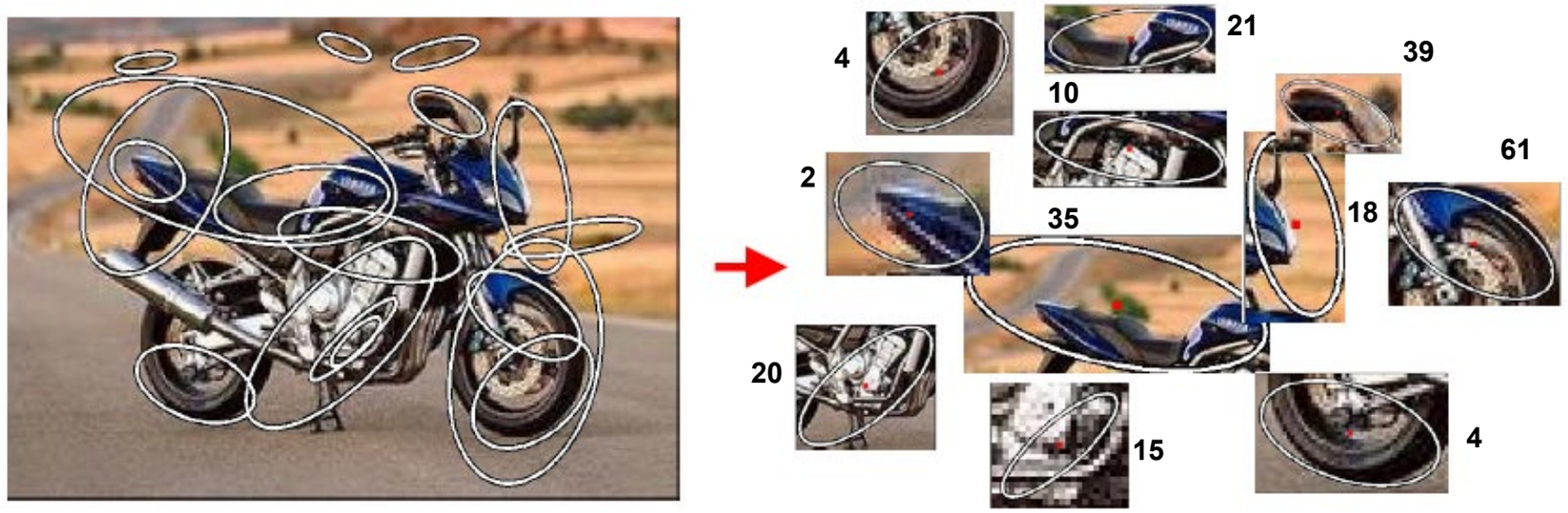






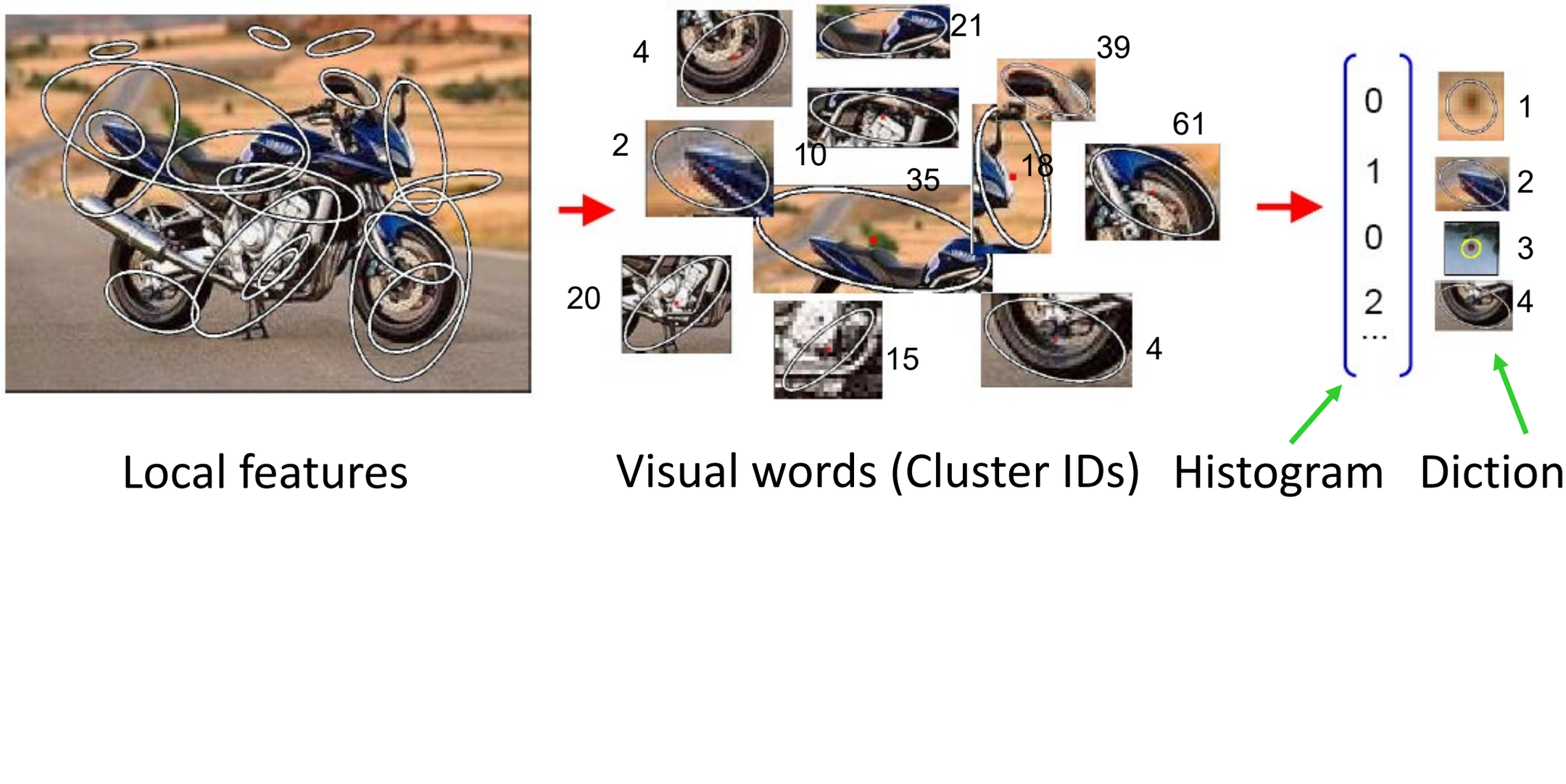
Quantization

- The visual vocabulary is used for quantizing features
  - Takes a feature vector (e.g. a SIFT descriptor from a local feature point) and map it to the **index of the nearest visual word (=cluster ID) in the vocabulary**
  - Frequent synonyms:
    - Codebook = visual vocabulary
    - Codevector = visual word



Local feature → Cluster IDs

Generation of Codeword Histograms

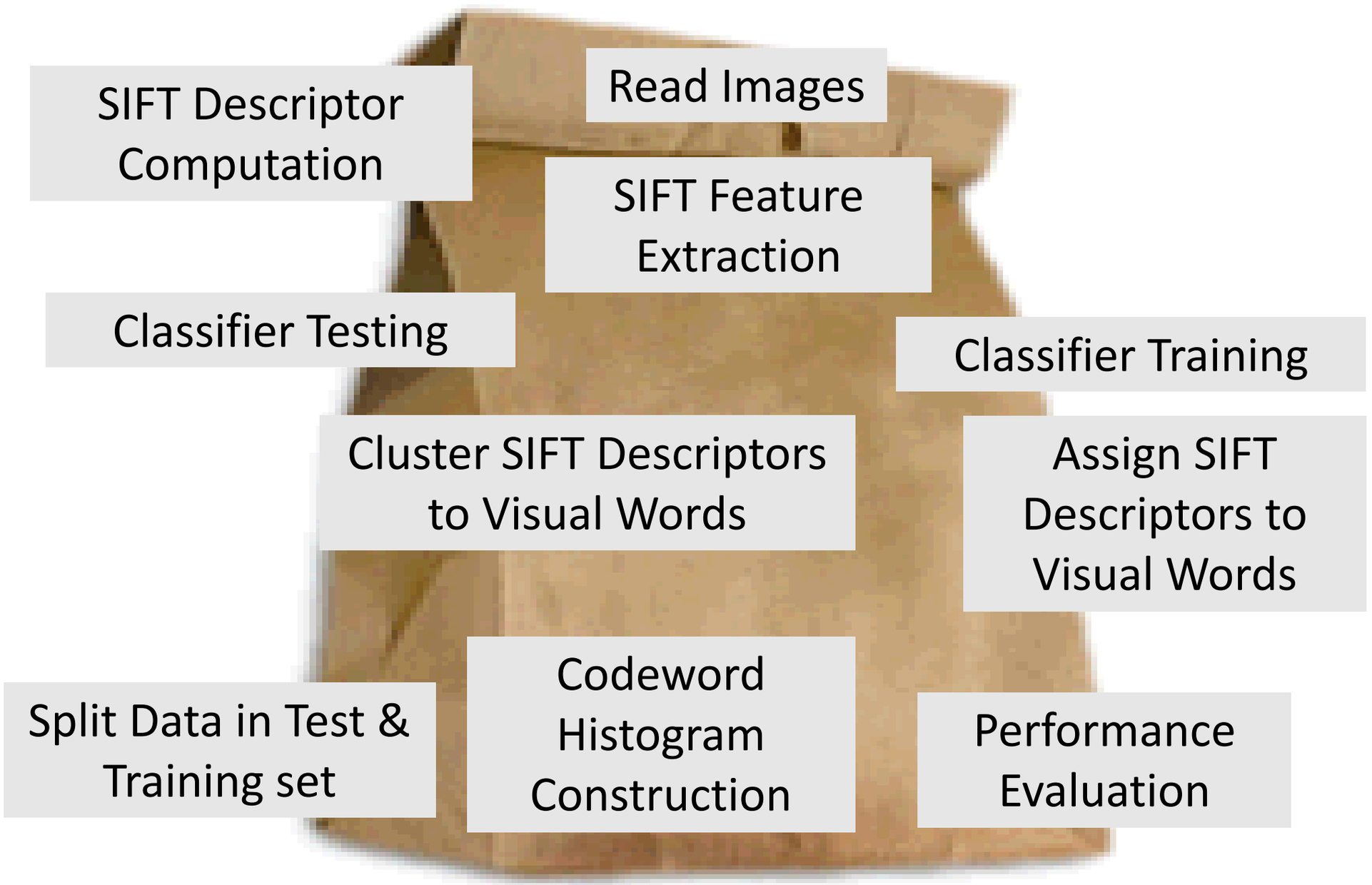


Local features Visual words (Cluster IDs) Histogram Dictionary

Slide credit: Bryan Russell & Josef Sivic

The “Bag of Words” Bag of Words Game:

Sketch the BoW Processing Pipeline for **training and testing** from the components below (components can be used multiple times):



The “Bag of Words” Bag of Words Game: Solution

Sketch the BoW Processing Pipeline from the components below:

Read Images

SIFT Feature Extraction

SIFT Descriptor Computation

Split Data in Test & Training set

Cluster SIFT descriptors to Visual Words

Assign SIFT Descriptors to Visual Words

Assign SIFT Descriptors to Visual Words

Codeword Histogram Construction

Codeword Histogram Construction

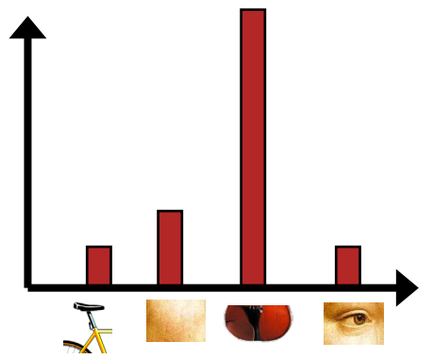
Classifier Training

Classifier Testing

Performance Evaluation

Object Recognition with BoW

Every image is represented by a codebook vector → how to recognize objects from this information?



Object Recognition with BoVW

- **Represent images by frequencies of “visual words”**

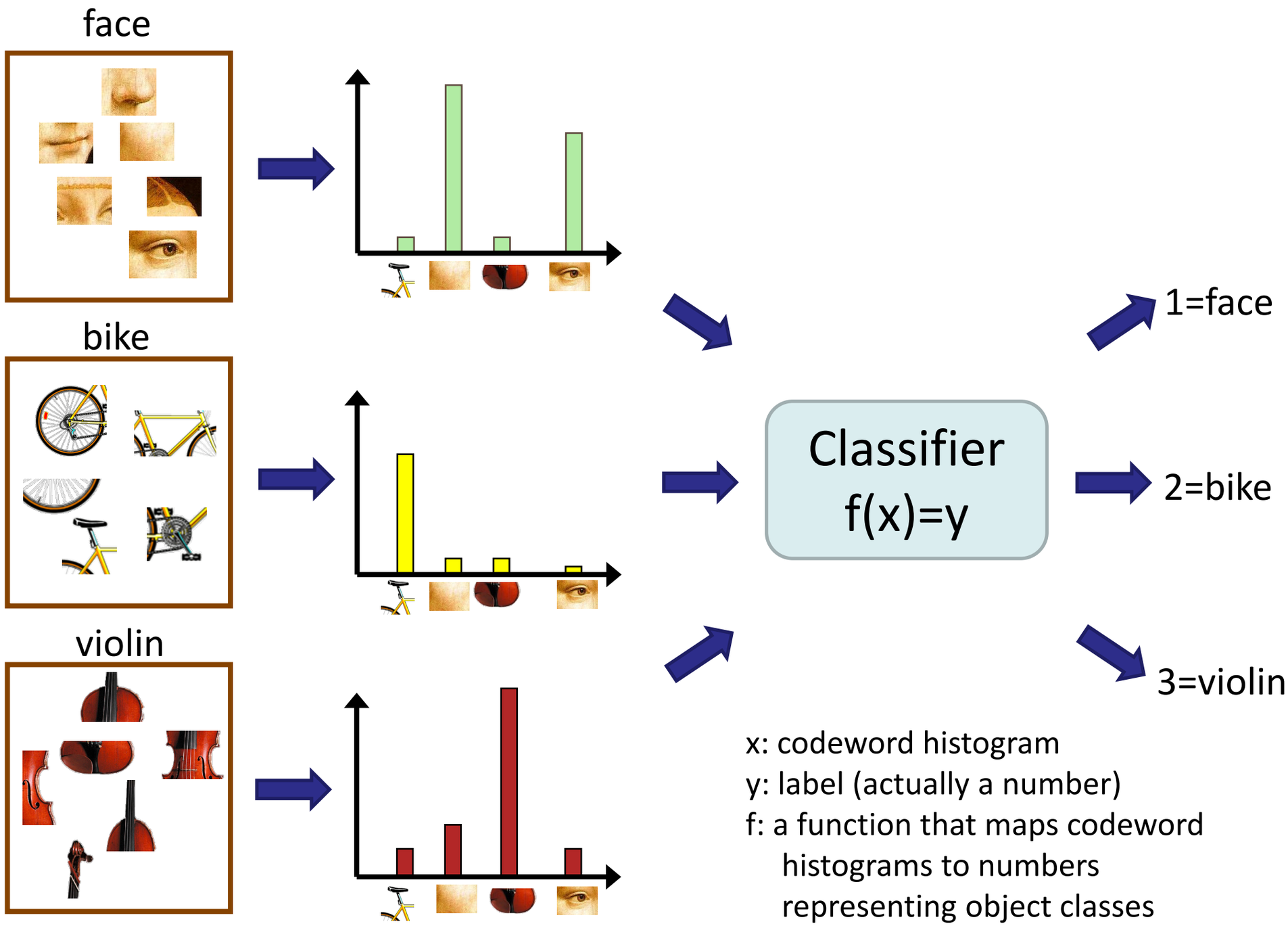
- Each image (or part of image) can be represented by a histogram of codewords
- Histograms have **FIXED number of elements**
- Images can be compared directly via their histograms → **matching**

  → Can be used for similarity search

  → And for object classification or image classification

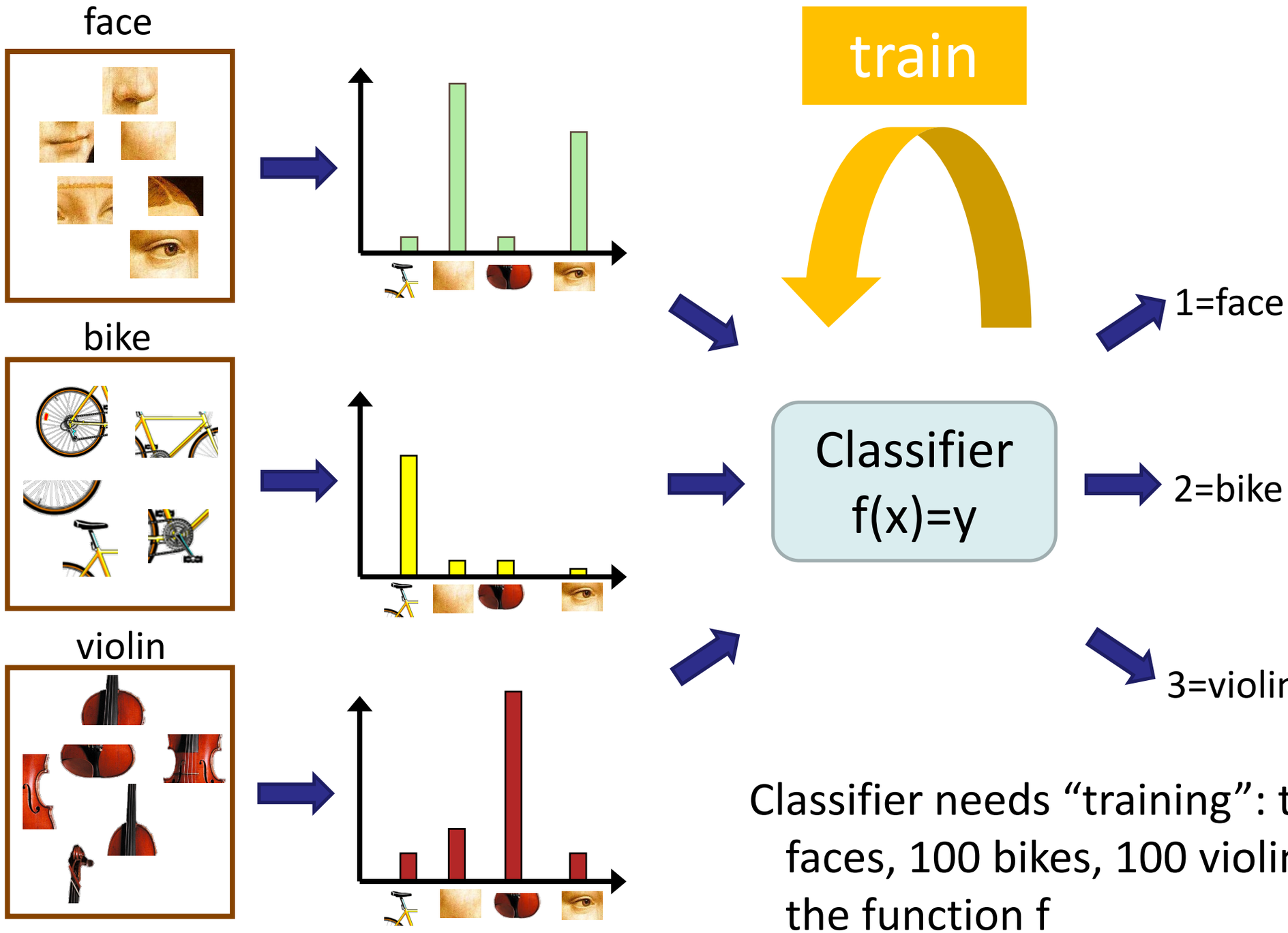
Object Recognition

- Train a classifier to predict image labels



Object Recognition

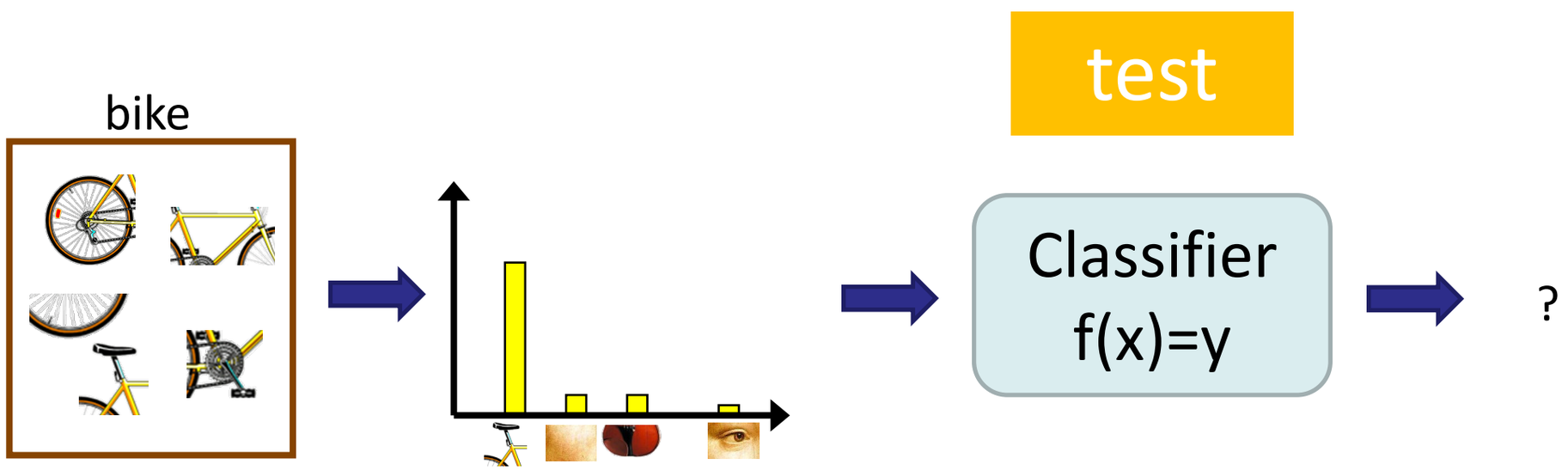
- Train a classifier to predict image labels



Classifier needs “training”: take 100 faces, 100 bikes, 100 violins to learn the function f

Object Recognition

- Use the trained classifier to predict image labels:



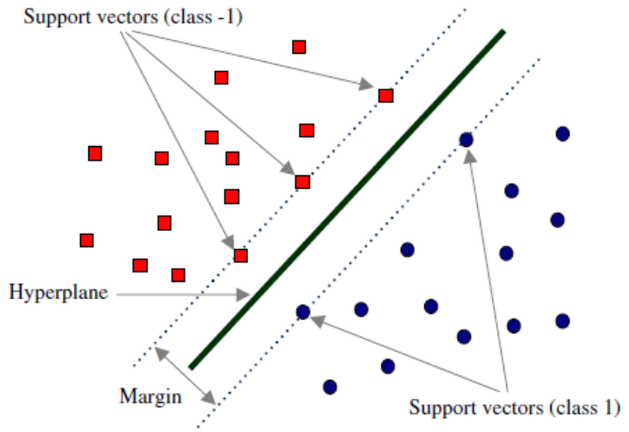
1. Take image, compute codeword histogram x
2. Apply classifier, i.e. compute f(x)
3. Result y is the id of the **predicted class**

Classifiers

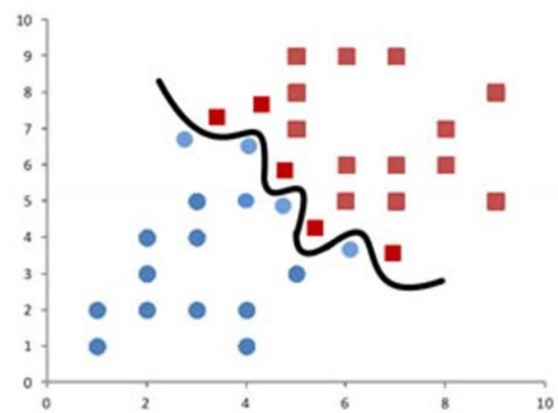
- Numerous classifiers exist
- The idea is always the same: f(x)=y
- The function f can look very different

Classifier
f(x)=y

- Example: Support Vector Machine
  - linear classifier
  - find a line that separates the classes in the feature space



- Example: a non-linear classifier
  - more complex (e.g. polynomial function separates the classes



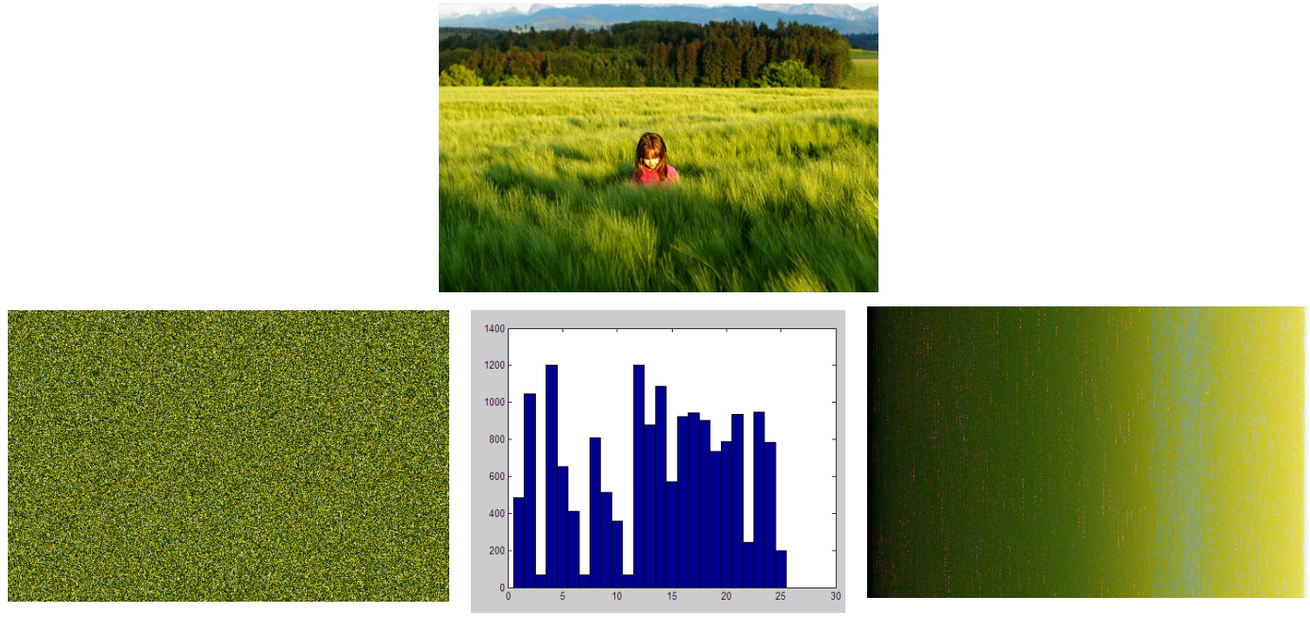
BoVW – Important Issues

- Select:
  - Sampling strategy (where to extract features): dense / sparse
  - Select clustering method: e.g. K-Means
  - Select K, the number of visual words (usually more is better)
- What about location information? There is no location in a Bag
  - Is this a problem?
    - Spatial BoVW

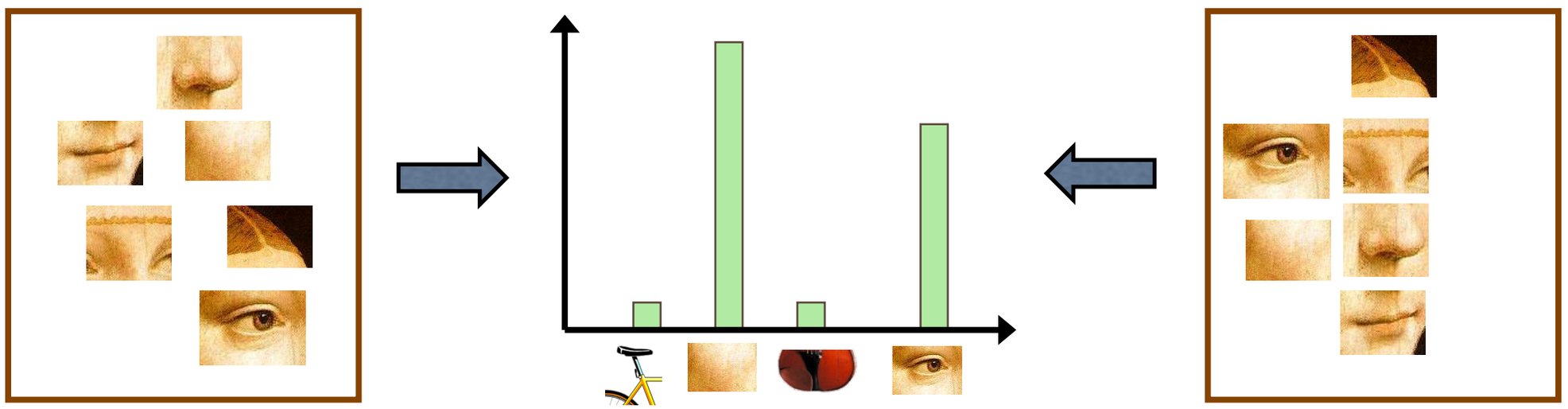
Kristen Grauman

Is throwing away spatial information a good idea?

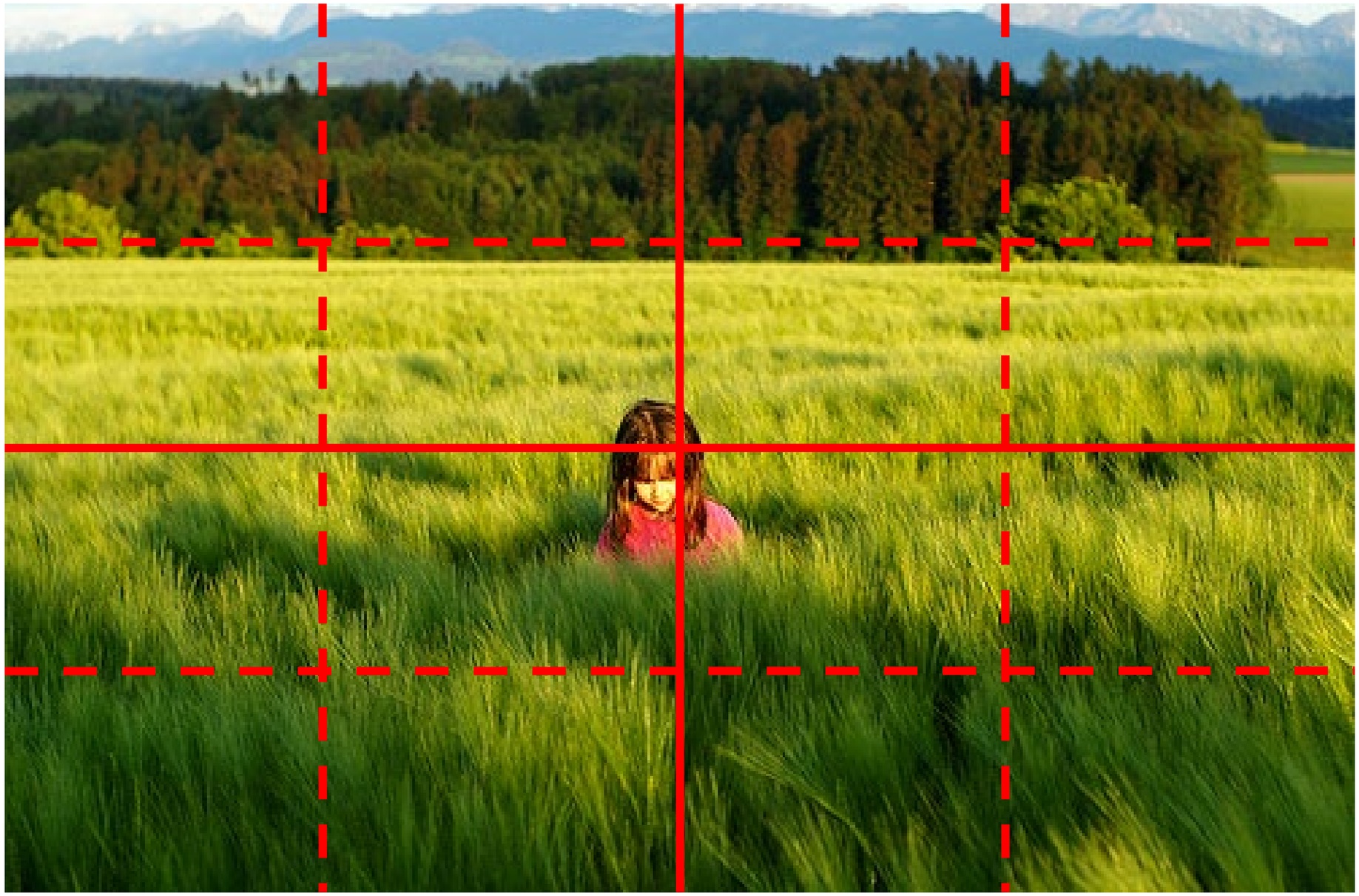
- In histograms we loose location information: all these images have the same color histogram



- The same applies for codeword histograms:



Spatial Pyramid

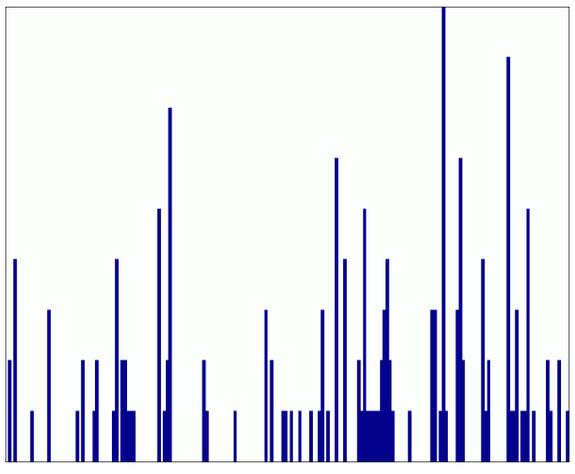


Compute histogram in each spatial bin!

Credit: James Hays

Spatial Pyramid Representation

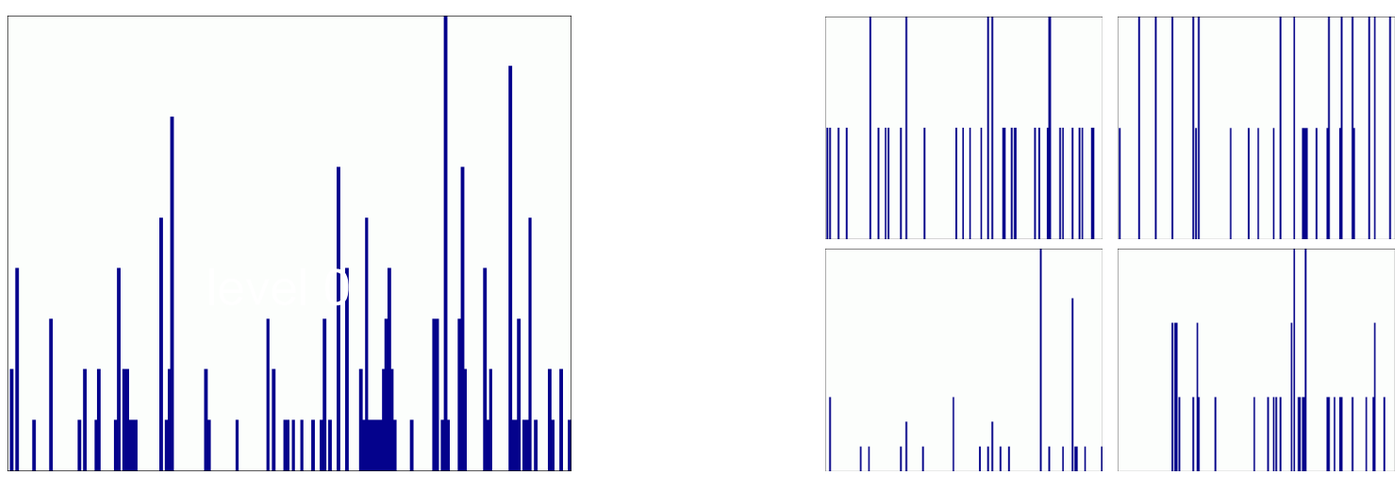
- Extension of a bag of features
- Locally orderless representation at several levels of resolution



Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramid representation

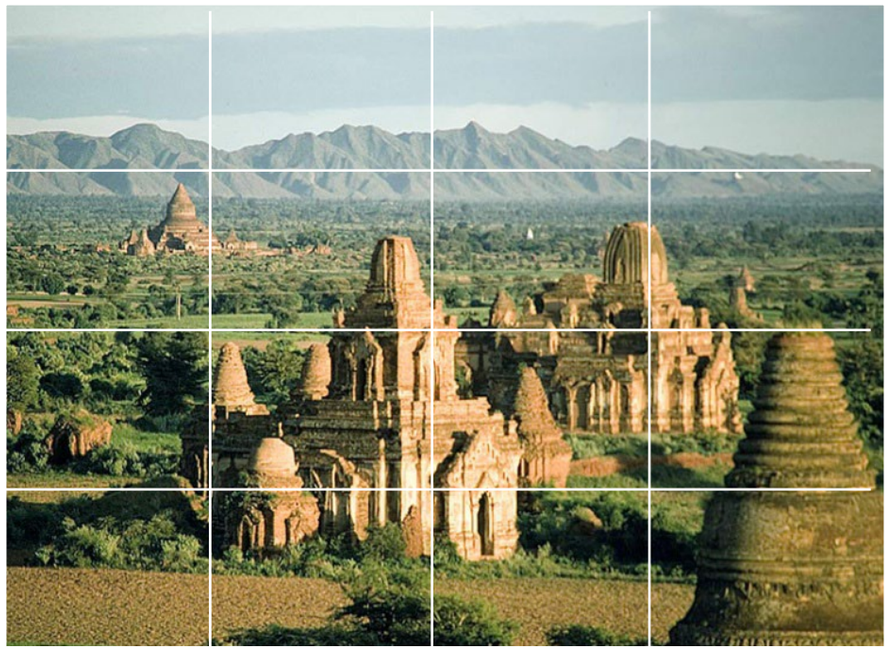
- Extension of a bag of features
- Locally orderless representation at several levels of resolution

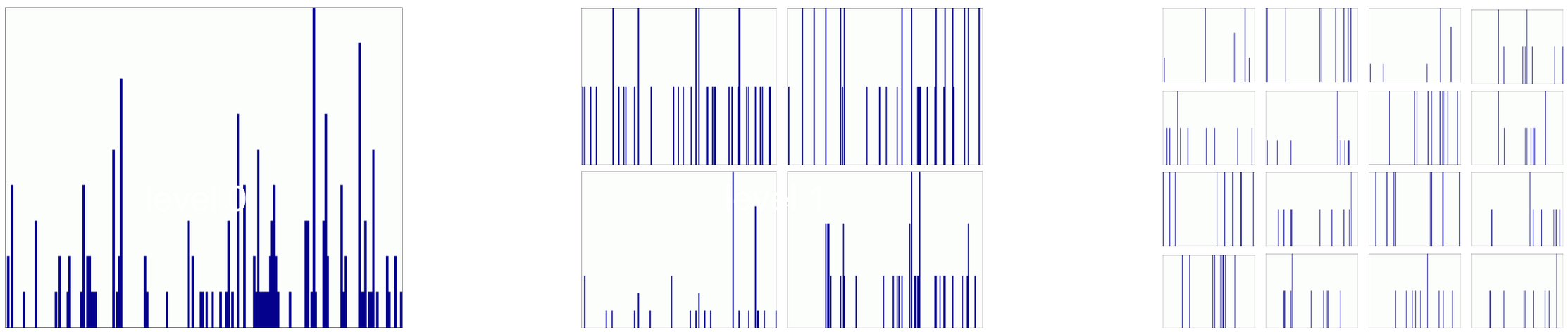


Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramid representation

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

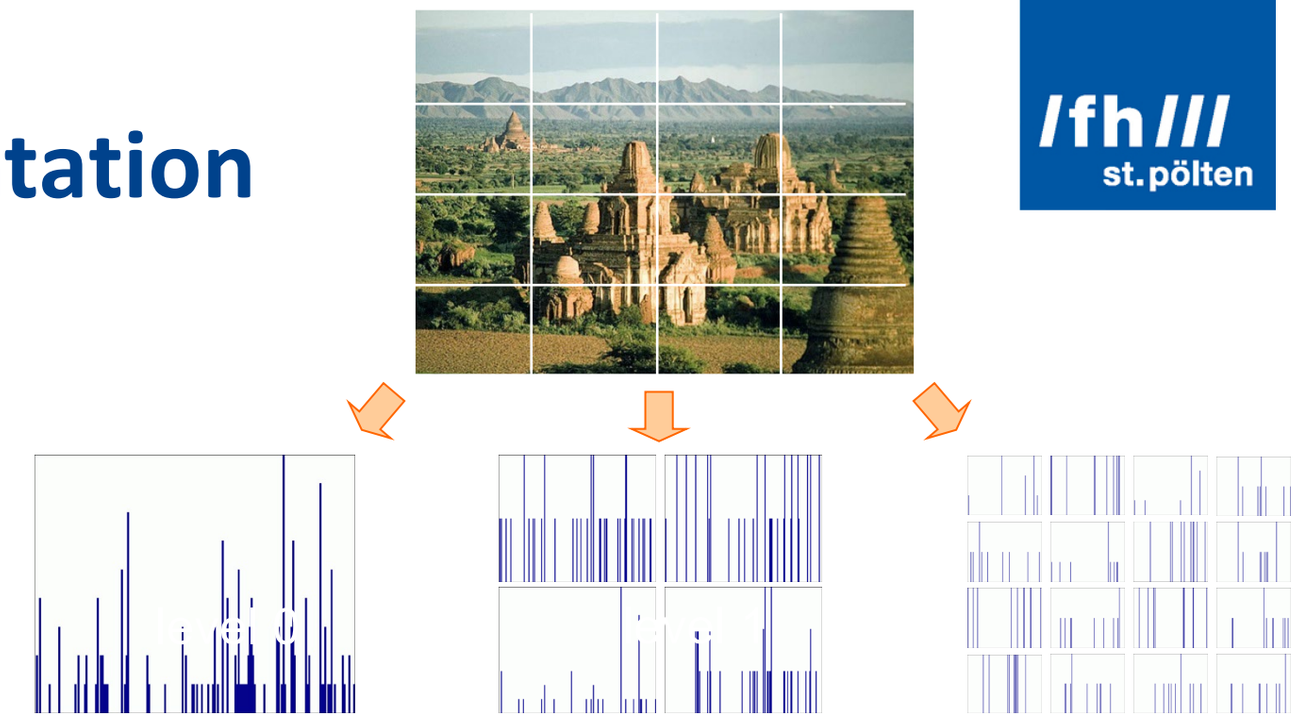




Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial Pyramid Representation

- Concatenate all codeword histograms → one large vector.
  Use this vector for matching



- Alternatively: for each pyramid level (=level of spatial division) concatenate the respective histograms → one vector per pyramid level
  Match at each pyramid level separately and combine matches to an overall match.

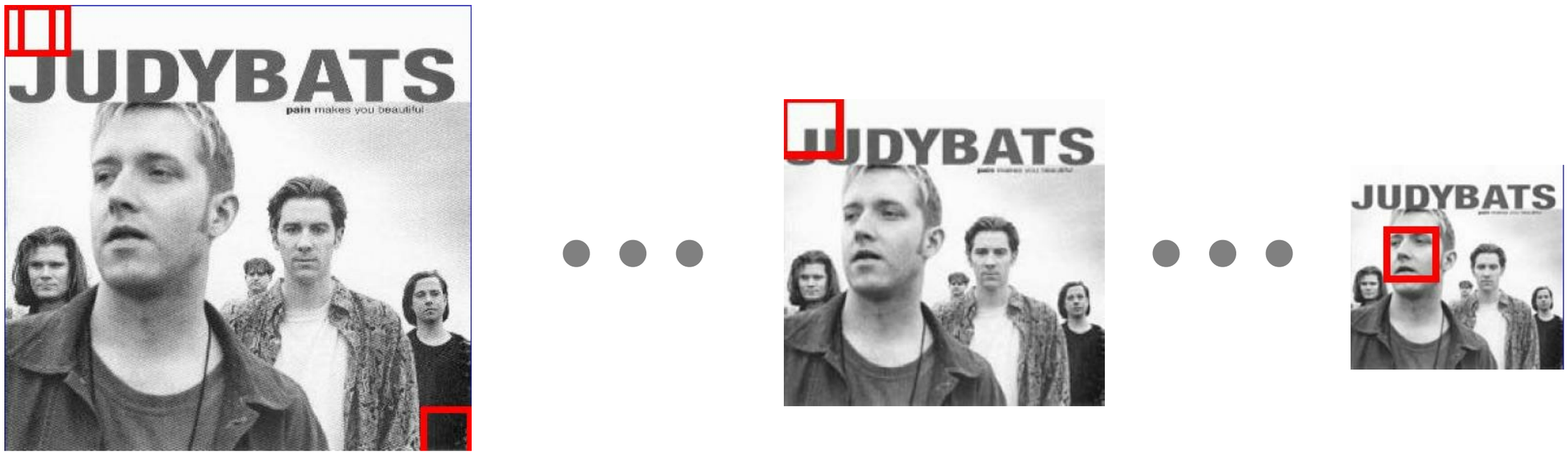
Object Recognition – In Practice

- BoVW is a robust representation for objects or entire images!
- How can object recognition be realized based on BoVW?

1. Training
   - Take many images of e.g. motorcycles, cars, dogs, etc., extract features, cluster them into a visual dictionary.
   - Train a classifier that takes codeword histograms as input and predicts the class (car, dog,..)
2. Detection/Recognition
   - Take an image as input → decide if it contains a motorcycle/car/horse/dog
   - Straight-forward approach: **sliding window:** move a detection window over image. At each position extract local features and build codeword histogram for current window
   - Feed the histogram into the classifier to obtain a prediction for the object visible in the window

What *is* a Sliding Window Approach?

- Search over space and scale



- Detection as sub-window classification problem

- “In the absence of a more intelligent strategy, any global image classification approach can be converted into a localization approach by using a sliding-window search.”

Slide by Bastian Liebe

Bags of words: pros and cons

- Pros
  - flexible to geometry / deformations / viewpoint
  - compact summary of image content
  - provides fixed vector representation for differently large sets of local feature points
  - + very good results in practice
- Cons
  - basic model ignores geometry – must verify afterwards, or encode via features
  - background and foreground mixed when bag covers whole image
  - optimal vocabulary formation remains unclear

Roadmap

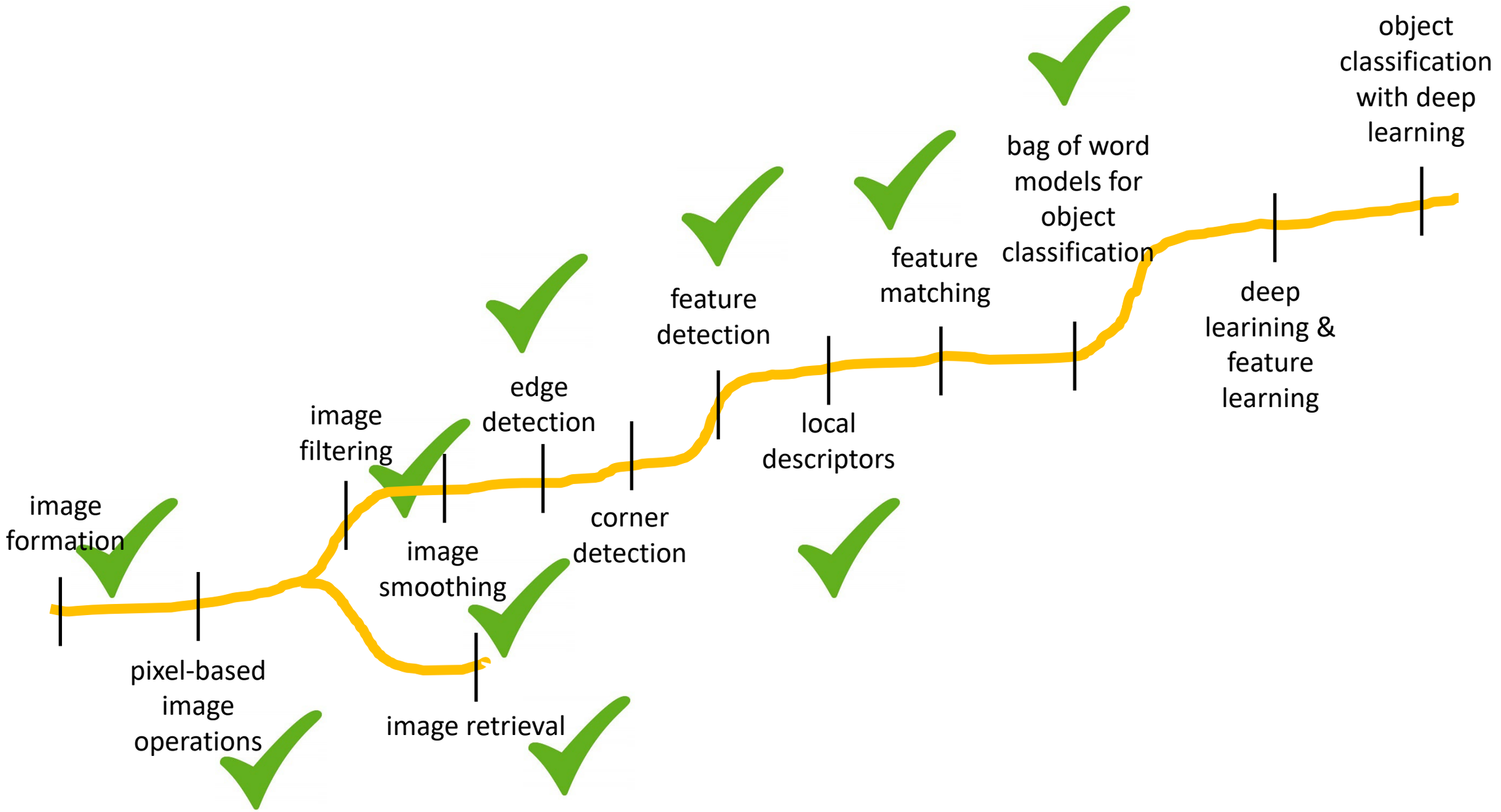


image formation
pixel-based image operations
image filtering
image smoothing
edge detection
corner detection
feature detection
image retrieval
feature matching
local descriptors
bag of word models for object classification
deep learining & feature learning
object classification with deep learning