13. Bag-of-Words

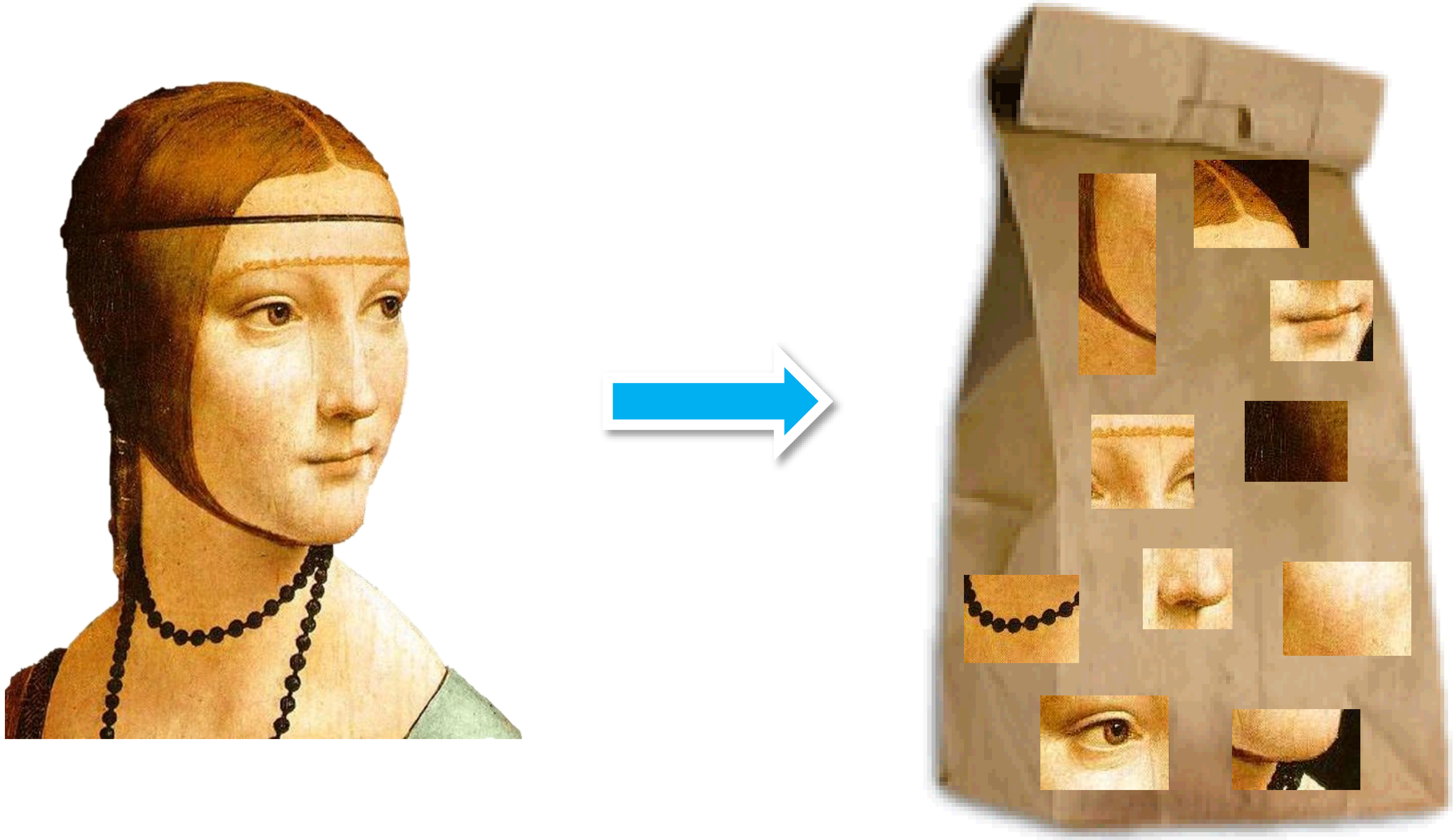


Image Classification

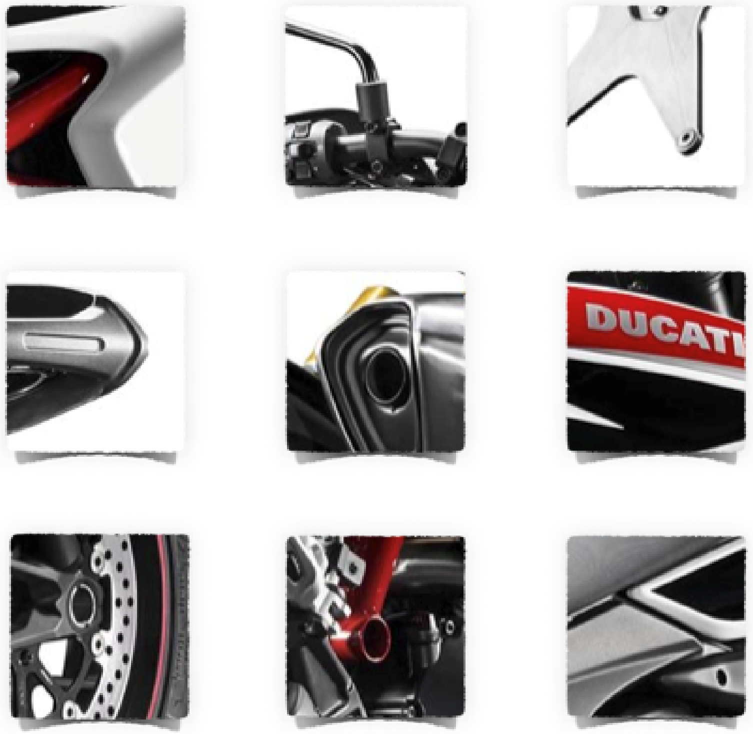
(assume given set of discrete labels)
{dog, cat, truck, plane, ...}

→ cat

More classification later
We focus on image search first

Some local feature
are very informative

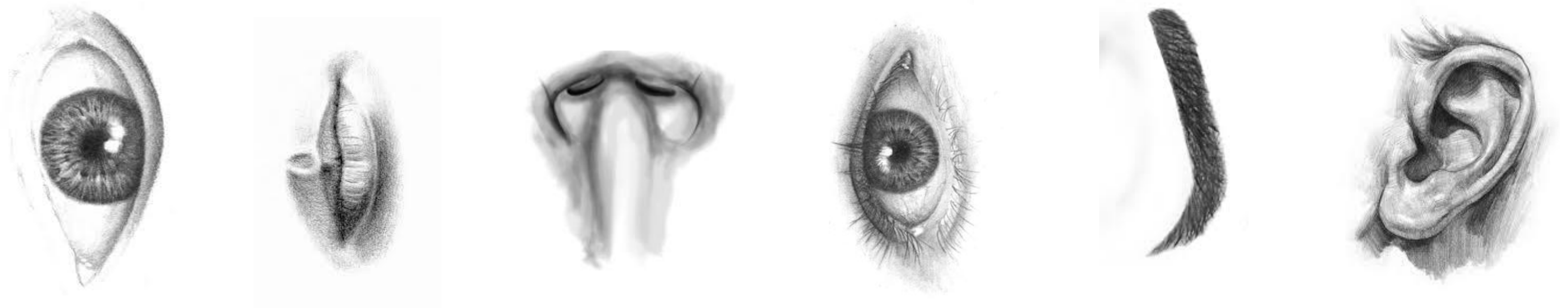
An object as



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)

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)

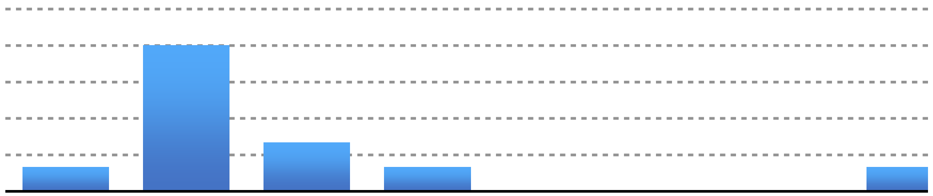
Vector Space Model

The Newsp

Sunday, December 22, 2013

DARPA Selects Carnegie Me

The Tartan Rescue Team from Carnegie Mellon University's National Robotics Engineering Center ranked third among teams competing in the Defense Advanced Research Projects Agency (DARPA) Robotics Challenge Trials this weekend in Homestead, Fla., and was selected by the agency as one of eight teams eligible for DARPA funding to prepare for next December's finals. The team's four-limbed CMU Highly Intelligent Mobile Platform, or CHIMP, robot scored 18 out of a possible 32 points during the two-day trials. It demonstrated its ability to perform such tasks as removing debris, cutting a hole through a wall and closing a series of valves.



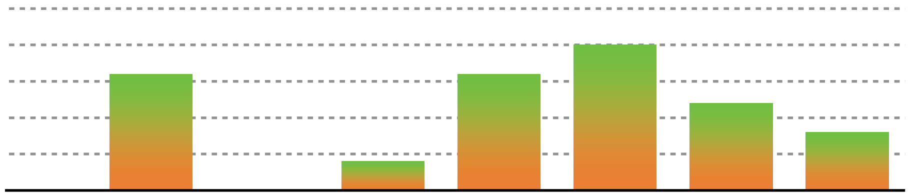
1	6	2	1	0	0	0	1
Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor

Tartan Tim

Monday, January 20, 2014

Bio-Inspired Robotic Device

PITTSBURGH—A soft, wearable device that mimics the muscles, tendons and ligaments of the lower leg could aid in the rehabilitation of patients with ankle-foot disorders such as drop foot, said Yong-Lae Park, an assistant professor of robotics at Carnegie Mellon University. Park, working with collaborators at Harvard University, the University of Southern California, MIT and BioSensics, developed an active orthotic device using soft plastics and composite materials, instead of a rigid exoskeleton. The soft materials, combined with pneumatic artificial muscles (PAMs), lightweight sensors and advanced control software, made it possible for the robotic device to achieve natural motions in the ankle.



0	4	0	1	4	5	3	2
Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor

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

Vector Space Model

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

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

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

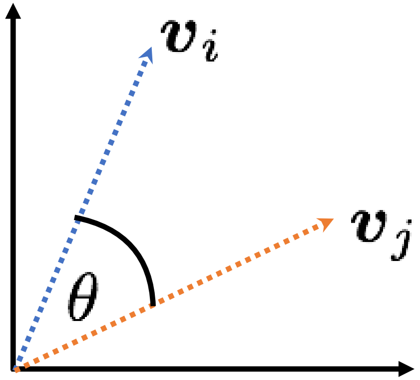
just a histogram over words


The Newspa
DARPA Selects Carnegie Me

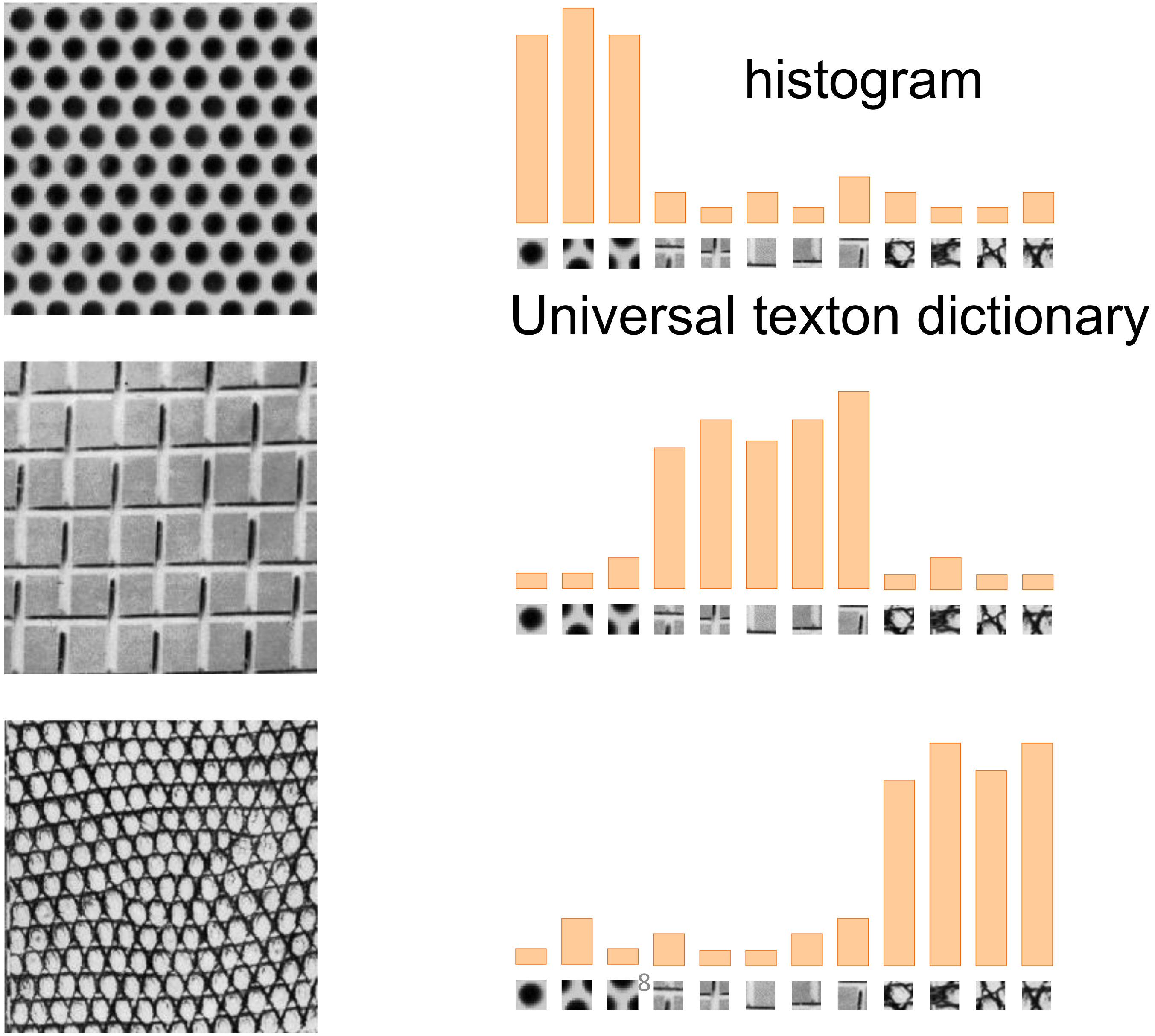

Tartan Tim
Bio-Inspired Robotic Device

- What is the similarity between two documents?
  - Use any distance you want but the cosine distance is fast

$$d(\boldsymbol{v}_i, \boldsymbol{v}_j) = \cos\theta = \frac{\boldsymbol{v}_i \cdot \boldsymbol{v}_j}{\|\boldsymbol{v}_i\|\|\boldsymbol{v}_j\|}$$

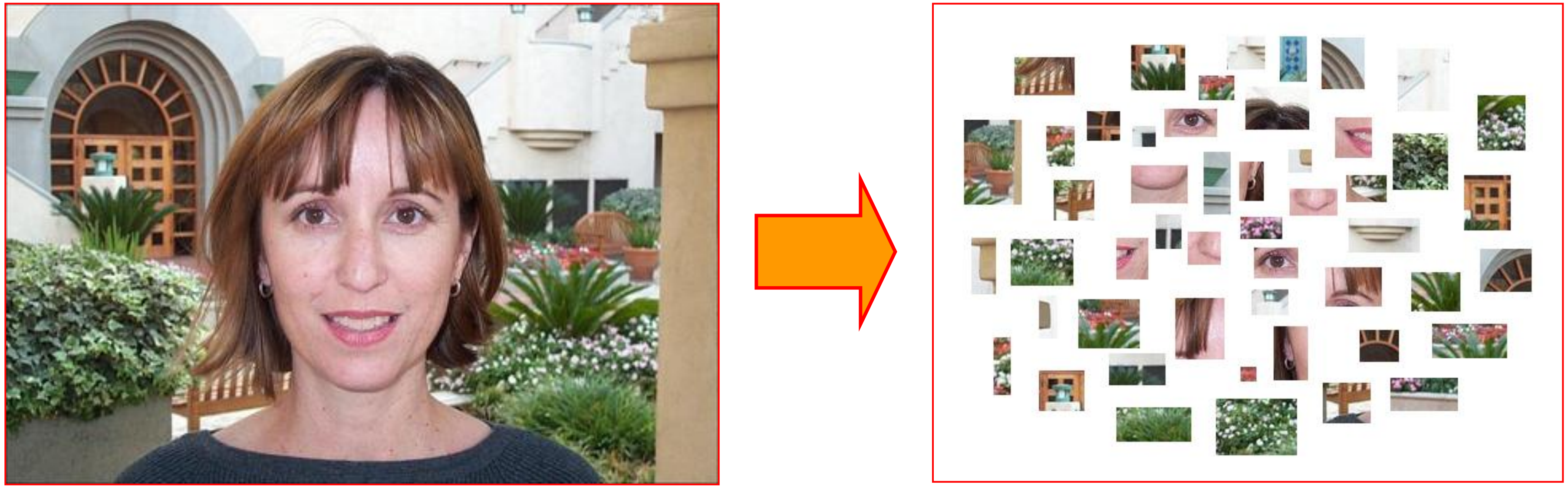


Texture Recognition



Bags of features for object recognition

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



face, flowers, building

Bags of features for object recognition






class	bag of features Zhang et al. (2005)	bag of features Willamowski et al. (2004)	Parts-and-shape model 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

Questions?

Bag of Words Pipeline

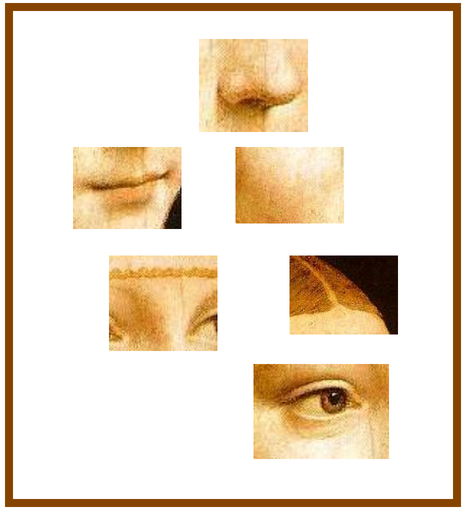
- Dictionary Learning: take a bunch of images, extract features, and build up a “dictionary” or “visual vocabulary” – a list of common features

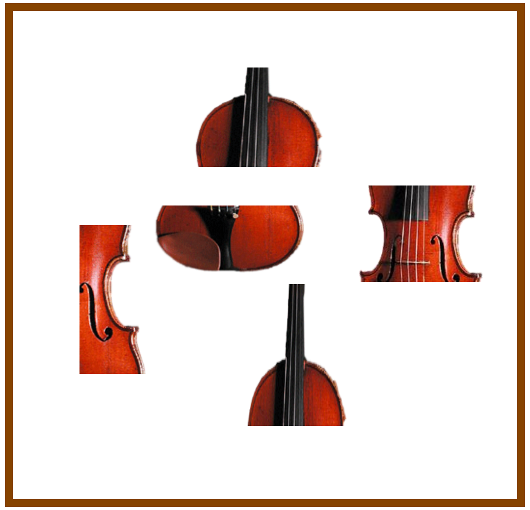
- Encode: given a new image, extract features and build a histogram – for each feature, find the closest visual word in the dictionary

- Classify: train and test data using BOW features (later)

Bag of Words: outline

1. Extract features





Bag of Words: outline

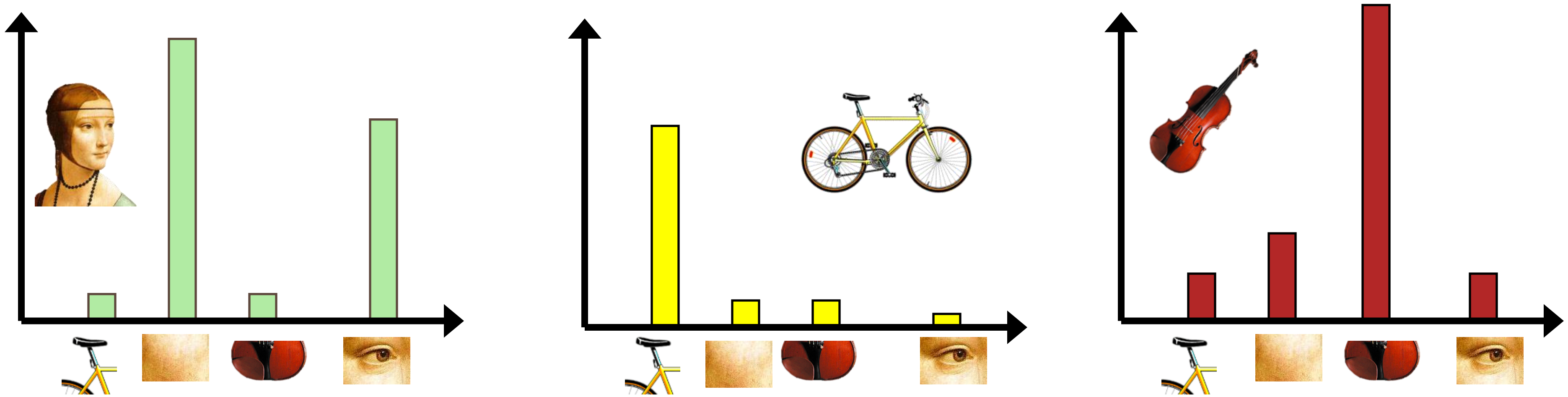
1. Extract features
2. Learn "visual vocabulary"

Bag of Words: outline

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

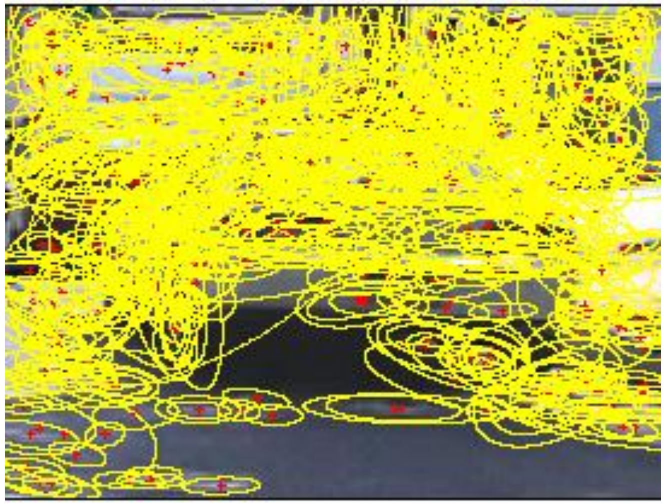
Bag of Words: outline

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

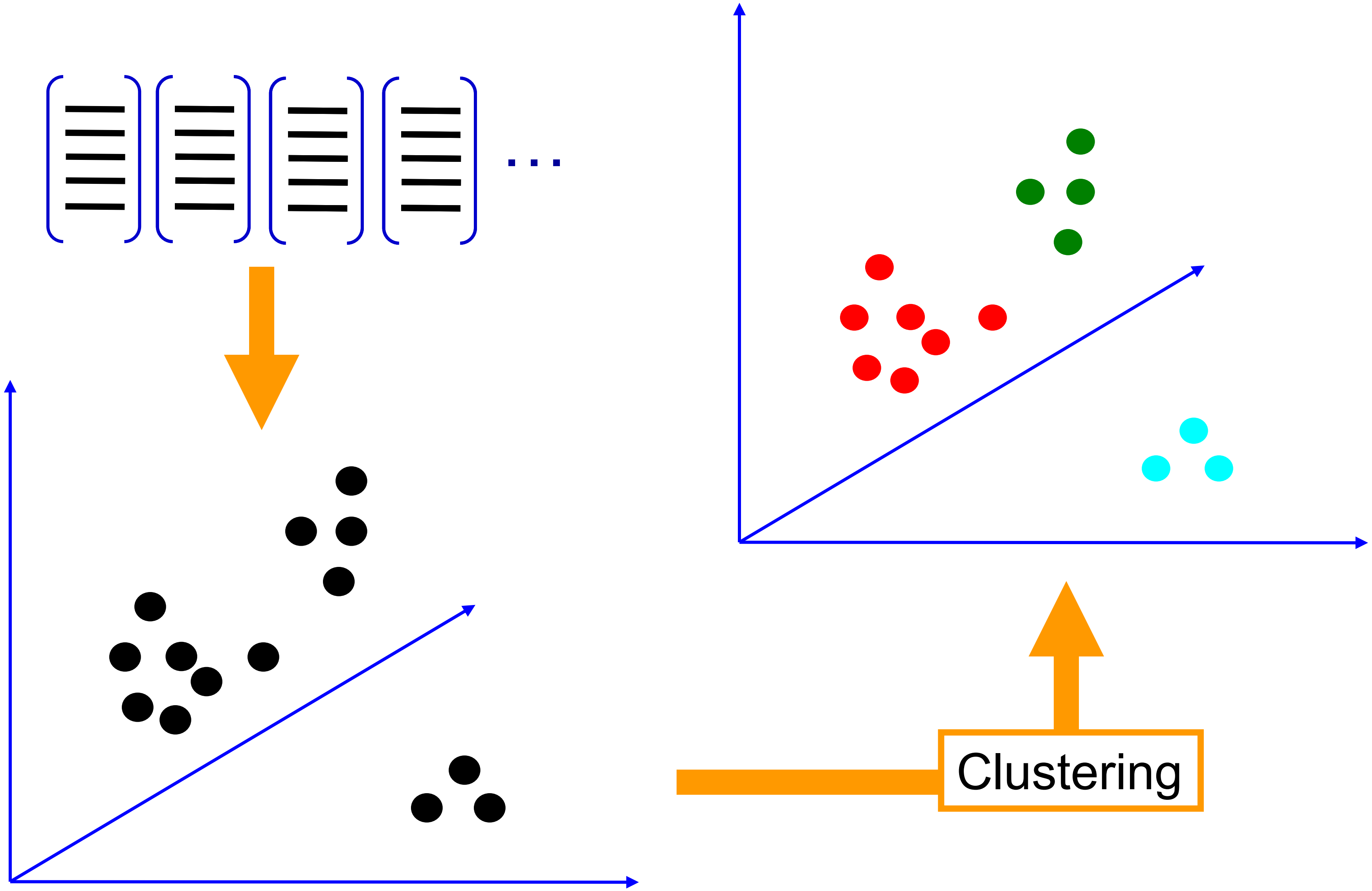


1. Feature extraction

- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
- 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)

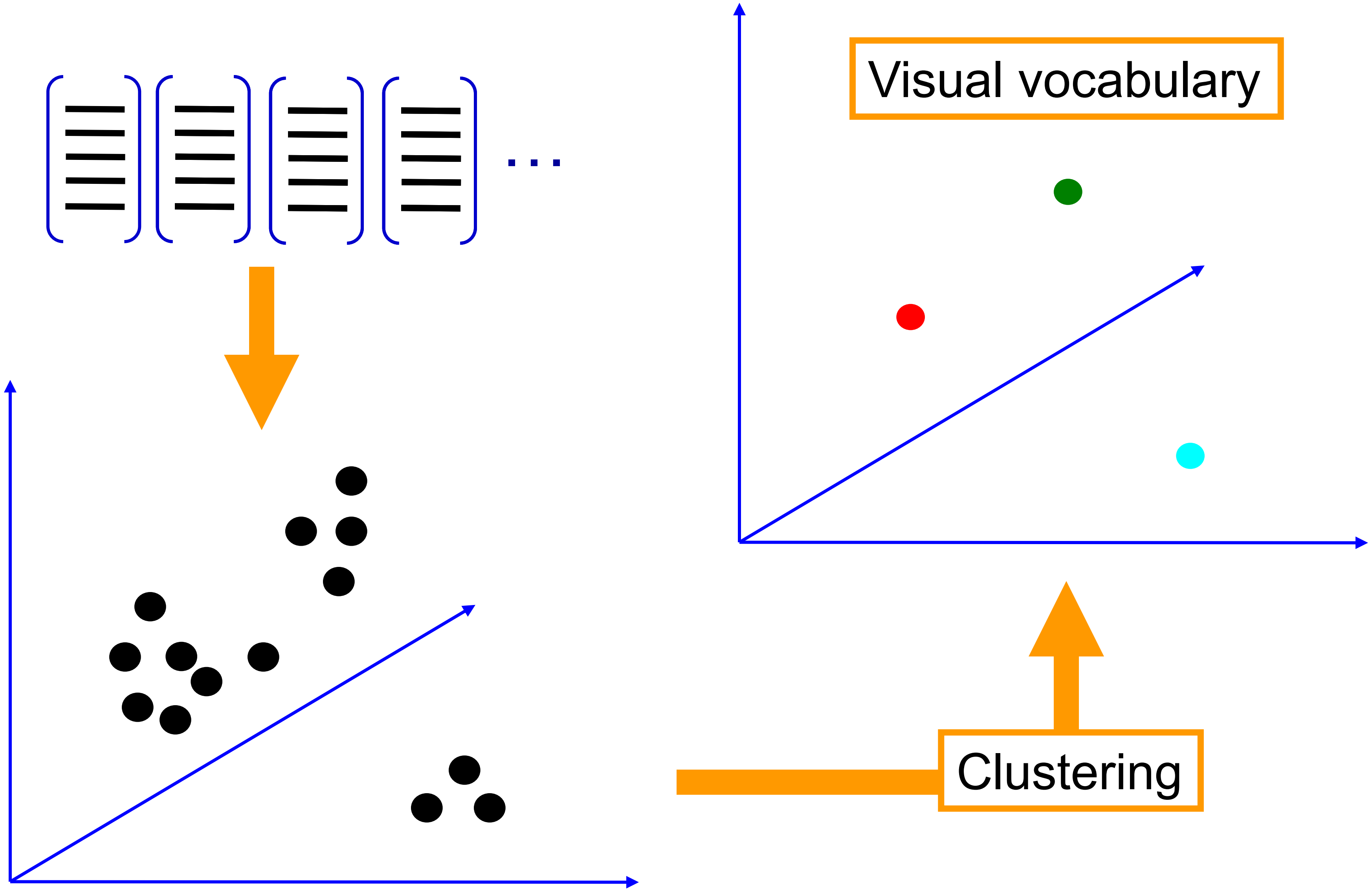


2. Learning the visual vocabulary



Slide credit: Josef Sivic

2. Learning the visual vocabulary

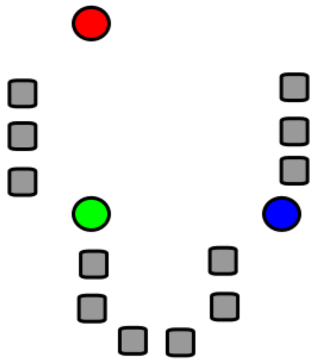


Slide credit: Josef Sivic

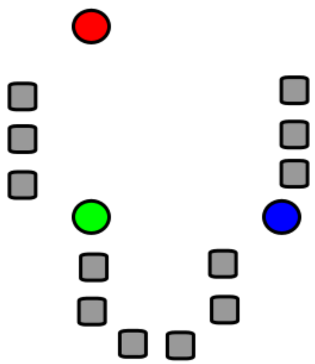
K-means Clustering

- Given k:

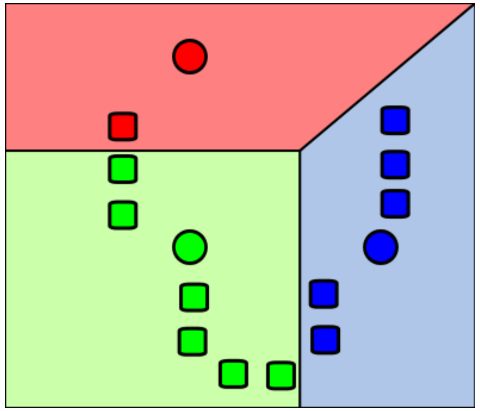
1. Select initial centroids at random.
2. Assign each object to the cluster with the nearest centroid.
3. Compute each centroid as the mean of the objects assigned to it.
4. Repeat previous 2 steps until no change.



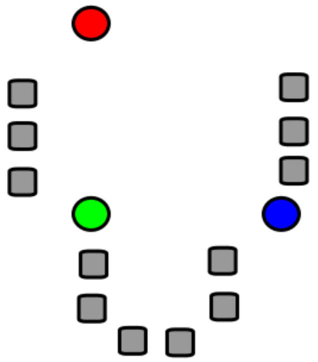
1. Select initial centroids at random



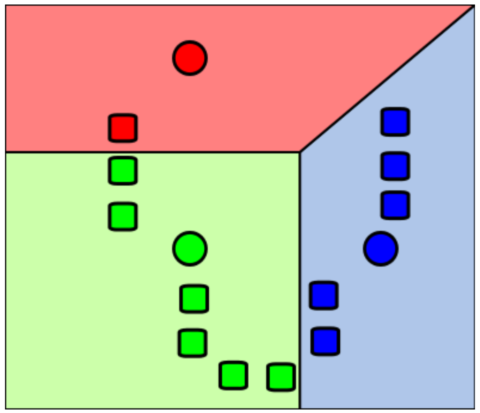
1. Select initial centroids at random



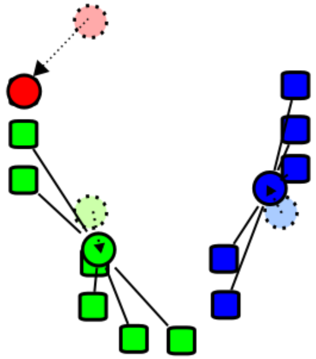
2. Assign each object to the cluster with the nearest centroid.



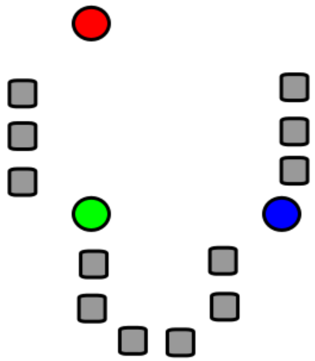
1. Select initial centroids at random



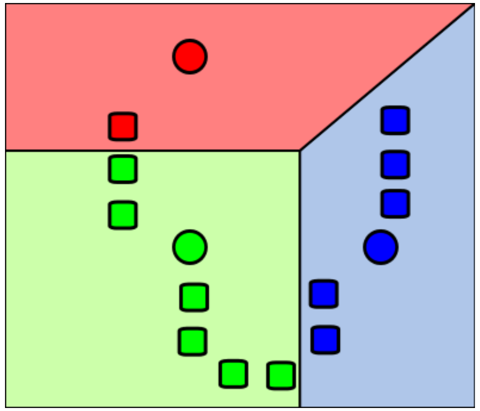
2. Assign each object to the cluster with the nearest centroid.



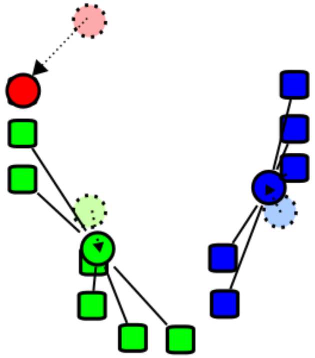
3. Compute each centroid as the mean of the objects assigned to it (go to 2)



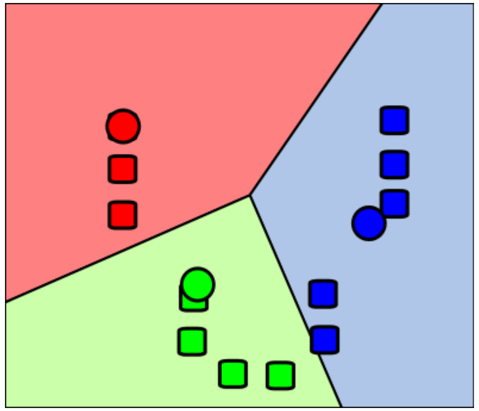
1. Select initial centroids at random



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.

Repeat previous 2 steps until no change

2. Learning the visual vocabulary

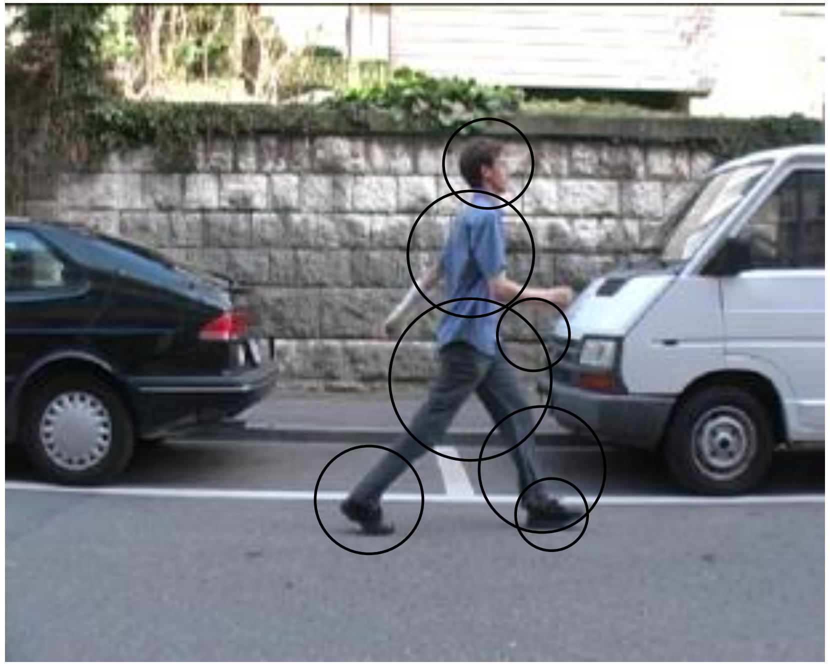
- From what data should I learn the dictionary?
  - Dictionary can be learned on separate training set
  - Provided the training set is sufficiently representative, the dictionary will be "universal"

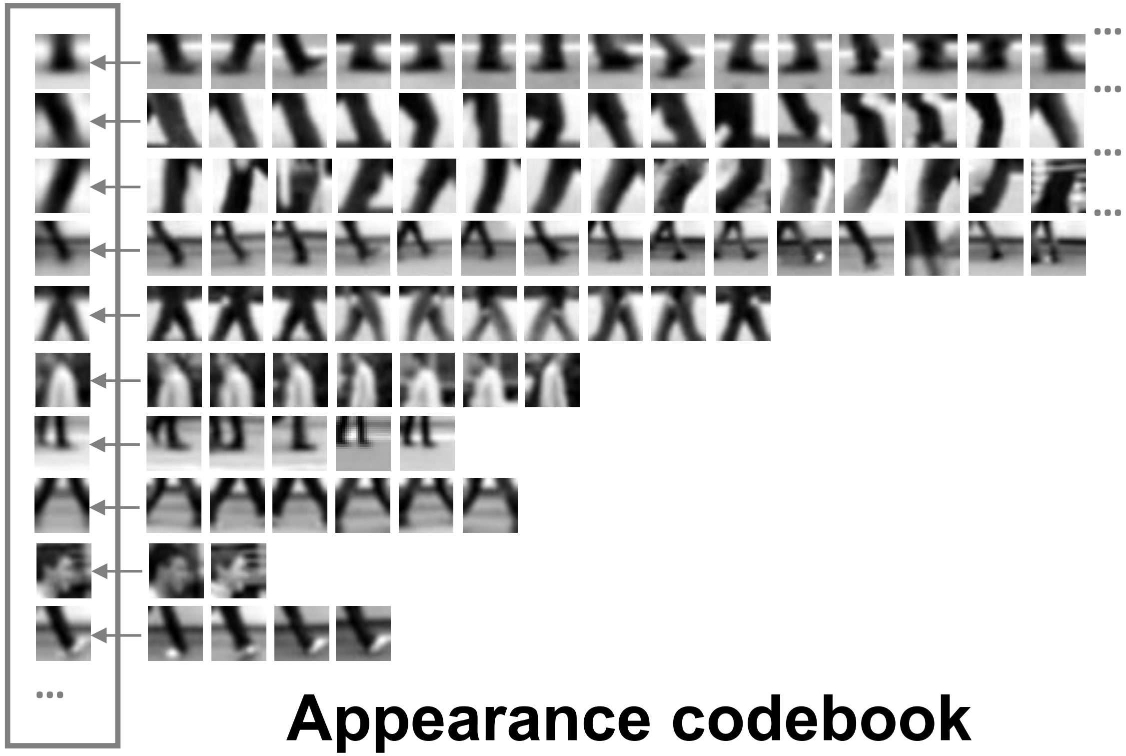
Example Dictionary

Appearance codebook

Source: B. Leibe

Another Dictionary



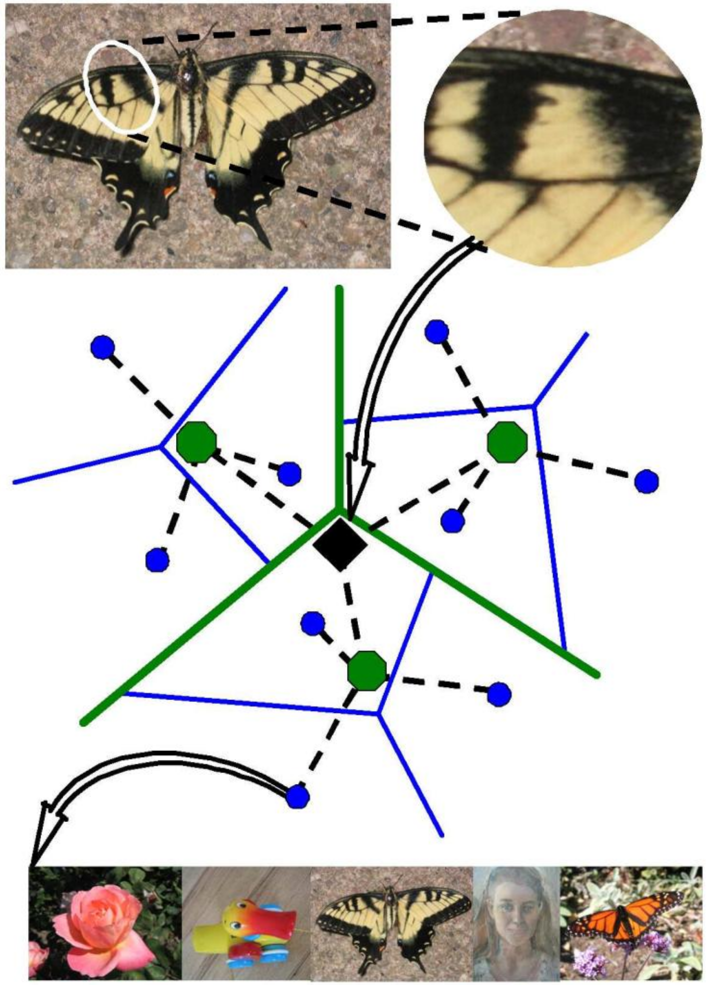


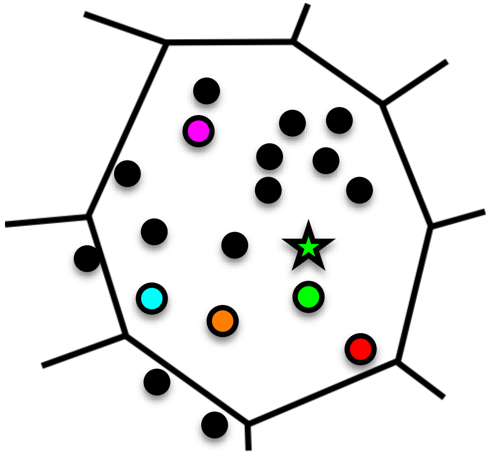
Appearance codebook

Source: B. Leibe

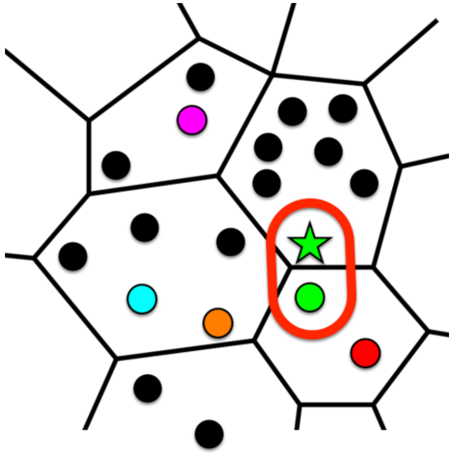
Visual Vocabularies: Issues

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





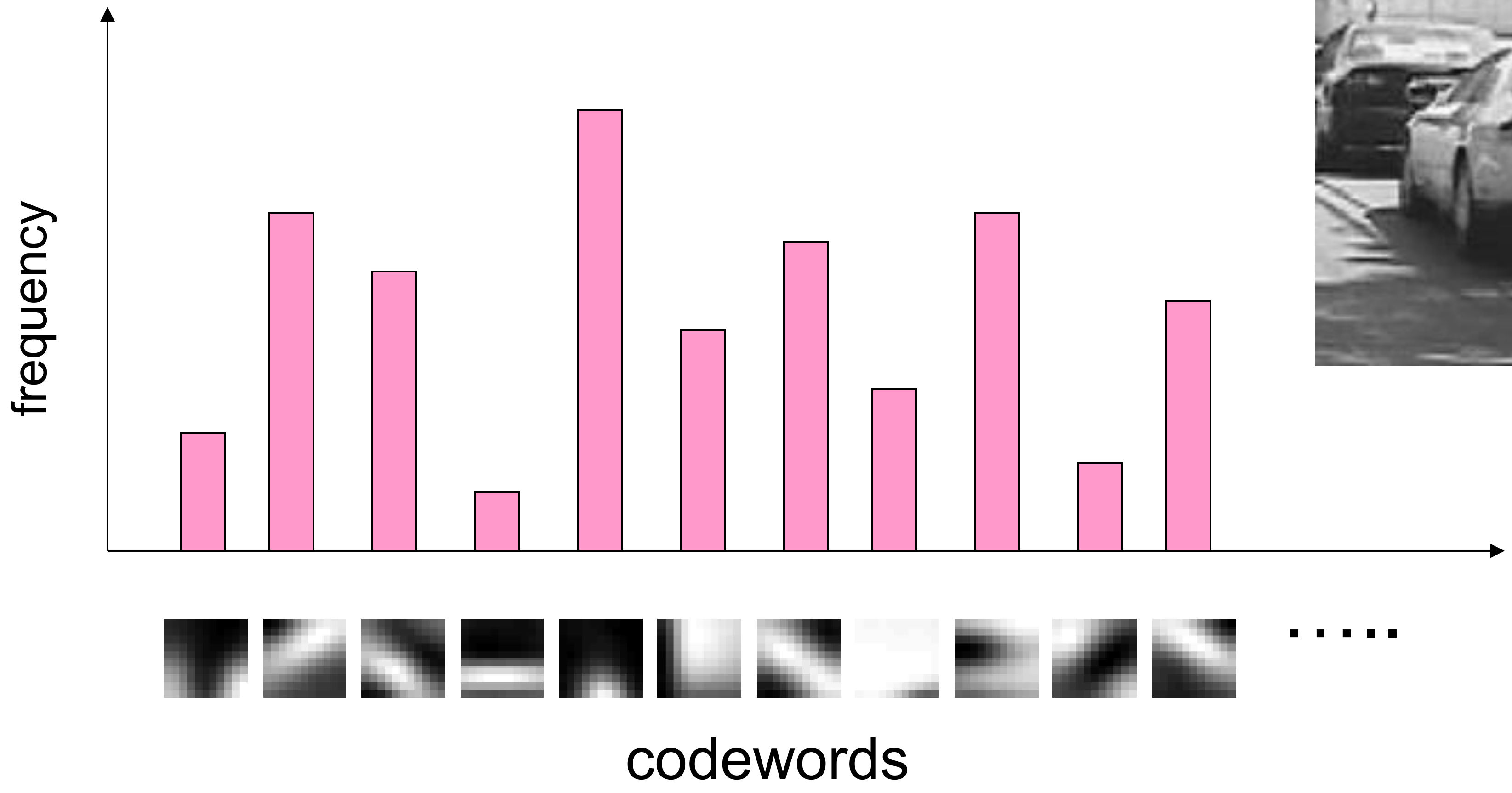
Smaller Vocabulary



Match missed
due to quantization!

3. Image Representation

- Histogram: count the number of visual word occurrences



Large-scale Image Search

11,400 images of game covers
(Caltech games dataset)

- Bag-of-words models have been useful in matching an image to a large database of object *instances*

how do I find this image in the database?

Large-scale Image Search

- Build the database:
  - Extract features from the database images
  - Learn a vocabulary using k-means (typical k: 100,000)
  - Compute *weights* for each word
  - Create an inverted file mapping words → images

Weighting the Words

- Just as with text, some visual words are more discriminative than others

the, and, or vs. *cow, AT&T, Cher*

- the bigger fraction of the documents a word appears in, the less useful it is for matching
  - e.g., a word that appears in *all* documents is not helping us

TF-IDF Weighting

- Instead of computing a regular histogram distance, we'll weight each word by its *inverse document frequency*

- inverse document frequency (IDF) of word *j* =

$$\log \frac{\text{number of documents}}{\text{number of documents in which } j \text{ appears}}$$

TF-IDF Weighting

- To compute the value of bin *j* in image *I*:

term frequency of *j* in *I* x *inverse document frequency* of *j*

Inverted File

- Each image has ~1,000 features
- We have ~100,000 visual words

  →each histogram is extremely sparse (mostly zeros)

- Inverted file
  - mapping from words to documents

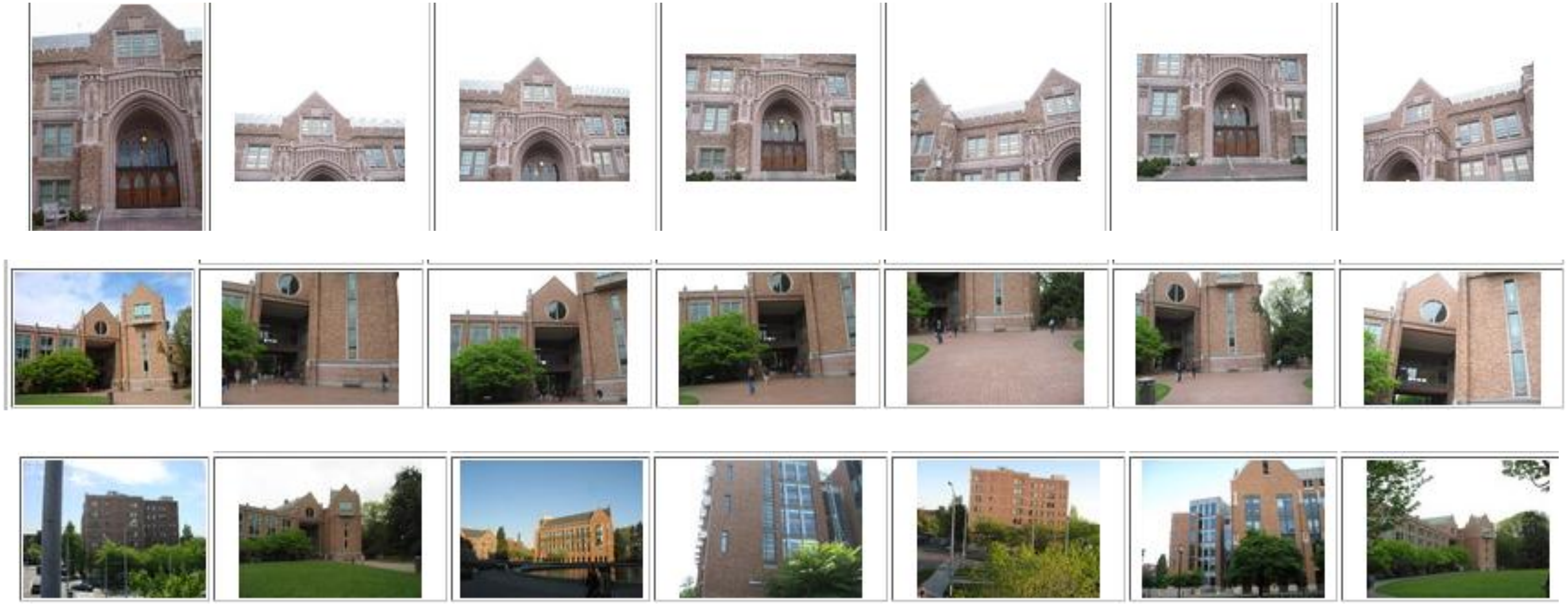
```
"a":      {2}
"banana": {2}
"is":     {0, 1, 2}
"it":     {0, 1, 2}
"what":   {0, 1}
```

Inverted File

- Can quickly use the inverted file to compute similarity between a new image and all the images in the database
  - Only consider database images whose bins overlap the query image

Large-scale image search

query image top 6 results



- Cons:
  - not as accurate as per-image-pair feature matching
  - performance degrades as the database grows

Large-scale Image Search

- Pros:
  - Works well for CD covers, movie posters
  - Real-time performance possible

real-time retrieval from a database of 40,000 CD covers

Scalable Recognition with a Vocabulary Tree

David Nistér and Henrik Stewénius
Center for Visualization and Virtual Environments
Department of Computer Science, University of Kentucky
http://www.vis.uky.edu/~dnister/ http://www.vis.uky.edu/~stewe/

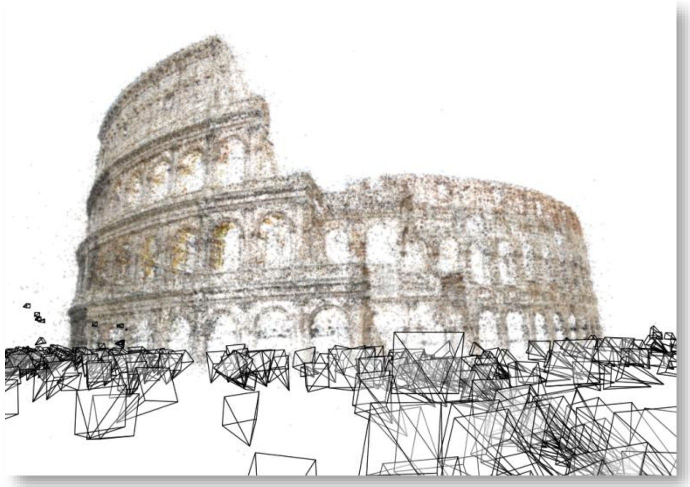
[CVPR 2006]

Large-scale image matching

Turn 1,000,000 images of Rome…

…into 3D models

St. Peter's Basilica



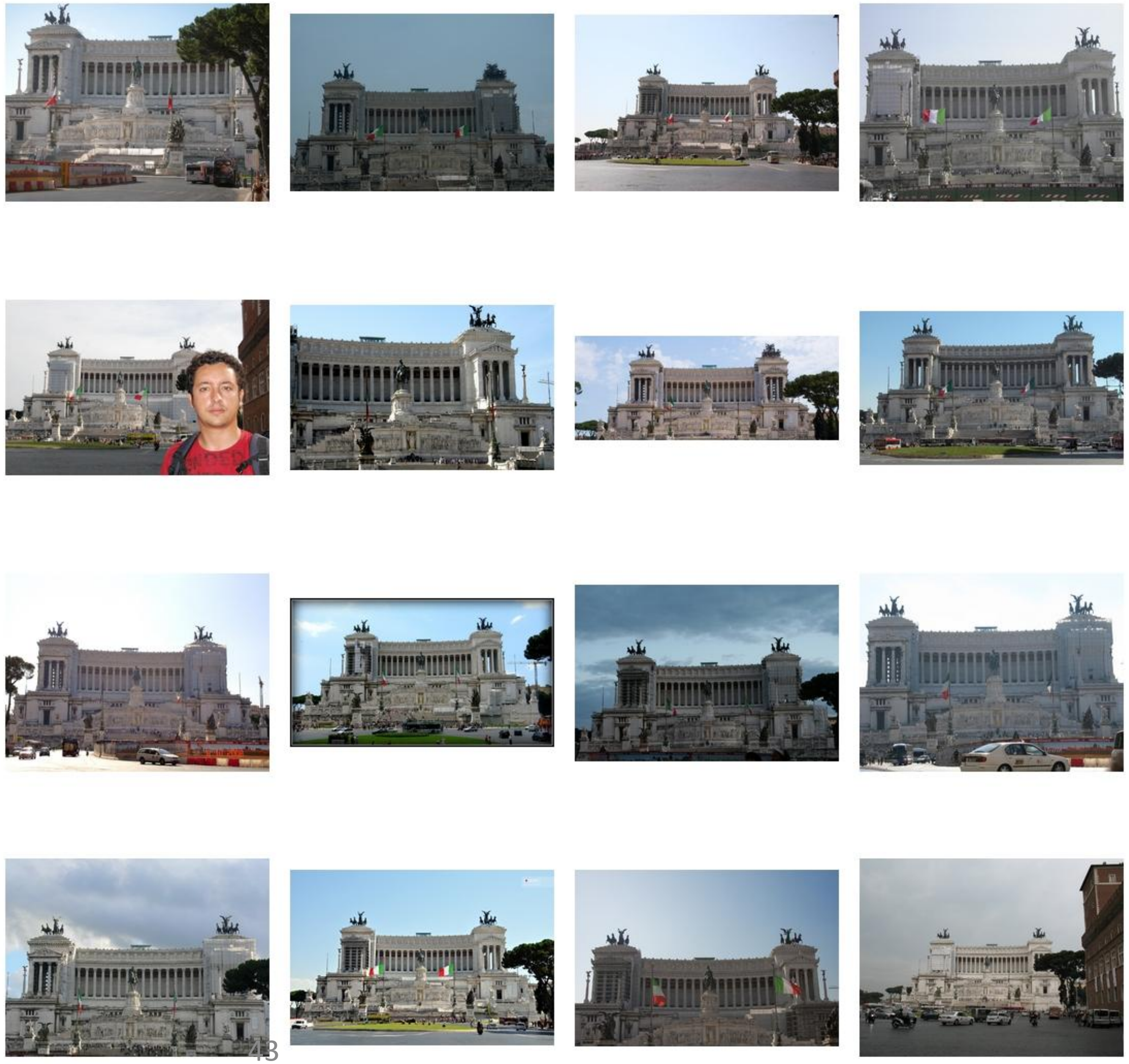
Colosseum

Trevi Fountain

Large-scale image matching

- How can we match 1,000,000 images to each other?
- Brute force approach: 500,000,000,000 pairs
  - won't scale
- Better approach: use bag-of-words technique to find *likely* matches
- For each image, find the top M scoring other images, do detailed SIFT matching with those

Example bag-of-words matches



Example bag-of-words matches

Matching Statistics

Dataset	Size	Matches possible	Matches Tried	Matches Found	Time
Dubrovnik	58K	1.6 Billion	2.6M	0.5M	5 hrs
Rome	150K	11.2 Billion	8.8M	2.7M	13 hrs
Venice	250K	31.2 Billion	35.5M	6.2M	27 hrs

NetVLAD: CNN architecture for weakly supervised place recognition

Relja Arandjelović
INRIA *

Petr Gronat
INRIA*

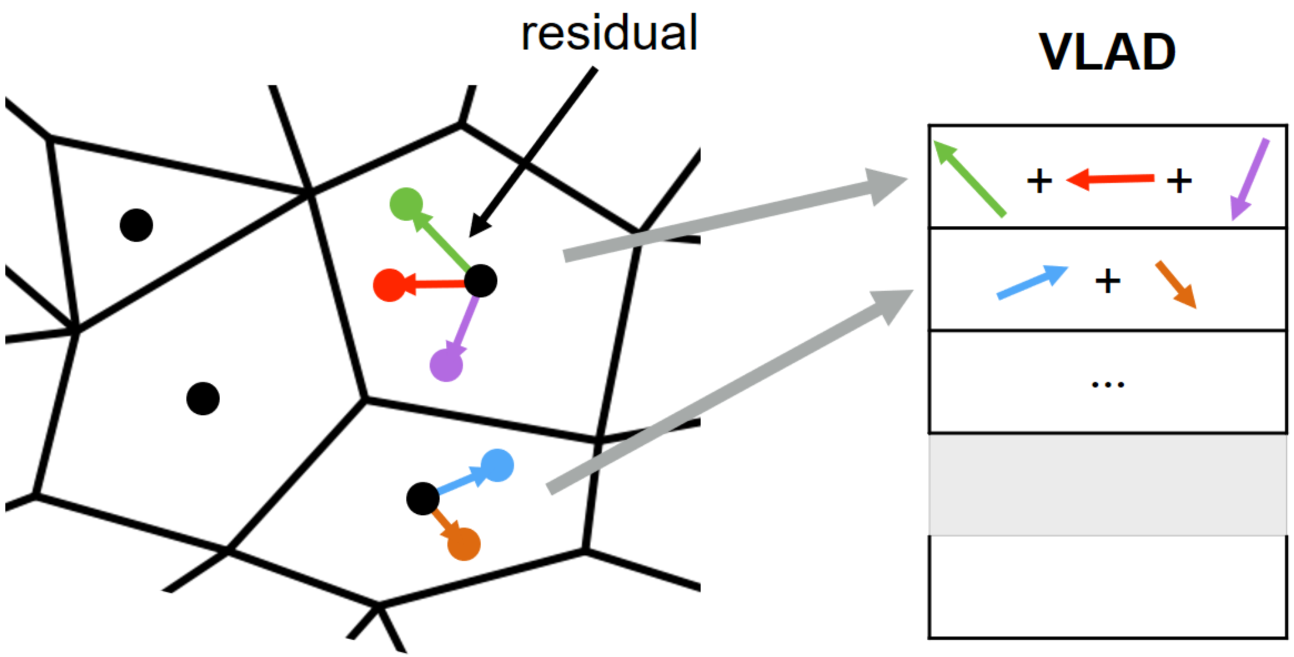
Akihiko Torii
Tokyo Tech †

Tomas Pajdla
CTU in Prague ‡

Josef Sivic
INRIA*

Vector of Locally Aggregated Descriptors (VLAD)

image with **local features**



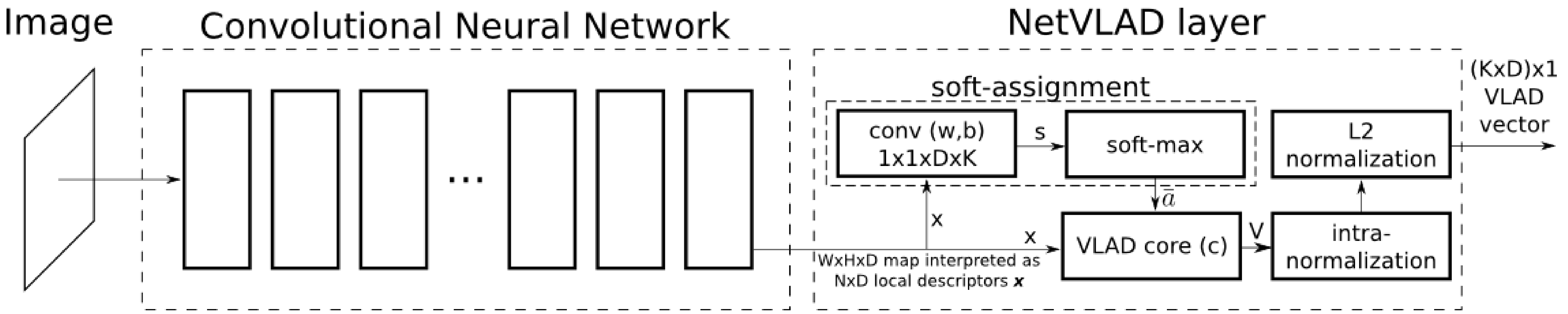
visual words =
quantization of descriptor space

+ PCA

$$V(j,k) = \sum_{i=1}^{N} \alpha_k(x_i)(x_i(j) - c_k(j))$$

Jégou et al. Aggregating local descriptors into a compact image representation. CVPR 2010
Arandjelovic & Zisserman. All about VLAD. CVPR 2013.

NetVLAD



$$V(j,k)=\sum_{i=1}^{N}\frac{e^{w_k^T x_i+b_k}}{\sum_{k'} e^{w_{k'}^T x_i+b_{k'}}}(x_i(j)-c_k(j))$$

[Arandjelović, Gronat, Torii, Pajdla, Sivic, NetVLAD: CNN architecture for weakly supervised place recognition. CVPR 2016]