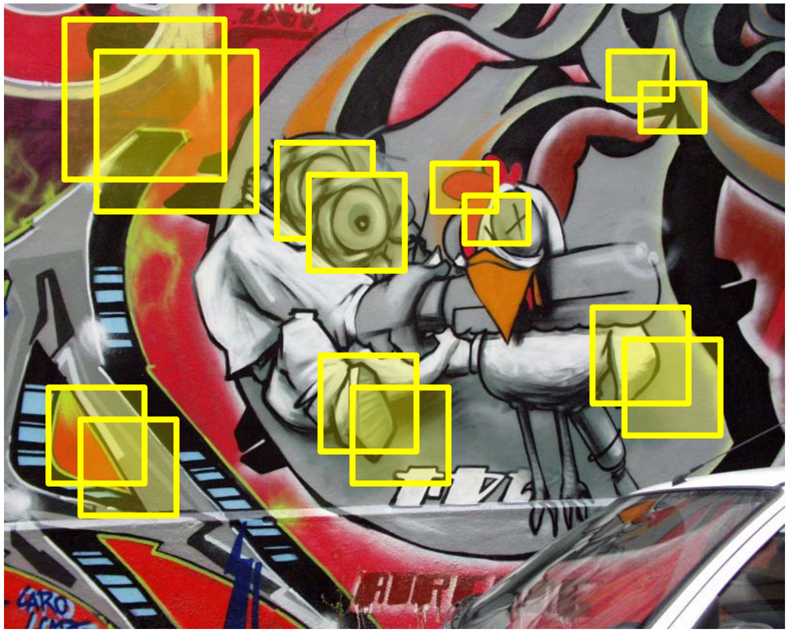
Patch Descriptors

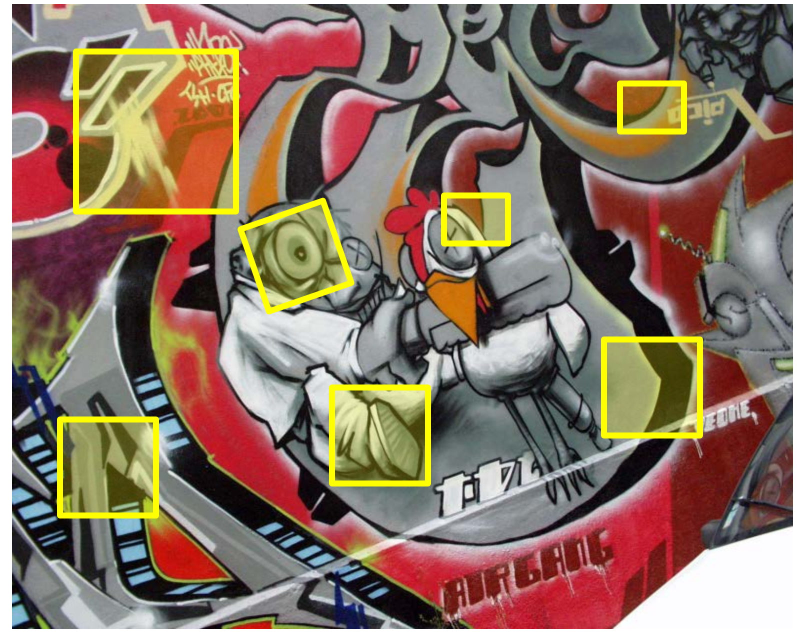
CSE 455

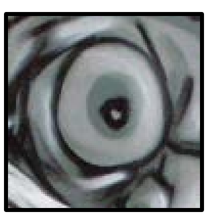
Linda Shapiro

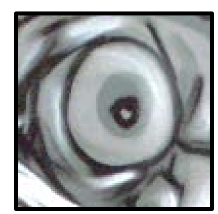
How can we find corresponding points?

How can we find correspondences?

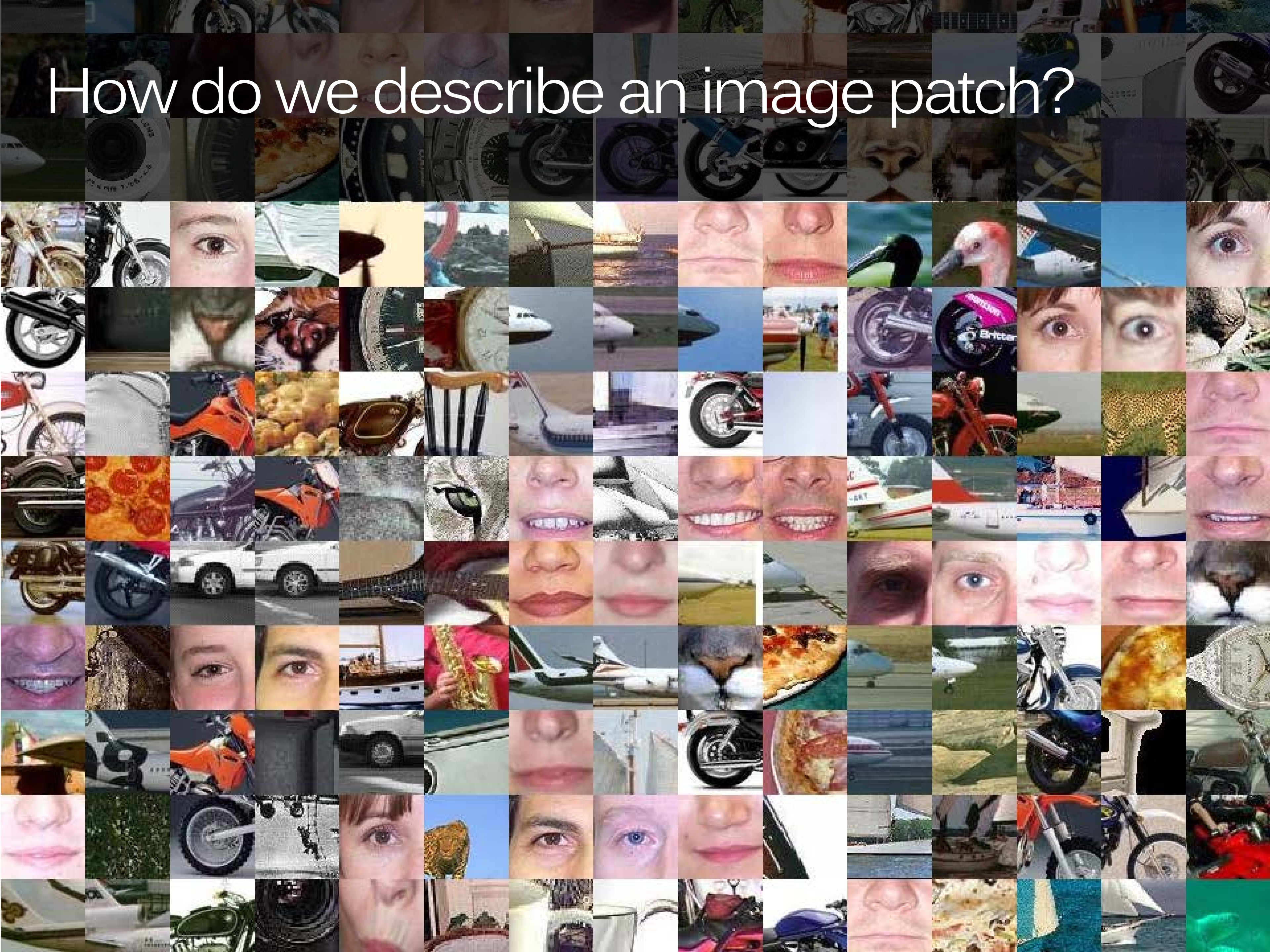








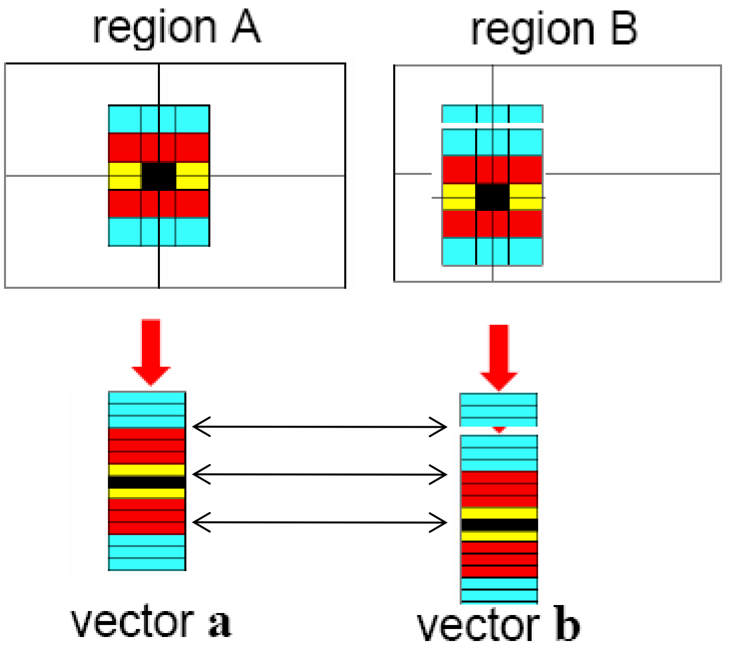
How do we describe an image patch?



How do we describe an image patch?

Patches with similar content should have similar descriptors.

Raw patches as local descriptors



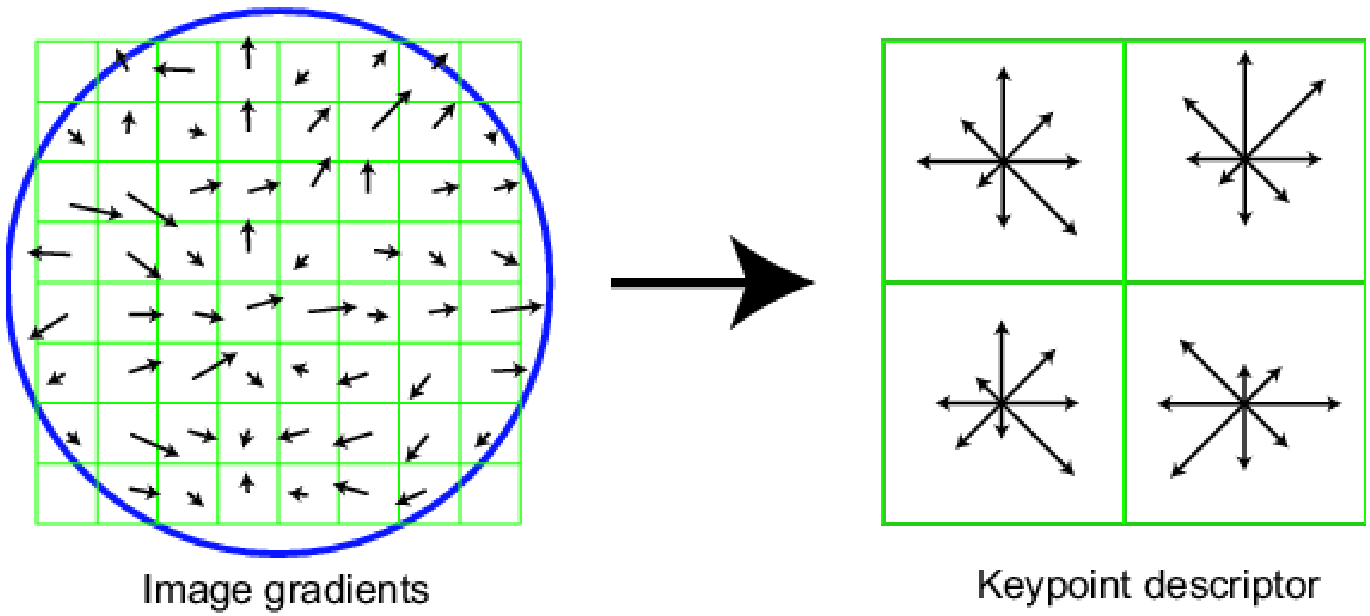
The simplest way to describe the neighborhood around an interest point is to write down the list of intensities to form a feature vector.

But this is very sensitive to even small shifts, rotations.

SIFT descriptor

Full version

- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor

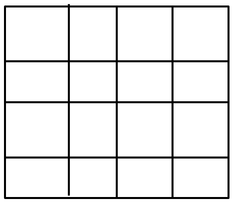


Adapted from slide by David Lowe

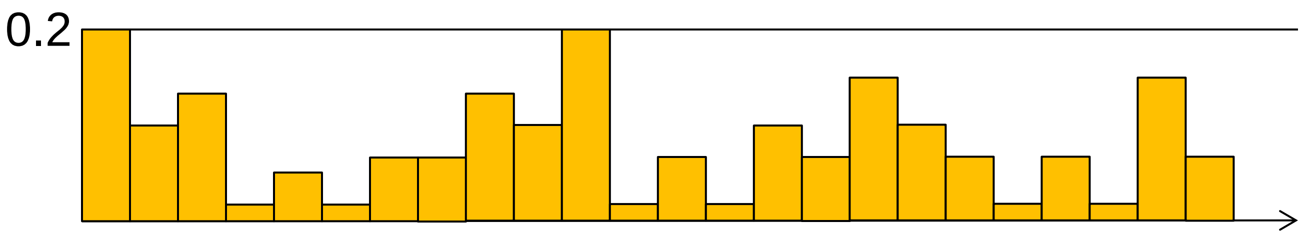
SIFT descriptor

Full version

- Start with a 16x16 window (256 pixels)
- Divide the 16x16 window into a 4x4 grid of cells (16 cells)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor
- Threshold normalize the descriptor:



$$\sum_i d_i^2 = 1 \quad \text{such that:} \quad d_i < 0.2$$



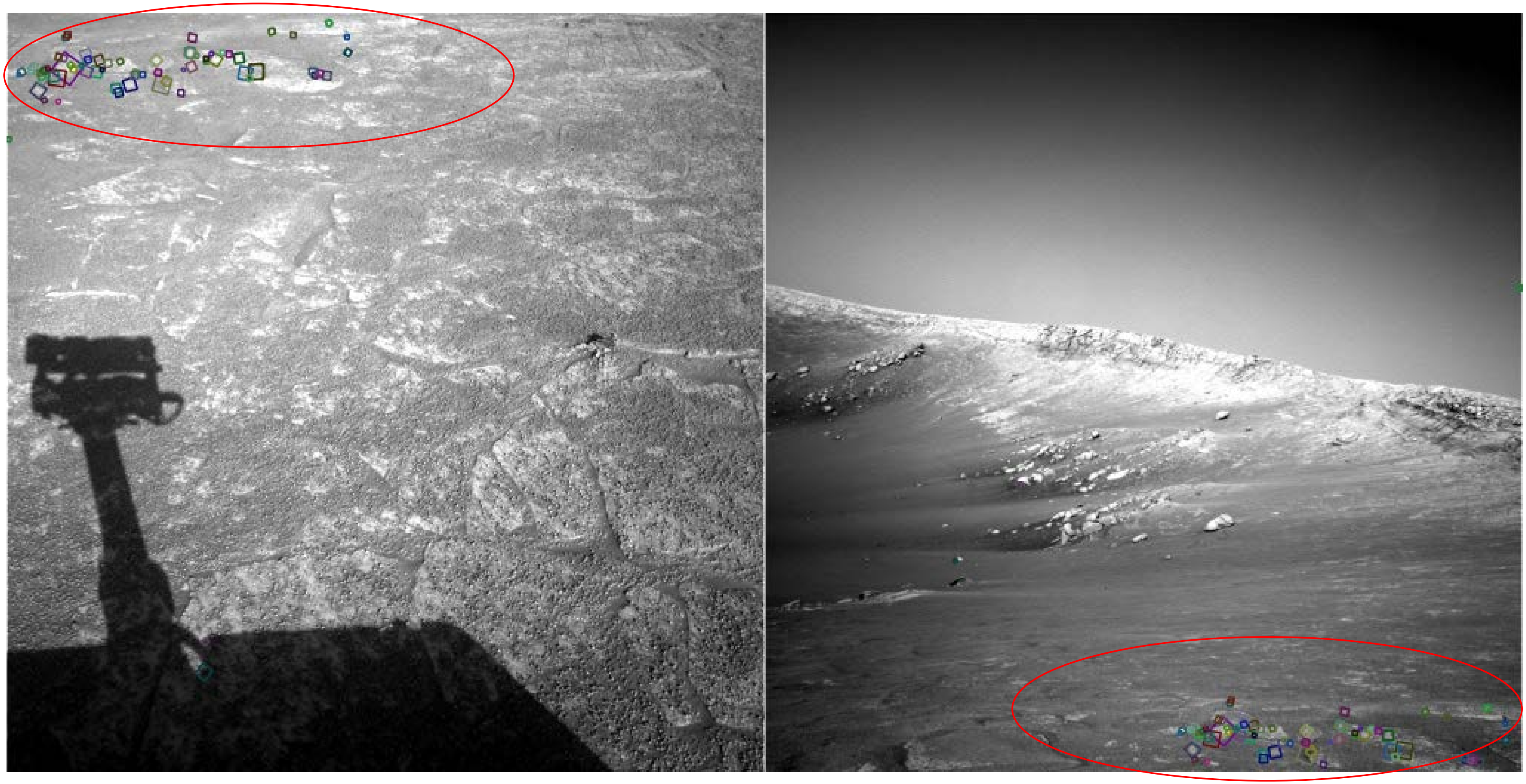
Adapted from slide by David Lowe

Properties of SIFT

Extraordinarily robust matching technique

- Can handle changes in viewpoint
  - Up to about 30 degree out of plane rotation
- Can handle significant changes in illumination
  - Sometimes even day vs. night (below)
- Fast and efficient—can run in real time
- Various code available
  - http://www.cs.ubc.ca/~lowe/keypoints/

Example



NASA Mars Rover images
with SIFT feature matches
Figure by Noah Snavely

Example: Object Recognition

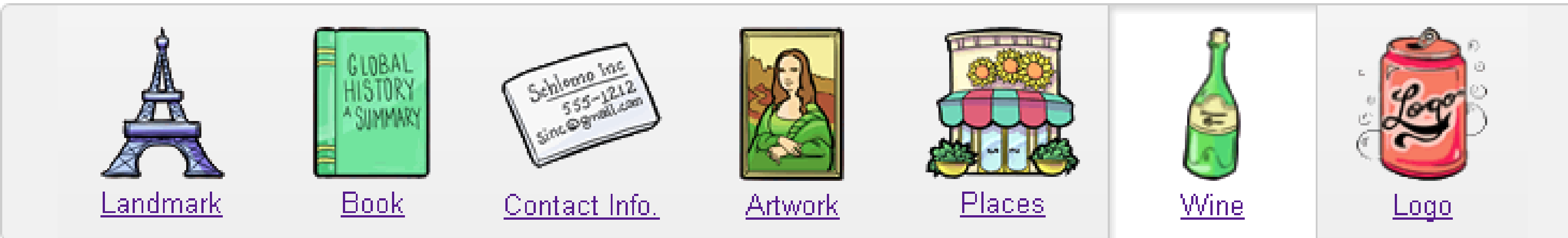
SIFT is extremely powerful for object instance recognition, especially for well-textured objects

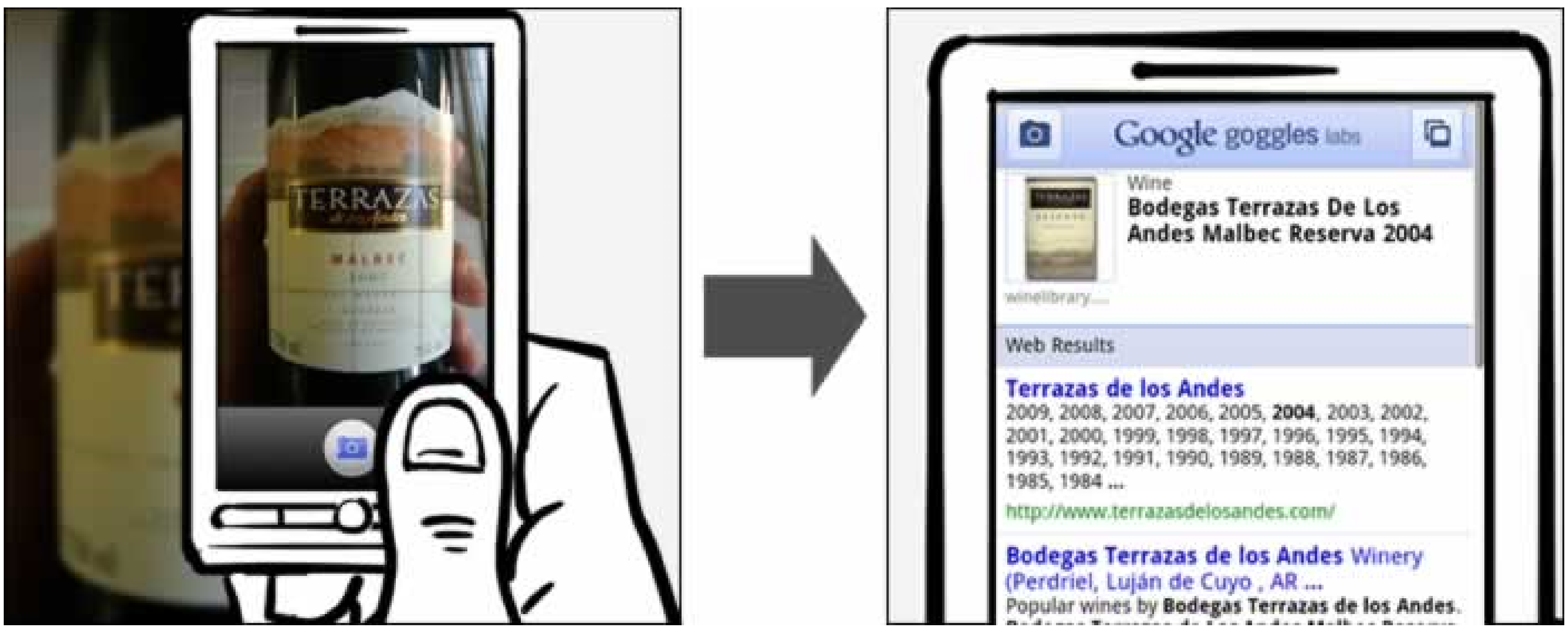
Lowe, IJCV04

Example: Google Goggle

Google Goggles in Action

Click the icons below to see the different ways Google Goggles can be used.



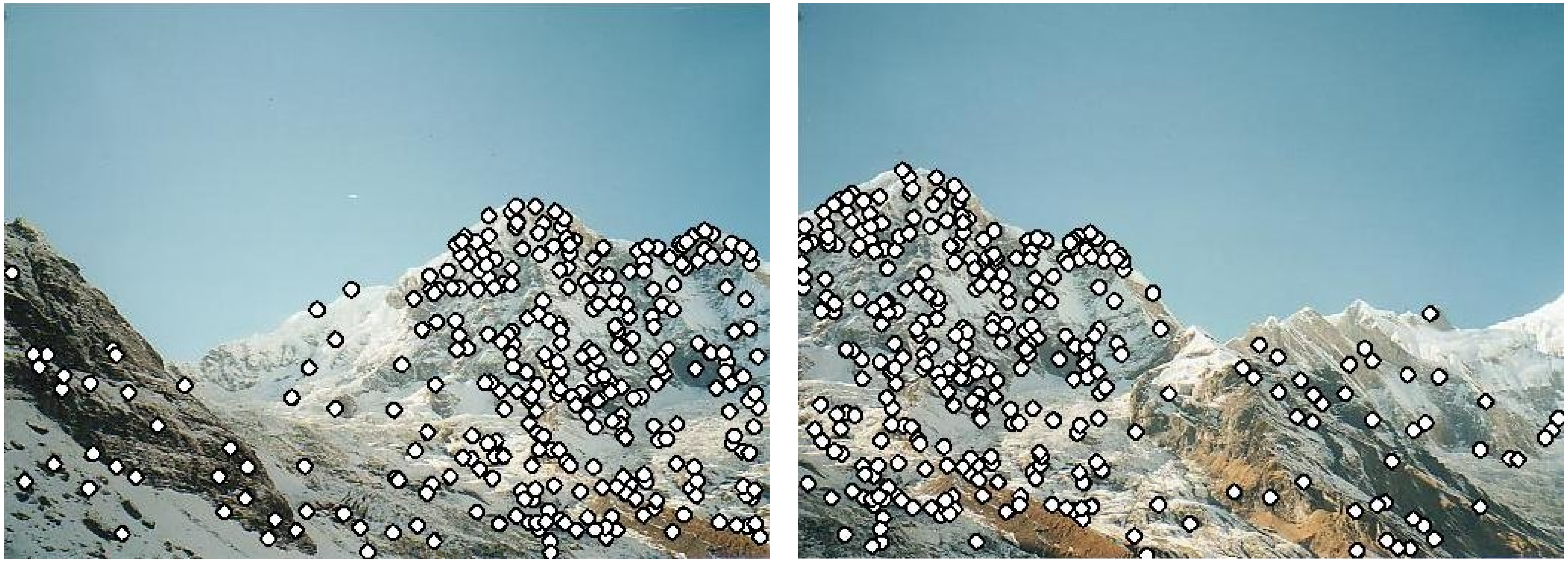


panorama?

- We need to match (align) images

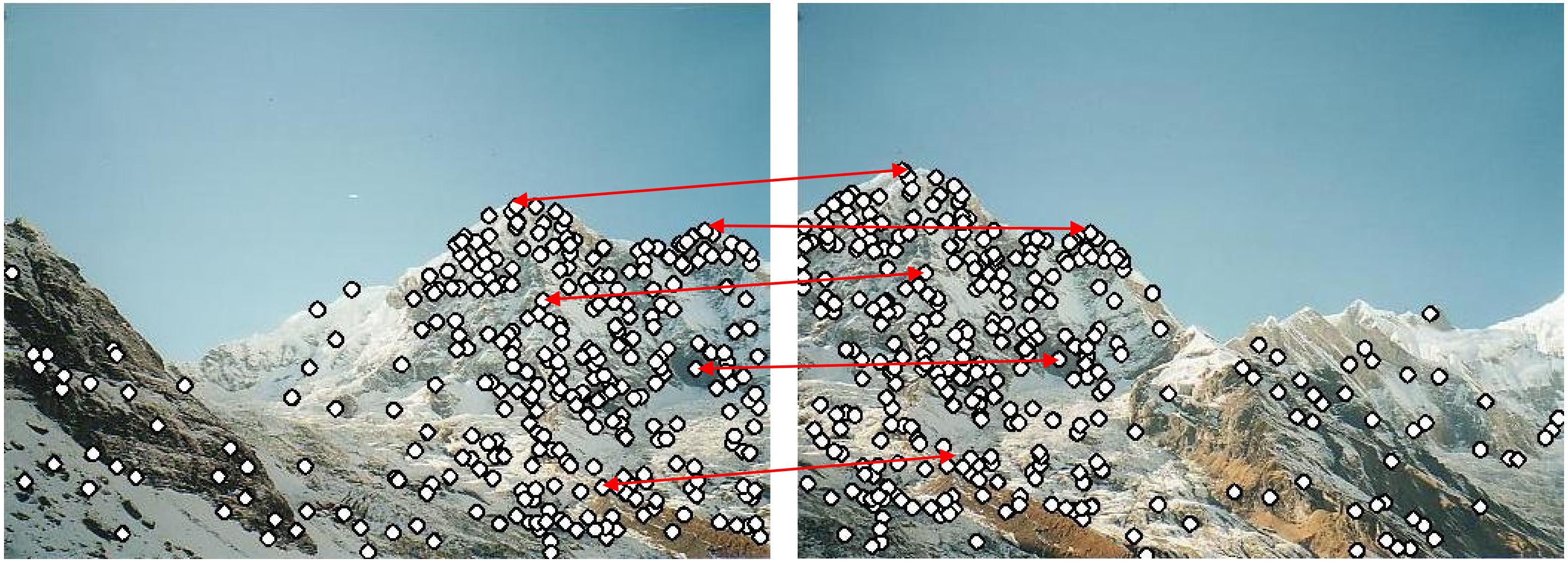
Matching with Features

- Detect feature points in both images



Matching with Features

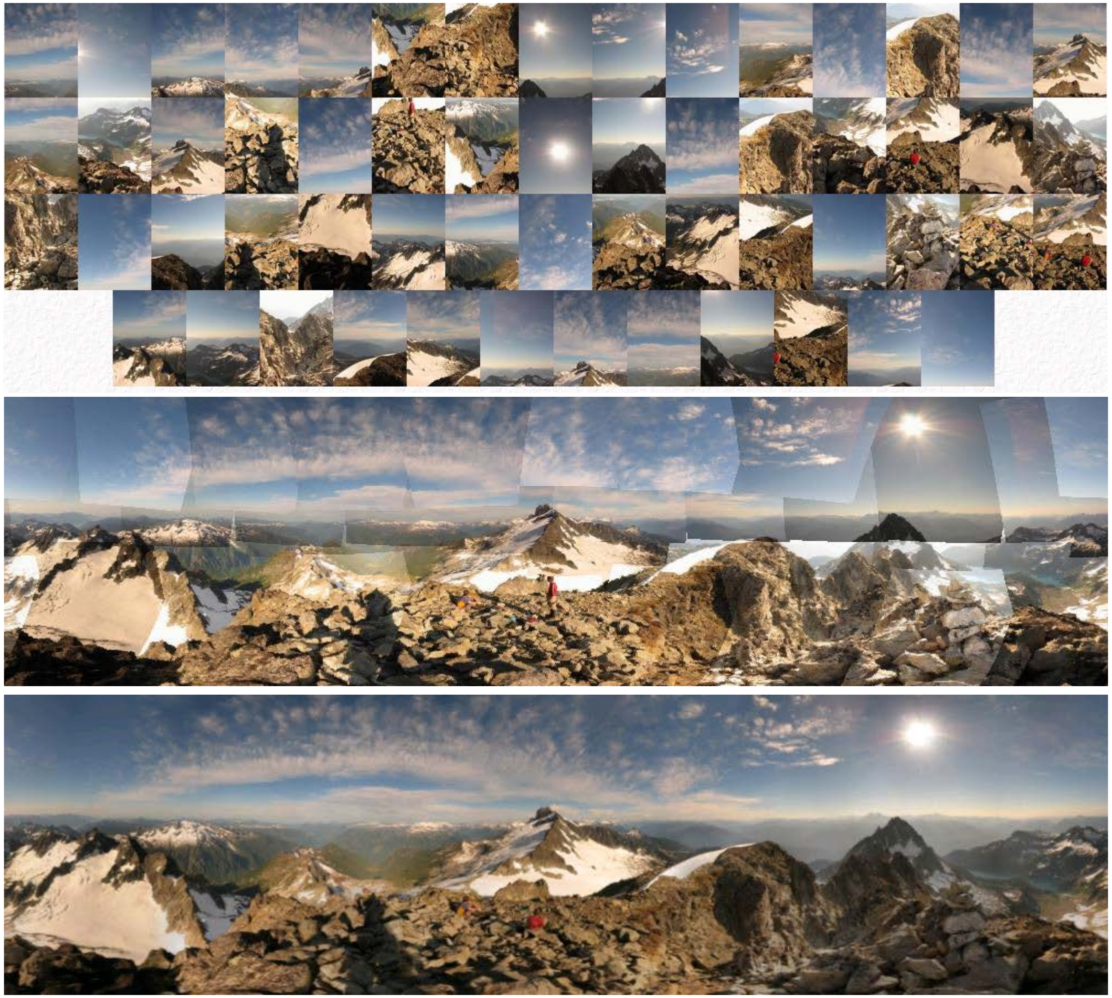
- Detect feature points in both images
- Find corresponding pairs



Matching with Features

- Detect feature points in both images
- Find corresponding pairs
- Use these matching pairs to align images - the required mapping is called a homography.

Automatic mosaicing



http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

Recognition of specific objects, scenes

Schmid and Mohr 1997

Sivic and Zisserman, 2003

Rothganger et al. 2003

Lowe 2002

Kristen Grauman

Example: 3D Reconstructions

- Photosynth

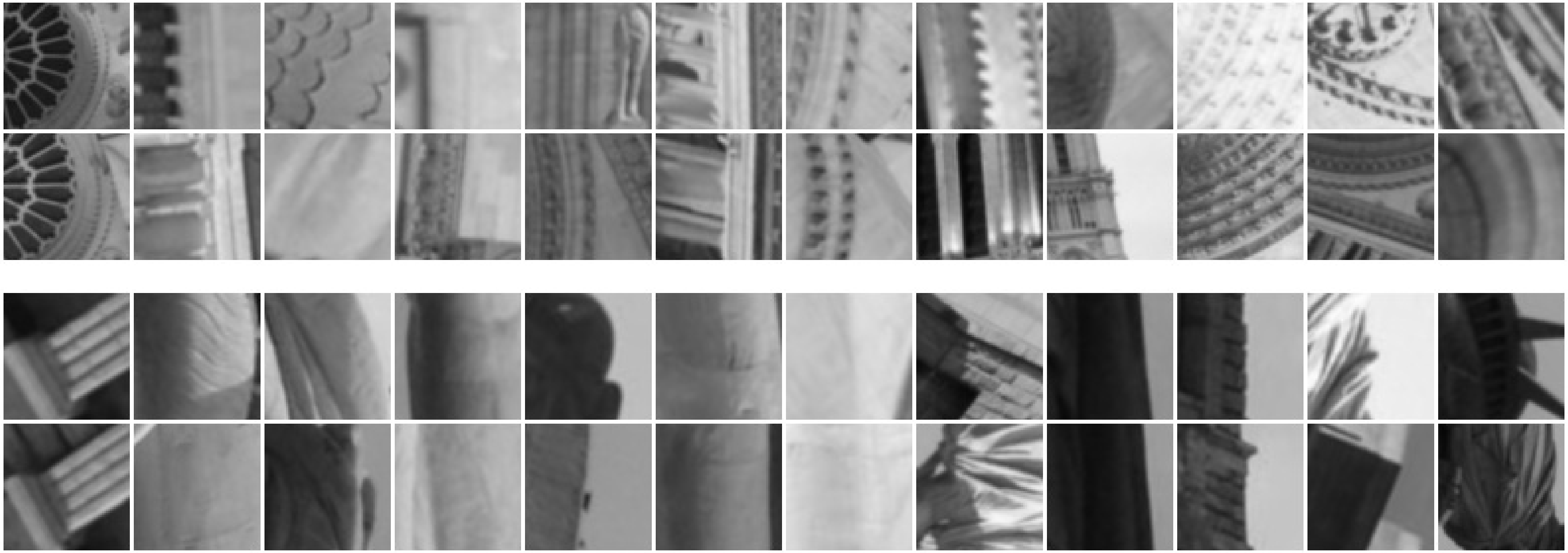
http://www.youtube.com/watch?v=p16frKJLVi0

- Building Rome in a day

http://www.youtube.com/watch?v=kxtQqYLRaSQ&feature=player_embedded

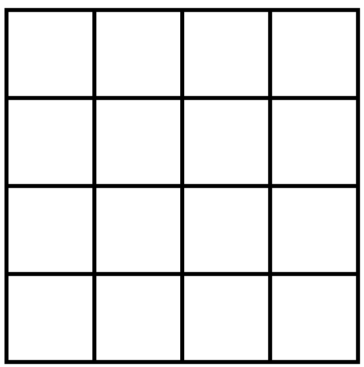
When does the SIFT descriptor fail?

Patches SIFT thought were the same but aren't:

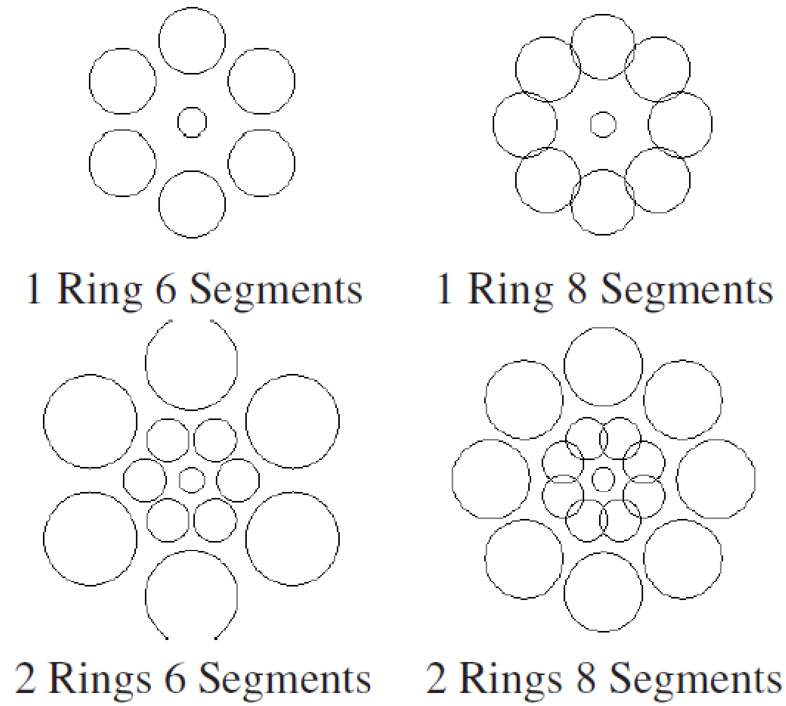


Other methods: Daisy

Circular gradient binning



SIFT



Daisy

Picking the best DAISY, S. Winder, G. Hua, M. Brown, CVPR 09

Other methods: SURF

For computational efficiency only compute gradient histogram with 4 bins:

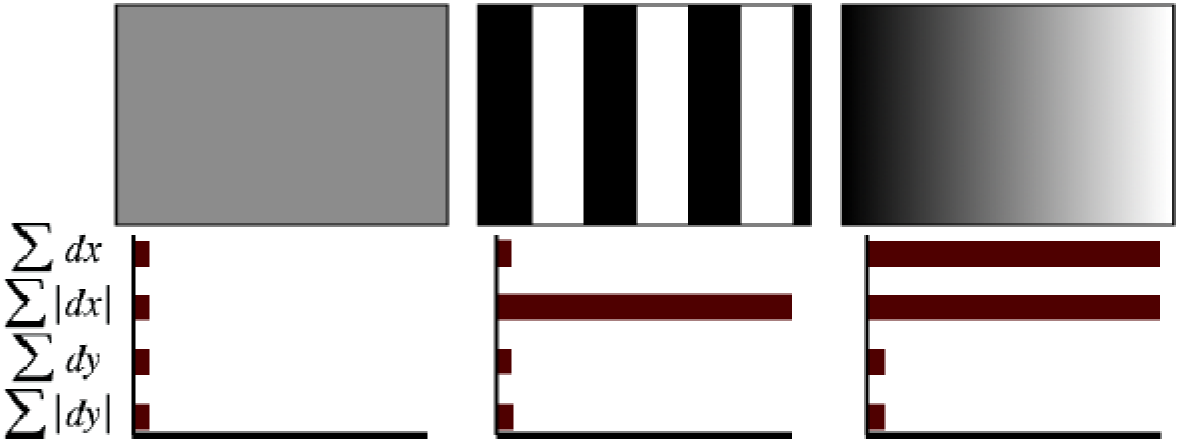


Fig. 3. The descriptor entries of a sub-region represent the nature of the underlying intensity pattern. Left: In case of a homogeneous region, all values are relatively low. Middle: In presence of frequencies in x direction, the value of $\sum |d_x|$ is high, but all others remain low. If the intensity is gradually increasing in x direction, both values $\sum d_x$ and $\sum |d_x|$ are high.

SURF: Speeded Up Robust Features
Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, ECCV 2006

Other methods: BRIEF

Randomly sample pair of pixels a and b.
1 if a > b, else 0. Store binary vector.

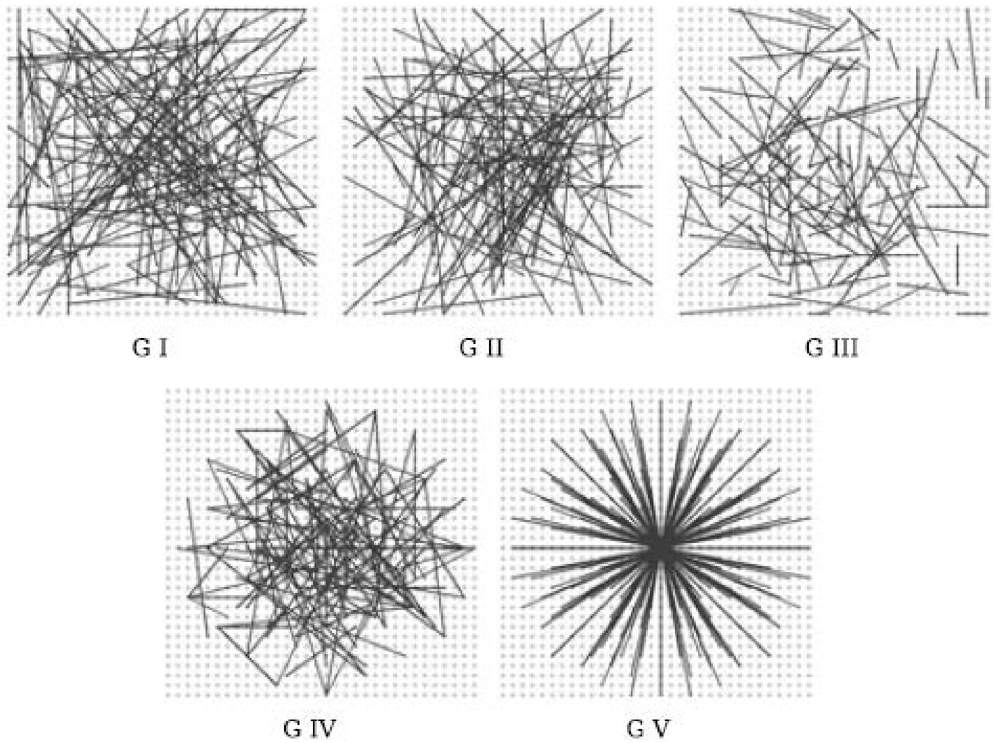


Fig. 2. Different approaches to choosing the test locations. All except the righmost one are selected by random sampling. Showing 128 tests in every image.

BRIEF: binary robust independent elementary features,
Calonder, V Lepetit, C Strecha, ECCV 2010

Descriptors and Matching

- The SIFT descriptor and the various variants are used to describe an image patch, so that we can match two image patches.

- In addition to the descriptors, we need a distance measure to calculate how different the two patches are?

?

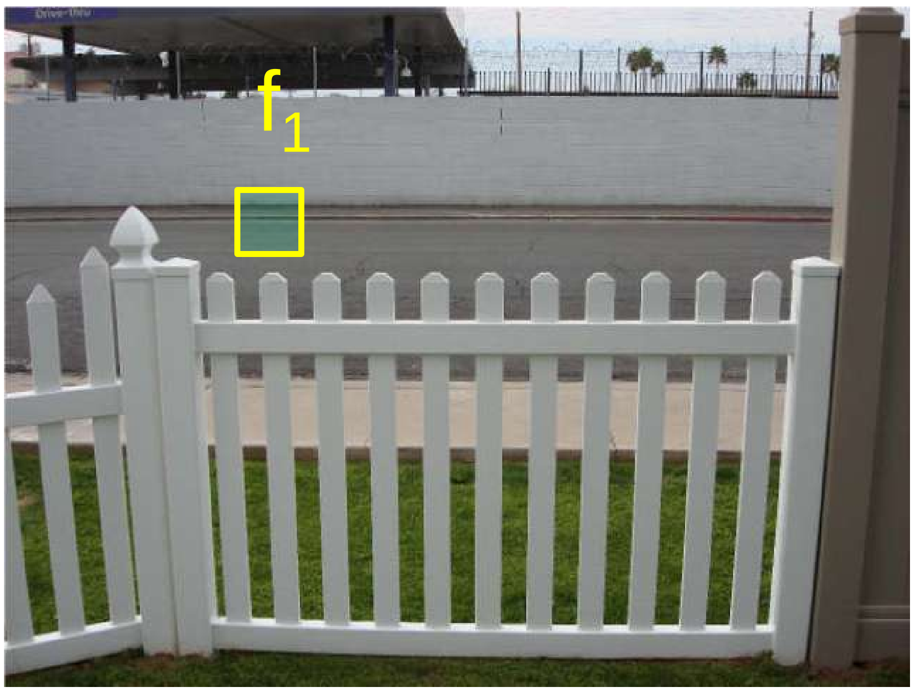
Feature distance

How to define the difference between two features f_1, f_2?

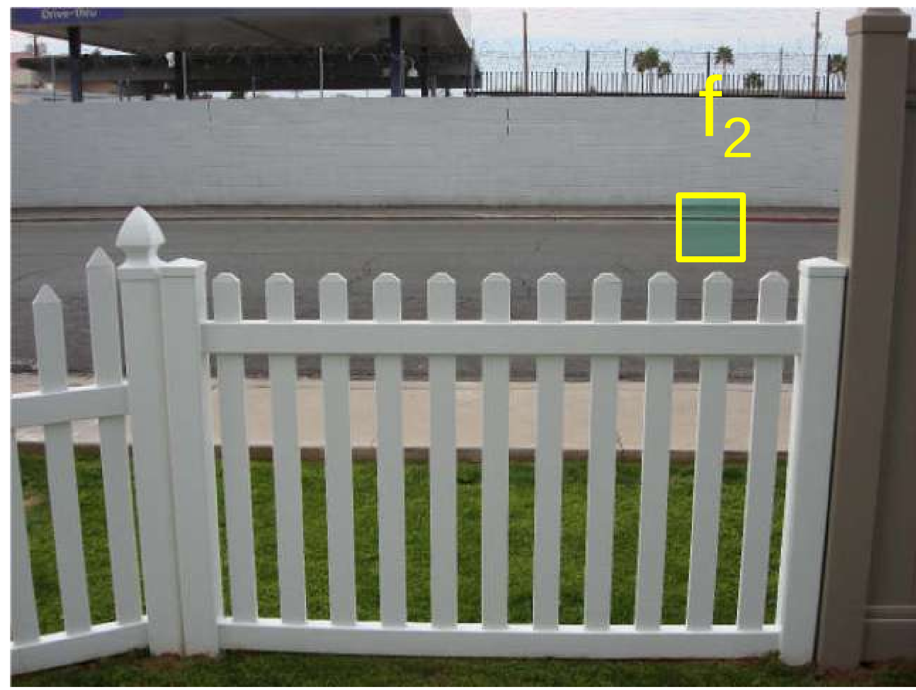
- Simple approach is $SSD(f_1, f_2)$
  - sum of square differences between entries of the two descriptors

$$\sum_i (f_{1i} - f_{2i})^2$$

  - But it can give good scores to very ambiguous (bad) matches



I_1

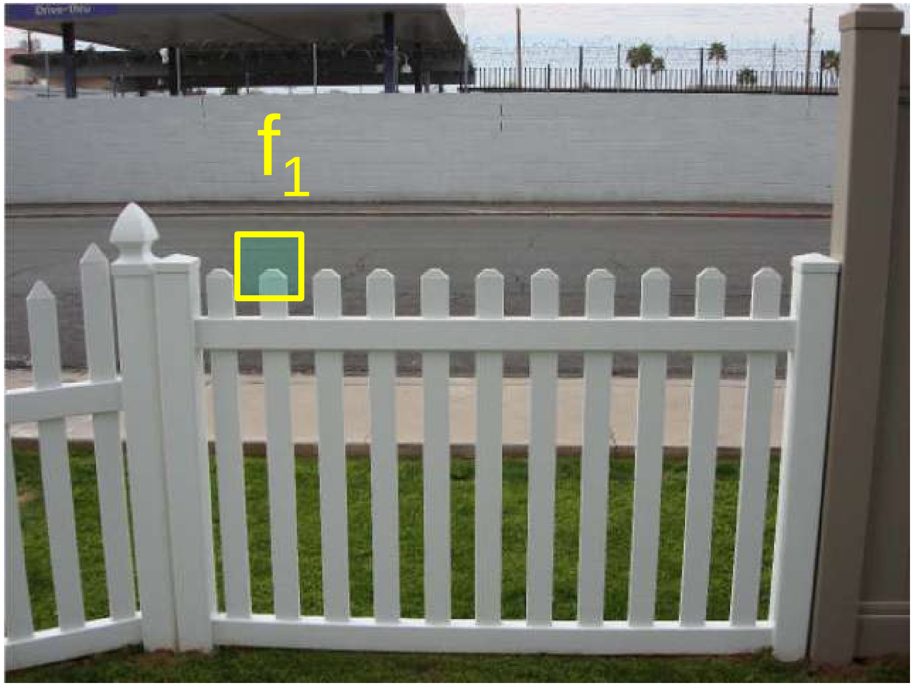


I_2

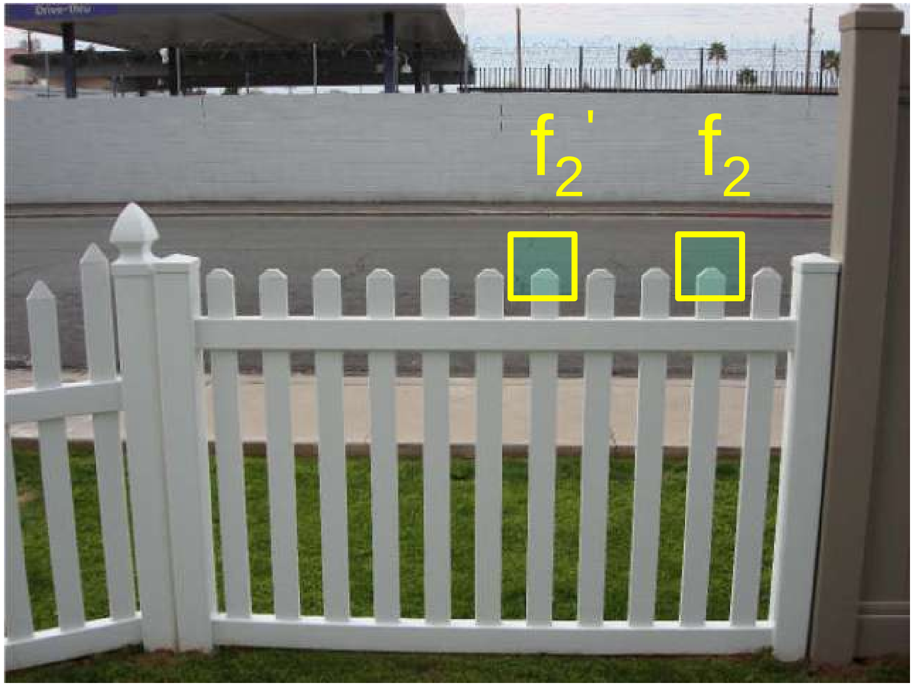
Feature distance in practice

How to define the difference between two features f_1, f_2?

- Better approach: ratio distance = $SSD(f_1, f_2) / SSD(f_1, f_2')$
  - f_2 is best SSD match to f_1 in I_2
  - f_2' is 2^{nd} best SSD match to f_1 in I_2
  - gives large values (~1) for ambiguous matches WHY?

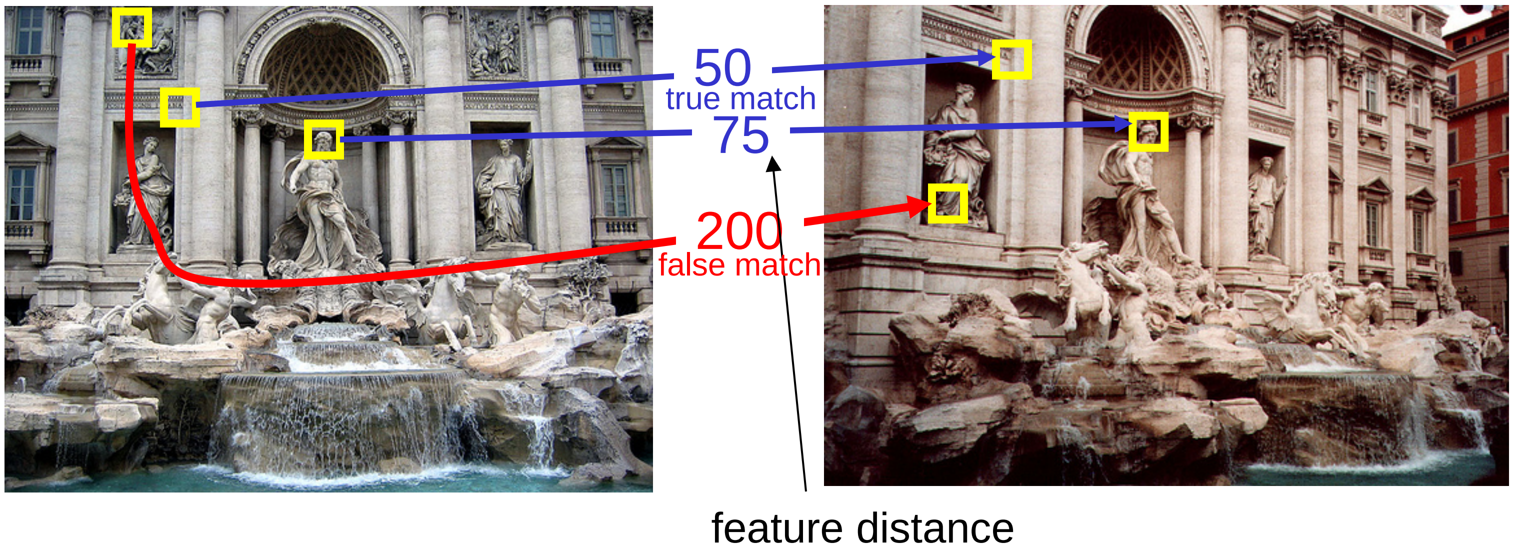


I_1



I_2

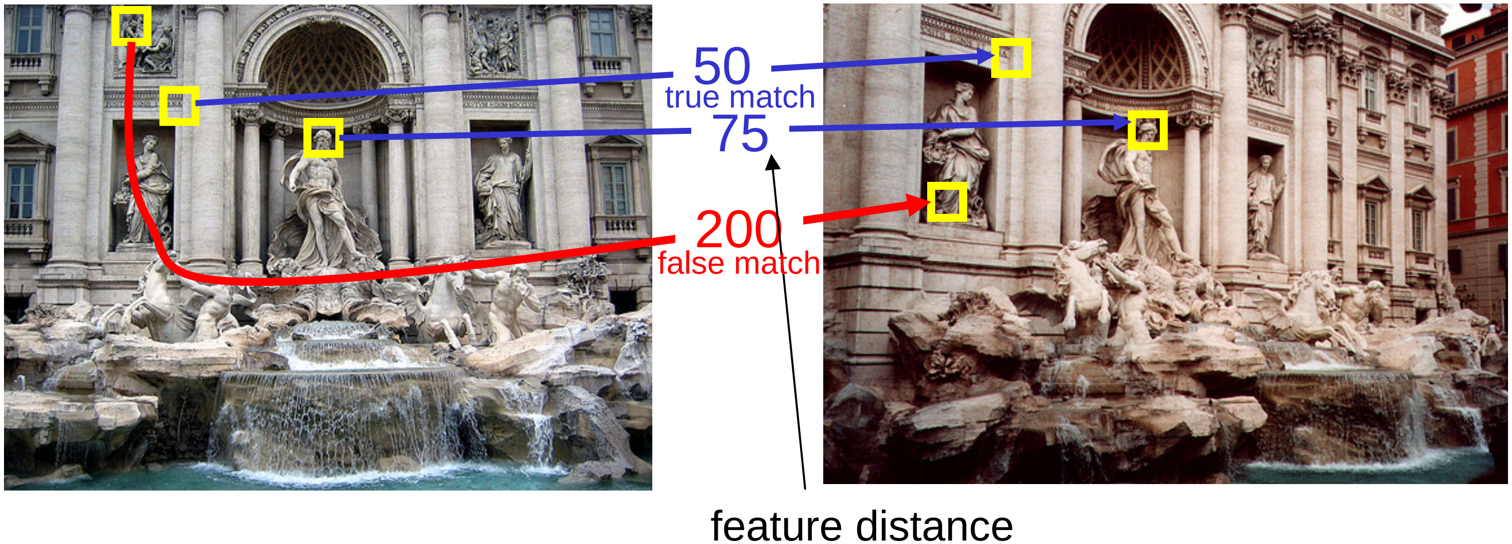
Eliminating more bad matches



Throw out features with distance > threshold

- How to choose the threshold?

True/false positives



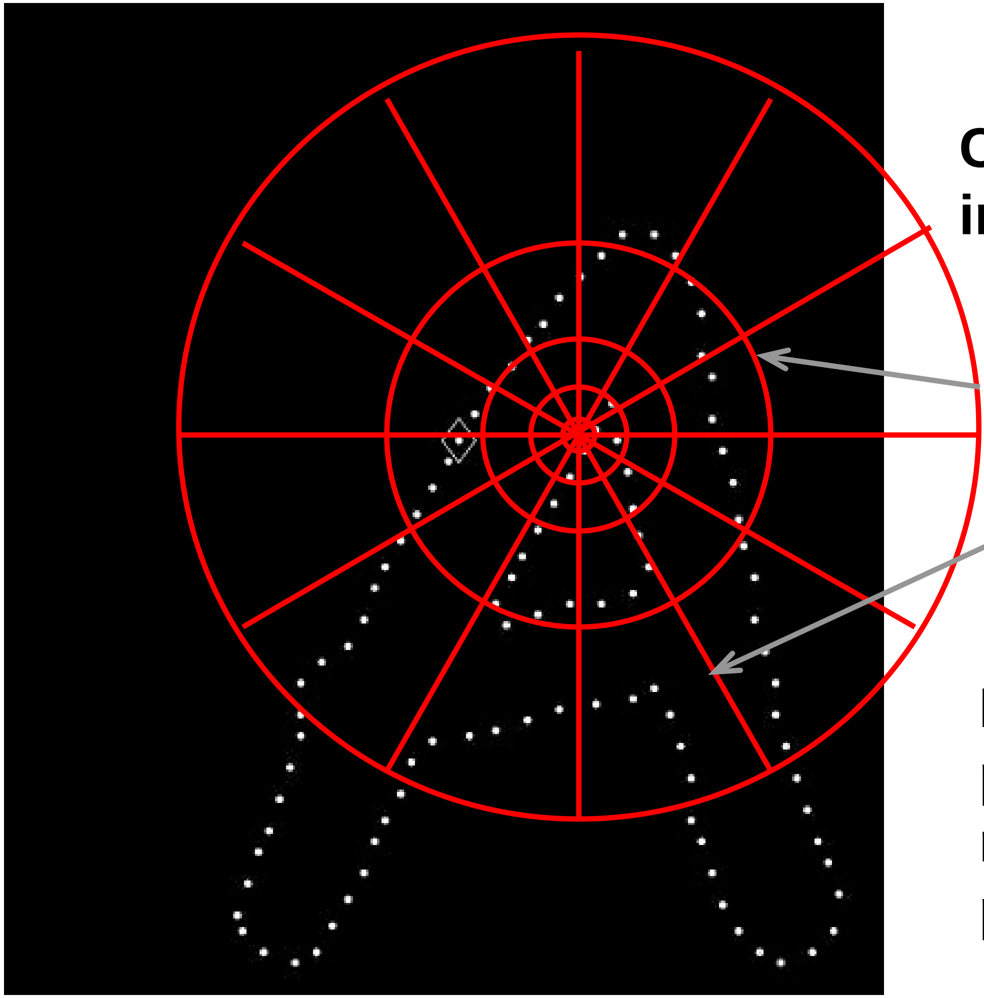
The distance threshold affects performance

- True positives = # of detected matches that are correct
  - Suppose we want to maximize these—how to choose threshold?
- False positives = # of detected matches that are incorrect
  - Suppose we want to minimize these—how to choose threshold?

Other kinds of descriptors

- There are descriptors for other purposes
  - Describing shapes
  - Describing textures
  - Describing features for image classification
  - Describing features for a code book

Local Descriptors: Shape Context



Count the number of points inside each bin, e.g.:

Count = 4

⋮

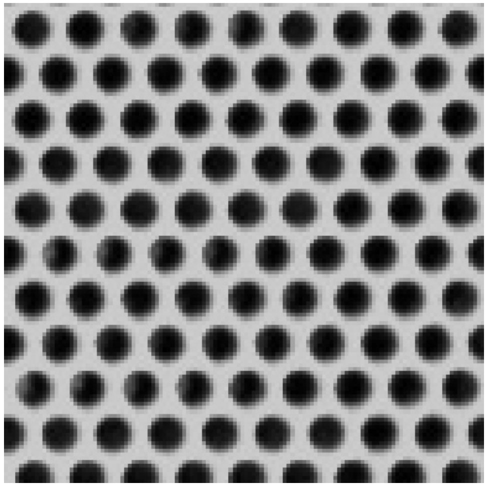
Count = 10

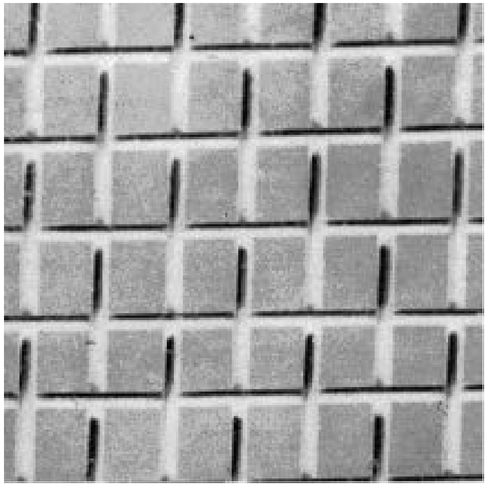
Log-polar binning: more precision for nearby points, more flexibility for farther points.

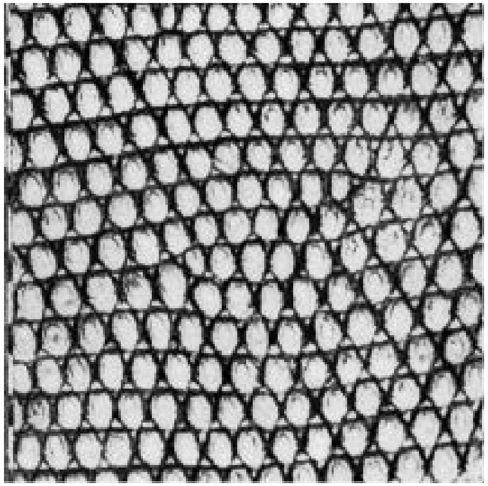
Belongie & Malik, ICCV 2001

Texture

- The texture features of a patch can be considered a descriptor.







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

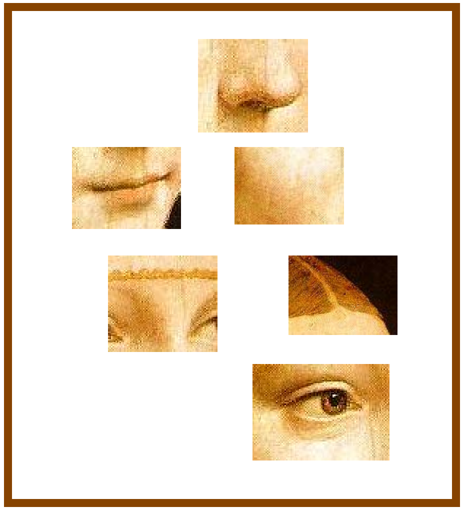
- Orderless document representation: frequencies of words from a dictionary   Salton & McGill (1983)

Bag-of-words models

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

Bags of features for image classification

1. Extract features



Bags of features for image classification

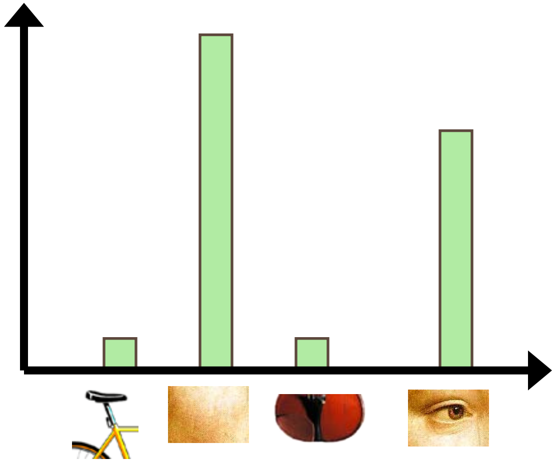
1. Extract features
2. Learn "visual vocabulary"

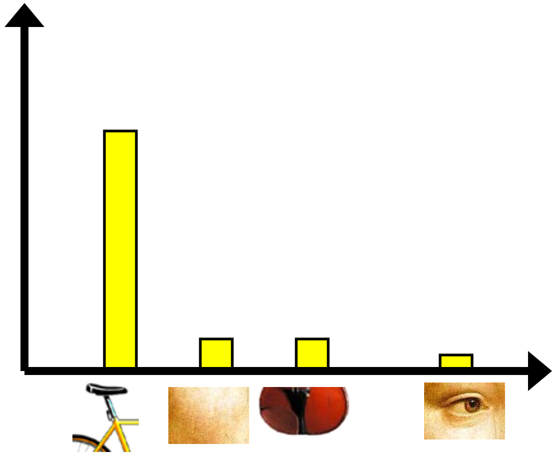
Bags of features for image classification

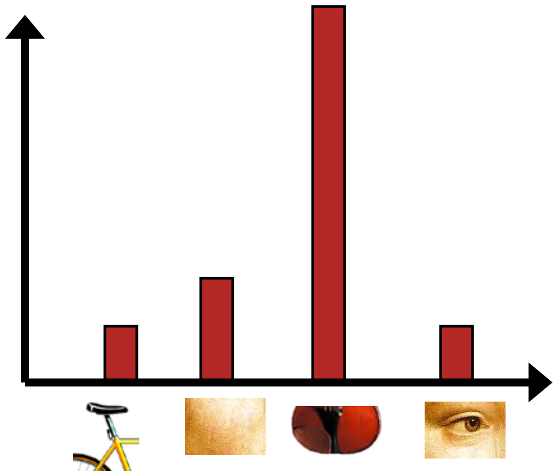
1. Extract features
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary

Bags of features for image classification

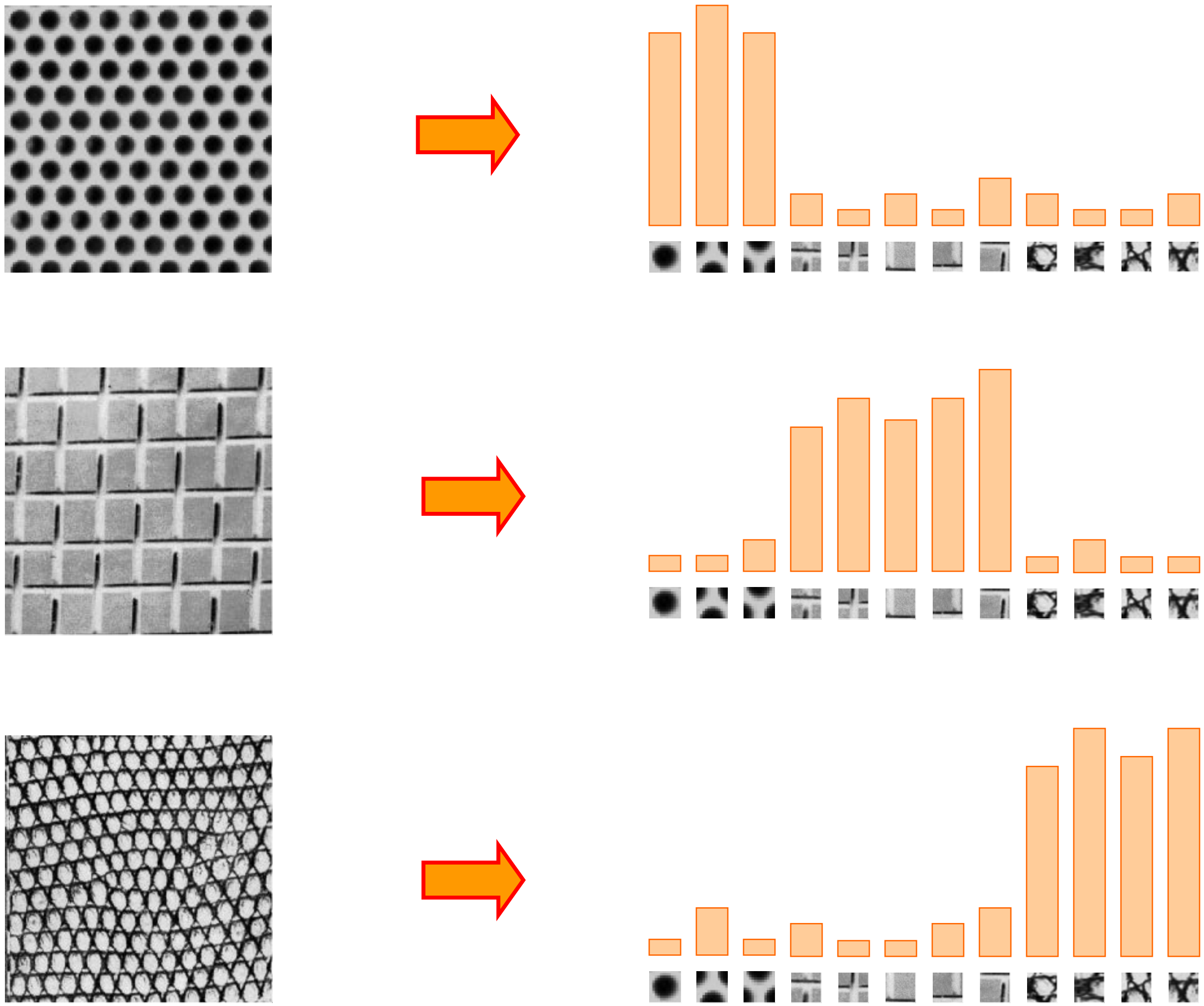
1. Extract features
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary
4. Represent images by frequencies of “visual words”





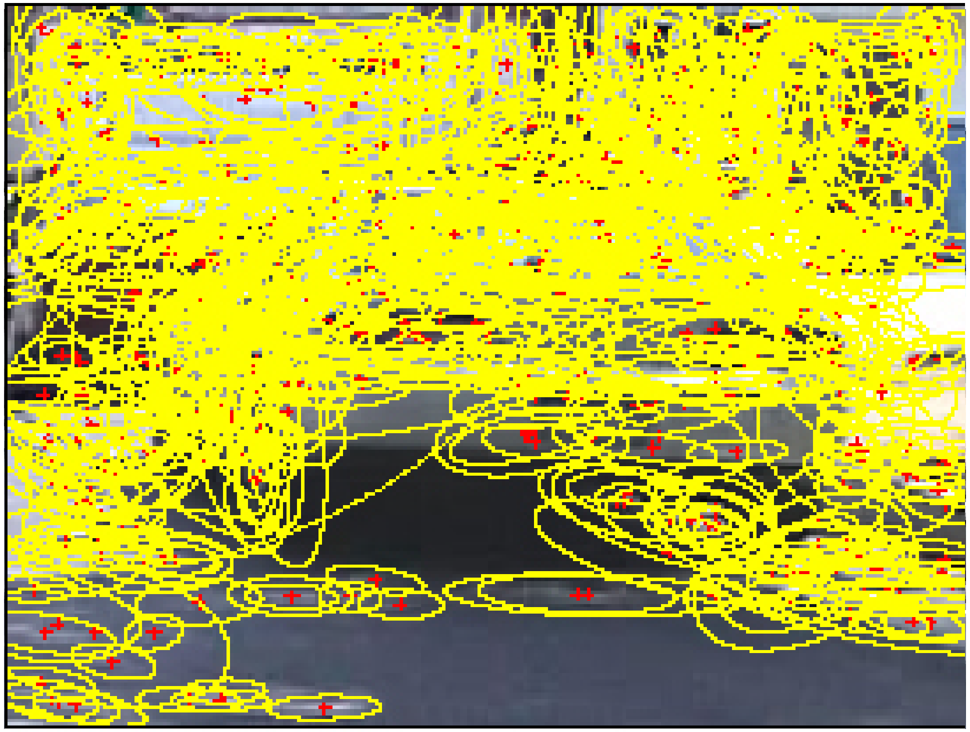


Texture representation

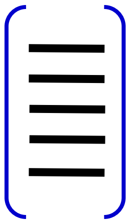


1. Feature extraction

- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
- Interest point detector
  - Csurka et al. 2004
  - Fei-Fei & Perona, 2005
  - Sivic et al. 2005

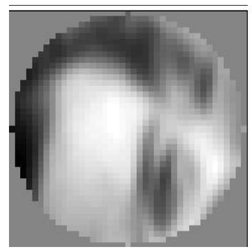


1. Feature extraction

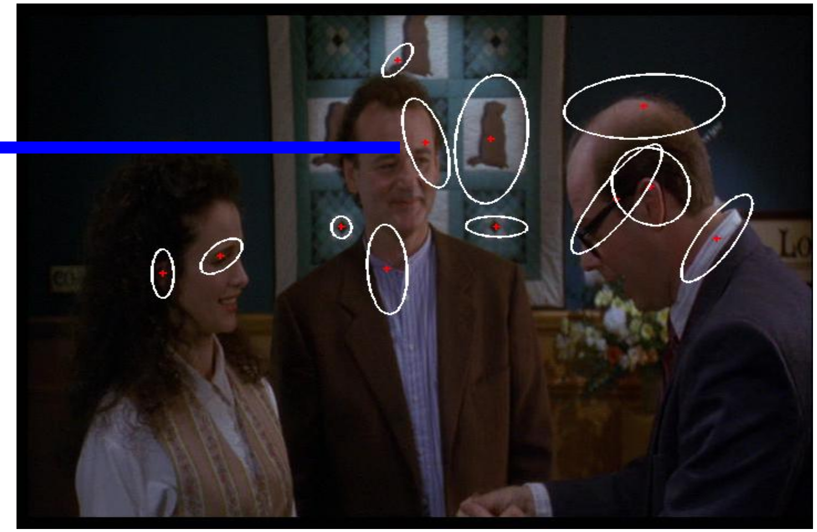


Compute SIFT descriptor

[Lowe'99]



Normalize patch



Detect patches

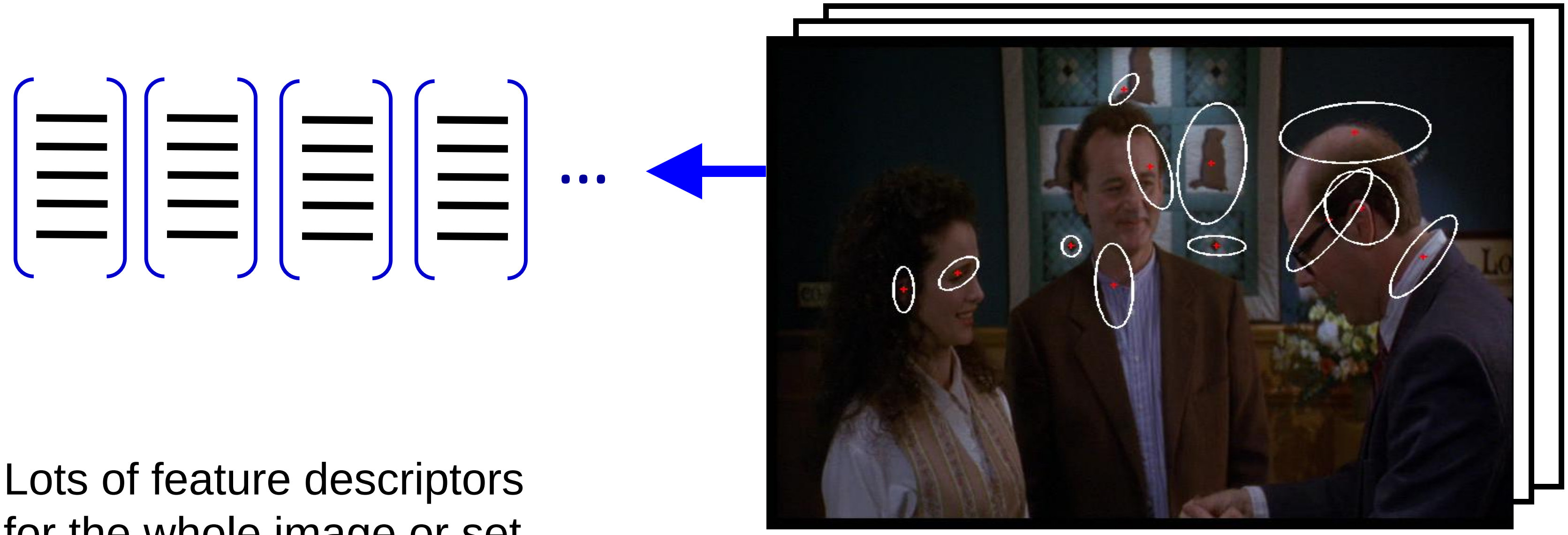
[Mikojaczyk and Schmid '02]

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

[Sivic & Zisserman, '03]

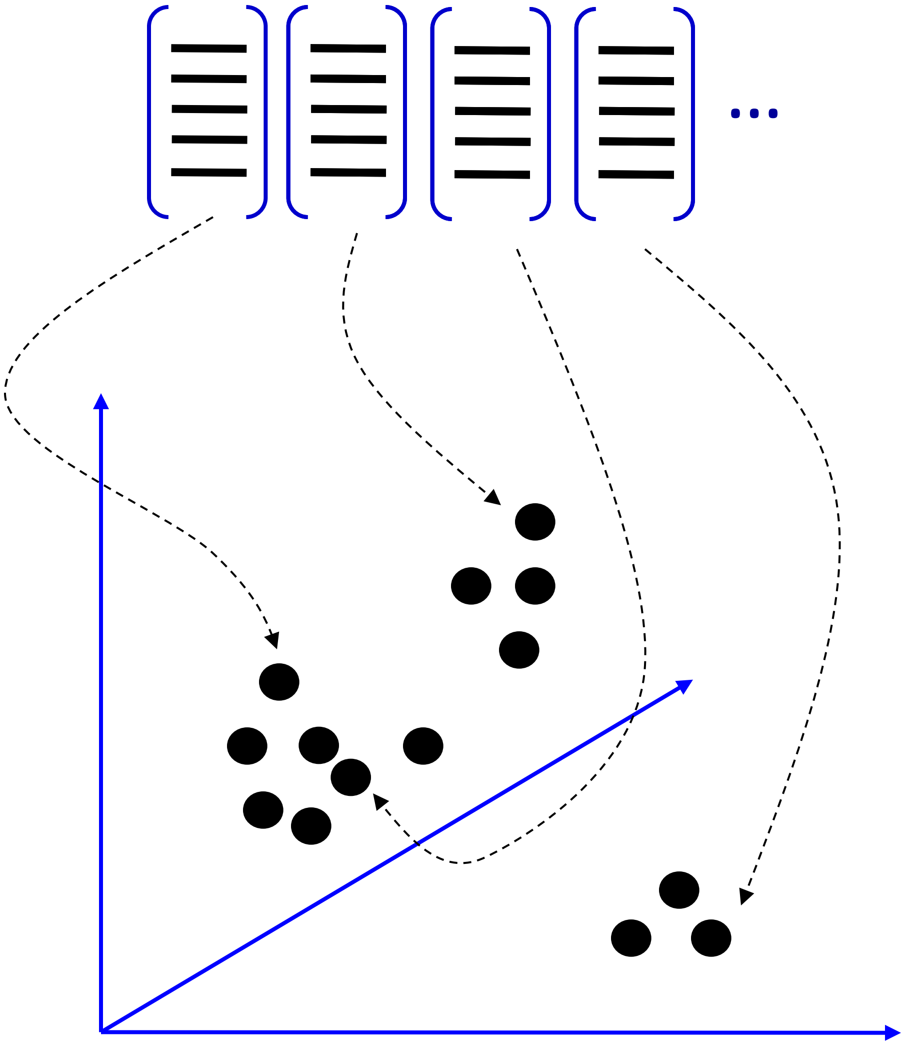
Slide credit: Josef Sivic

1. Feature extraction



Lots of feature descriptors for the whole image or set of images.

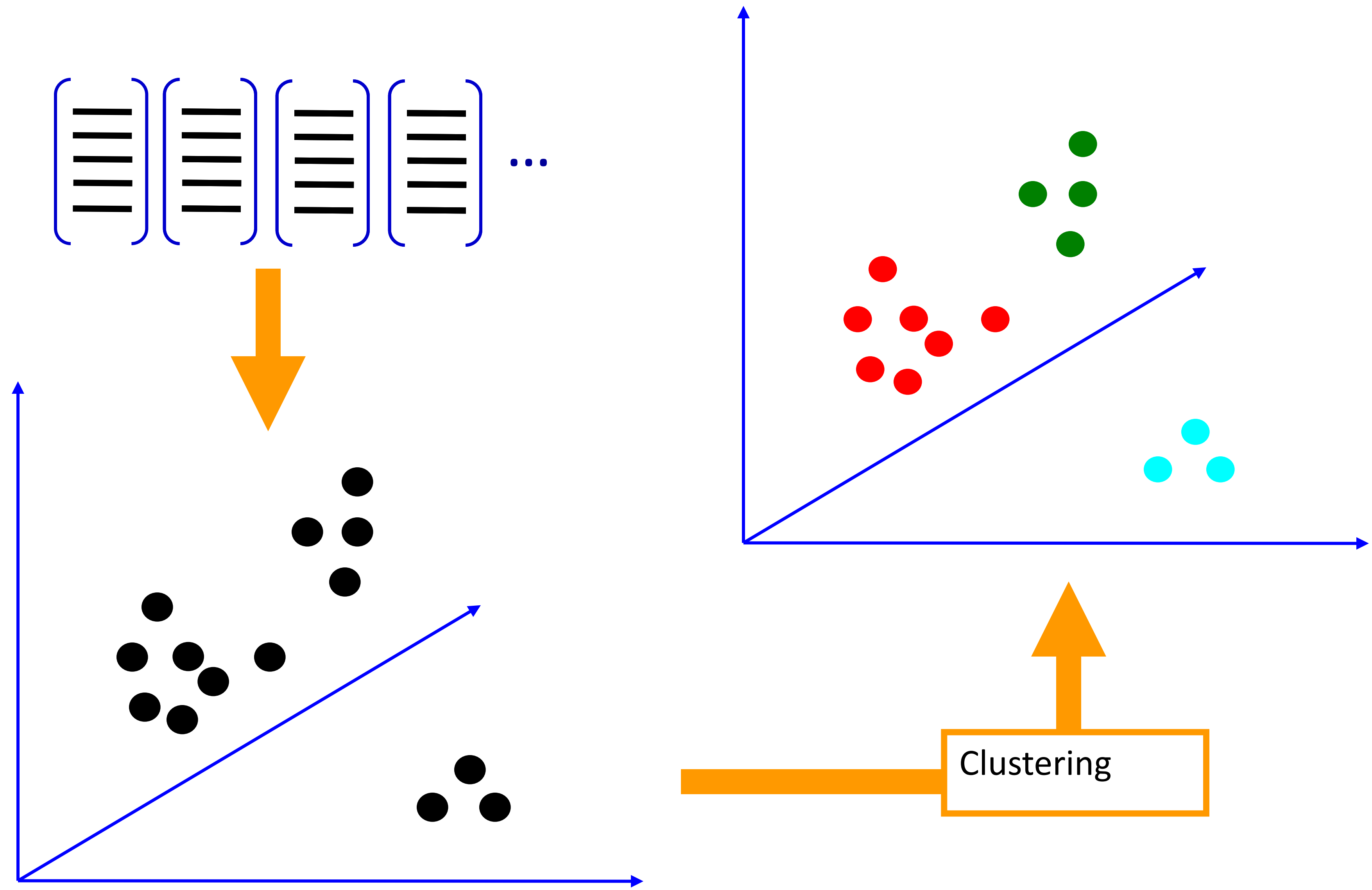
2. Discovering the visual vocabulary



feature vector space

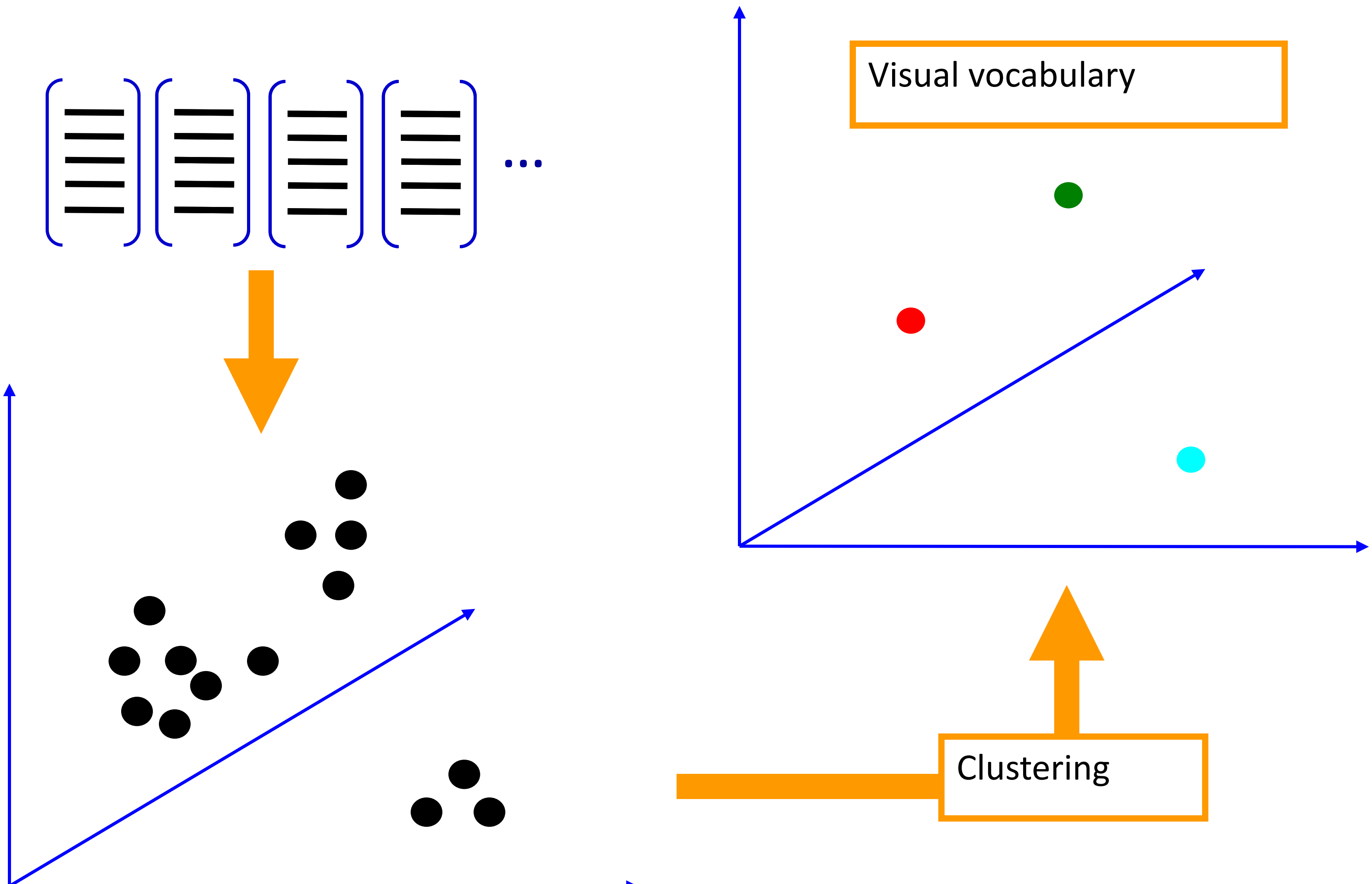
What is the dimensionality?

2. Discovering the visual vocabulary



Slide credit: Josef Sivic

2. Discovering the visual vocabulary



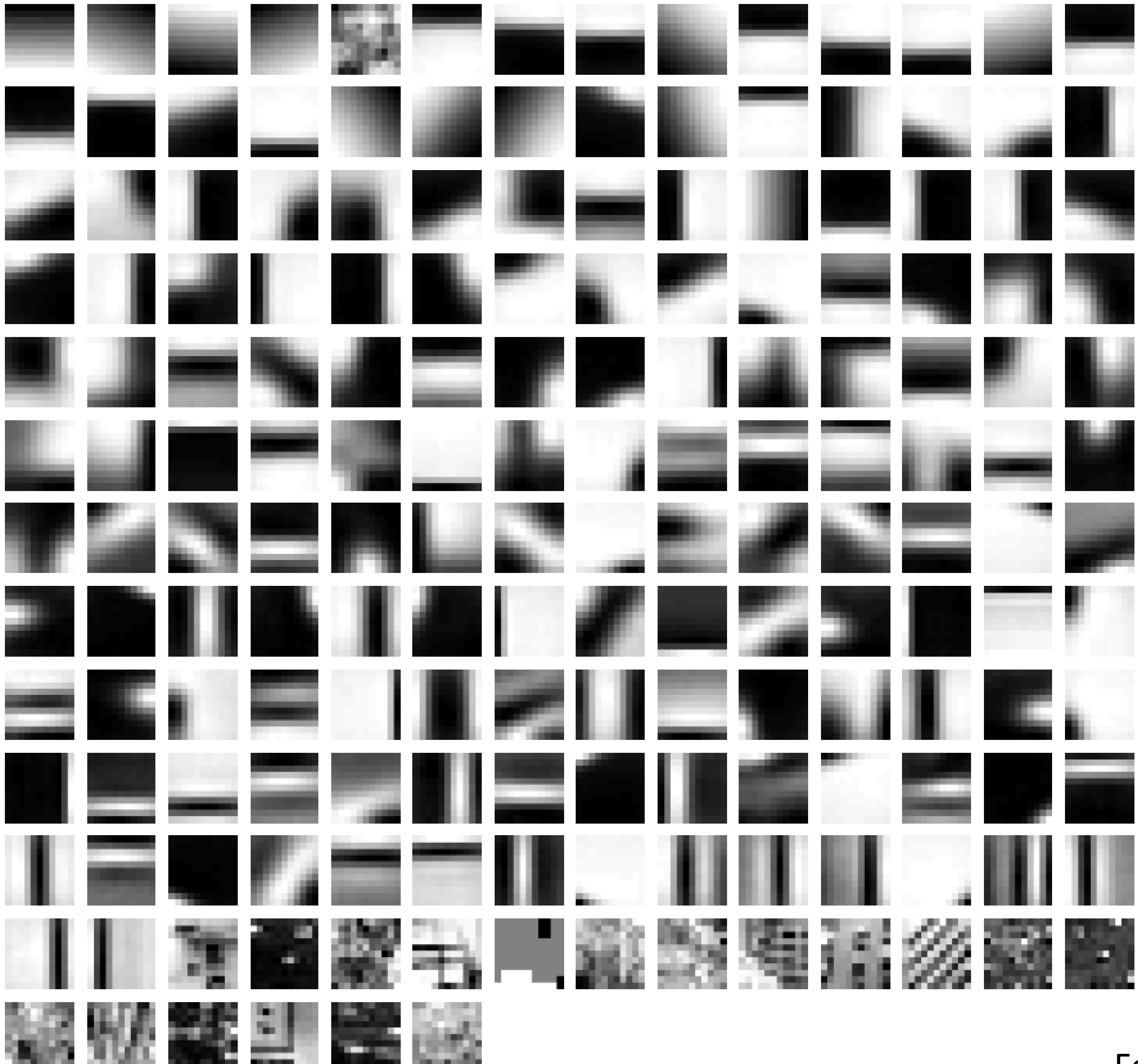
Slide credit: Josef Sivic

Clustering and 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"

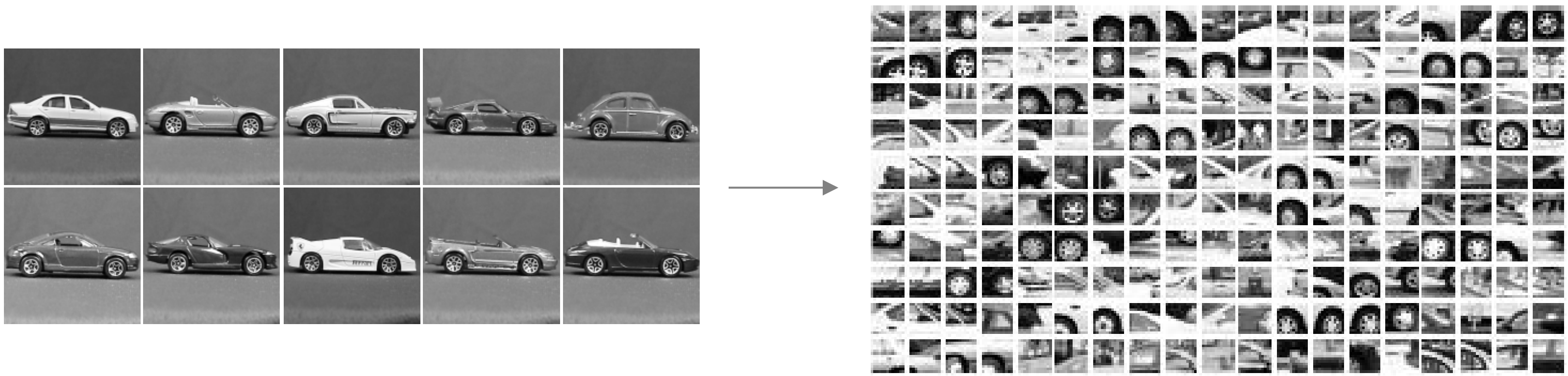
- The codebook is used for quantizing features
  - A *vector quantizer* takes a feature vector and maps it to the index of the nearest codevector in a codebook
  - Codebook = visual vocabulary
  - Codevector = visual word

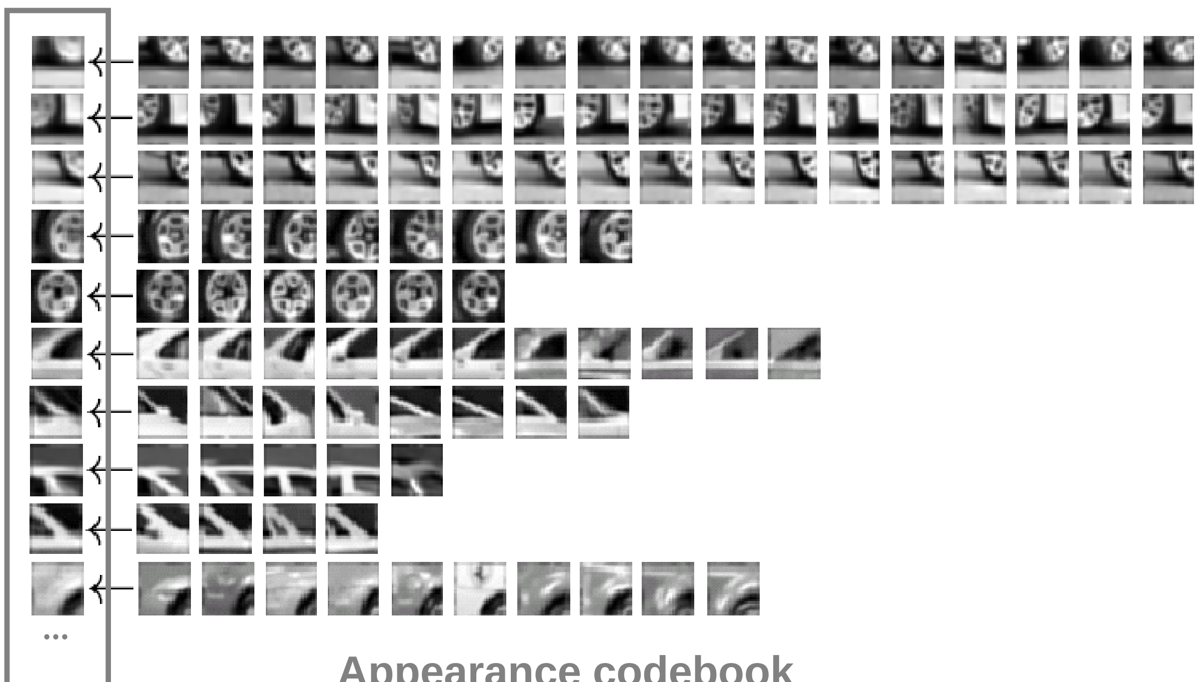
Example visual vocabulary



Fei-Fei et al. 2005

Example codebook

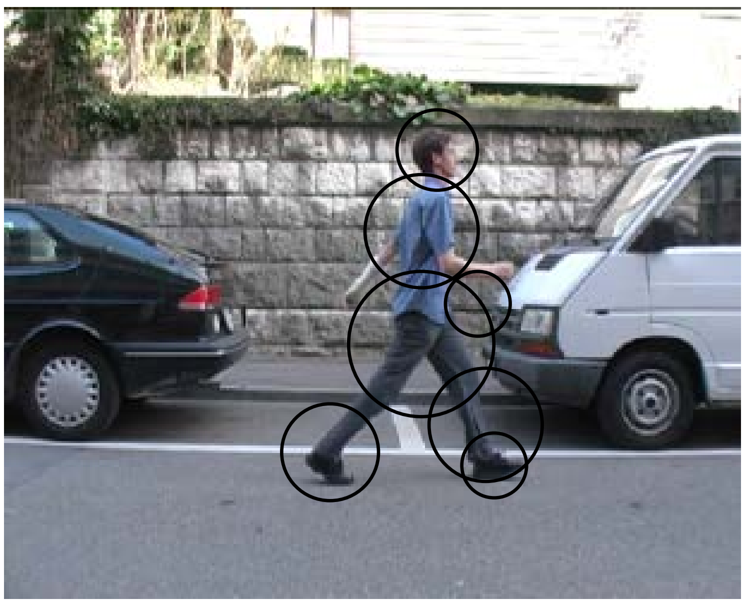


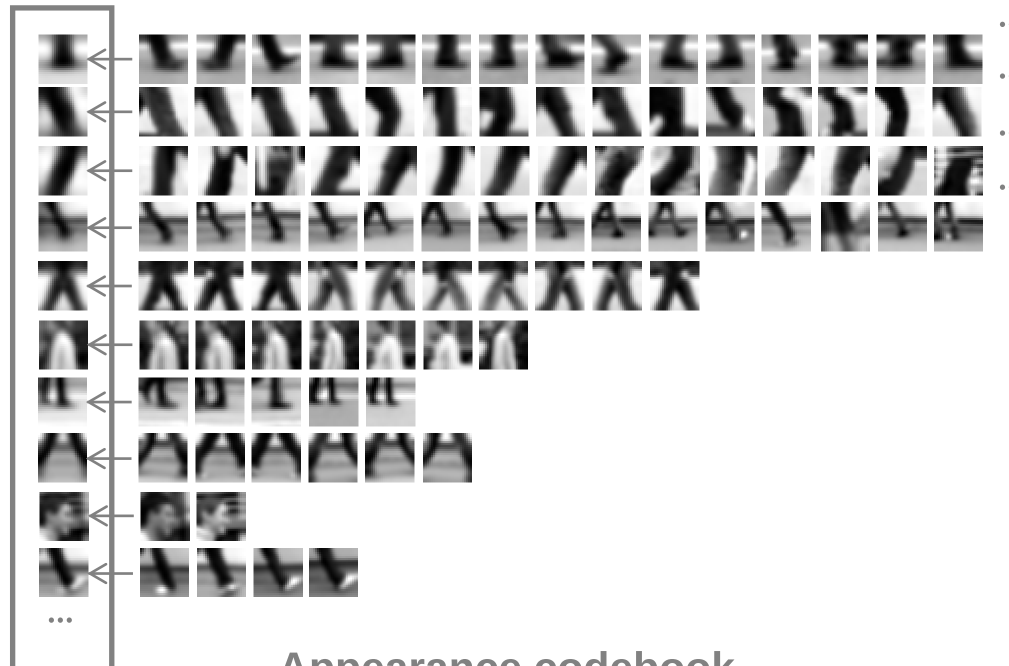


Appearance codebook

Source: B. Leibe

Another codebook





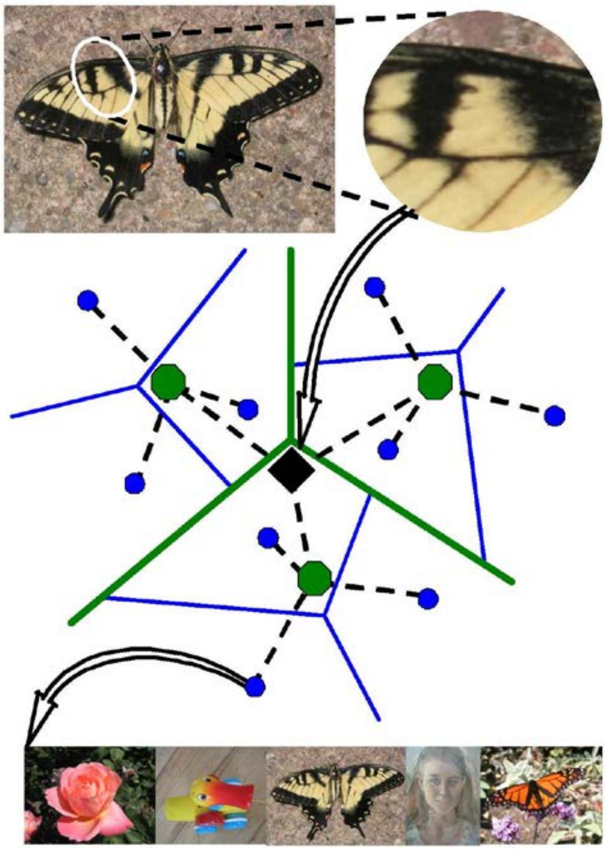
Appearance codebook

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)



3. Image representation: histogram of codewords

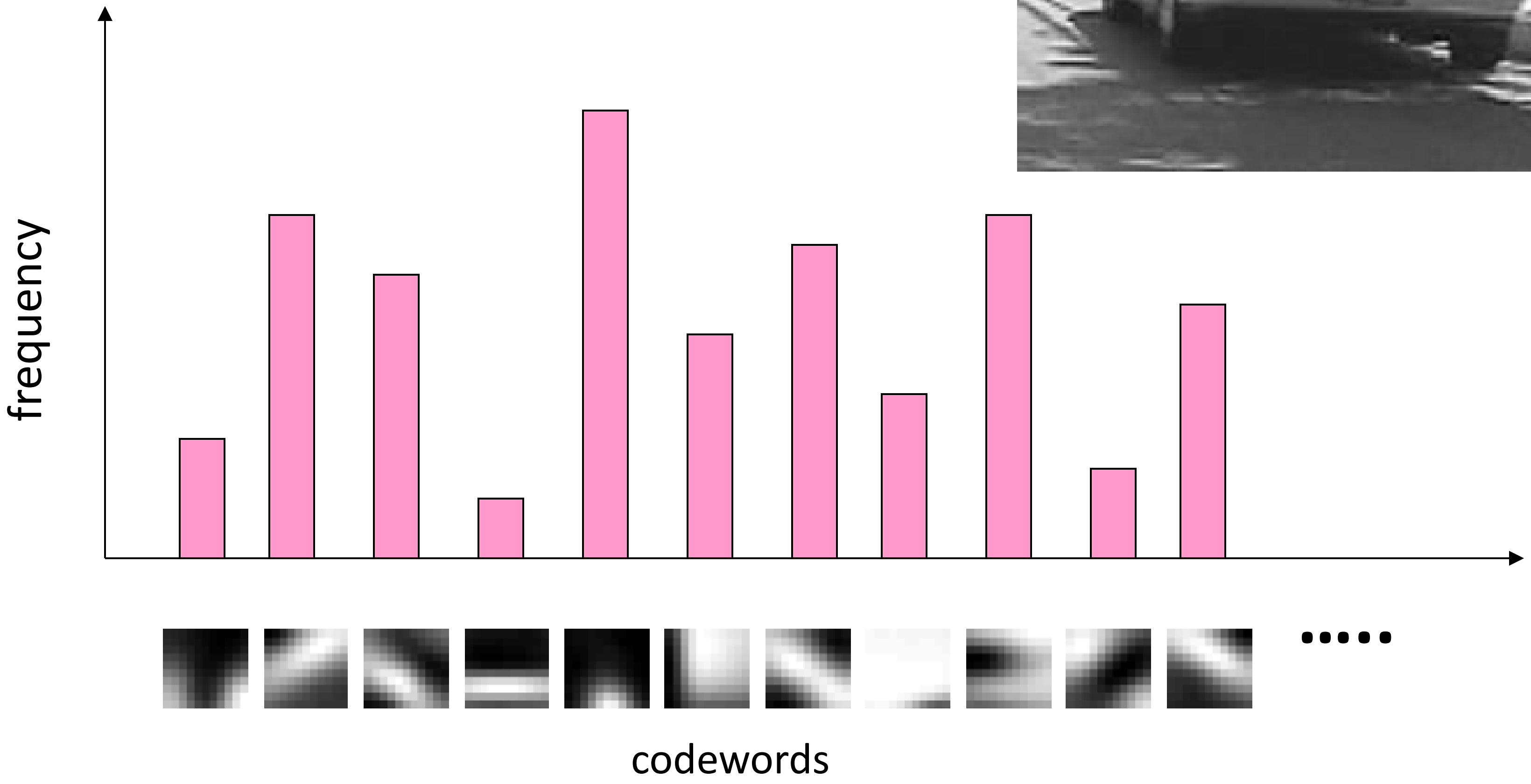
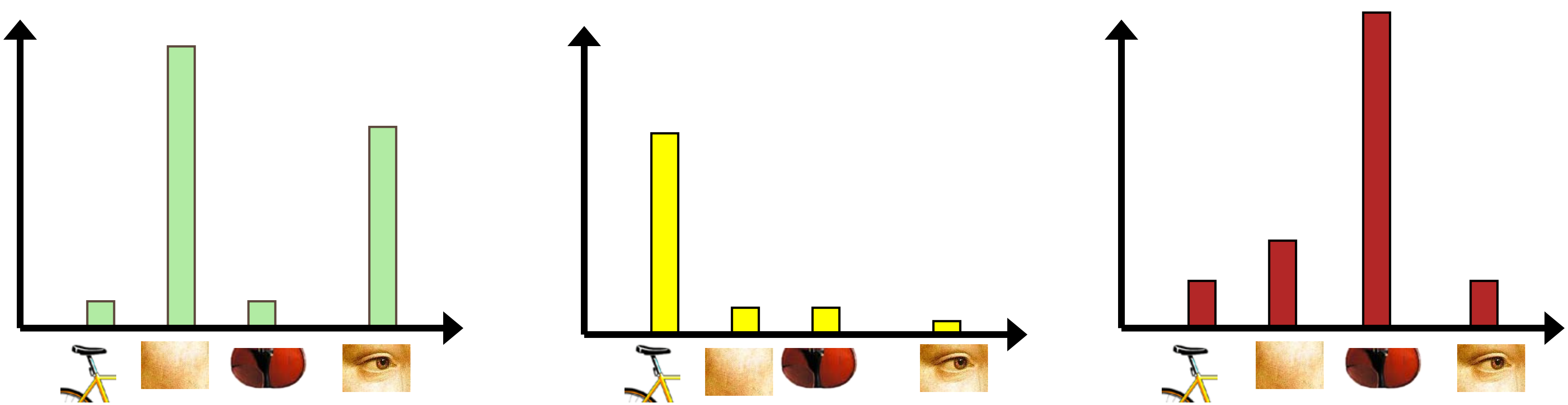
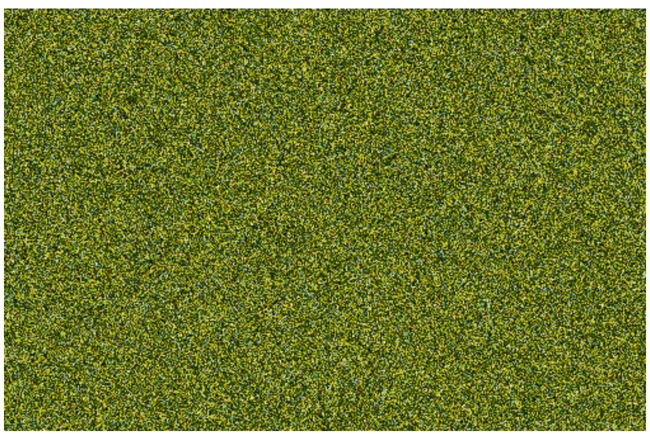


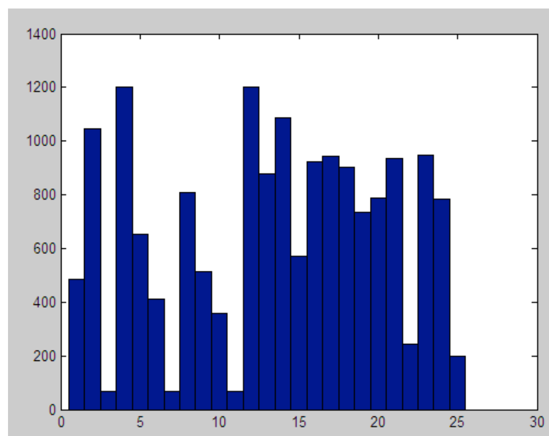
Image classification

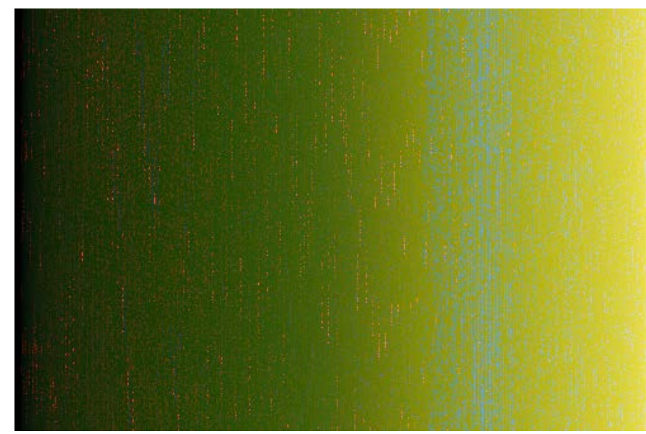
- Given the bag-of-features representations of images from different classes, learn a classifier using machine learning



But what about layout?



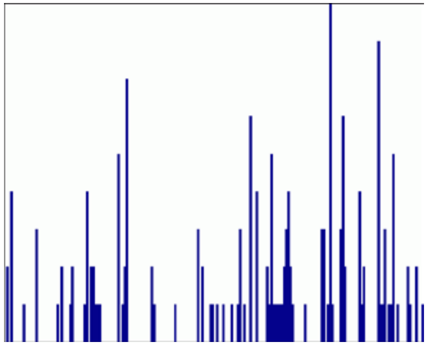




All of these images have the same color histogram

Spatial pyramid representation

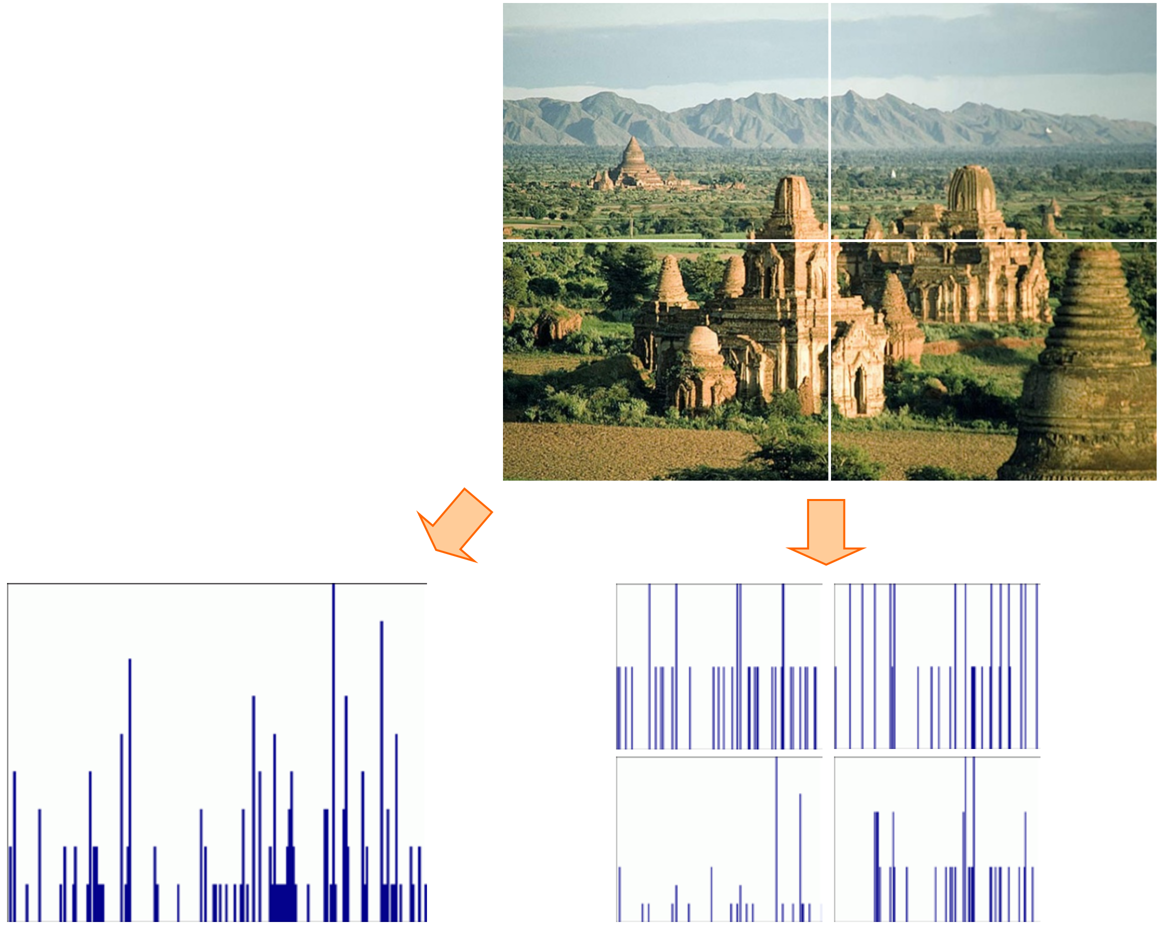
- Extension of a bag of features
- Locally orderless representation at several levels of resolution



Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramid representation

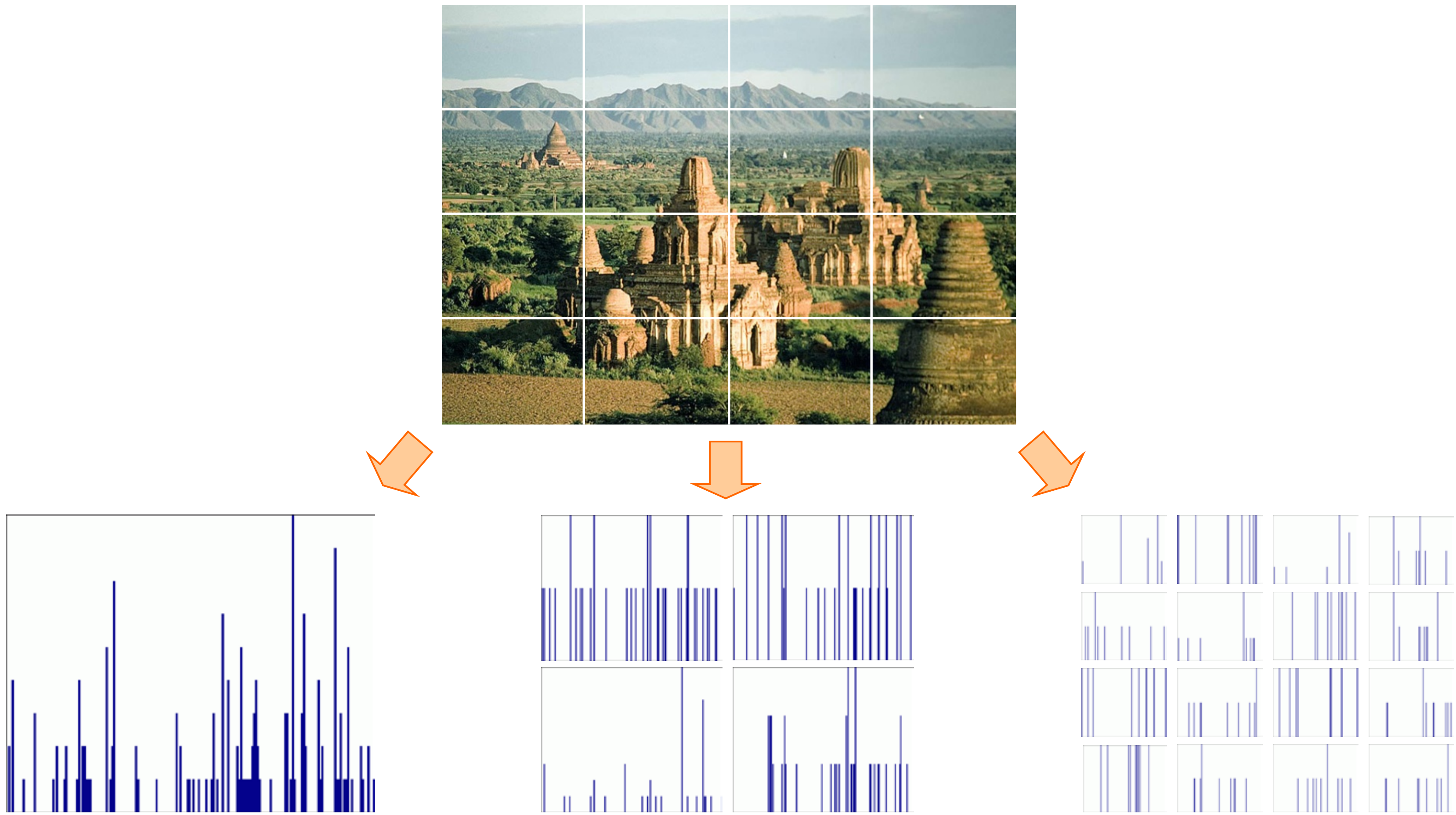
- Extension of a bag of features
- Locally orderless representation at several levels of resolution



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)