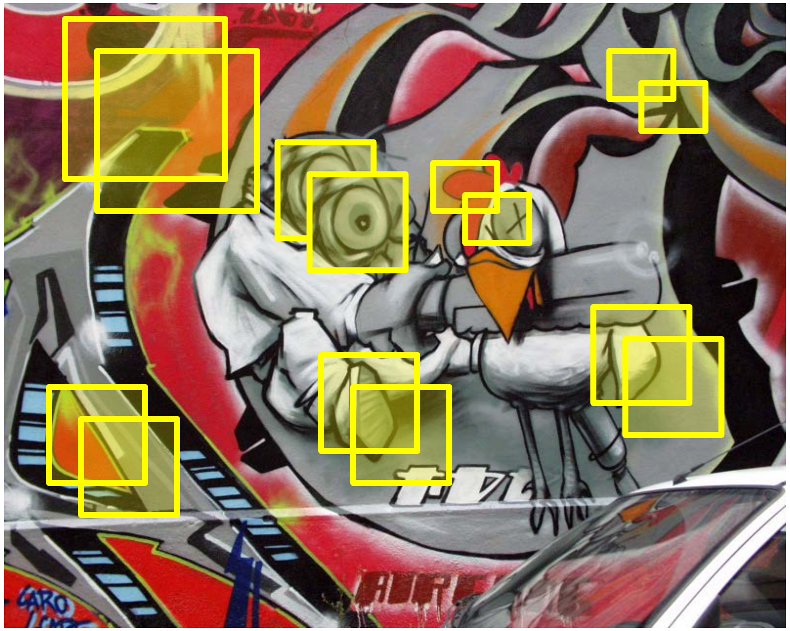
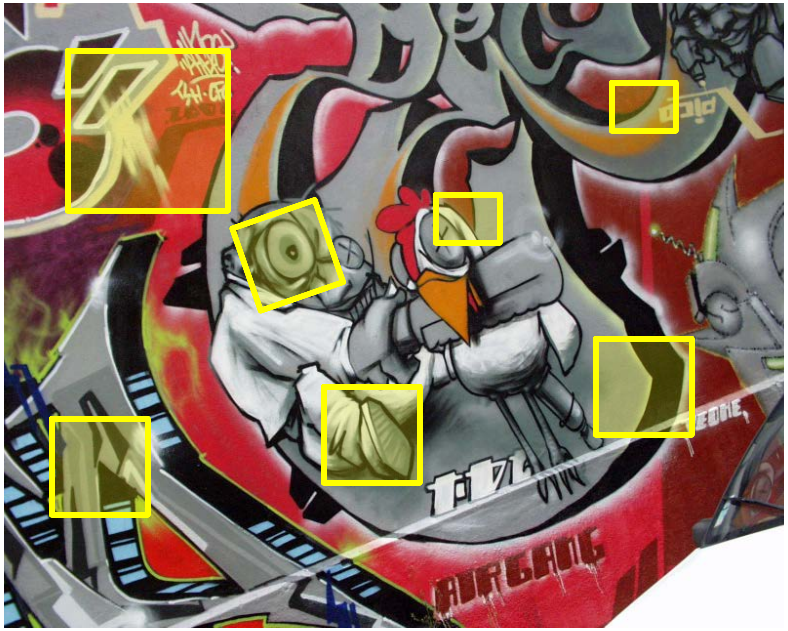
Patch Descriptors

EE/CSE 576

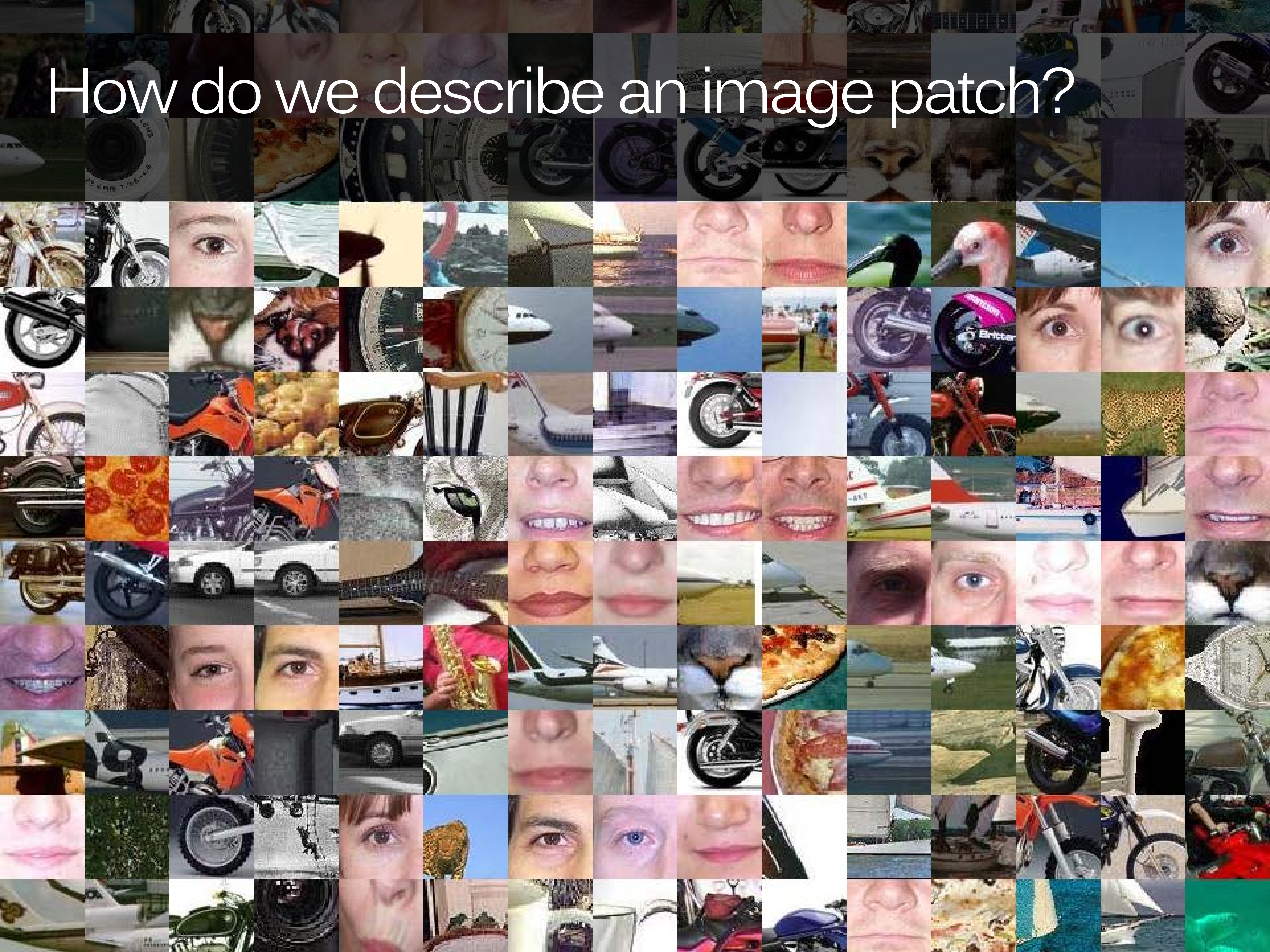
Linda Shapiro

How can we find corresponding points?

How can we find correspondences?

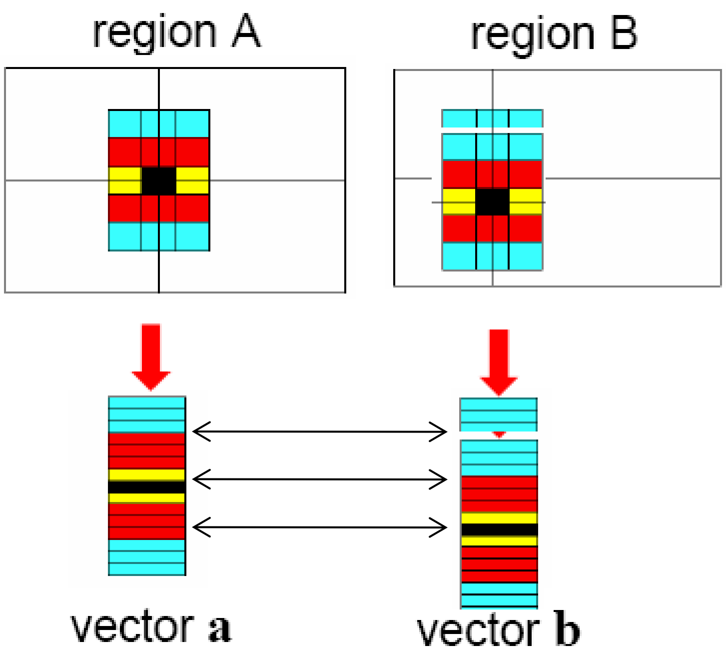

How do we describe an image patch?



How do we describe an image patch?

Patches with similar content should have similar descriptors.

Raw patches as local descriptors



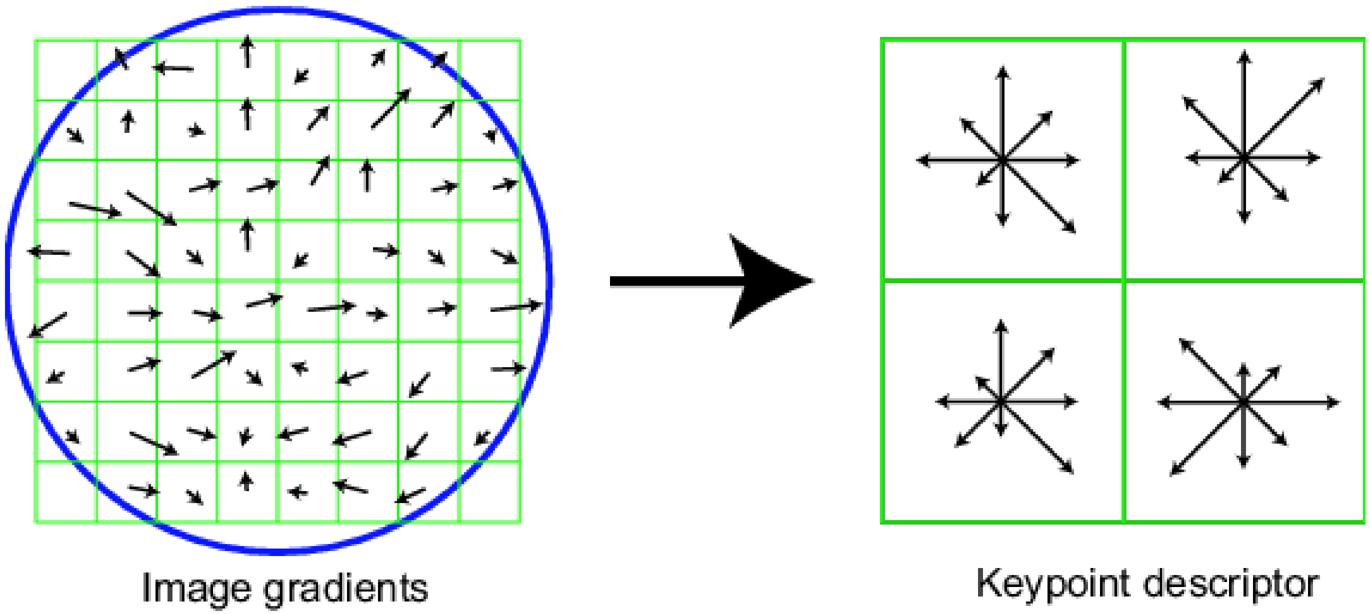
The simplest way to describe the neighborhood around an interest point is to write down the list of intensities to form a feature vector.

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

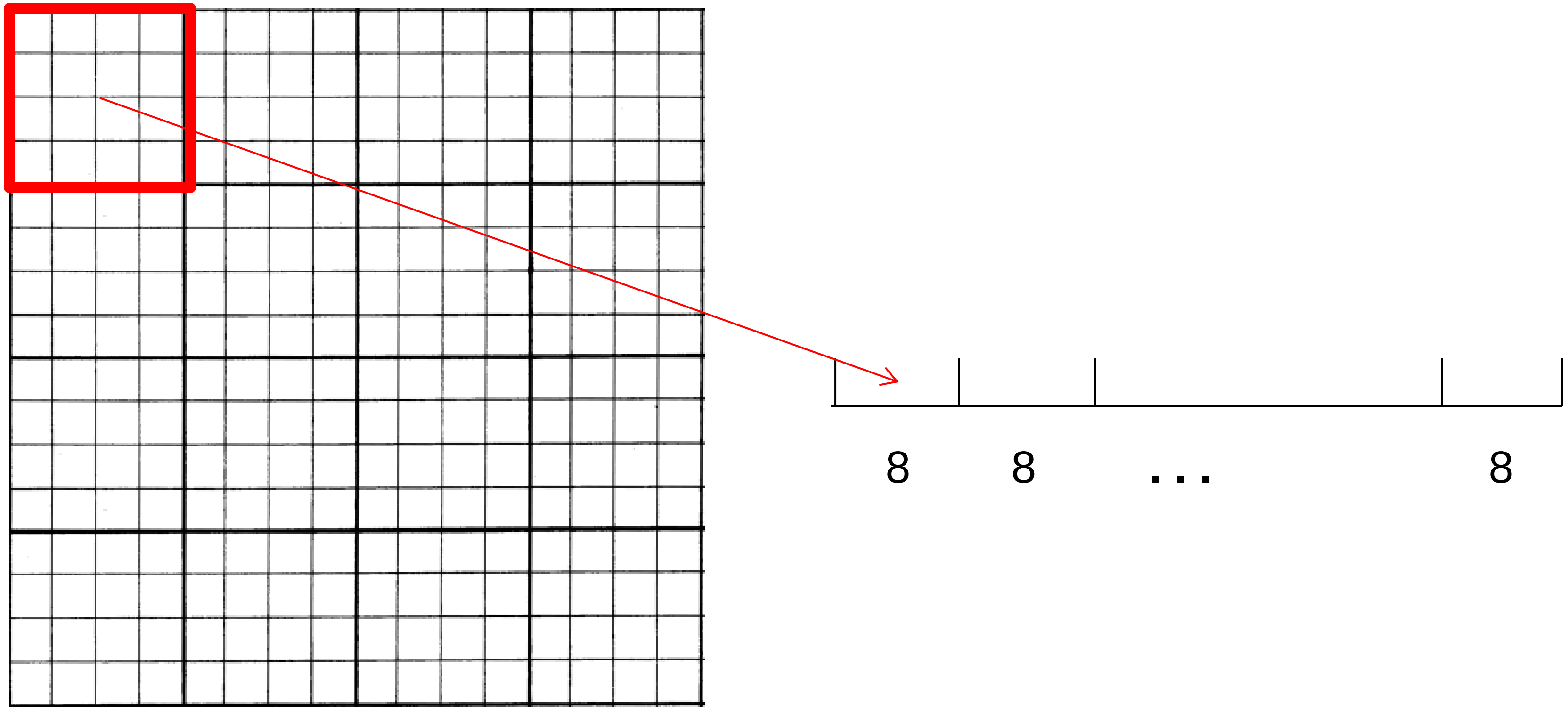


Adapted from slide by David Lowe

SIFT descriptor

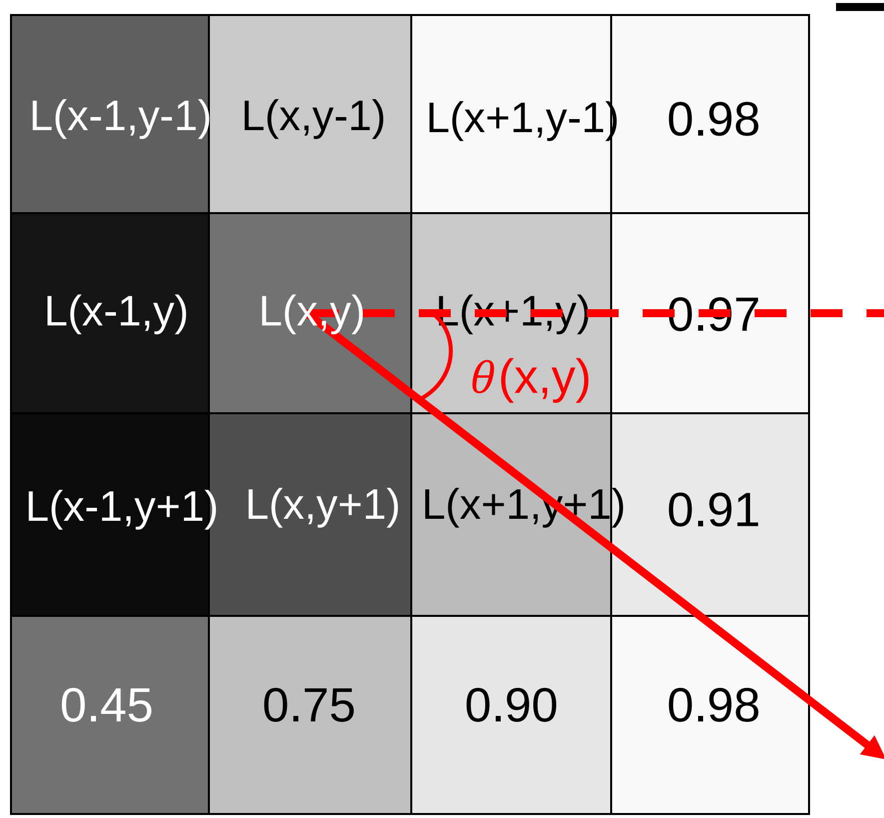
Full version

- Divide the 16x16 window into a 4x4 grid of cells
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor



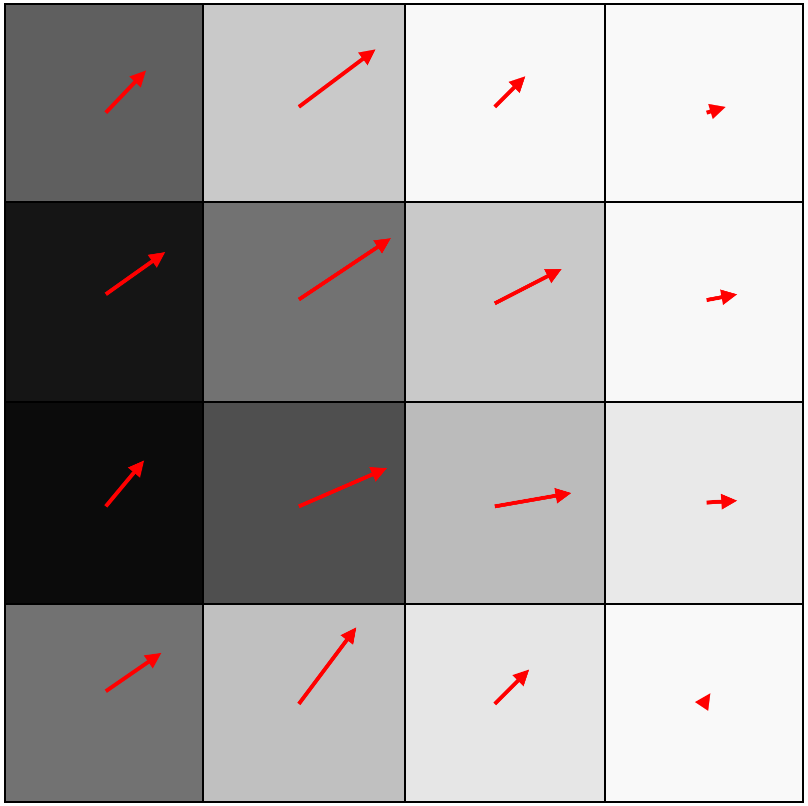
Numeric Example

0.37	0.79	0.97	0.98
0.08	0.45	0.79	0.97
0.04	0.31	0.73	0.91
0.45	0.75	0.90	0.98

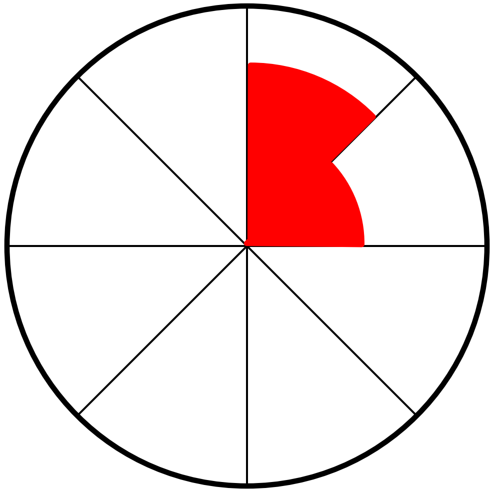


$$\text{magnitude}(x,y)=\sqrt{\left(L(x+1,y)-L(x-1,y)\right)^2+\left(L(x,y+1)-L(x,y-1)\right)^2}$$

$$\theta(x,y)=atan\left(\left(\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}\right)\right)$$



Orientations in each of the 16 pixels of the cell



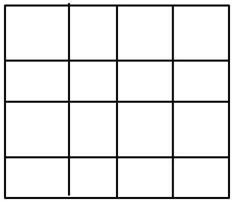
The orientations all ended up in two bins: 11 in one bin, 5 in the other. (rough count)

5 11 0 0 0 0 0 0

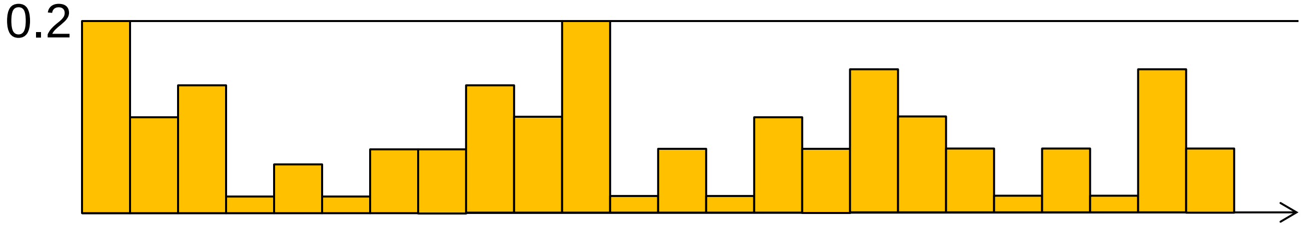
SIFT descriptor

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:



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



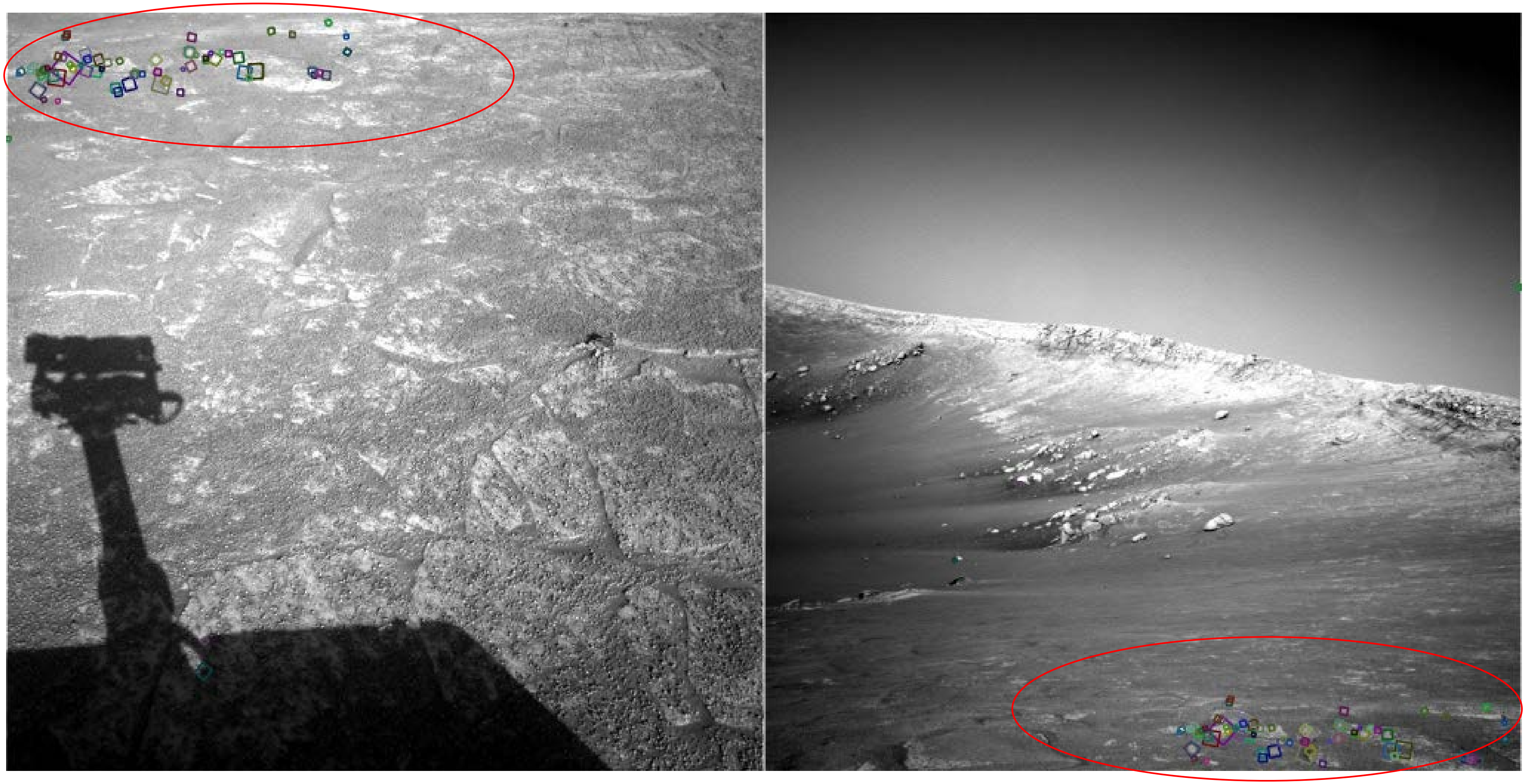
Adapted from slide by David Lowe

Properties of SIFT

Extraordinarily robust matching technique

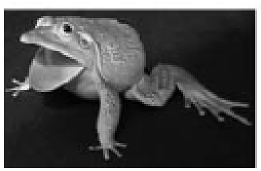
- Can handle changes in viewpoint
  - Up to about 30 degree out of plane rotation
- Can handle significant changes in illumination
  - Sometimes even day vs. night (below)
- Fast and efficient—can run in real time
- Various code available
  - http://www.cs.ubc.ca/~lowe/keypoints/

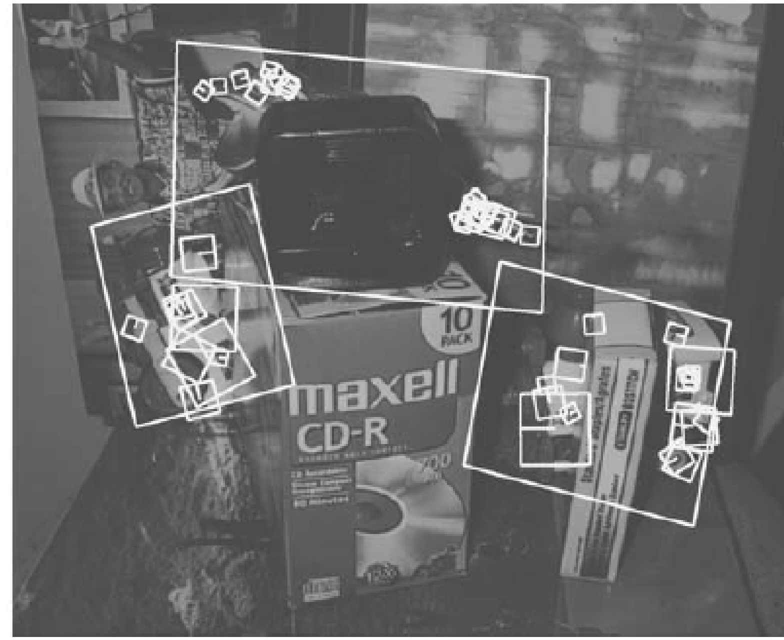
Example



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.

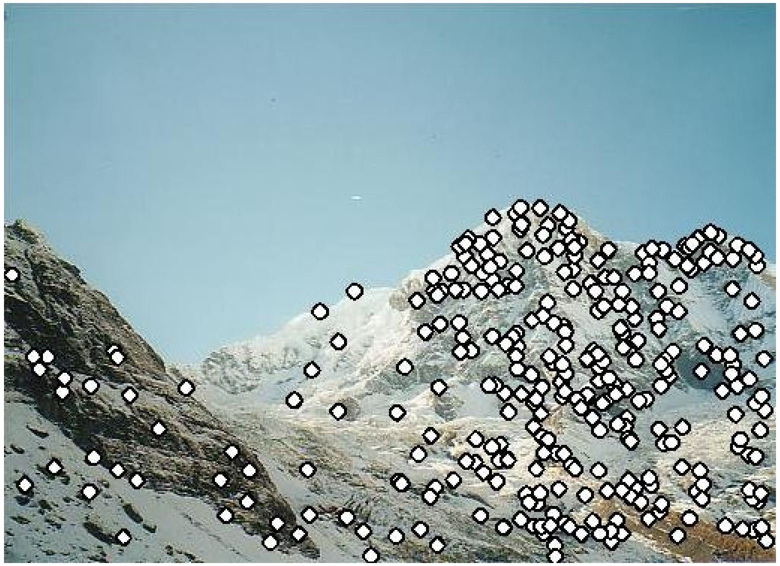
panorama?

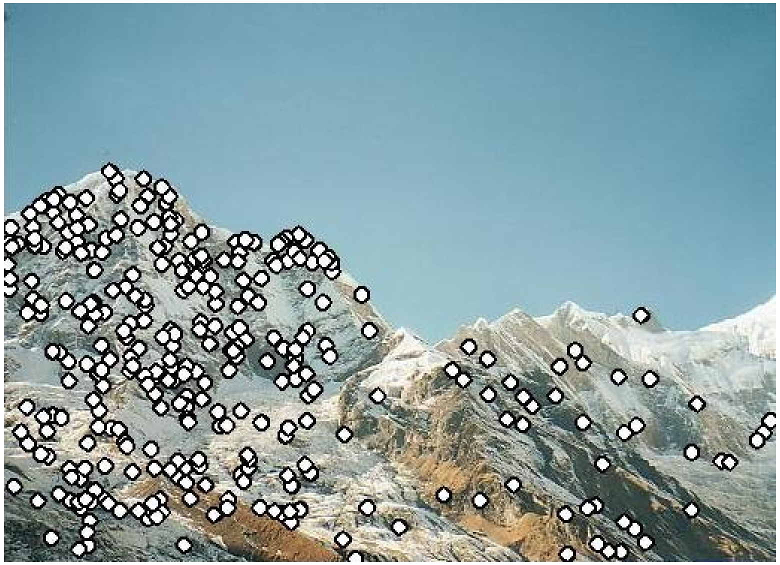
- We need to match (align) images

Matching with Features

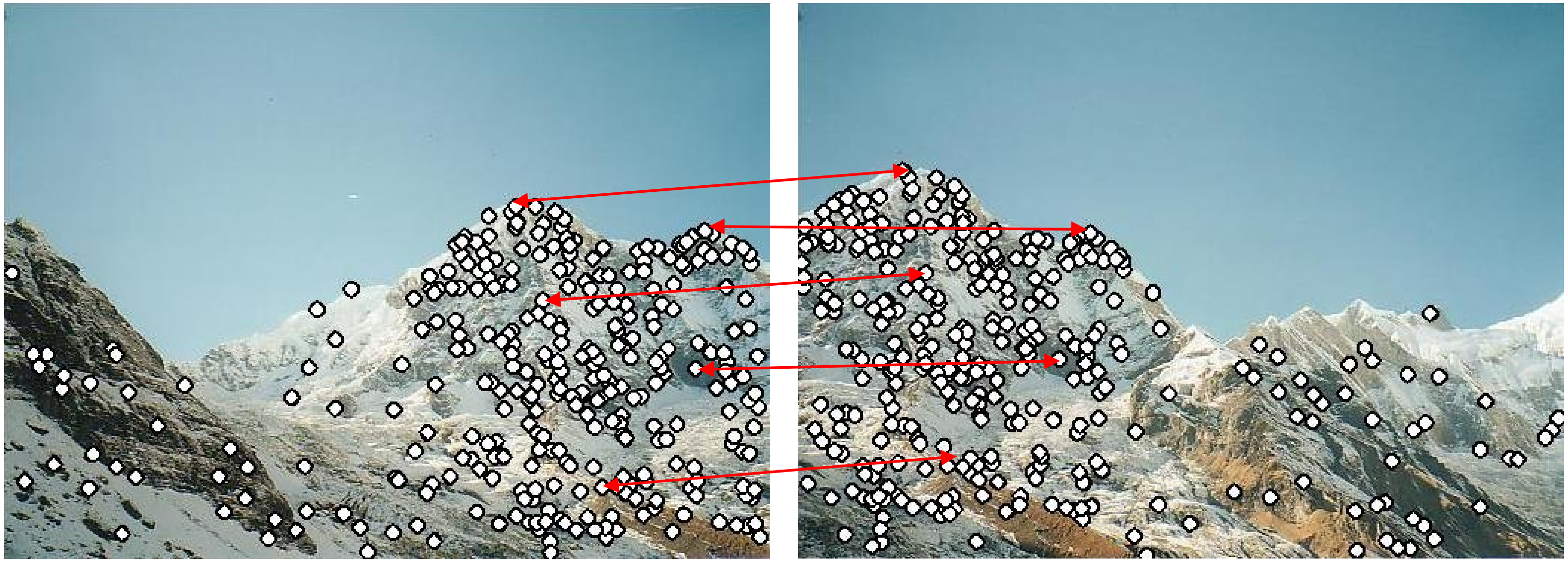
•Detect feature points in both images





Matching with Features

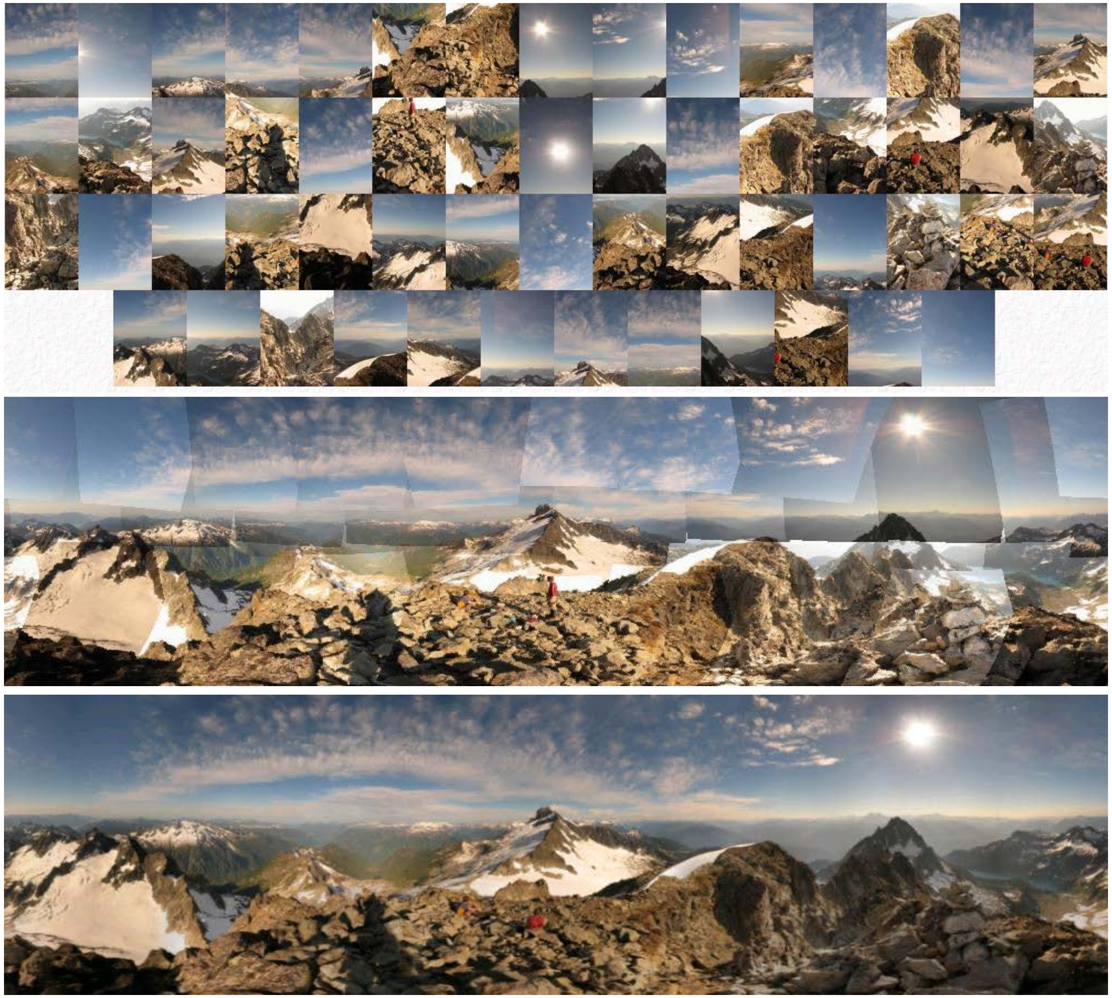
- Detect feature points in both images
- Find corresponding pairs



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



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 (also called Photo Tourism) developed at UW by Noah Snavely, Steve Seitz, Rick Szeliski and others

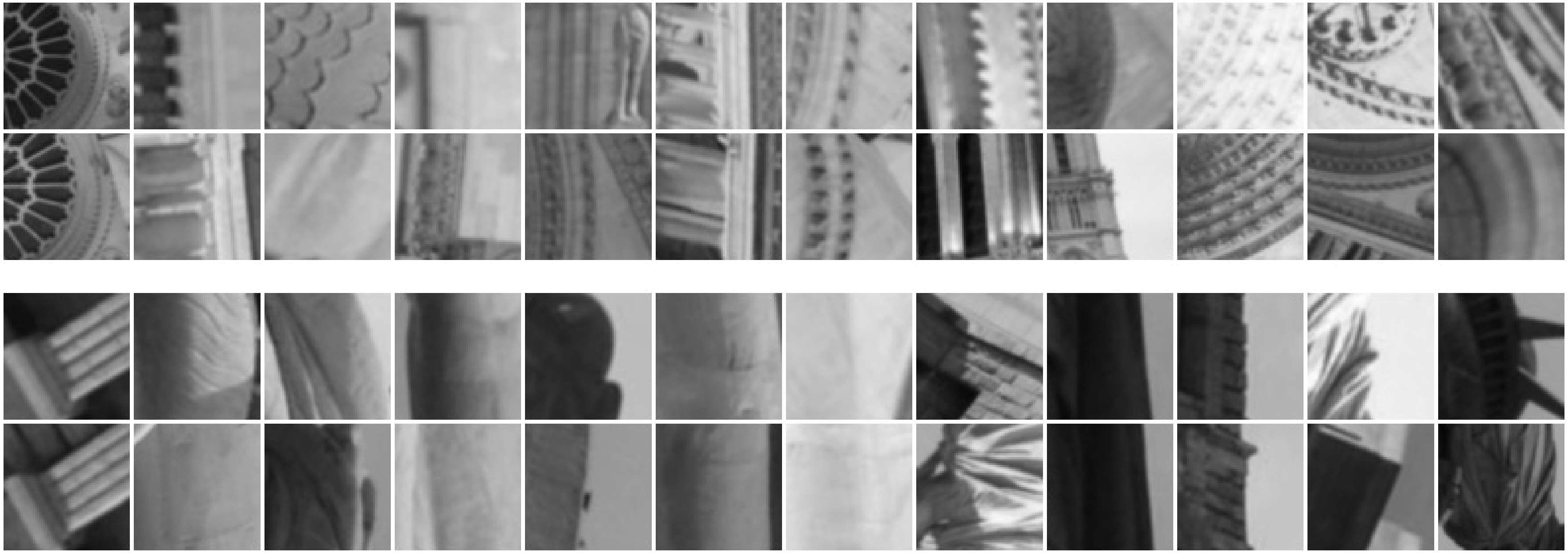
http://www.youtube.com/watch?v=p16frKJLVi0

- Building Rome in a day, developed at UW by Sameer Agarwal, Noah Snavely, Steve Seitz and others

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

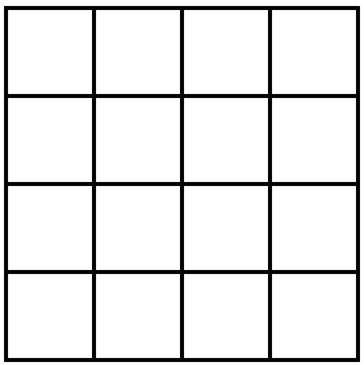
When does the SIFT descriptor fail?

Patches SIFT thought were the same but aren't:

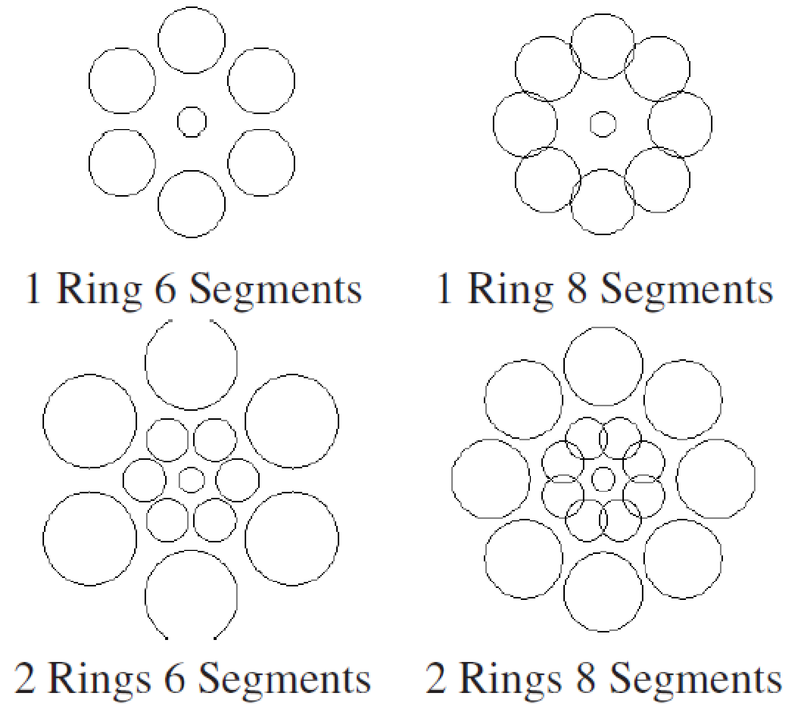


Other methods: Daisy

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:

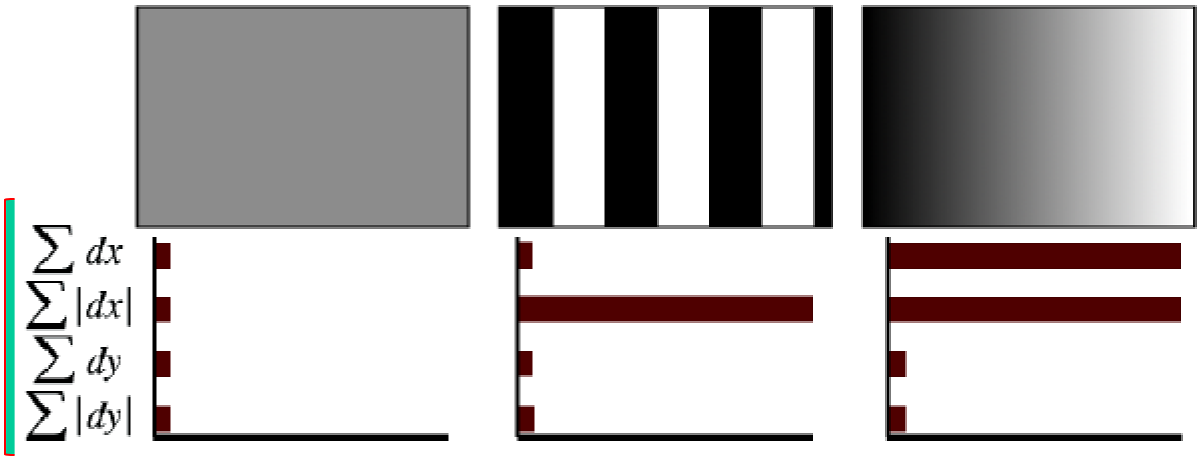


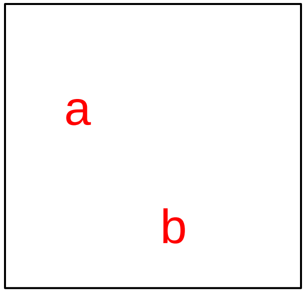
Fig. 3. The descriptor entries of a sub-region represent the nature of the underlying intensity pattern. Left: In case of a homogeneous region, all values are relatively low. Middle: In presence of frequencies in x direction, the value of $\sum |d_x|$ is high, but all others remain low. If the intensity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

011000111000...



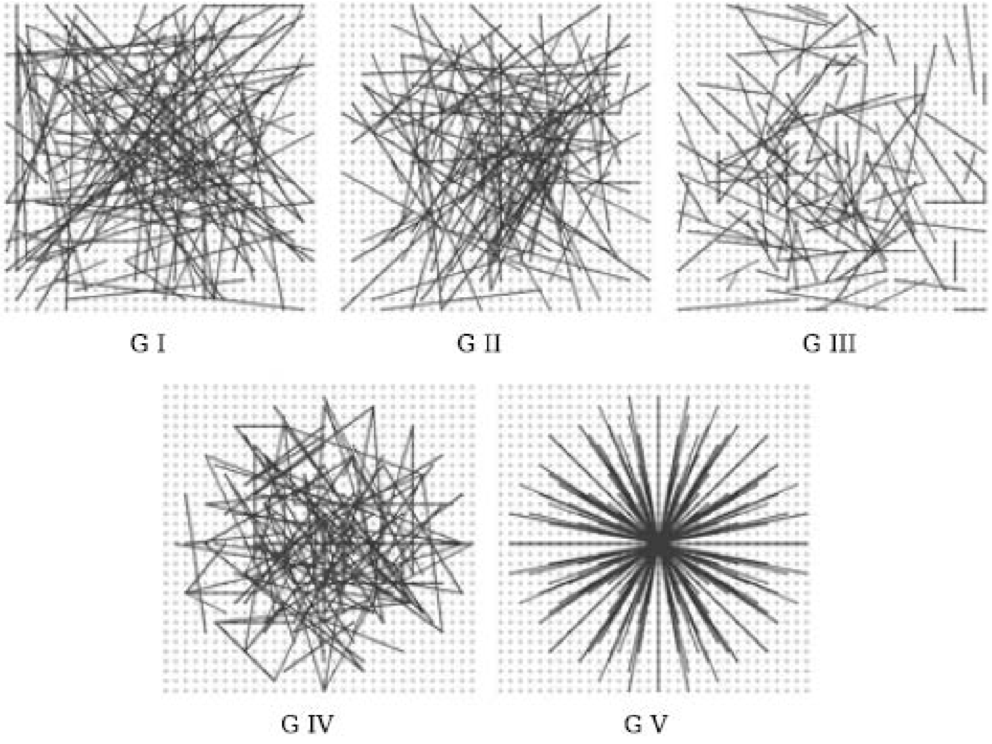


Fig. 2. Different approaches to choosing the test locations. All except the righmost one are selected by random sampling. Showing 128 tests in every image.

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?

?

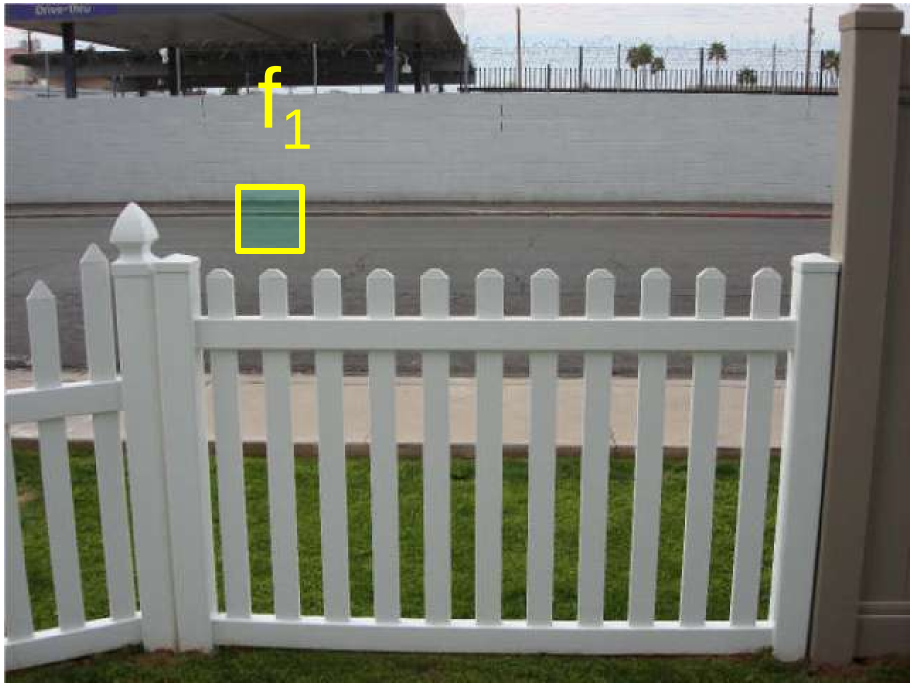
Feature distance

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

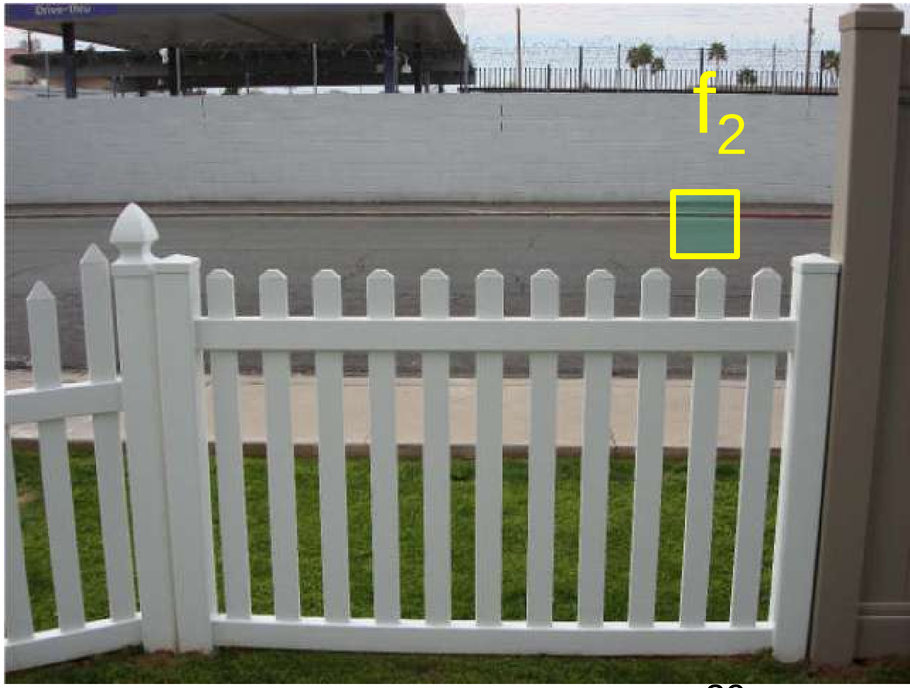
- Simple approach is SSD(f_1, f_2)
  - sum of square differences between entries of the two descriptors

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

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



I_1

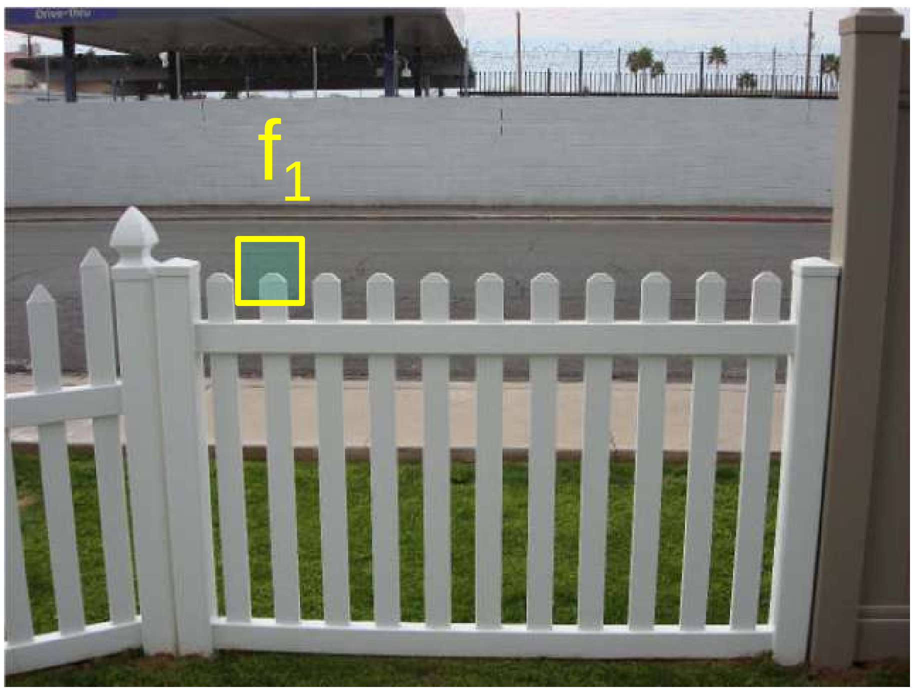


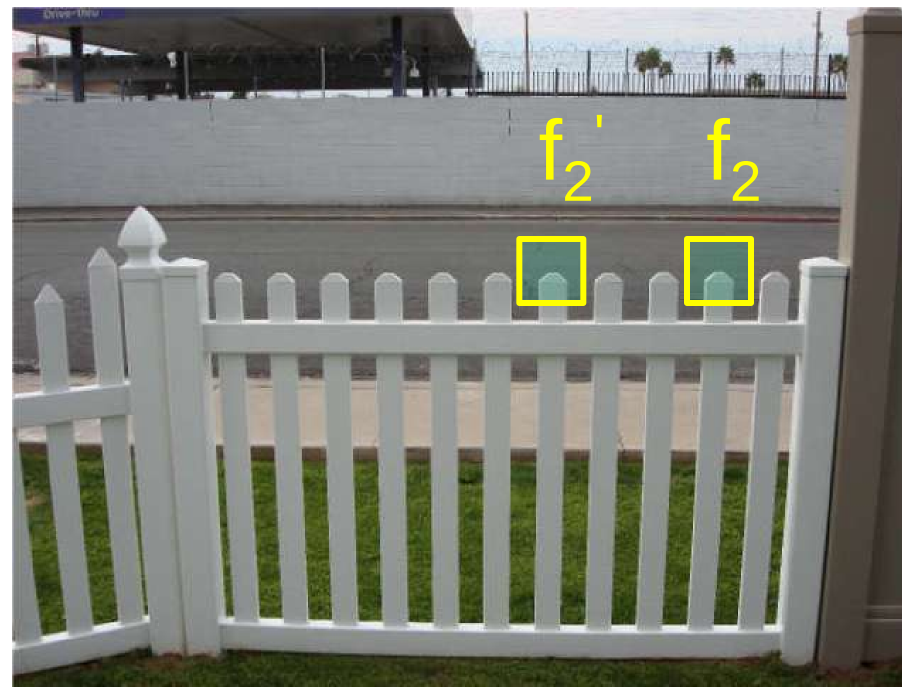
I_2

Feature distance in practice

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

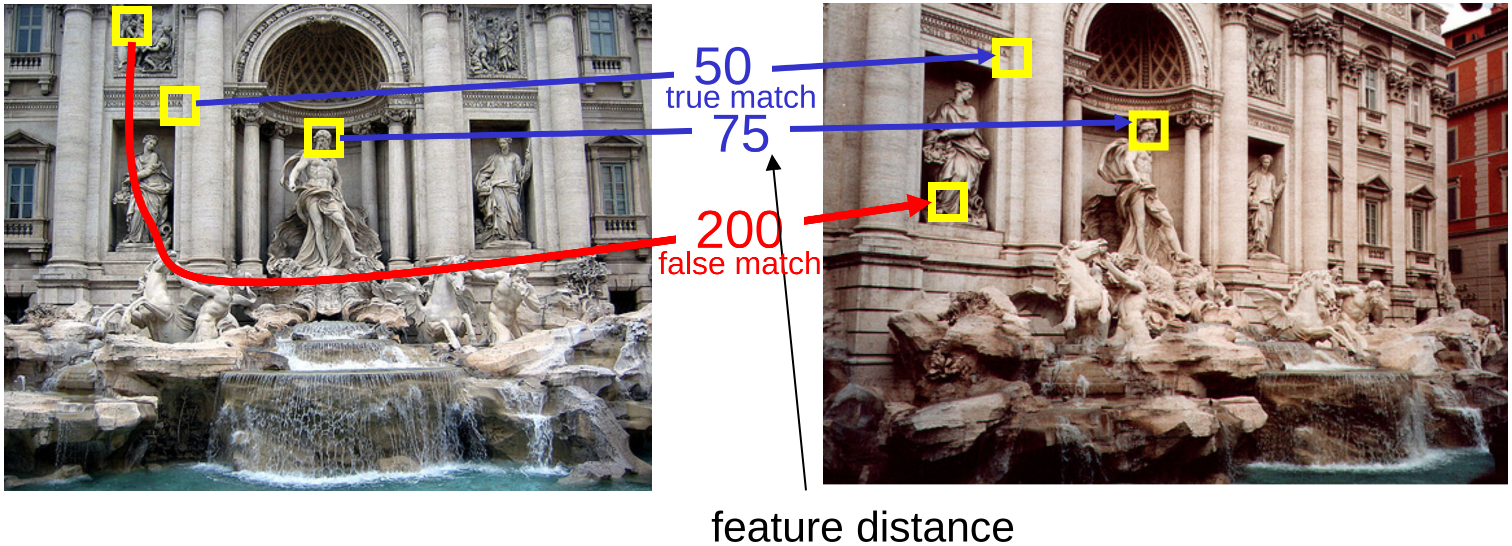
- Better approach: ratio distance = $SSD(f_1, f_2) / SSD(f_1, f_2')$
  - f_2 is best SSD match to f_1 in I_2
  - f_2' is 2^{nd} best SSD match to f_1 in I_2
  - gives large values (~1) for ambiguous matches WHY?





I_1 I_2

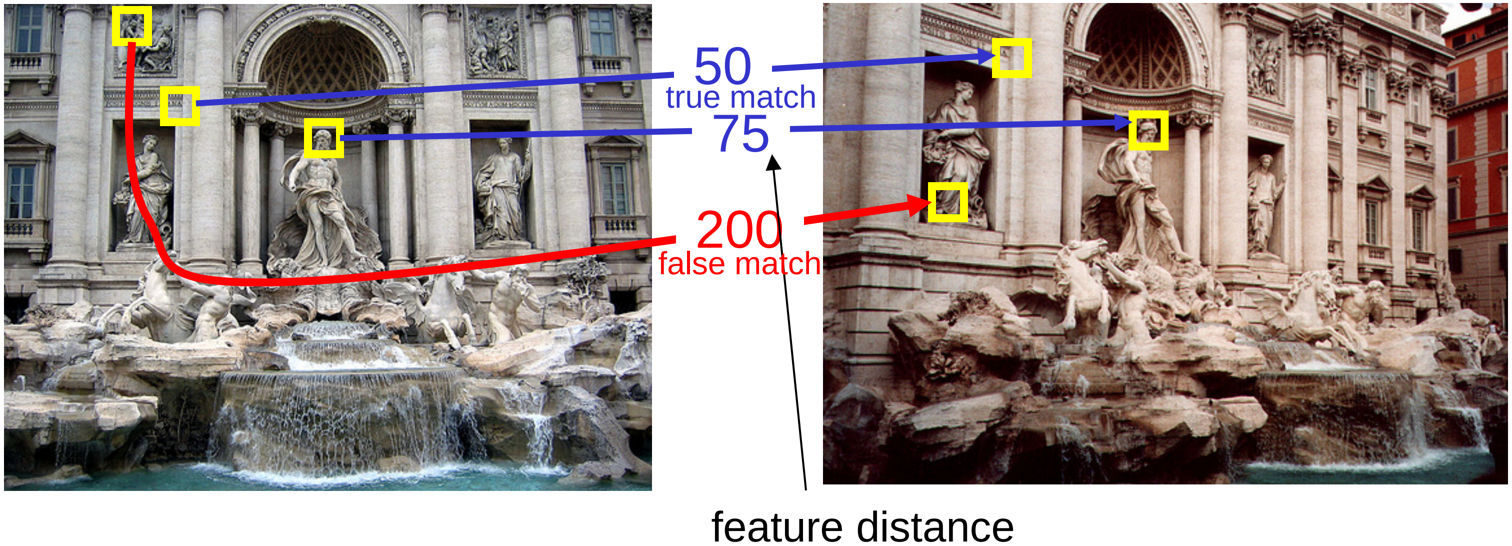
Eliminating more bad matches



Throw out features with distance > threshold

- How to choose the threshold?

True/false positives



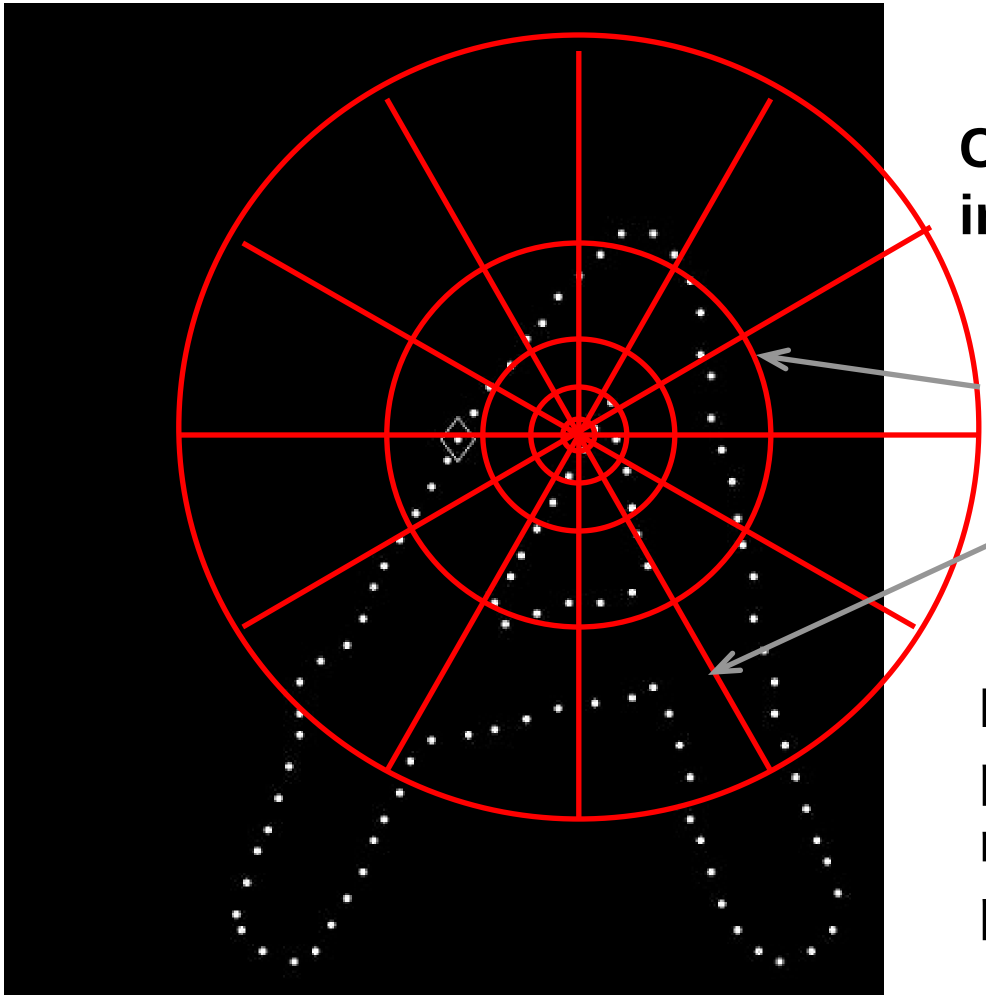
The distance threshold affects performance

- 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 = ?

⋮

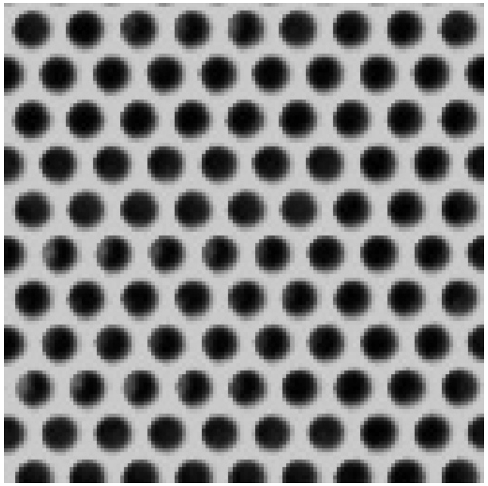
Count = ?

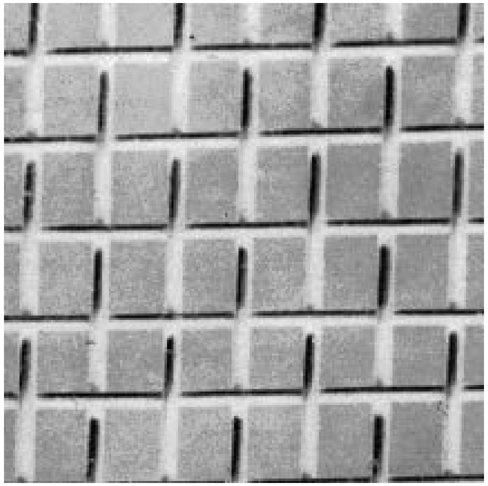
Log-polar binning: more precision for nearby points, more flexibility for farther points.

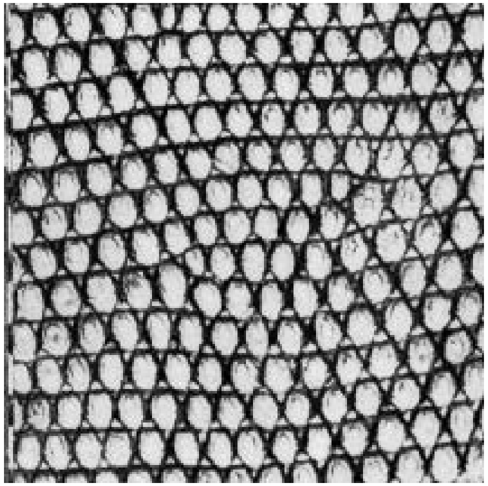
Belongie & Malik, ICCV 2001

Texture

- The texture features of a patch can be considered a descriptor.
- E.g. the LBP histogram is a texture descriptor for a patch.







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

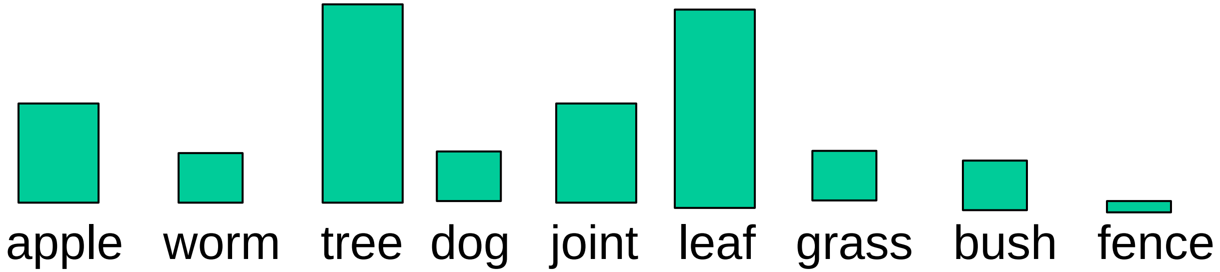
- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

Bag-of-words models

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

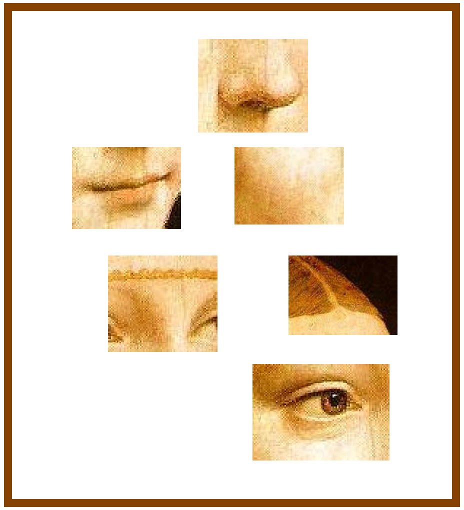
What is a bag-of-words representation?

- For a text document
- Have a dictionary of non-common words
- Count the occurrence of each word in that document
- Make a histogram of the counts
- Normalize the histogram by dividing each count by the sum of all the counts
- The histogram is the representation.



Bags of features for image classification

1. Extract features



Bags of features for image classification

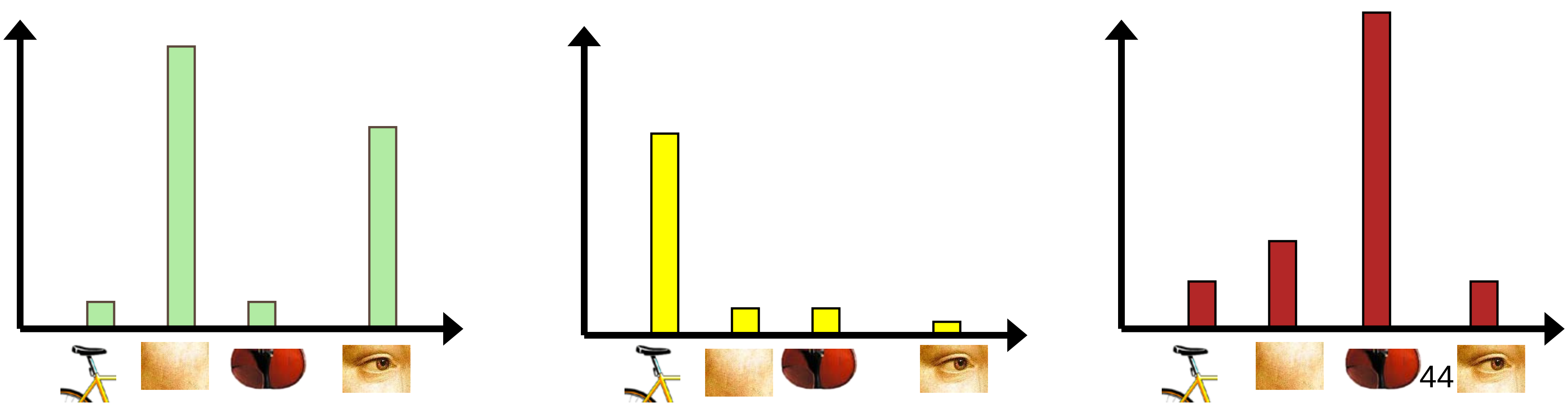
1. Extract features
2. Learn "visual vocabulary"

Bags of features for image classification

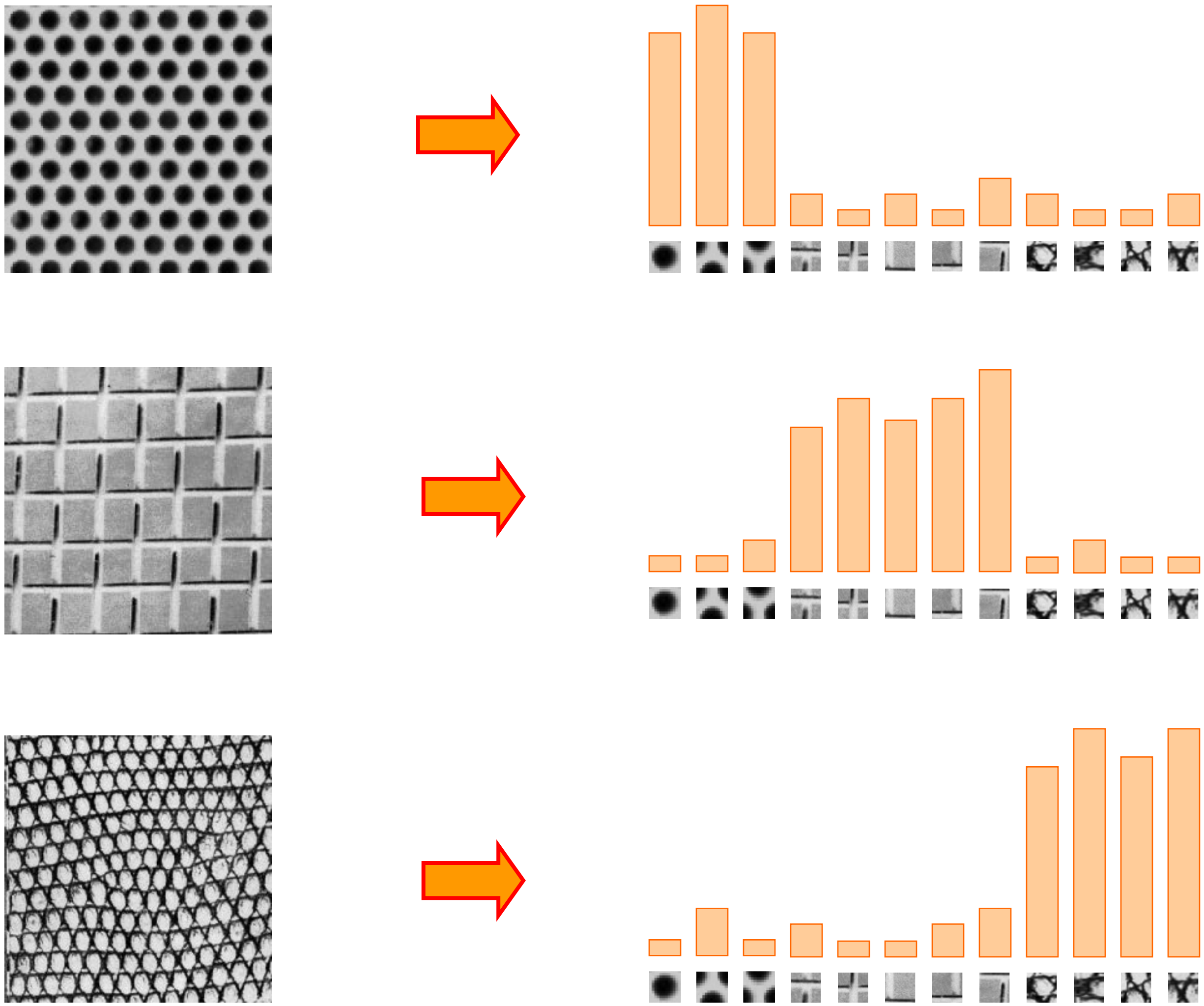
1. Extract features
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary

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"

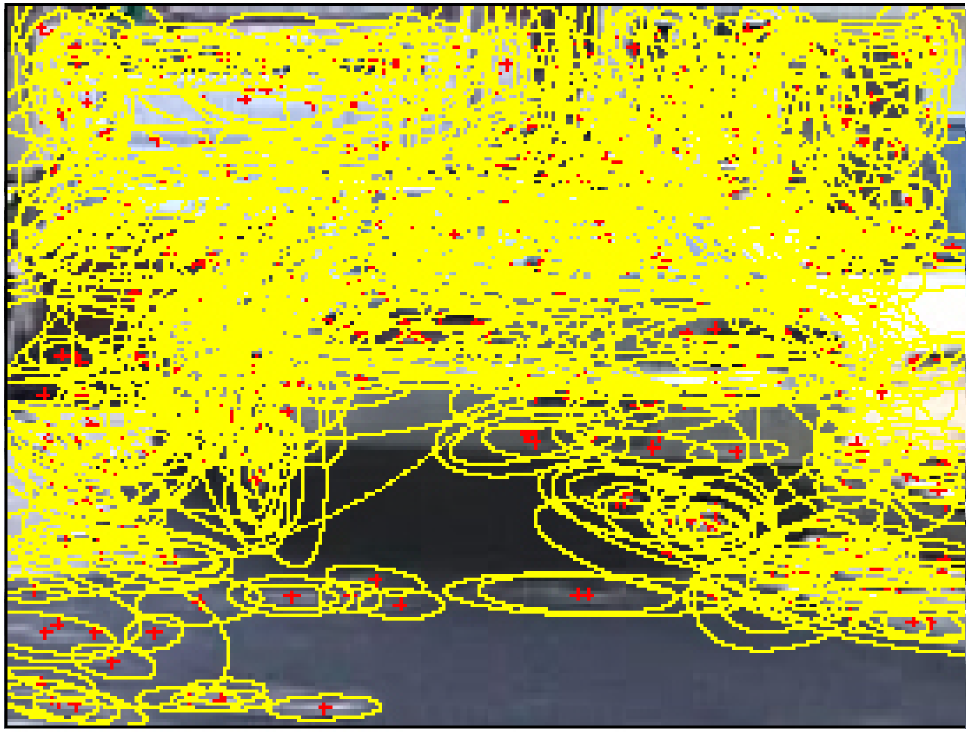


A possible texture representation



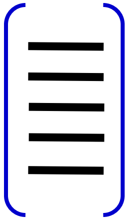
1. Feature extraction

- Regular grid: every grid square is a feature
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
- Interest point detector: the region around each point
  - Csurka et al. 2004
  - Fei-Fei & Perona, 2005
  - Sivic et al. 2005



1. Feature extraction

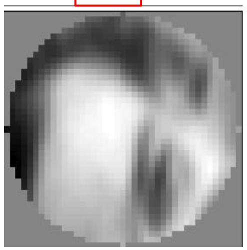
3



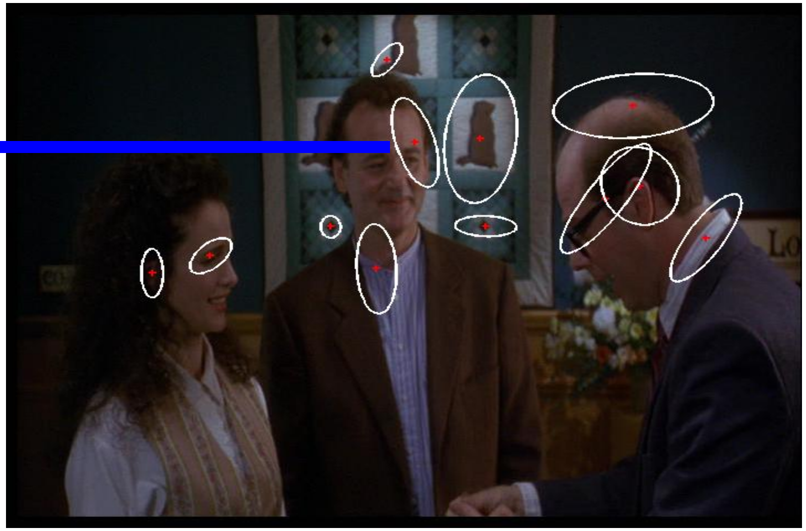
Compute SIFT descriptor

[Lowe'99]

2



Normalize patch



1 Detect patches

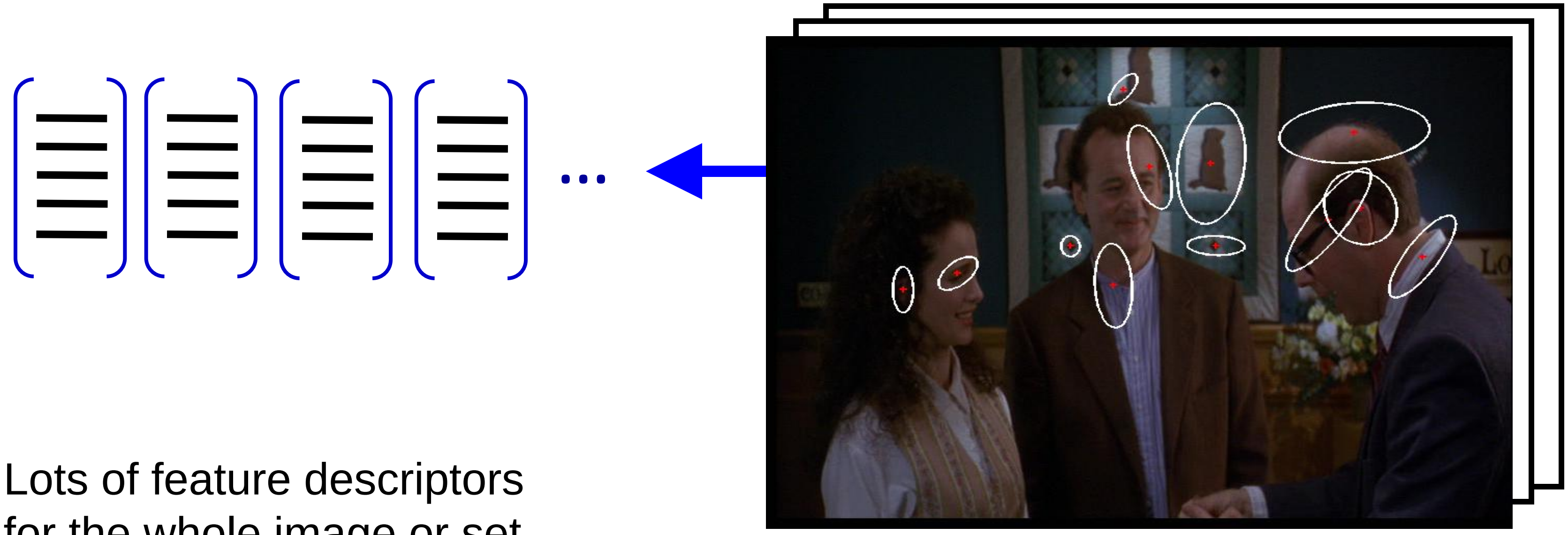
[Mikojaczyk and Schmid '02]

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

[Sivic & Zisserman, '03]

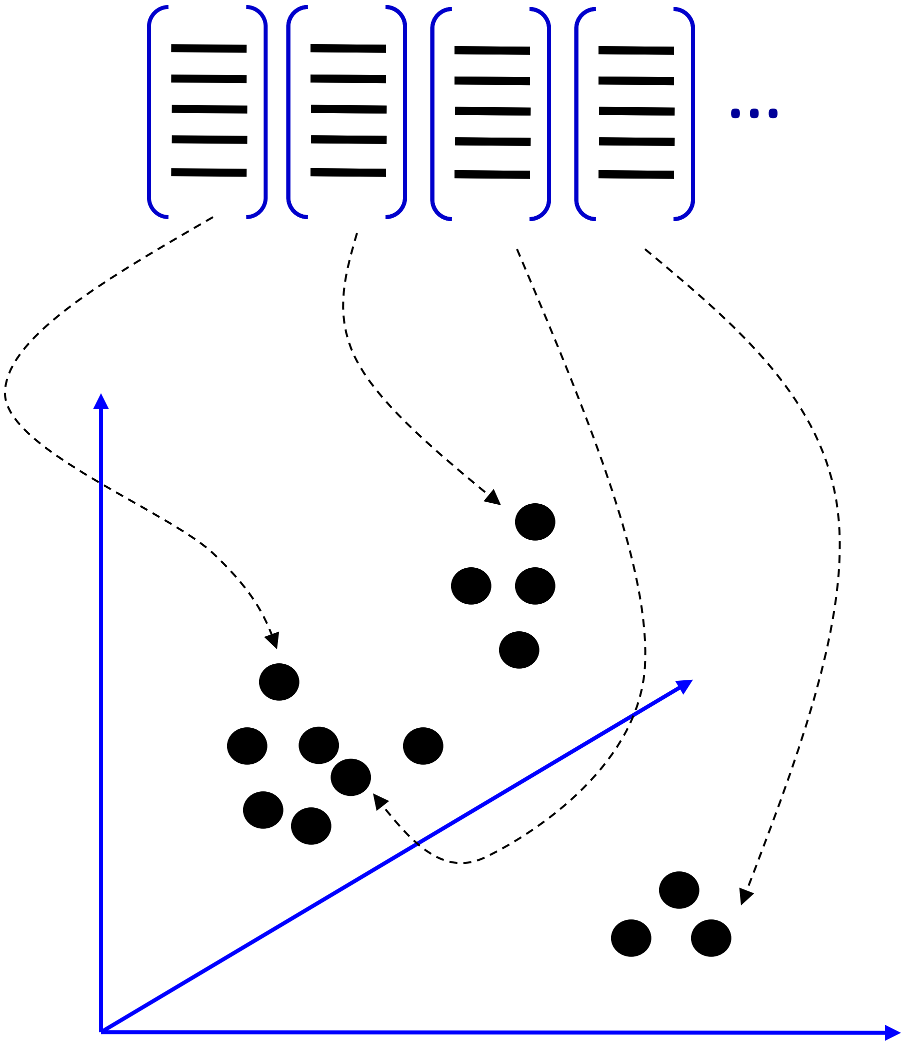
Slide credit: Josef Sivic

1. Feature extraction



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

2. Discovering the visual vocabulary

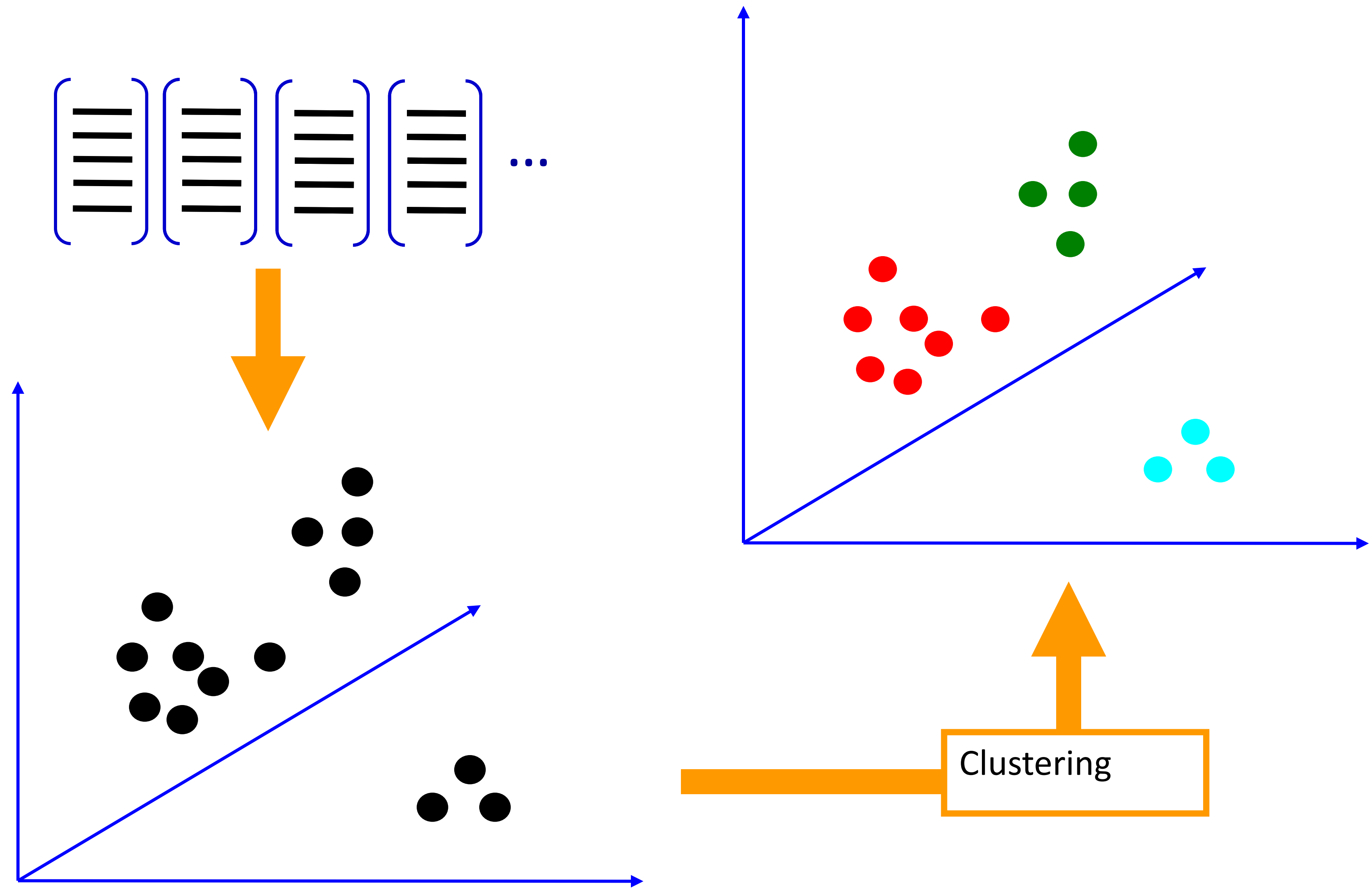


feature vector space

What is the dimensionality?

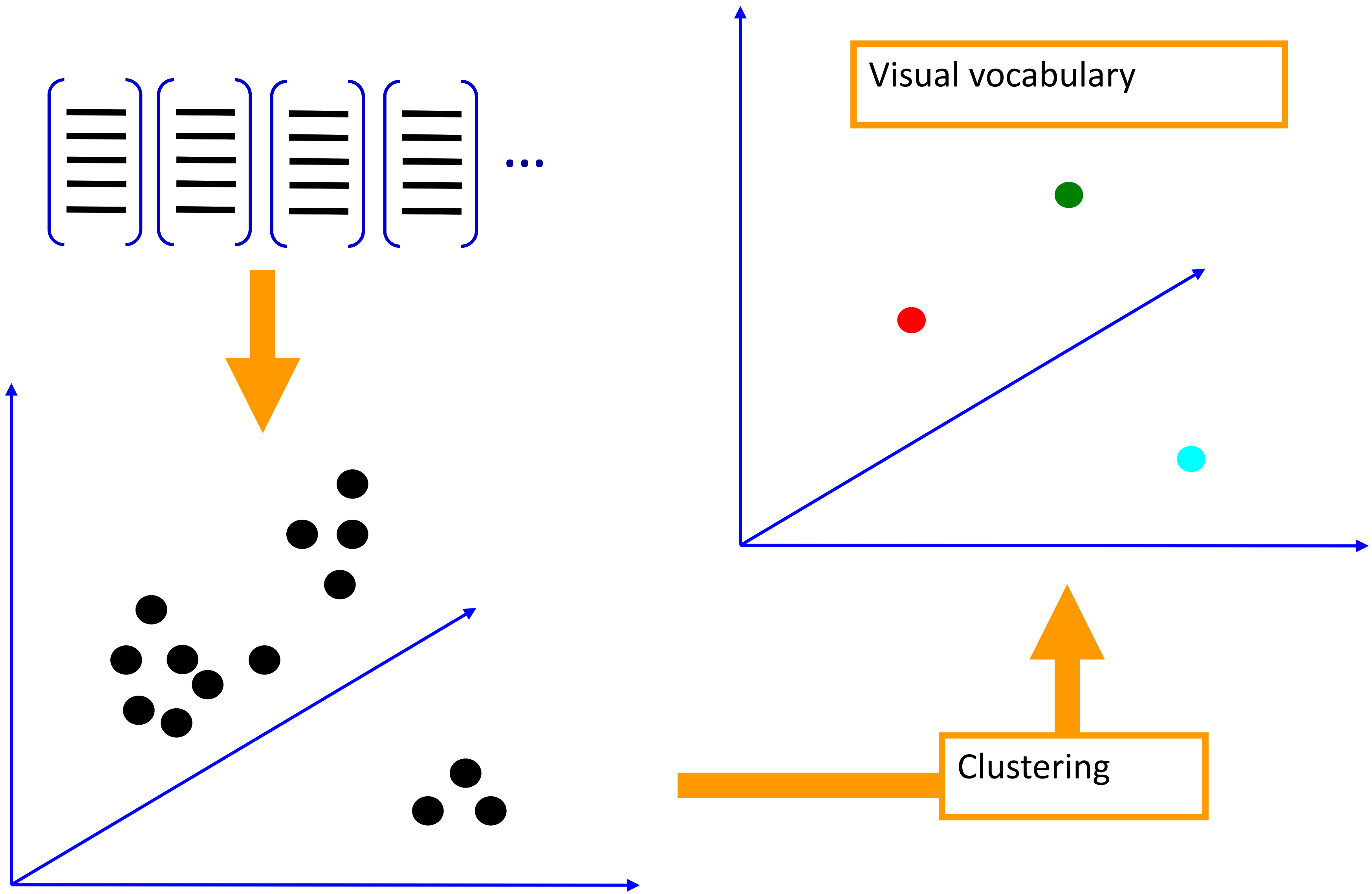
128D for SIFT

2. Discovering the visual vocabulary



Slide credit: Josef Sivic

2. Discovering the visual vocabulary



Slide credit: Josef Sivic

Viewpoint invariant description (Sivic)

- Two types of viewpoint covariant regions computed for each frame
  - Shape Adapted (SA) Mikolajczyk & Schmid
  - Maximally Stable (MSER) Matas *et al.*
- Detect different kinds of image areas
- Provide complimentary representations of frame
- Computed at twice originally detected region size to be more discriminating

Examples of Harris-Affine Operator

(Shape Adapted Regions)

140 K. Mikolajczyk and C. Schmid

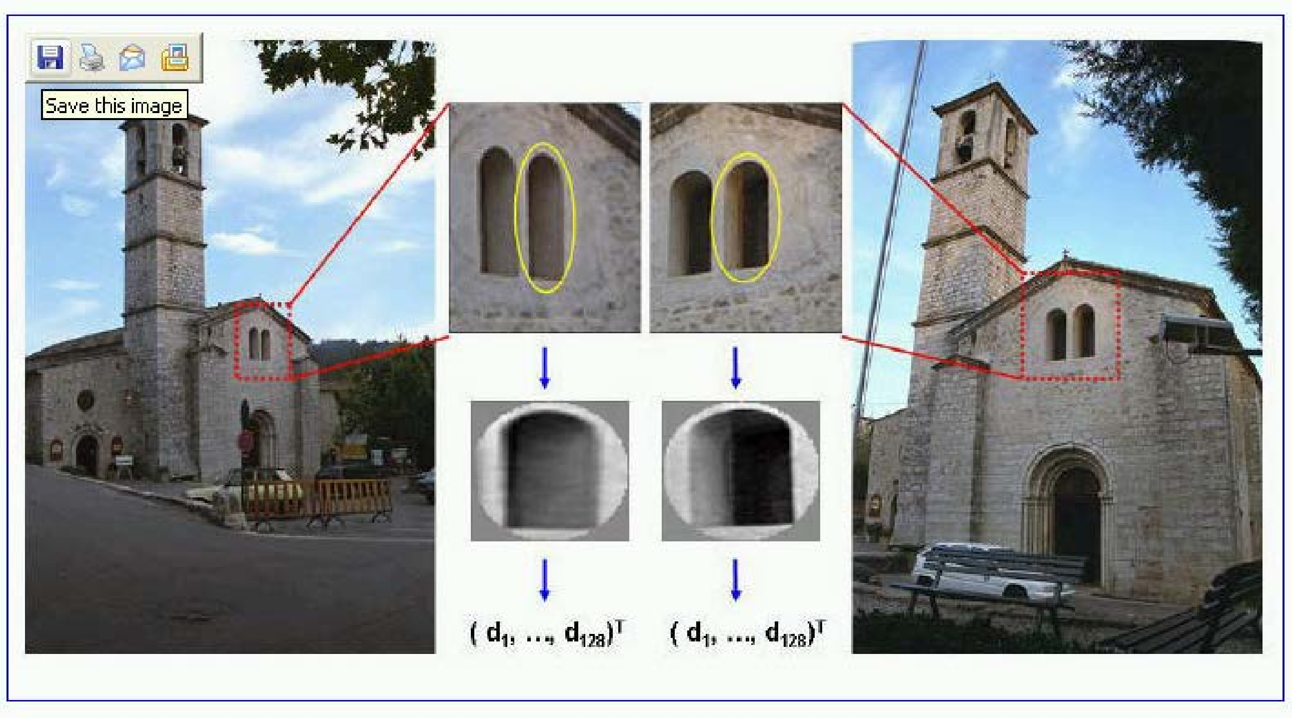
(a) b)

Fig. 6. (a) Example of a 3D scene observed from significantly different viewpoints. There are 14 inliers to a robustly estimated fundamental matrix, all of them correct. (b) An image pairs for which our method fails. There exist, however, corresponding points which we have selected manually.

Examples of Maximally Stable Regions

Feature Descriptor

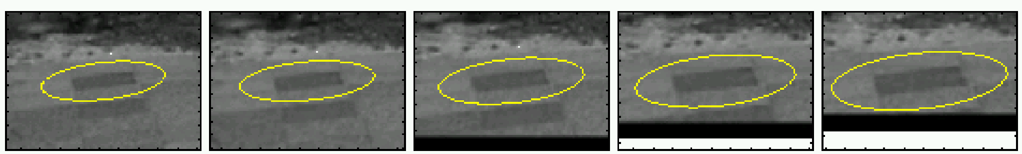
- Each region represented by 128 dimensional vector using SIFT descriptor



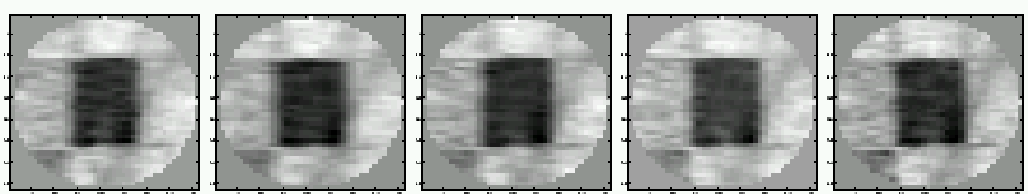
Noise Removal

- **Tracking region over 70 frames (must track over at least 3)**

First (left) and last (right) frame of the track.



Close-up of the 1st, 20th, 40th, 55th, 70th frame.



Visual Vocabulary for Sivic's Work

- Implementation: K-Means clustering
- Regions tracked through contiguous frames and average description computed
- 10% of tracks with highest variance eliminated, leaving about 1000 regions per frame
- Subset of 48 shots (~10%) selected for clustering
- Distance function: Mahalanobis
- 6000 SA clusters and 10000 MS clusters

Visual Vocabulary

Shape-Adapted

Maximally Stable

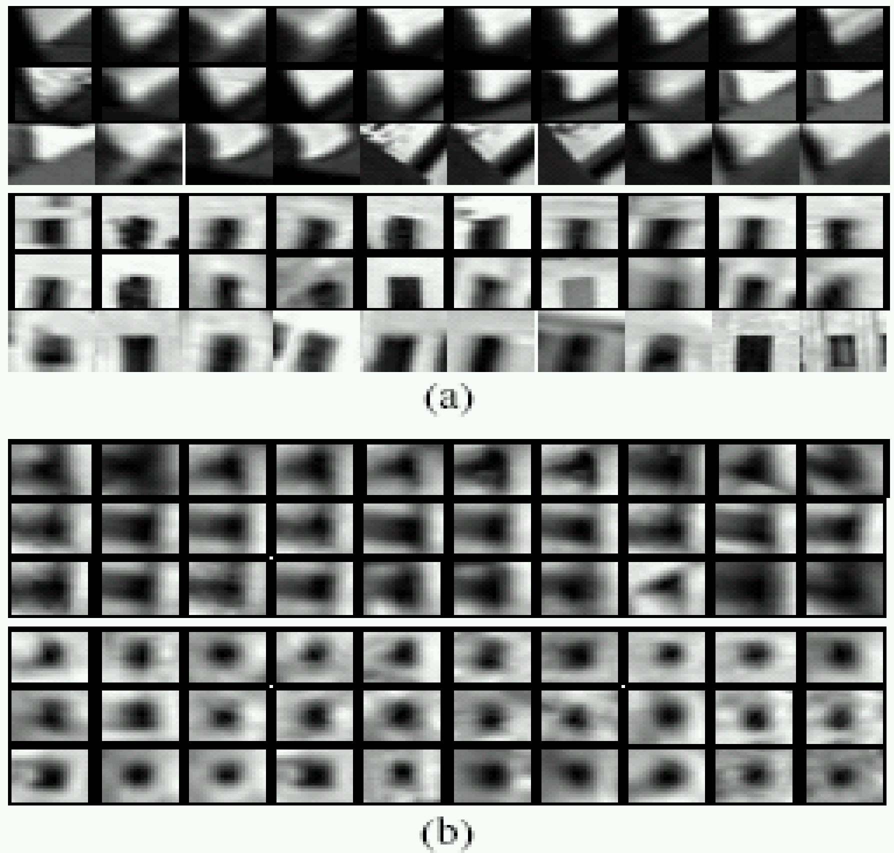
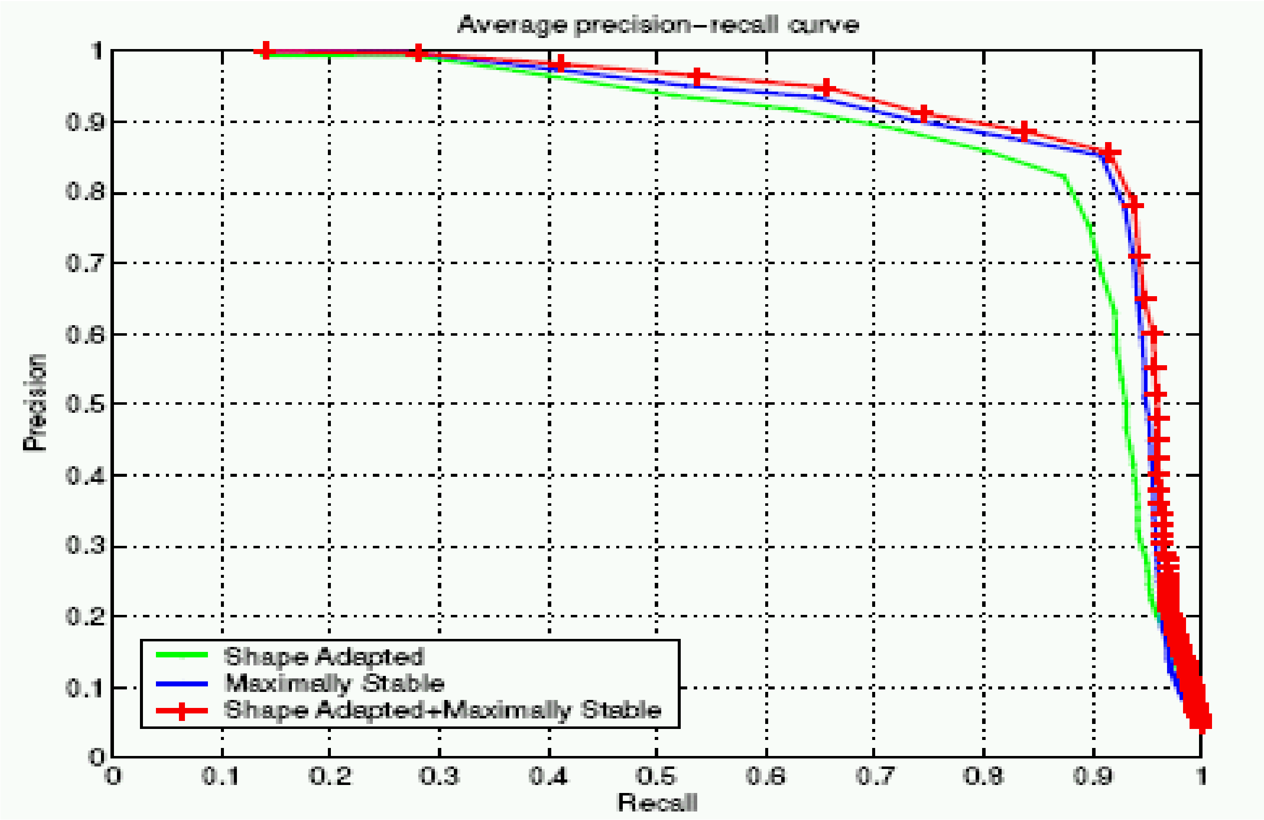


Figure 2: Samples from the clusters corresponding to a single visual word. (a) Two examples of clusters of Shape Adapted regions. (b) Two examples of clusters of Maximally Stable regions.

Sivic's Experiments on Video Shot Retrieval

- Goal: match scene locations within closed world of shots
- Data:164 frames from 48 shots taken at 19 different 3D locations; 4-9 frames from each location

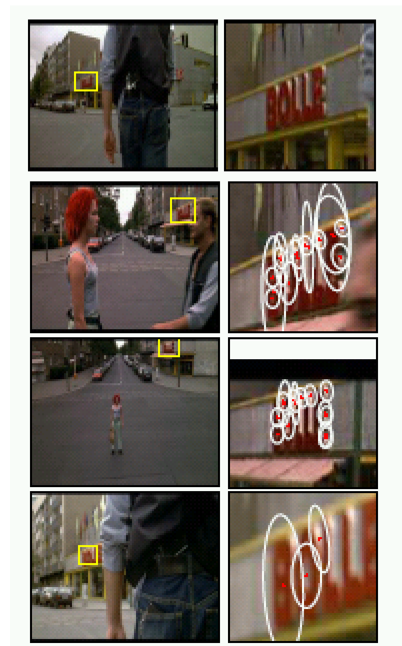
Experiments - Results

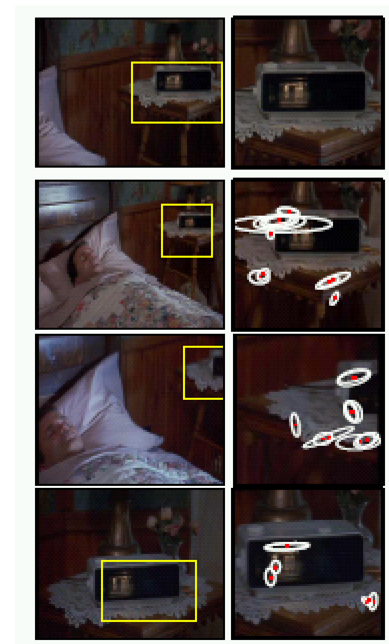


Precision = # relevant images/total # of frames retrieved

Recall = # correctly retrieved frames/ # relevant frames

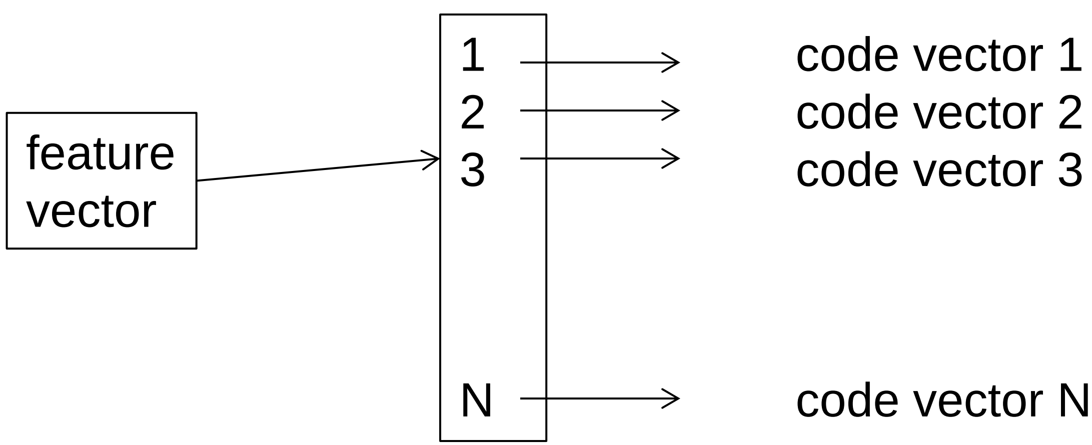
More Pictorial Results



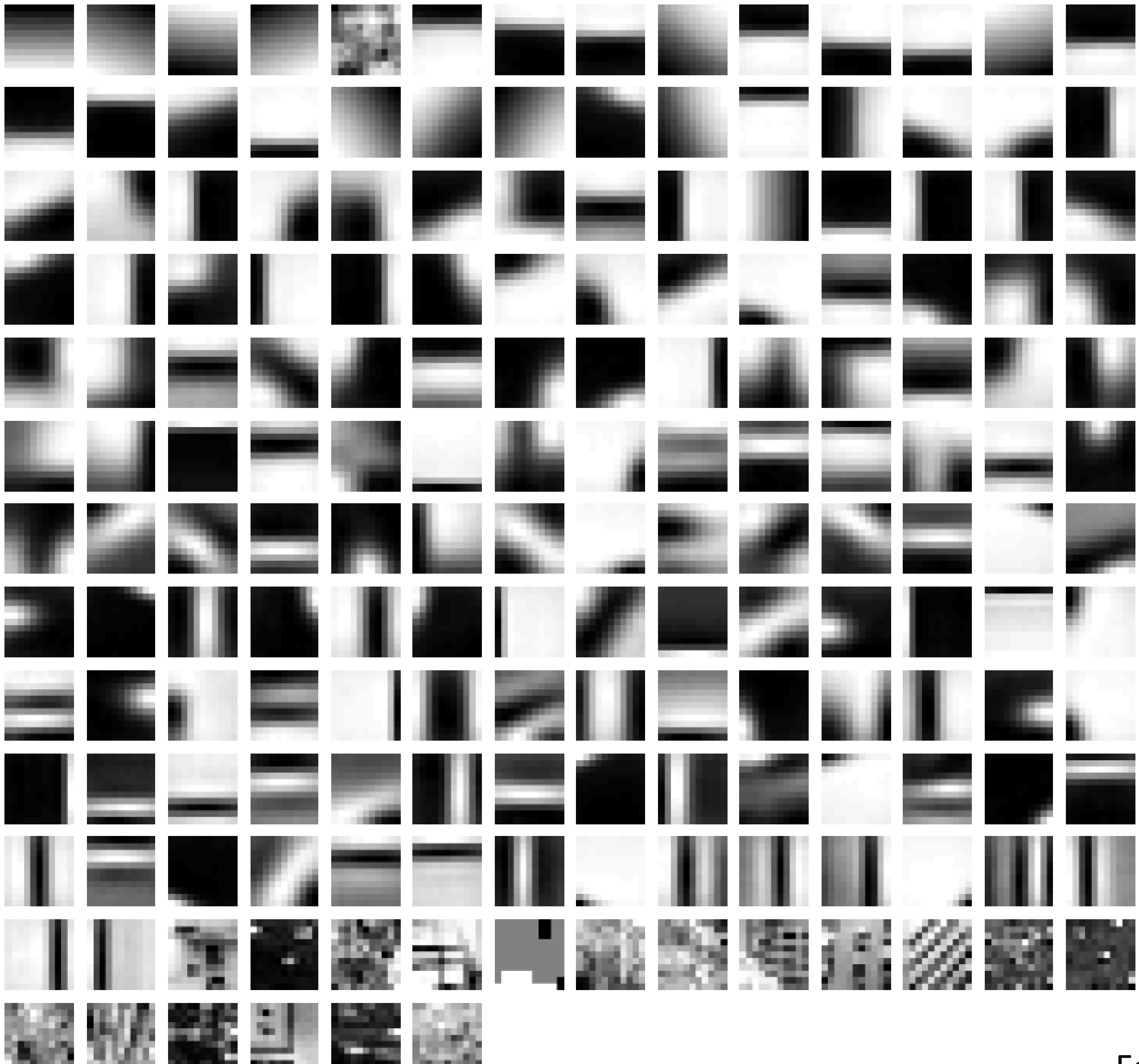


Clustering and vector quantization

- Clustering is a common method for learning a visual vocabulary or codebook
  - Each cluster center produced by k-means becomes a codevector
  - Codebook can be learned on separate training set
- The codebook is used for quantizing features
  - A *vector quantizer* takes a feature vector and maps it to the index of the nearest code vector in a codebook
  - Codebook = visual vocabulary
  - Code vector = visual word

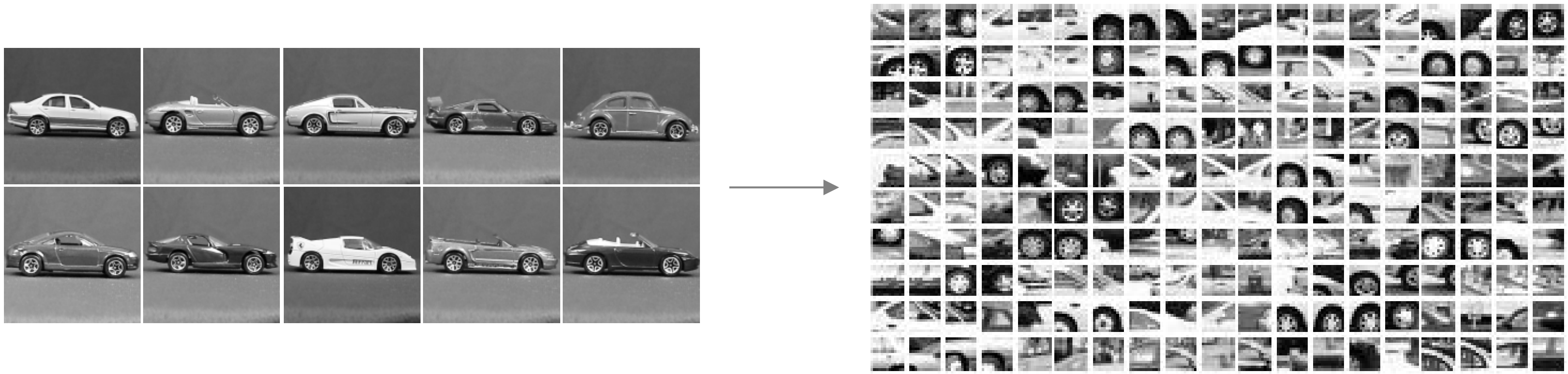


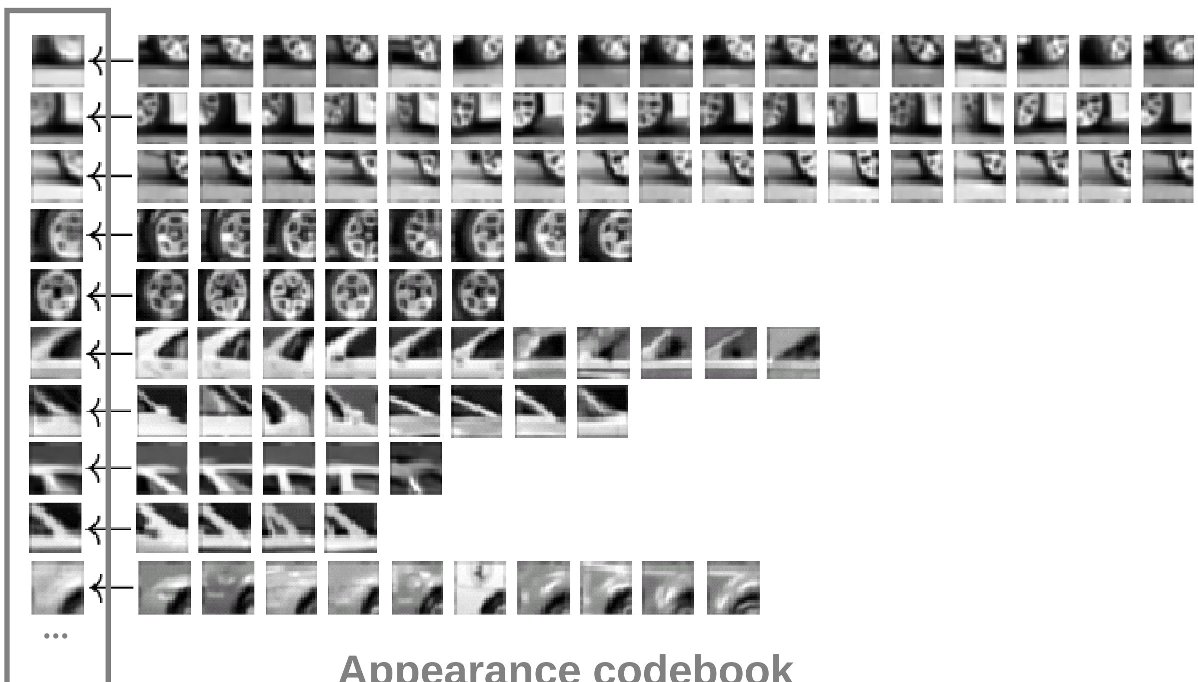
Another example visual vocabulary



Fei-Fei et al. 2005

Example codebook

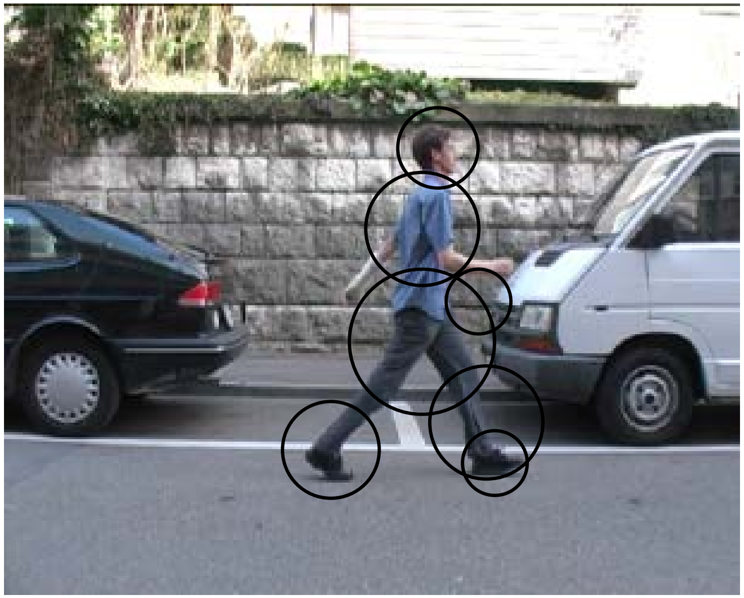




Appearance codebook

Source: B. Leibe

Another codebook

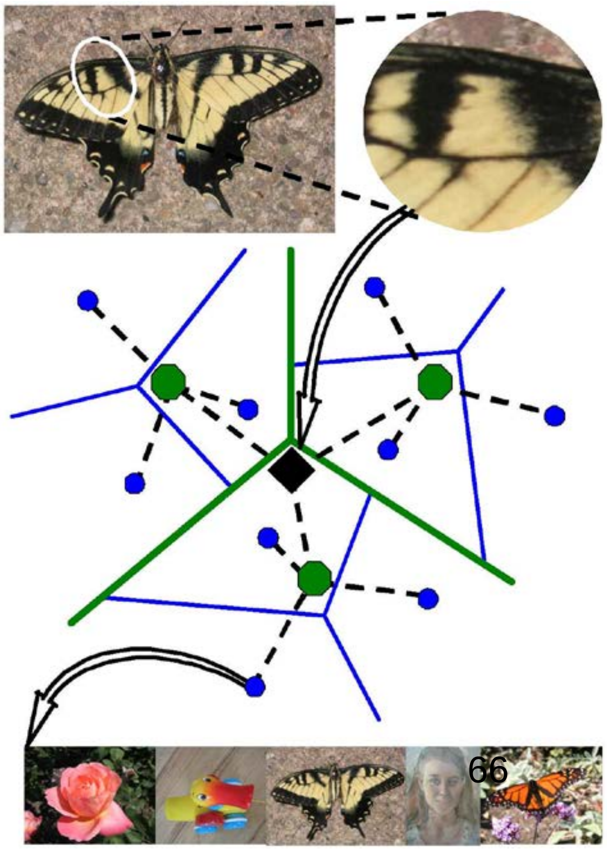


Appearance codebook

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

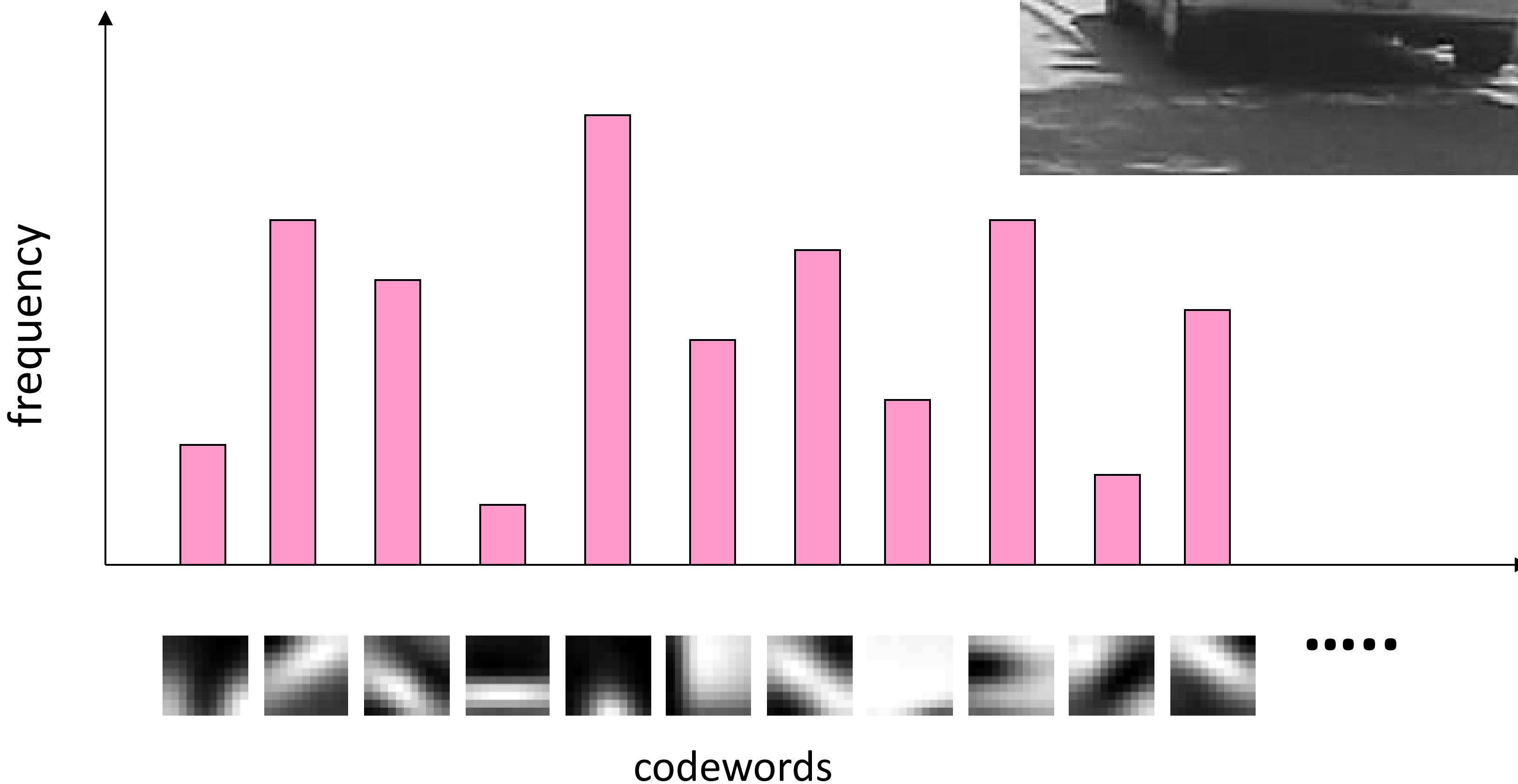
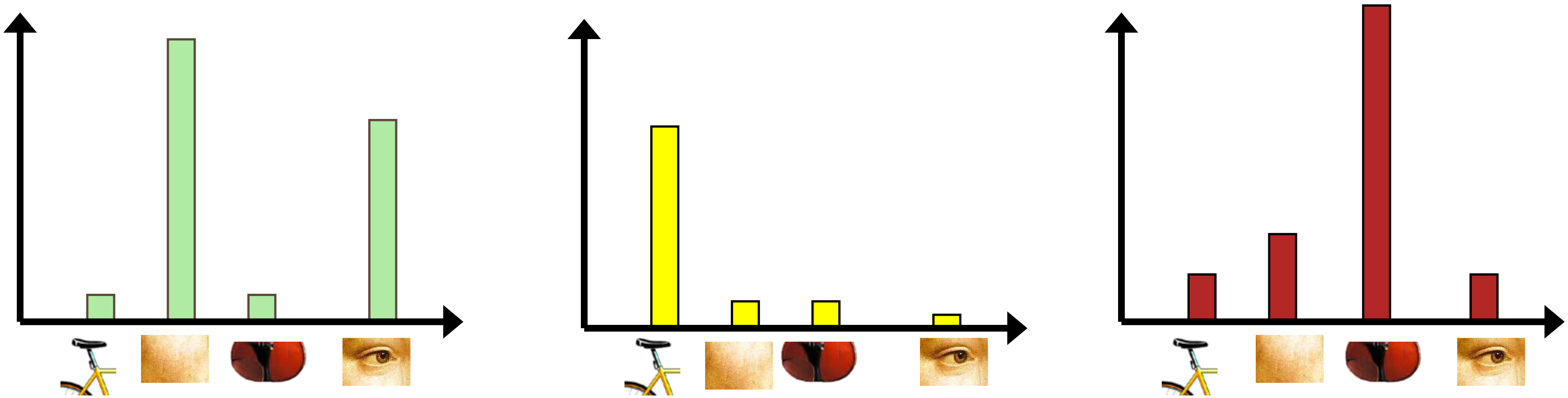
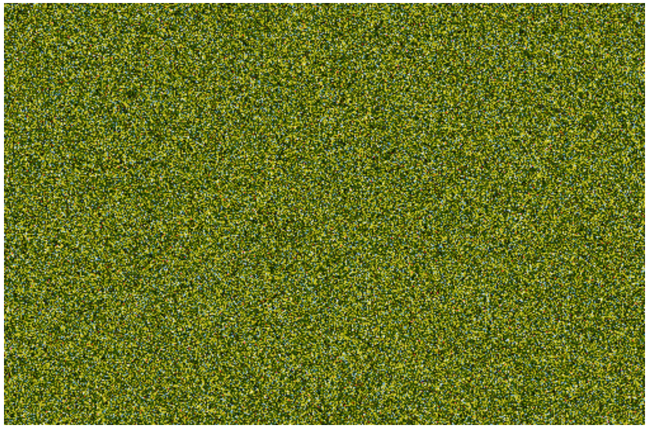


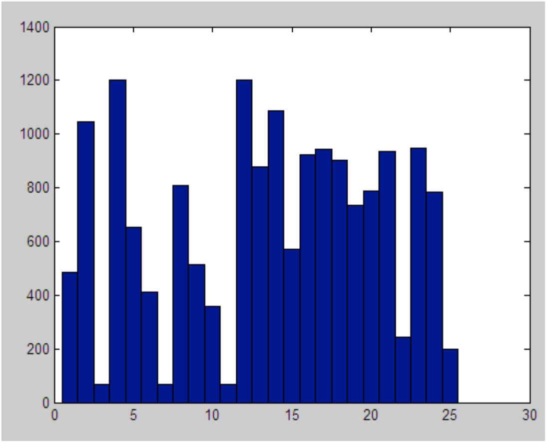
Image classification

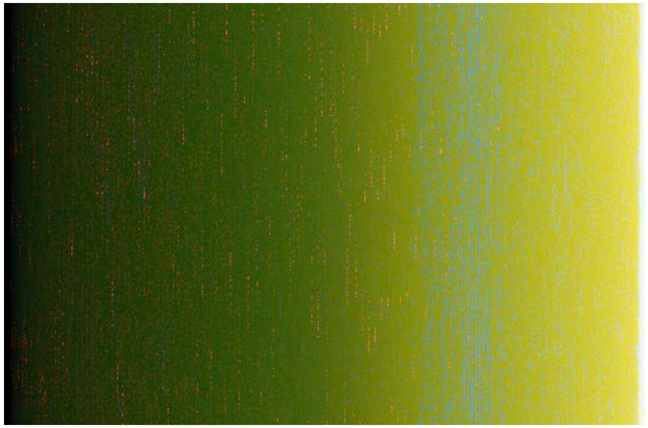
- Given the bag-of-features representations of images from different classes, learn a classifier using machine learning



But what about layout?



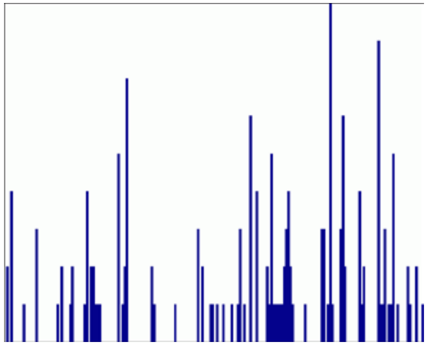




All of these images have the same color histogram

Spatial pyramid representation

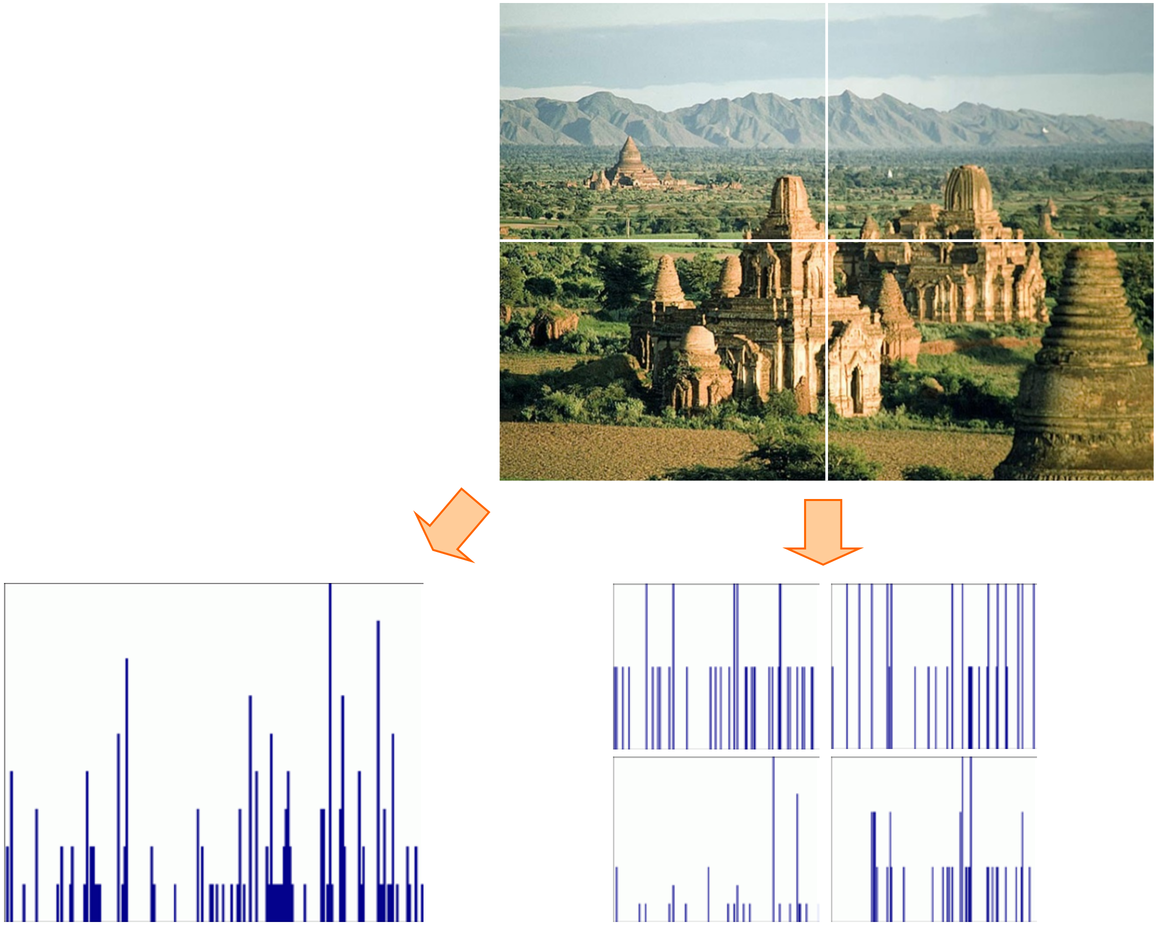
- Extension of a bag of features
- Locally orderless representation at several levels of resolution



Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramid representation

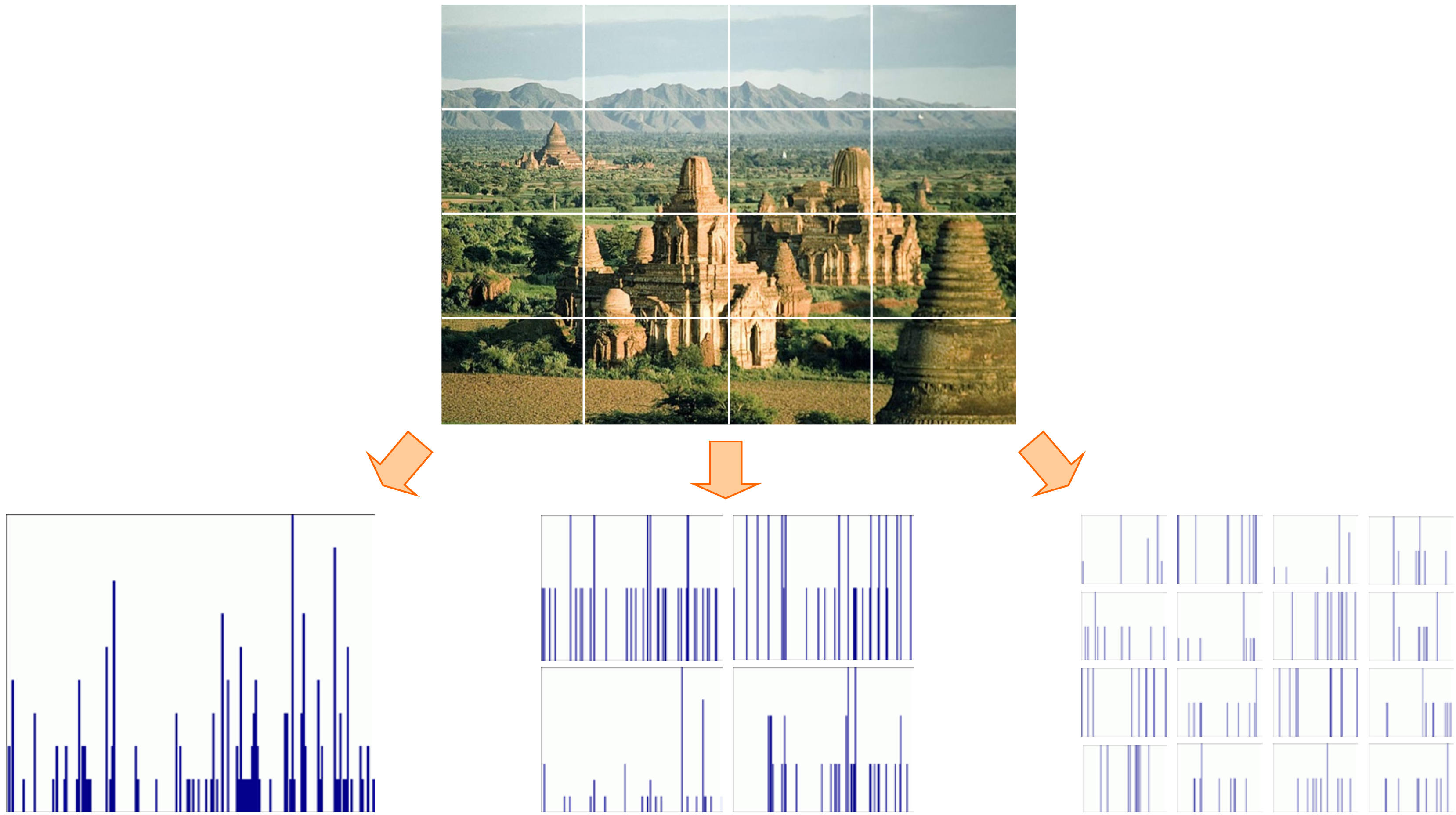
- Extension of a bag of features
- Locally orderless representation at several levels of resolution



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)

Finale

- Describing images or image patches is very important for matching and recognition
- The SIFT descriptor was invented in 1999 and is still very heavily used.
- Other descriptors are also available, some much simpler, but less powerful.
- Texture and shape descriptors are also useful.
- Bag-of-words is a handy technique borrowed from text retrieval. Lots of people use it to compare images or regions.
- Sivic developed a video frame retrieval system using this method, called it Video Google.
- The spatial pyramid allows us to describe an image as a whole and over its parts at multiple levels.