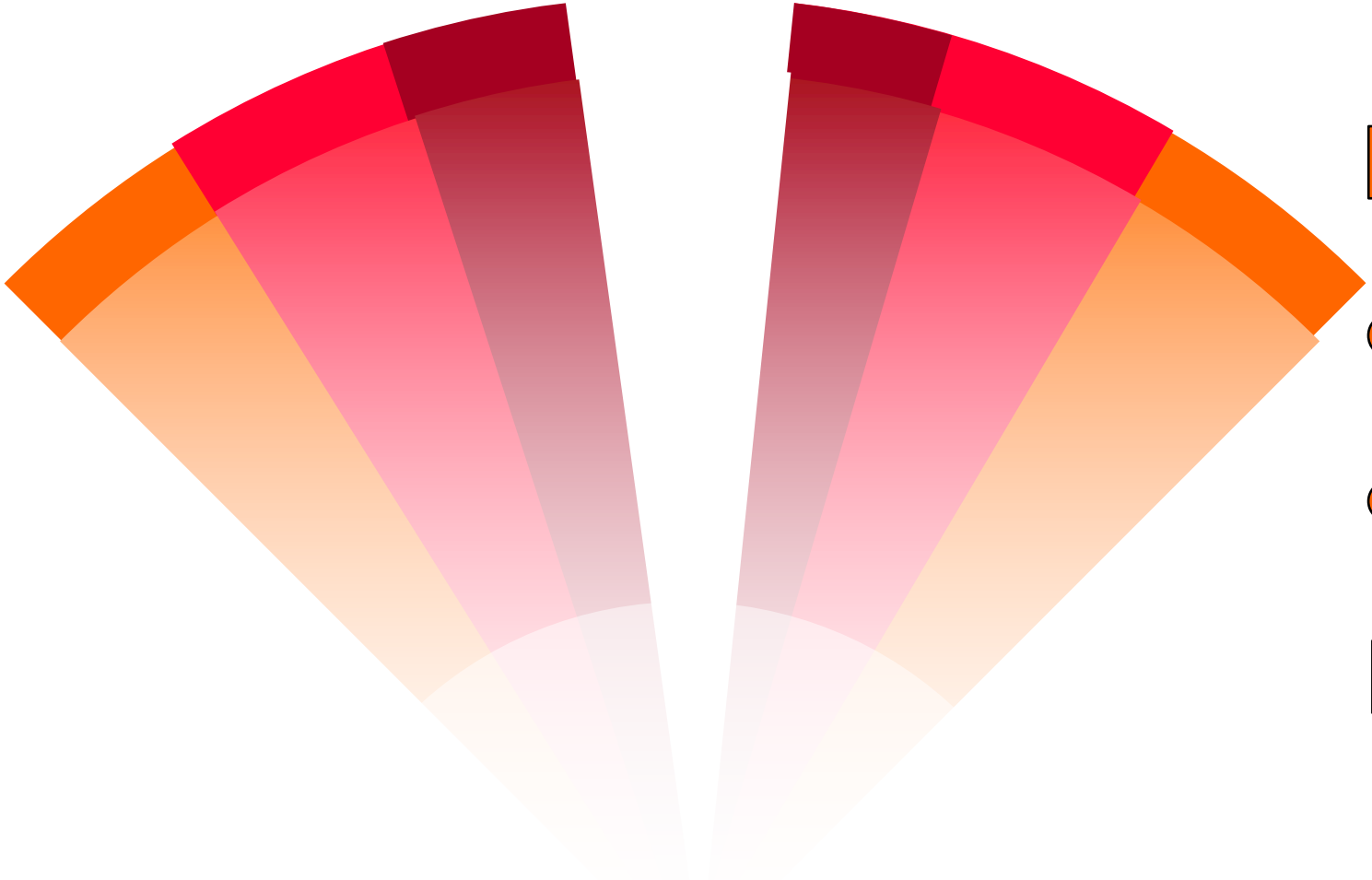
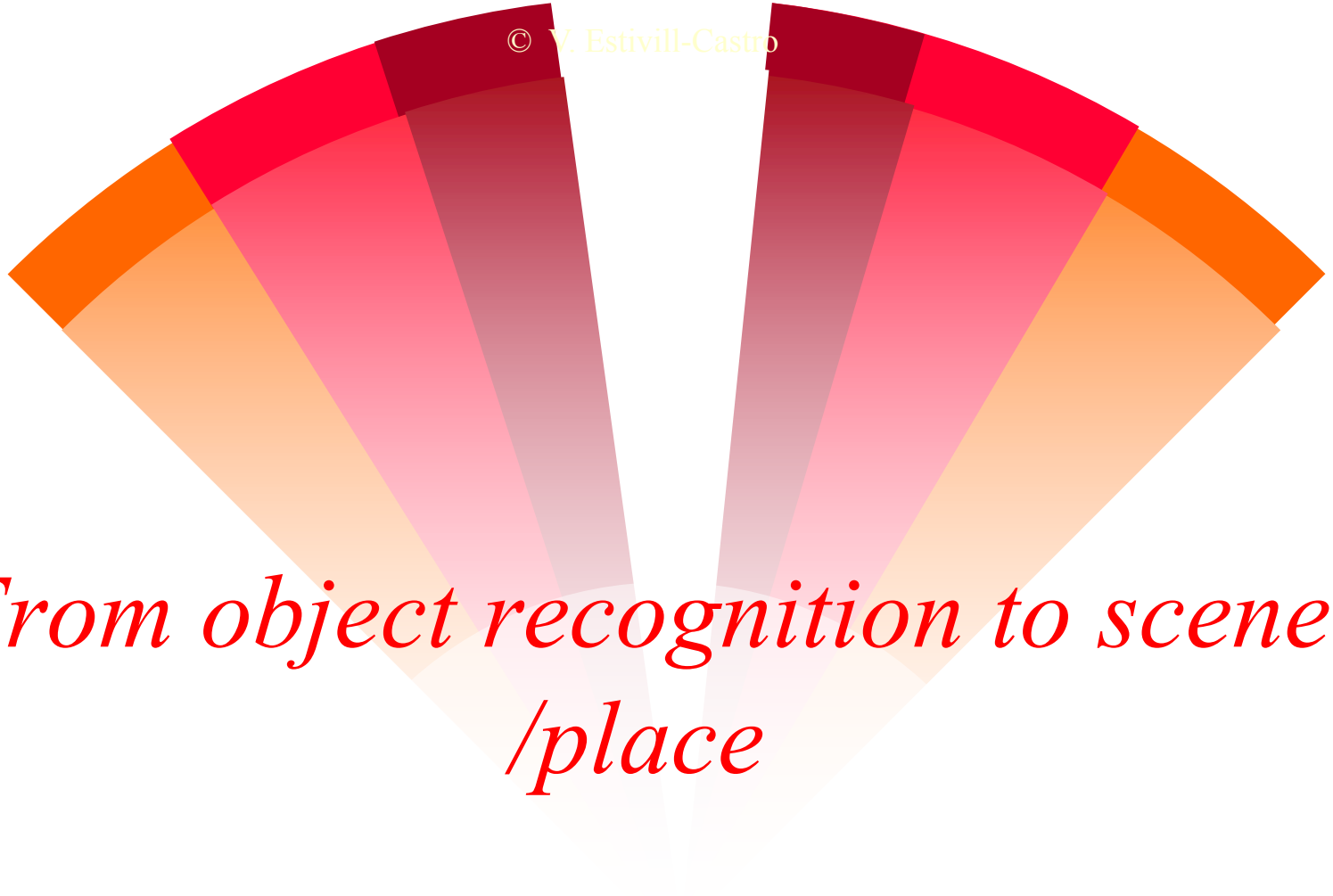


Robots and People

Vlad Estivill-Castro (2016)

Robots for People

--- A project for intelligent integrated systems



From object recognition to scene /place

Vision

Chapter 4 (textbook)

Sections 4.6 n 4.7

What is the course about?

- textbook
- **Introduction to Autonomous Mobile Robots**
- second edition

  Roland Siegwart, Illah R. Nourbakhsh, and Davide Scaramuzza

Many topics regarding vision, object recognition, place recognition

- Bag of words
  - a bit on vector model
  - a bit on clustering
  - a bit on hierarchical model

Visual Bag of words

- Image is recognized by the appearance information it holds.

Bag of visual words representing and image

… for detection or recognition tasks

Bag of words model

- Typical IR (Information Retrieval) data model:
  - Each document is just a bag (multiset) of words ("terms")
- Detail 1: "Stop Words"
  - Certain words are considered irrelevant and not placed in the bag
  - e.g., "the"
  - e.g., HTML tags like <H1>
- Detail 2: "Stemming" and other content analysis
  - Using English-specific rules, convert words to their basic form
  - e.g., "surfing", "surfed" --> "surf"

Document Vectors

- Documents are represented as "bags of words"
- Represented as vectors when used computationally
  - A vector is like an array of floating point
  - Has direction and magnitude
  - Each vector holds a place for every term in the collection
  - Therefore, most vectors are sparse

Document Vectors:

One location for each word

--- the vocabulary

	nova	galaxy	heat	h'wood	film	role	diet	fur
A	10	5	3					
B								
C	5	10						
D				10	8	7		
E				9	10	5		
F							10	10
G							9	10
H	5	7			9			
I		6	10	2	8			

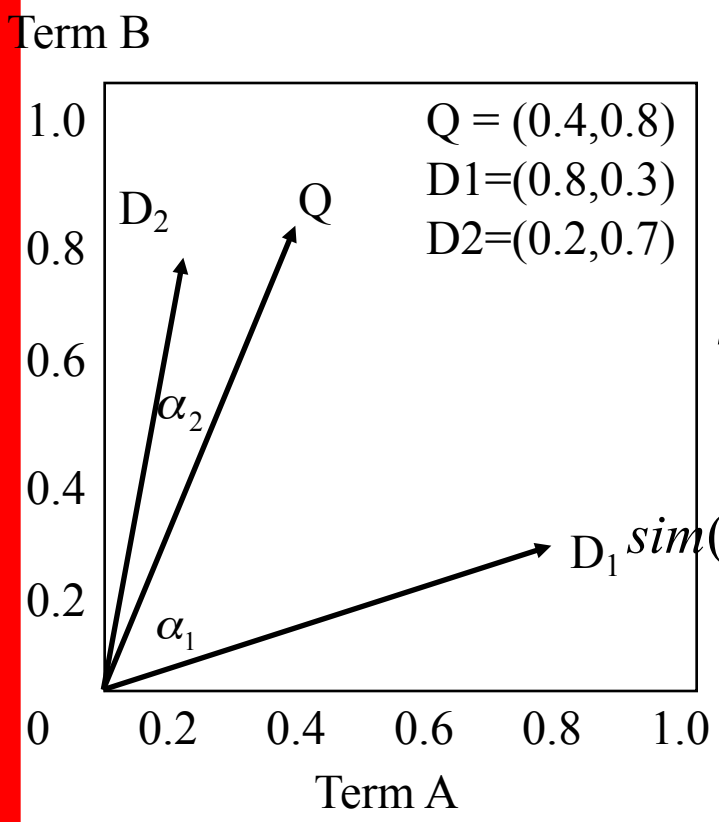
Nova" occurs 10 times in text A
"Galaxy" occurs 5 times in text A
"Heat" occurs 3 times in text A
" (Blank means 0 occurrences.)

3

Vector Space Model

- Documents are represented as *vectors* in term space
  - Terms are usually stems
  - Documents represented by binary vectors of terms
- Queries represented the same as documents
- A vector distance measure between the query and documents is used to rank retrieved documents
  - Query and Document similarity is based on length and direction of their vectors
  - Vector operations to capture Boolean query conditions
  - Terms in a vector can be "weighted" in many ways

Vector Space with Term Weights and Cosine Matching



$D_i=(d_{i1},w_{di1};d_{i2}, w_{di2};\ldots;d_{it}, w_{dit})$

$Q=(q_{i1},w_{qi1};q_{i2}, w_{qi2};\ldots;q_{it}, w_{qit})$

$$sim(Q,D_i)=\frac{\sum_{j=1}^{t} w_{q_j} w_{d_{ij}}}{\sqrt{\sum_{j=1}^{t}(w_{q_j})^2 \sum_{j=1}^{t}(w_{d_{ij}})^2}}$$

$$sim(Q,D2)=\frac{(0.4\cdot 0.2)+(0.8\cdot 0.7)}{\sqrt{[(0.4)^2+(0.8)^2]\cdot[(0.2)^2+(0.7)^2]}}$$

$$=\frac{0.64}{\sqrt{0.42}}=0.98$$

$$sim(Q,D_1)=\frac{.56}{\sqrt{0.58}}=0.74$$

TF x IDF Weights

- tf x idf measure:
  - Term Frequency (tf)
  - Inverse Document Frequency (idf) -- a way to deal with the problems of the Zipf distribution
- Goal: Assign a tf * idf weight to each term in each document

TF x IDF Calculation

$$w_{ik} = tf_{ik} * \log(N / n_k)$$

T_k = term k in document D_i

tf_{ik} = frequency of term T_k in document D_i

idf_k = inverse document frequency of term T_k in C

N = total number of documents in the collection C

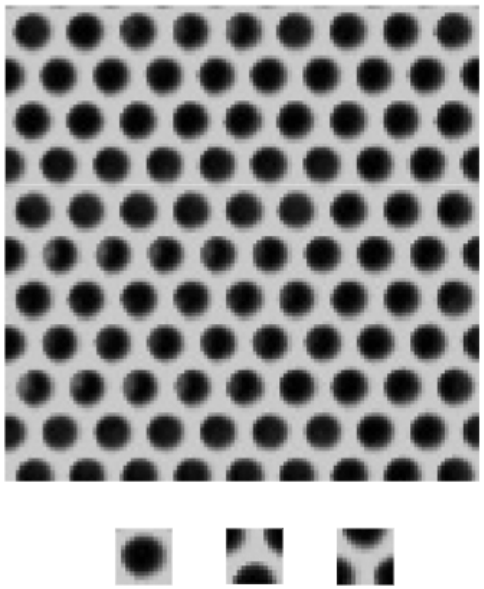
n_k = the number of documents in C that contain T_k

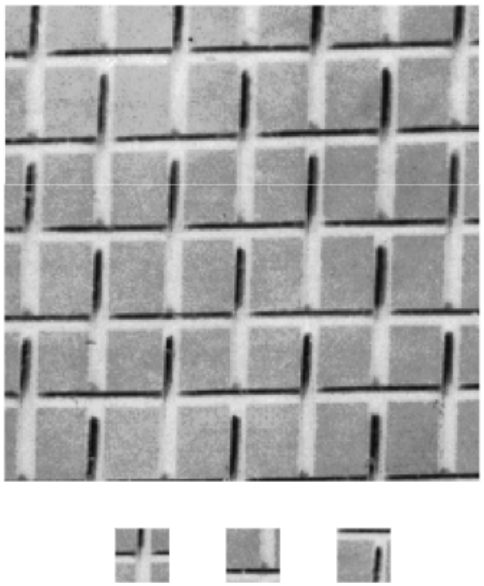
$$idf_k = \log\left(\frac{N}{n_k}\right)$$

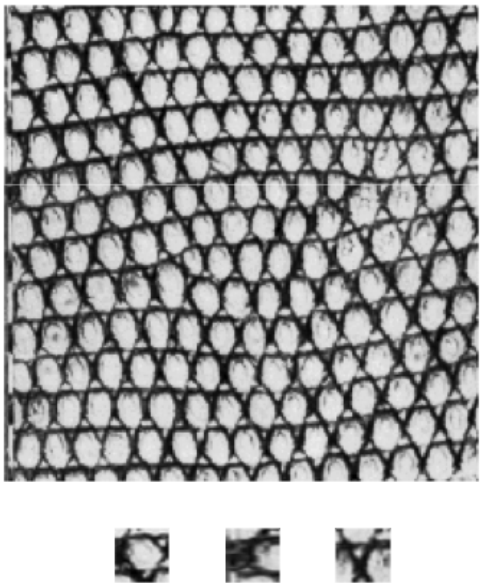
Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary (Salton & McGill , 1983)
- In vision: started with texture recognition
  - Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003
- For stochastic textures, it is the identity of the textons, not their spatial arrangement, that matters

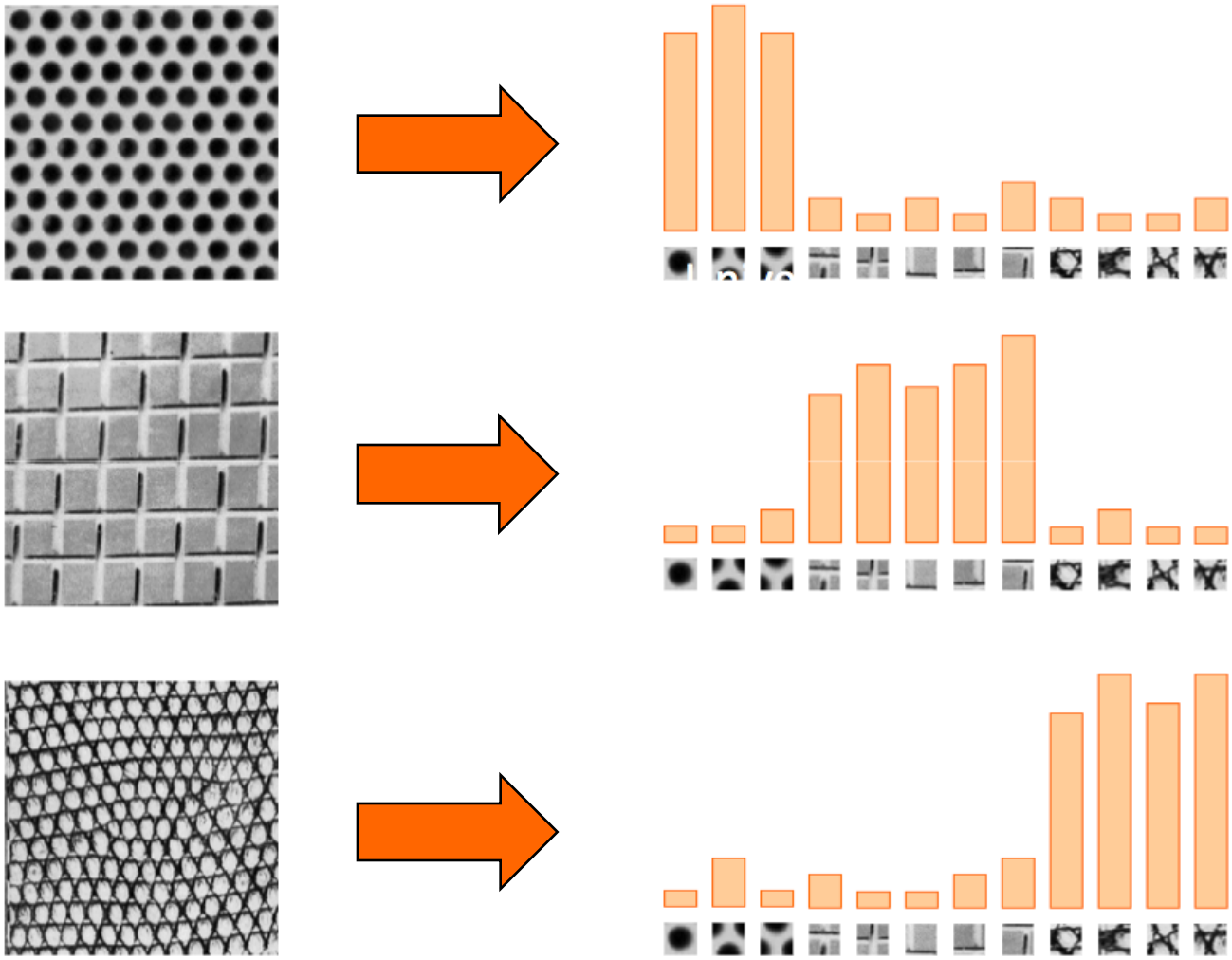
Recognizing texture





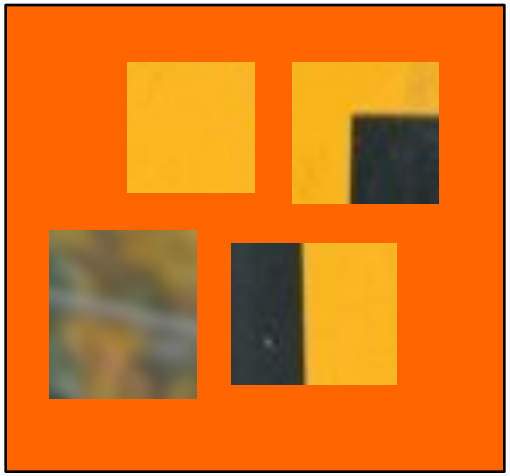


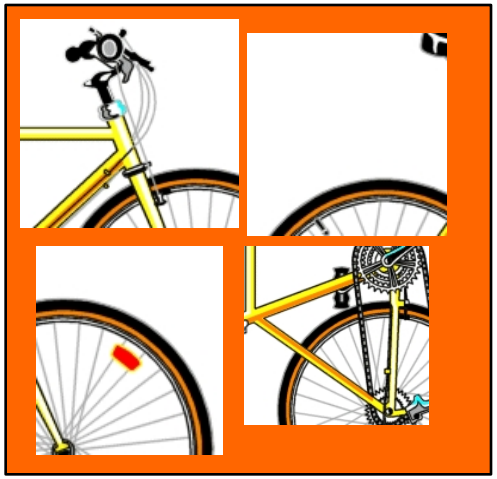
Texture as vectors (with respect to a vocabulary)



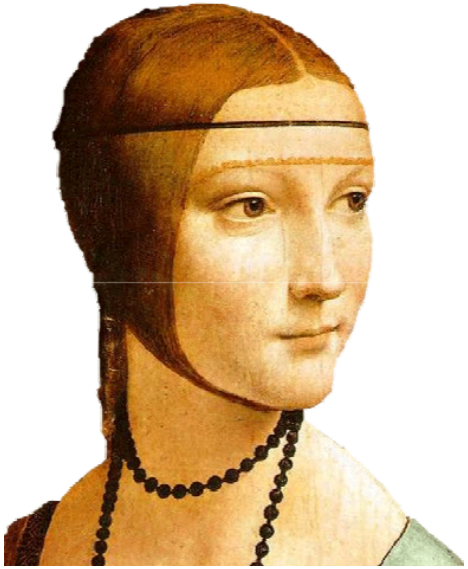
Bag of visual words representation

1. Extract features



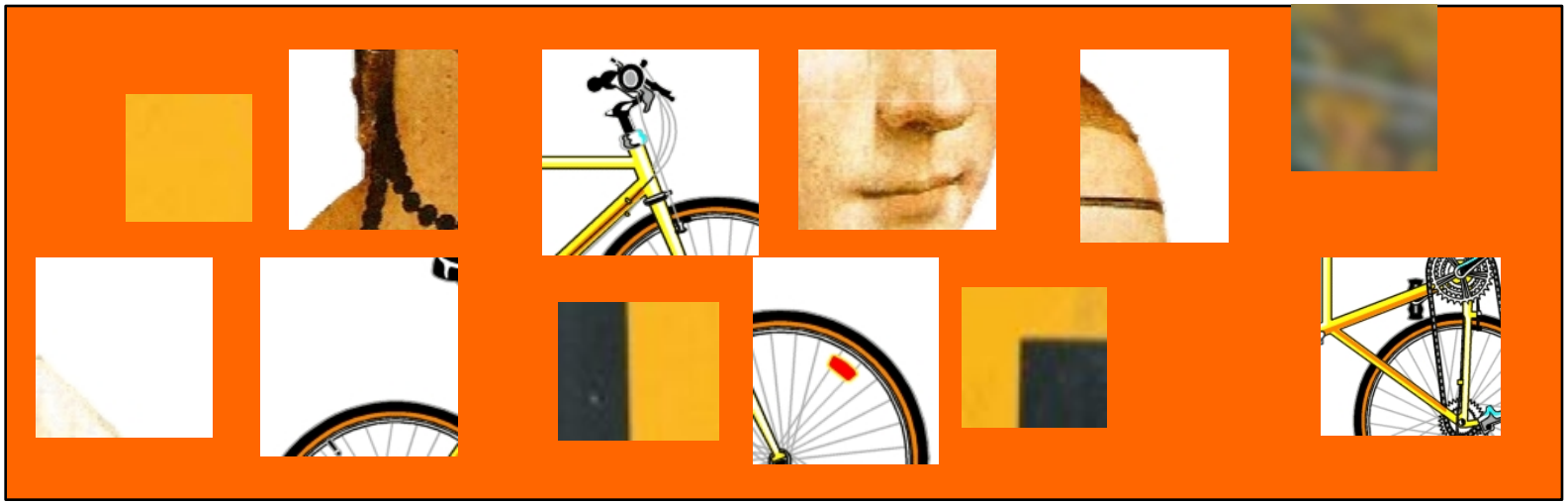


Robots and people



Bag of visual words representation

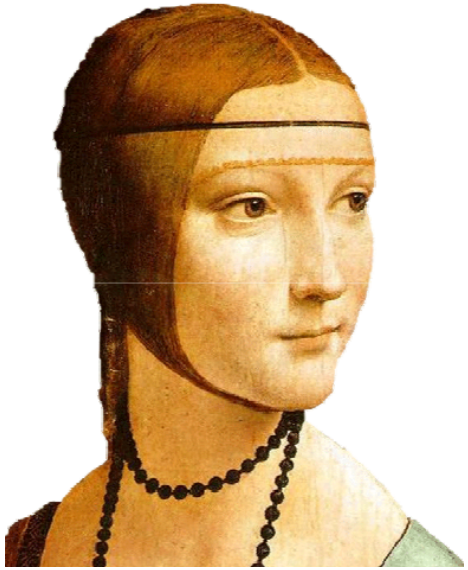
1. Extract features
2. Learn “visual vocabulary”

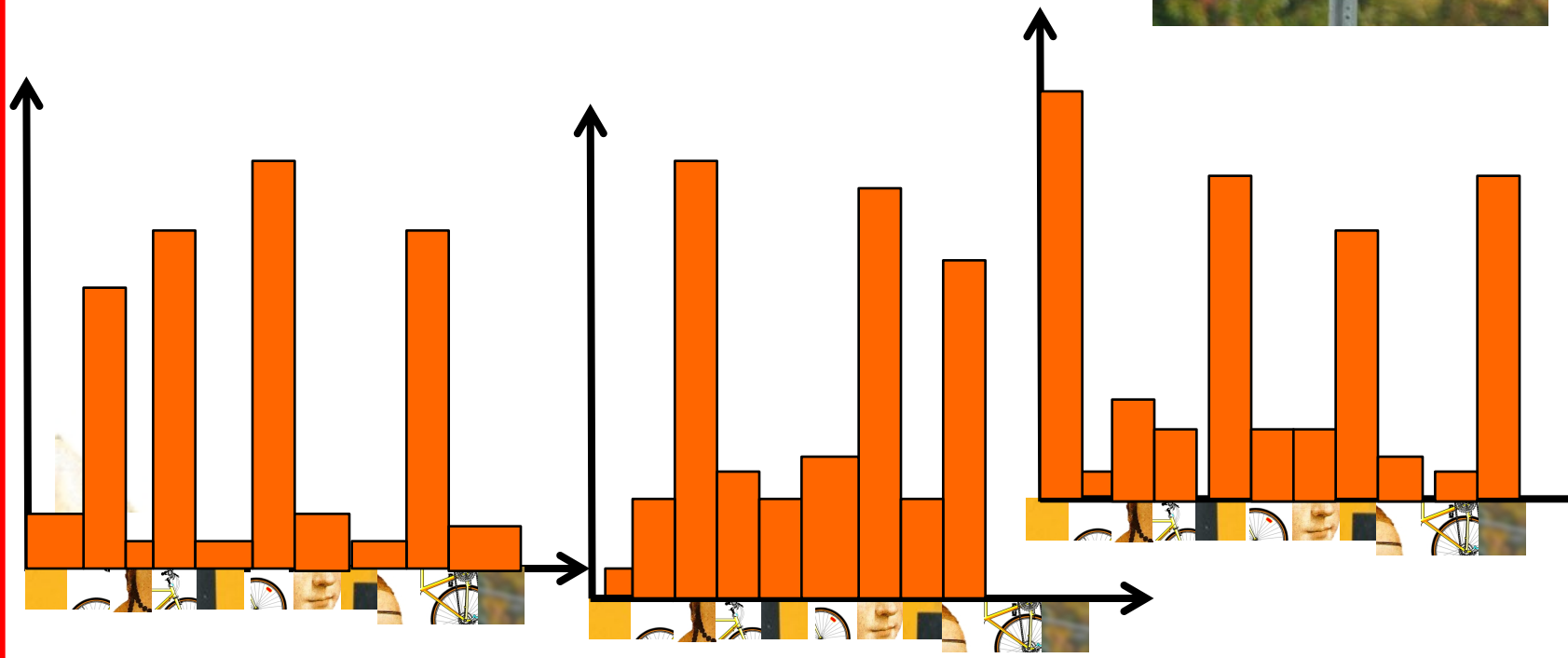


Bag of visual words representation

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

Robots and people

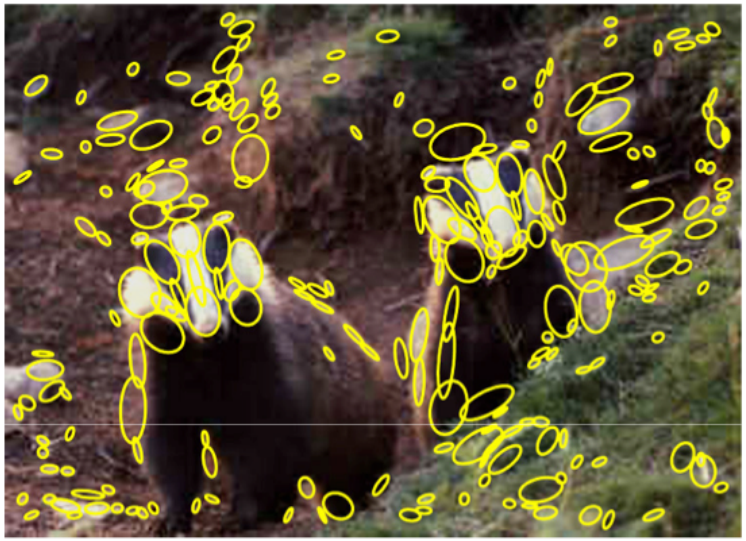




1.-Feature extraction (preference is SIFT descriptor)

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

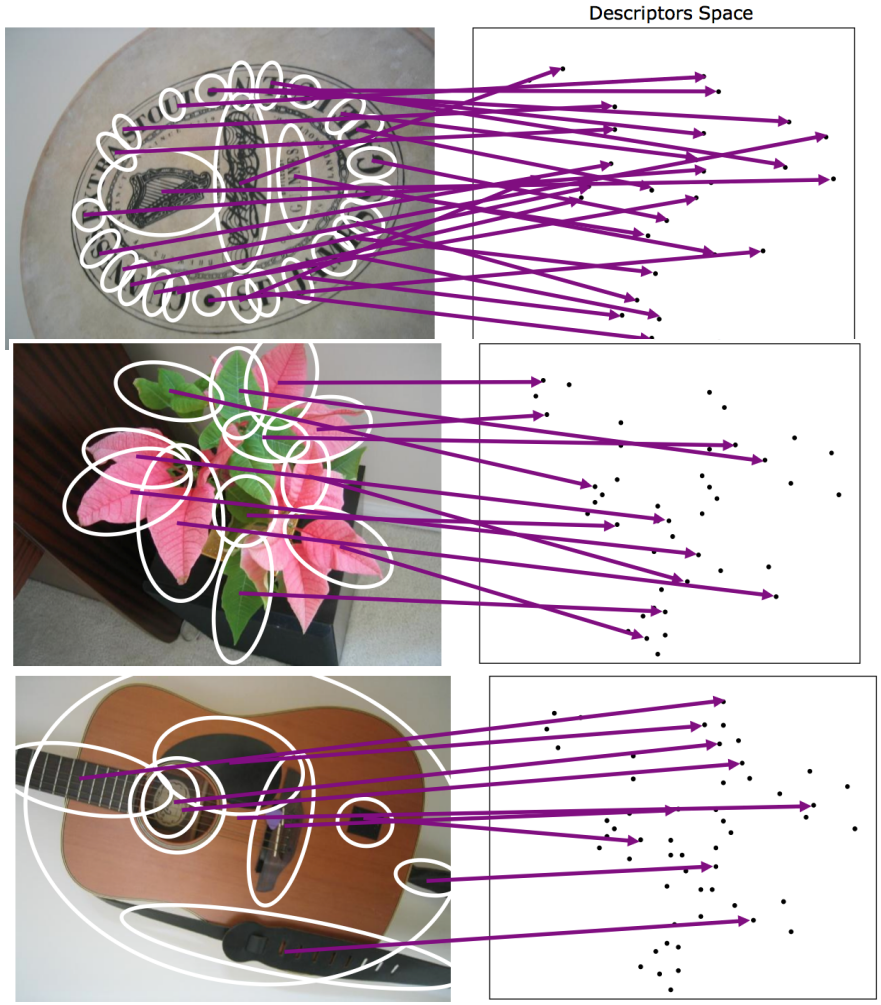
Random sampling (Vidal-Naquet & Ullman, 2002)

Segmentation-based patches (Barnard et al. 2003)

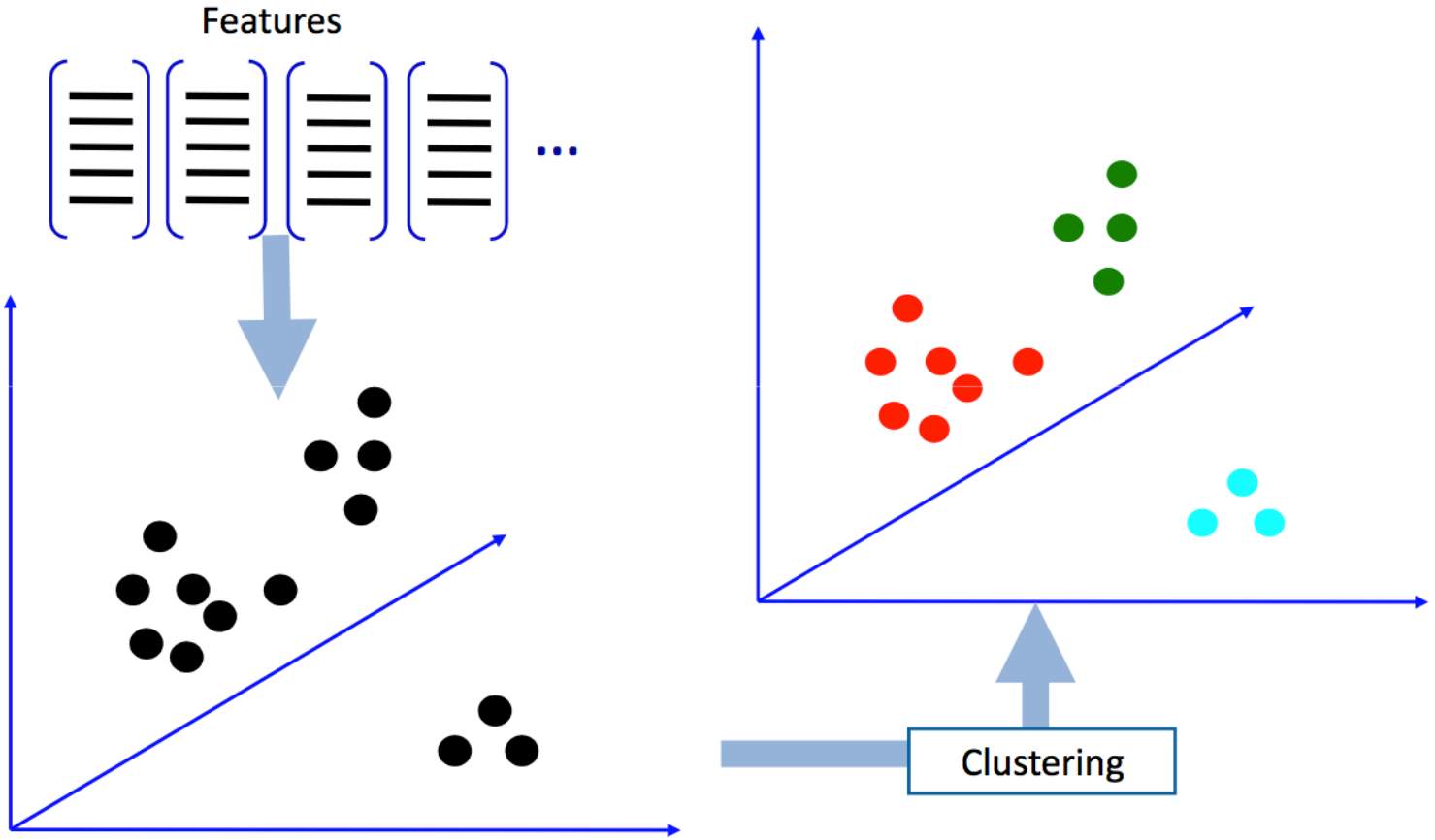
2.-Learn the visual vocabulary

- A visual word will be the patch around the interesting point
  - ore more succinctly, the SIFT descriptor for the interesting point.

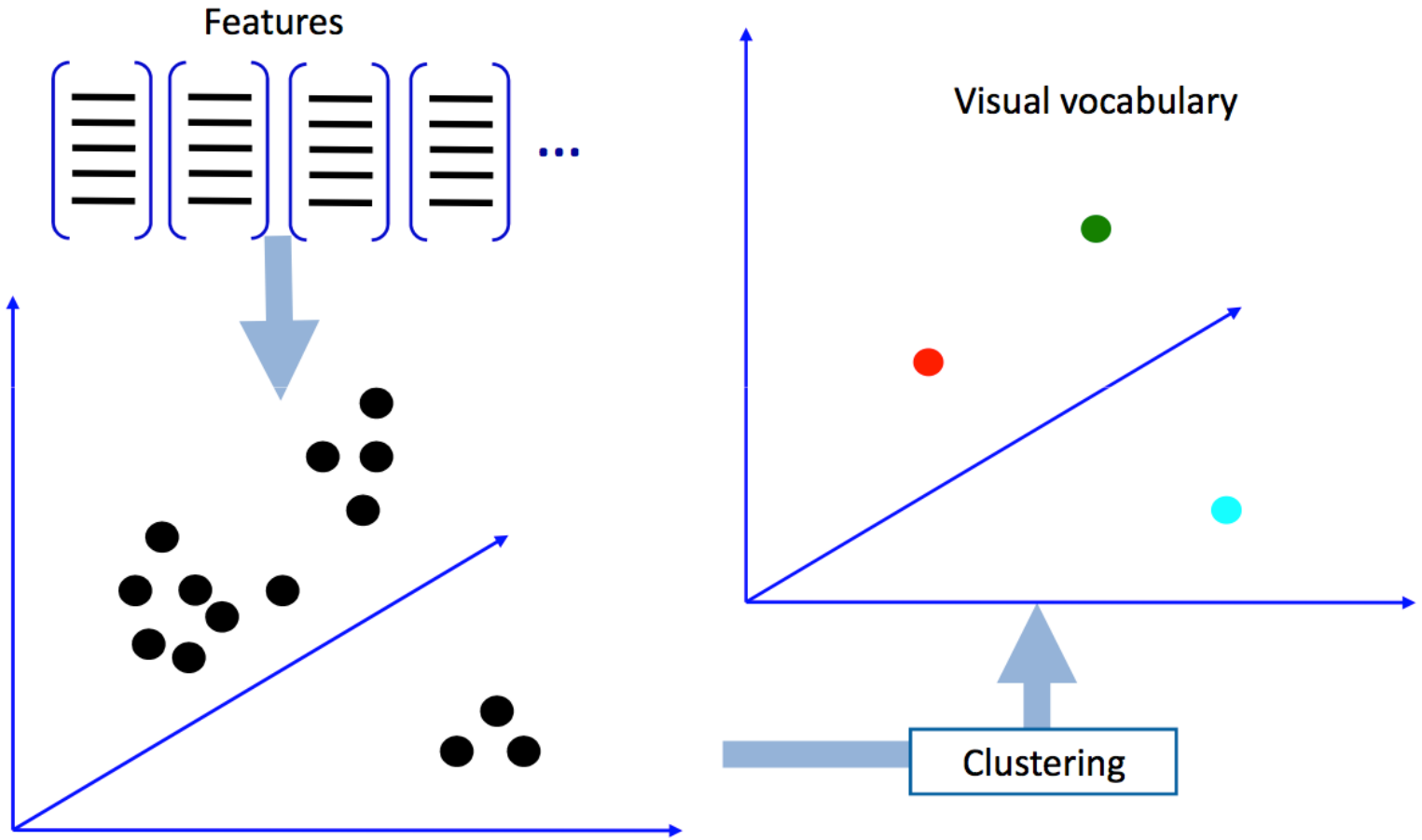
Some "visual words" are now points in space



Find the vocabulary as the centers of a center-based clustering method



Find the vocabulary as the centers of a center-based clustering method



Most popular clustering method

Initialize *k* cluster centers

Do

Assignment step: Assign each data point to its closest cluster center

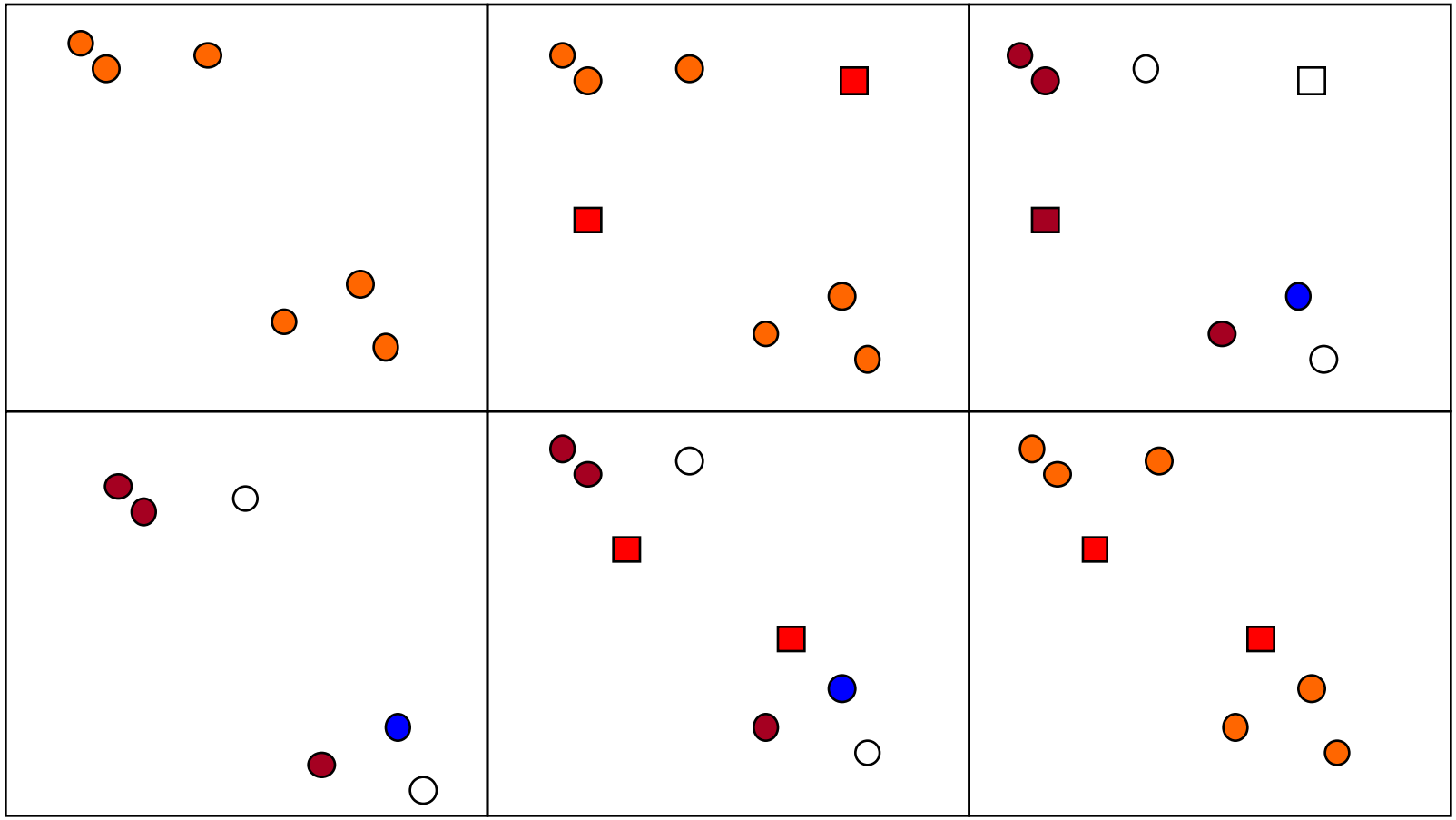
Re-estimation step: Re-compute cluster centers

While (there are still changes in the cluster centers)

Visualization at:

- http://www.delft-cluster.nl/textminer/theory/kmeans/kmeans.html

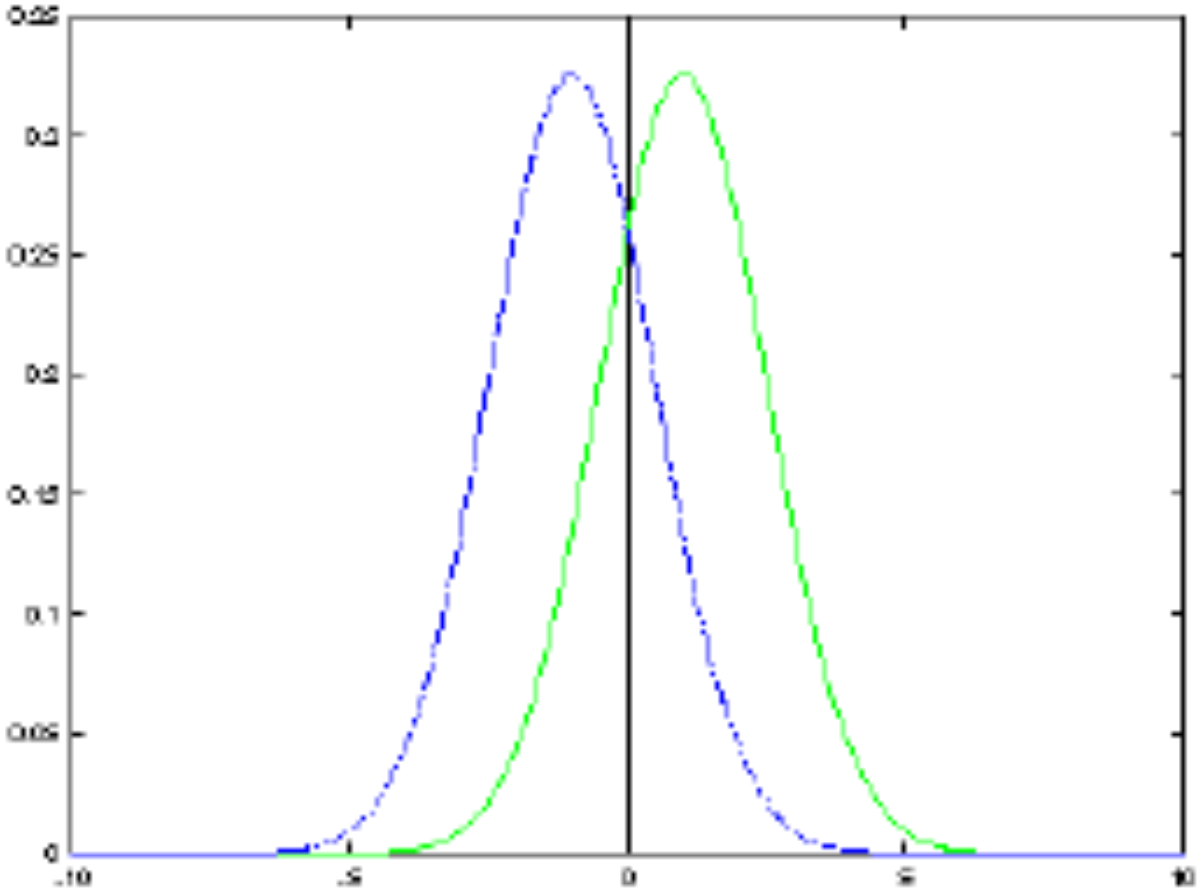
k-means



Properties of k-means

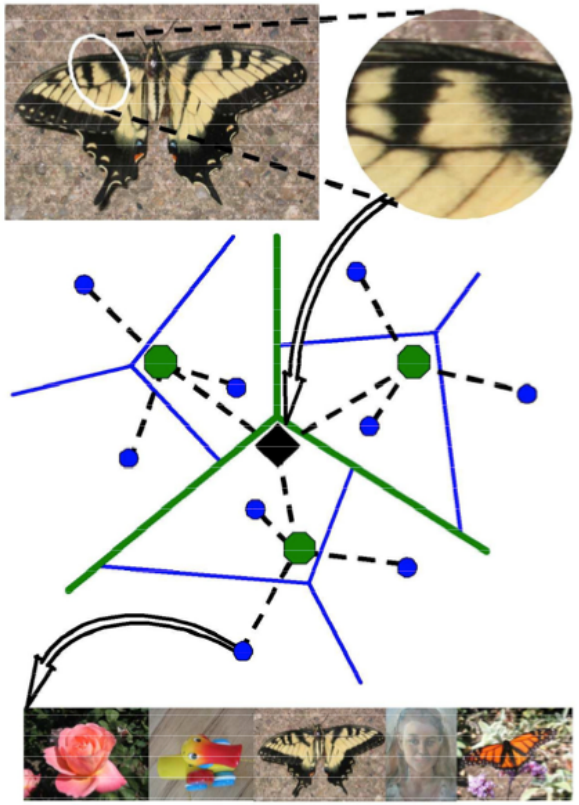
- When the number *n* of records is large the complexity is $\Theta(tmkn) = \Theta(n)$
  - IT IS LINEAR !
- The disadvantages
  - for numerical data
    - fixed with *k*-modes [Huang97a,Huang97b]
  - it is statistically biased
  - it is very sensitive to the presence of noise and outliers as well as to the initial random clustering.
  - representatives may not be valid records

Bias illustration



Vocabulary size (the k in k-means)

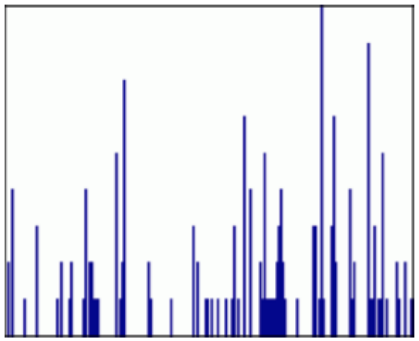
- How to choose vocabulary size?
  - Too small: visual words not representative of all patches
  - Too large: quantization artifacts, overfitting
- Computational efficiency
  - Vocabulary trees (Nister & Stewenius, 2006)



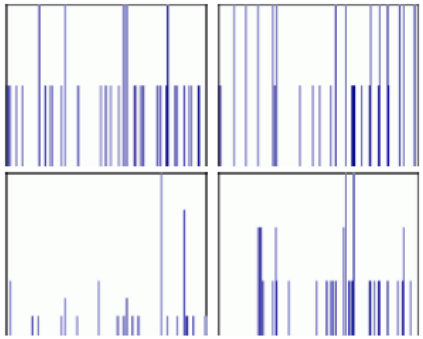
Overcome problems

- The order of patches (or position) in the image gives same result
- Include spatial information
  - Extension of a bag of features
  - Locally orderless representation at several levels of resolution
  - (Lazebnik, Schmid & Ponce, 2006)

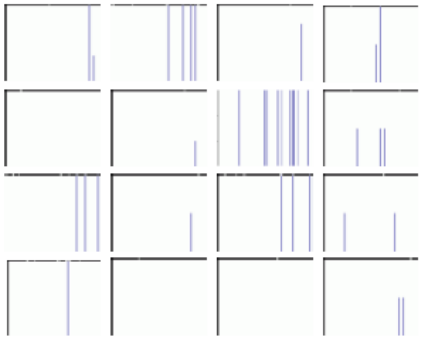
Hierarchy of information to build the vector



level 0



level 1



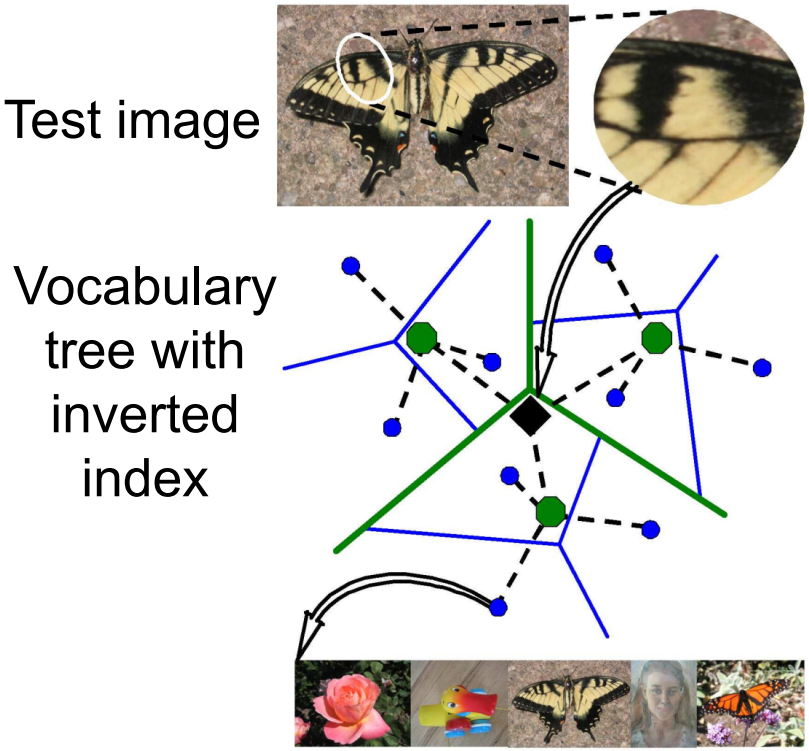
level 2

Object/Image recognition

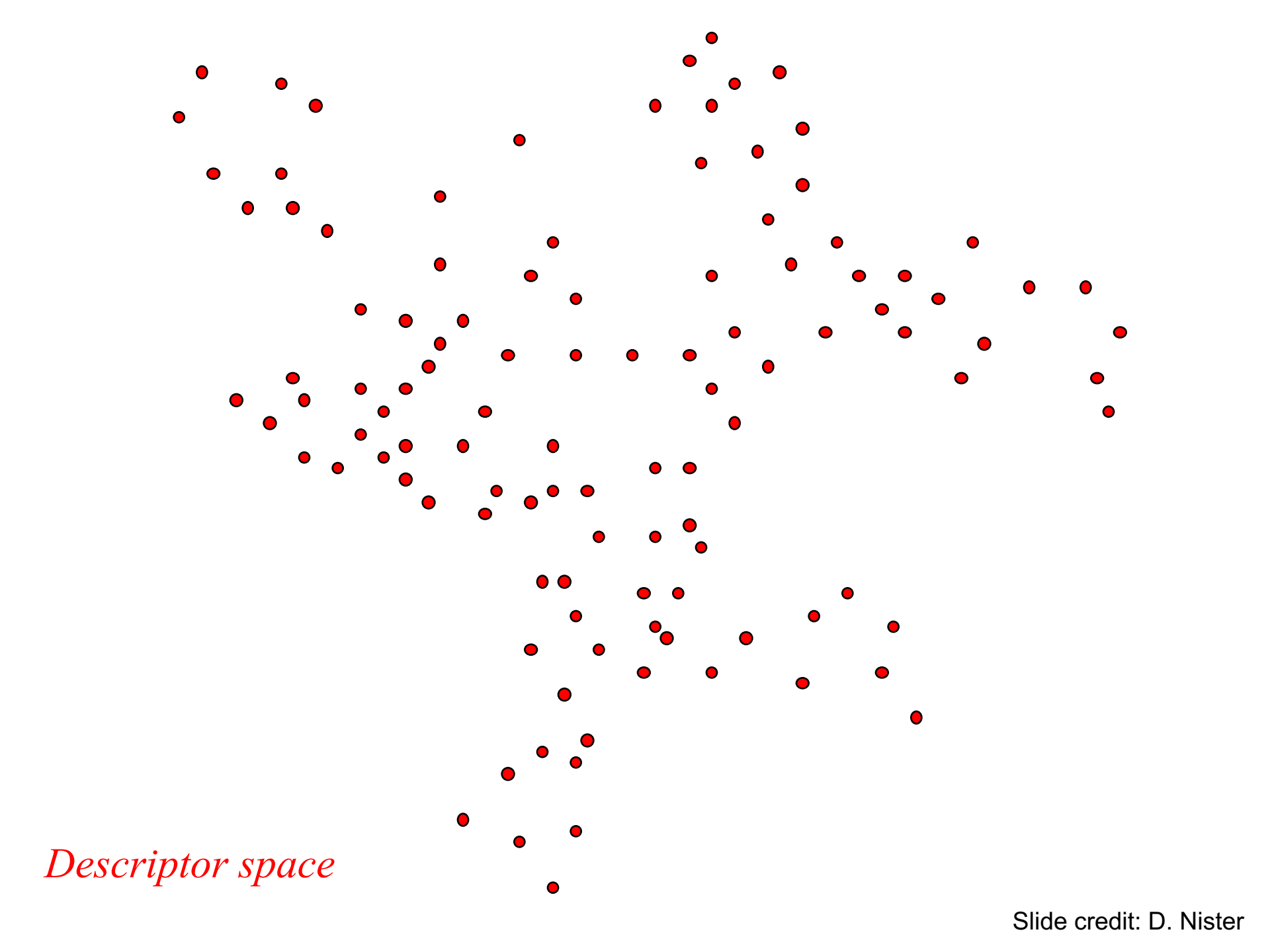
- we can describe a **scene as a collection of words** and look up in the database for **images with a similar collection of words**
- What if we need to find an object/scene in a database of millions of images?
  - Build Vocabulary **Tree** via hierarchical clustering
  - Use the **Inverted File** system: each node in the tree is associated with a list of images containing an instance of this node

Scalability: Alignment to large databases

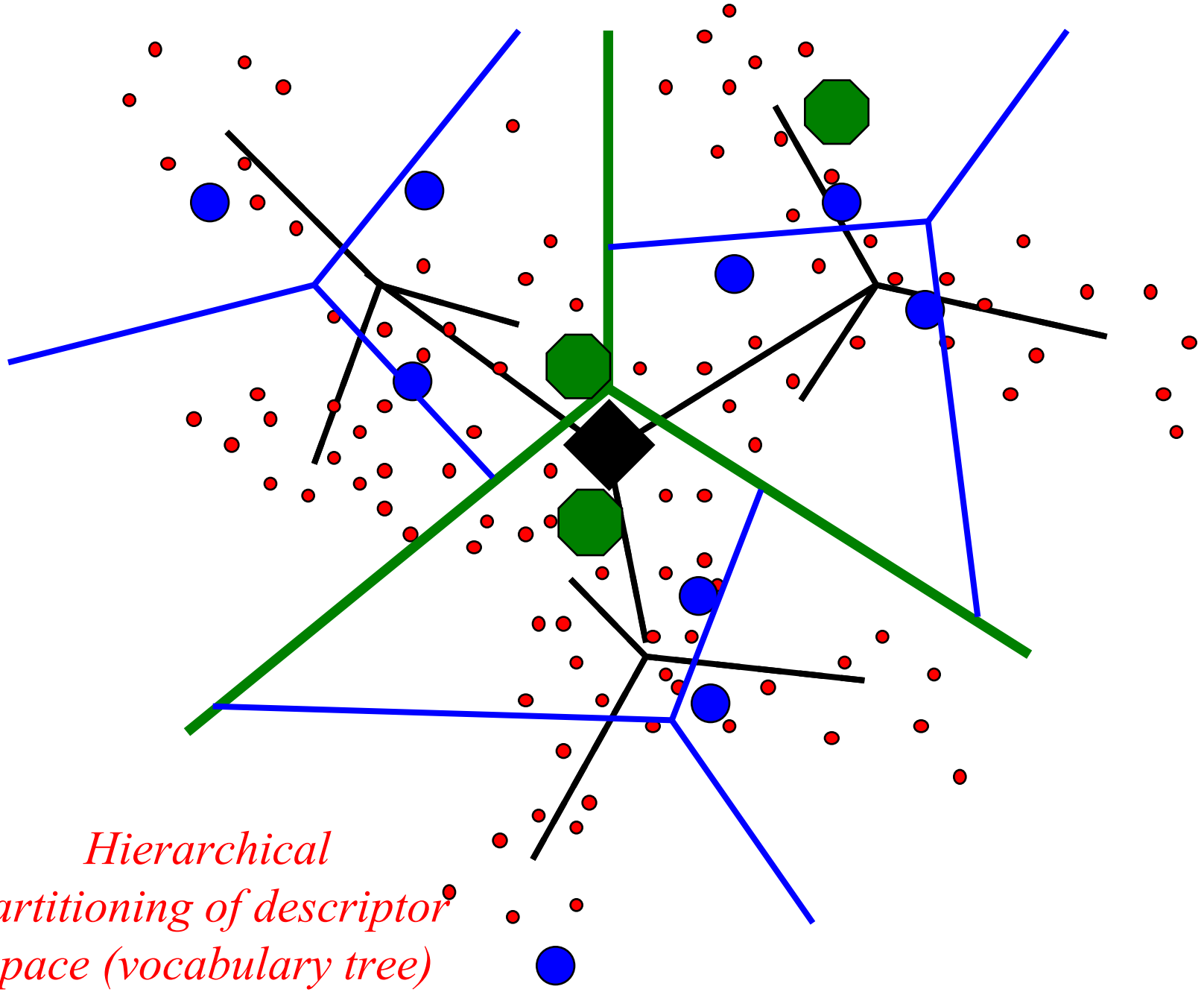
- Efficient putative match generation
  - Fast nearest neighbor search, inverted indexes



D. Nistér and H. Stewénius, Scalable Recognition with a Vocabulary Tree, CVPR 2006

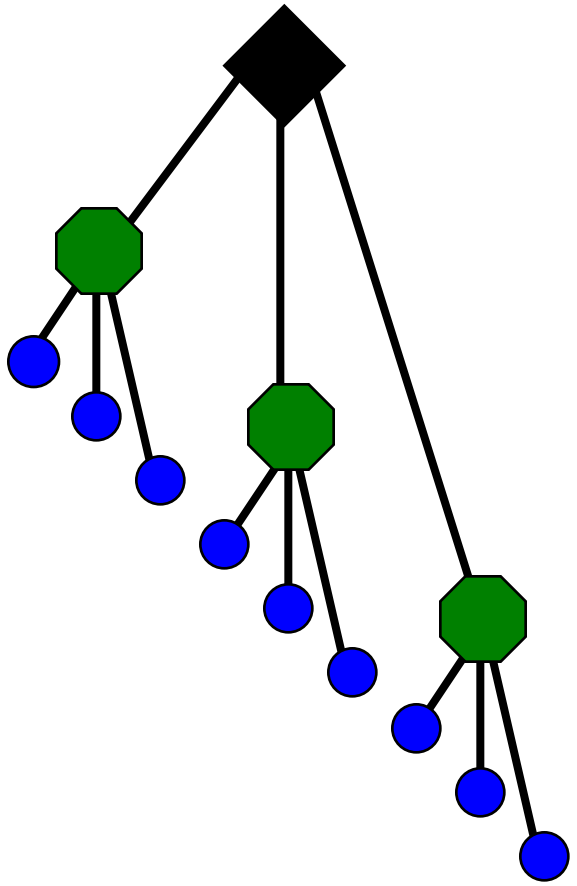


Slide credit: D. Nister



Hierarchical partitioning of descriptor space (vocabulary tree)

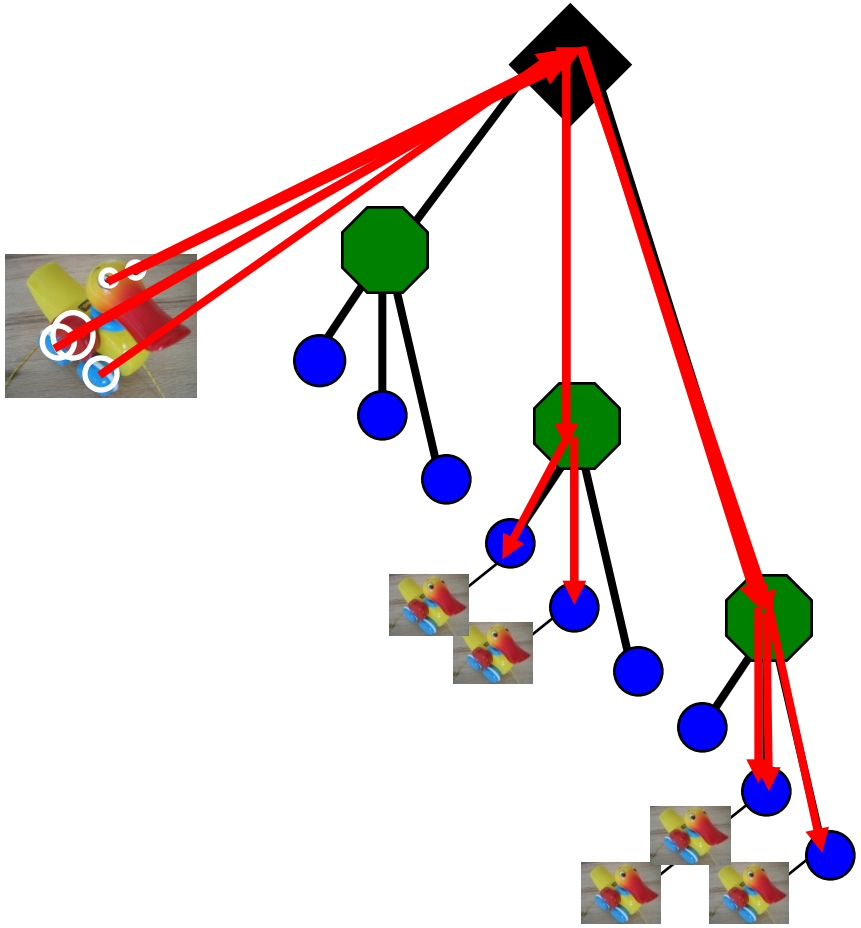
Slide credit: D. Nister



Vocabulary tree/inverted index

Slide credit: D. Nister

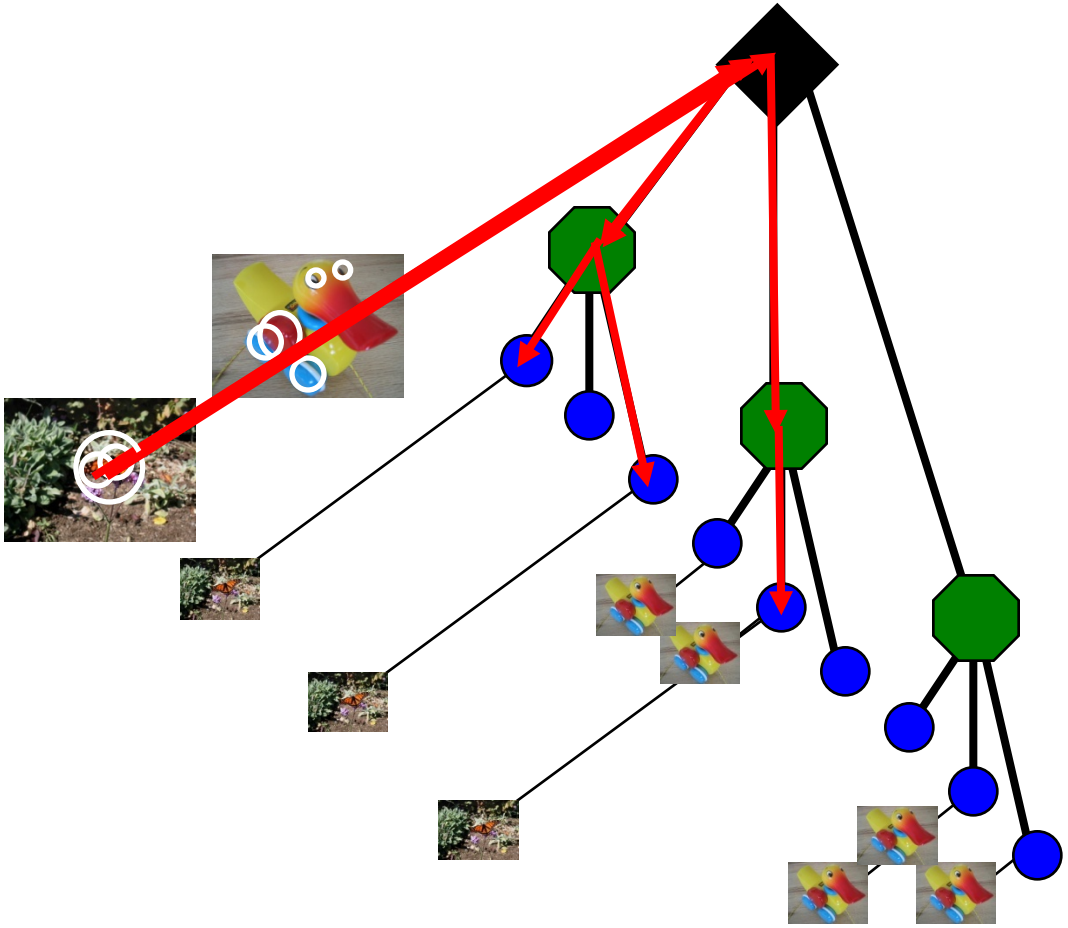
Model images



Populating the vocabulary tree/inverted index

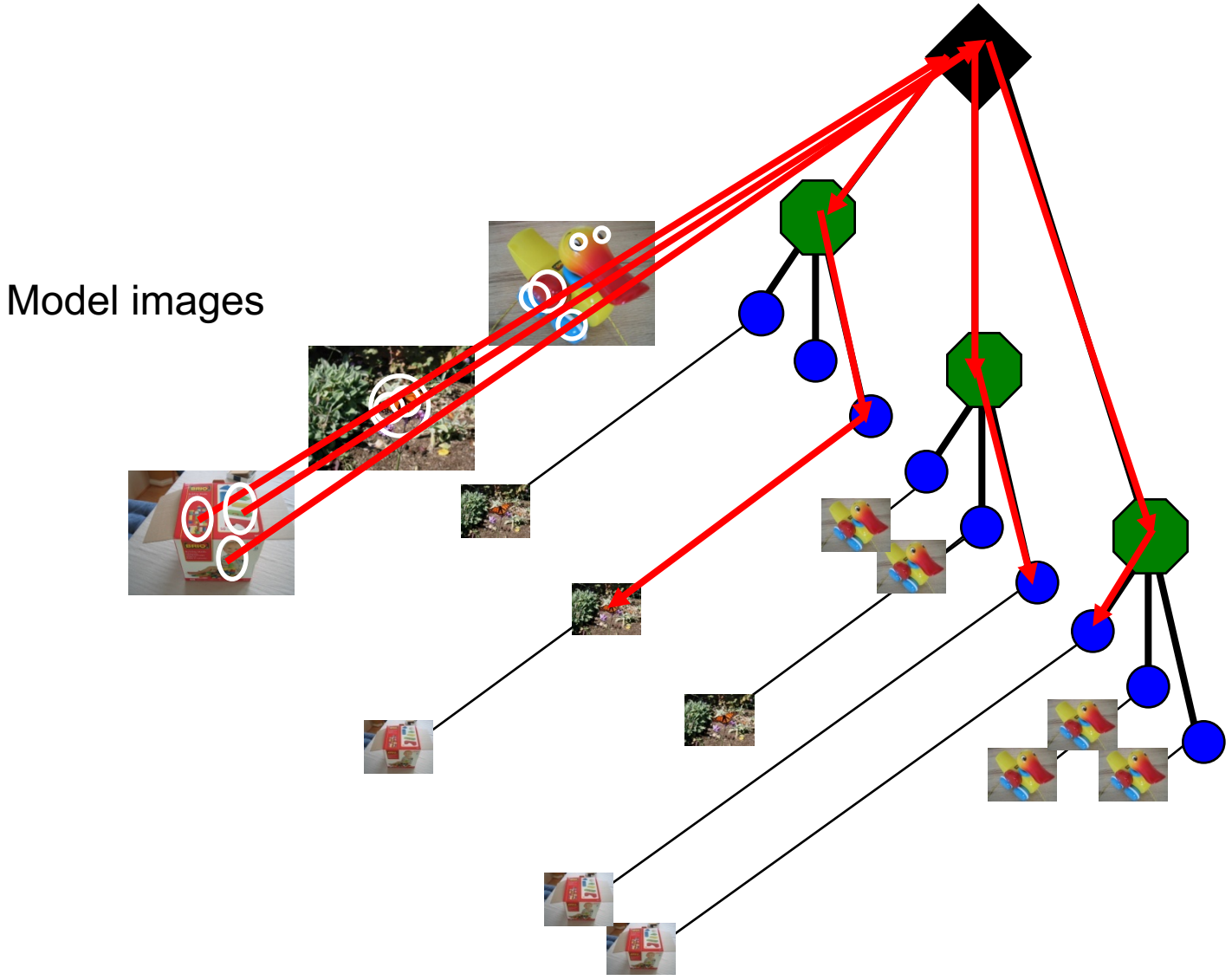
Slide credit: D. Nister

Model images



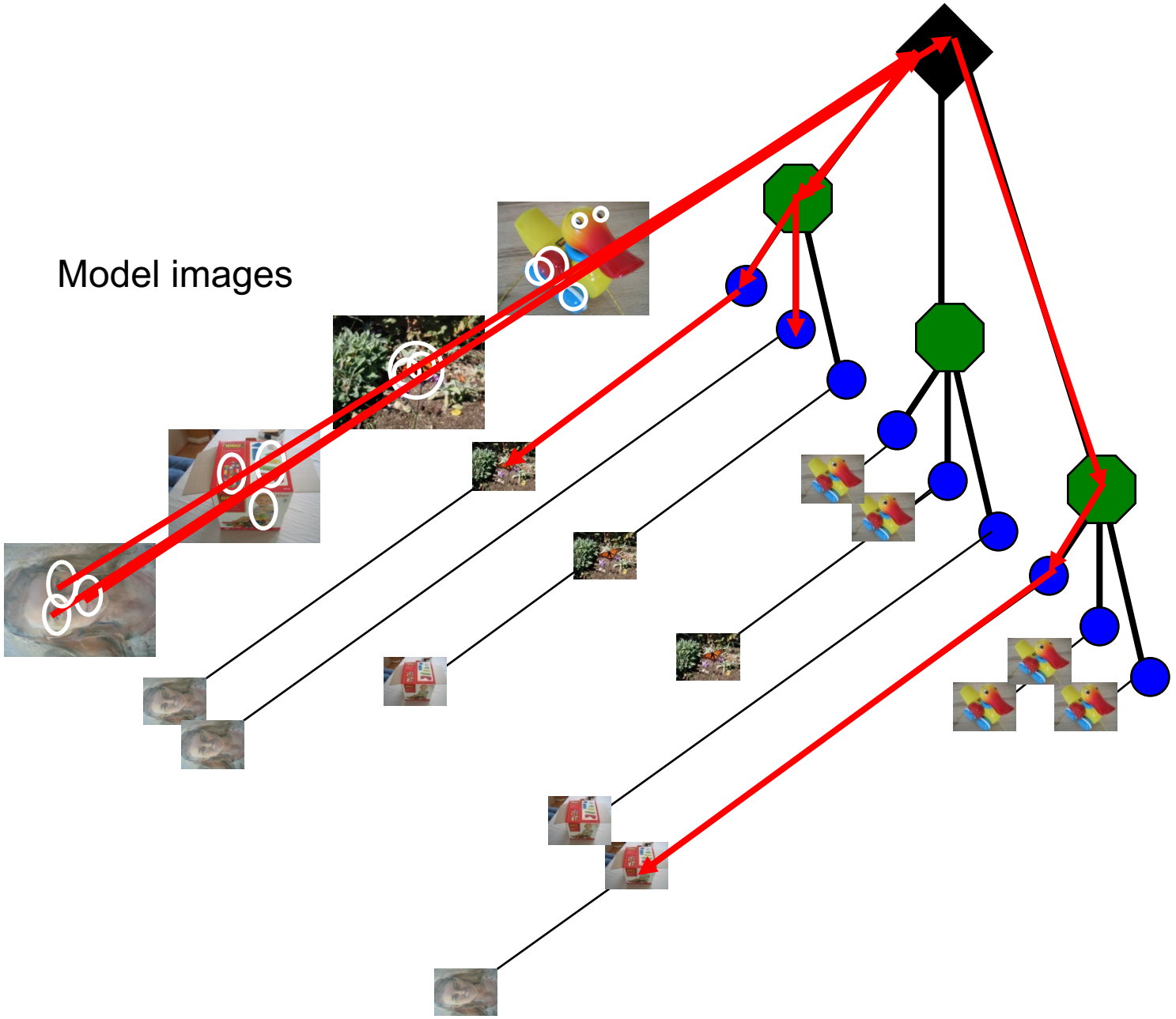
Populating the vocabulary tree/inverted index

Slide credit: D. Nister



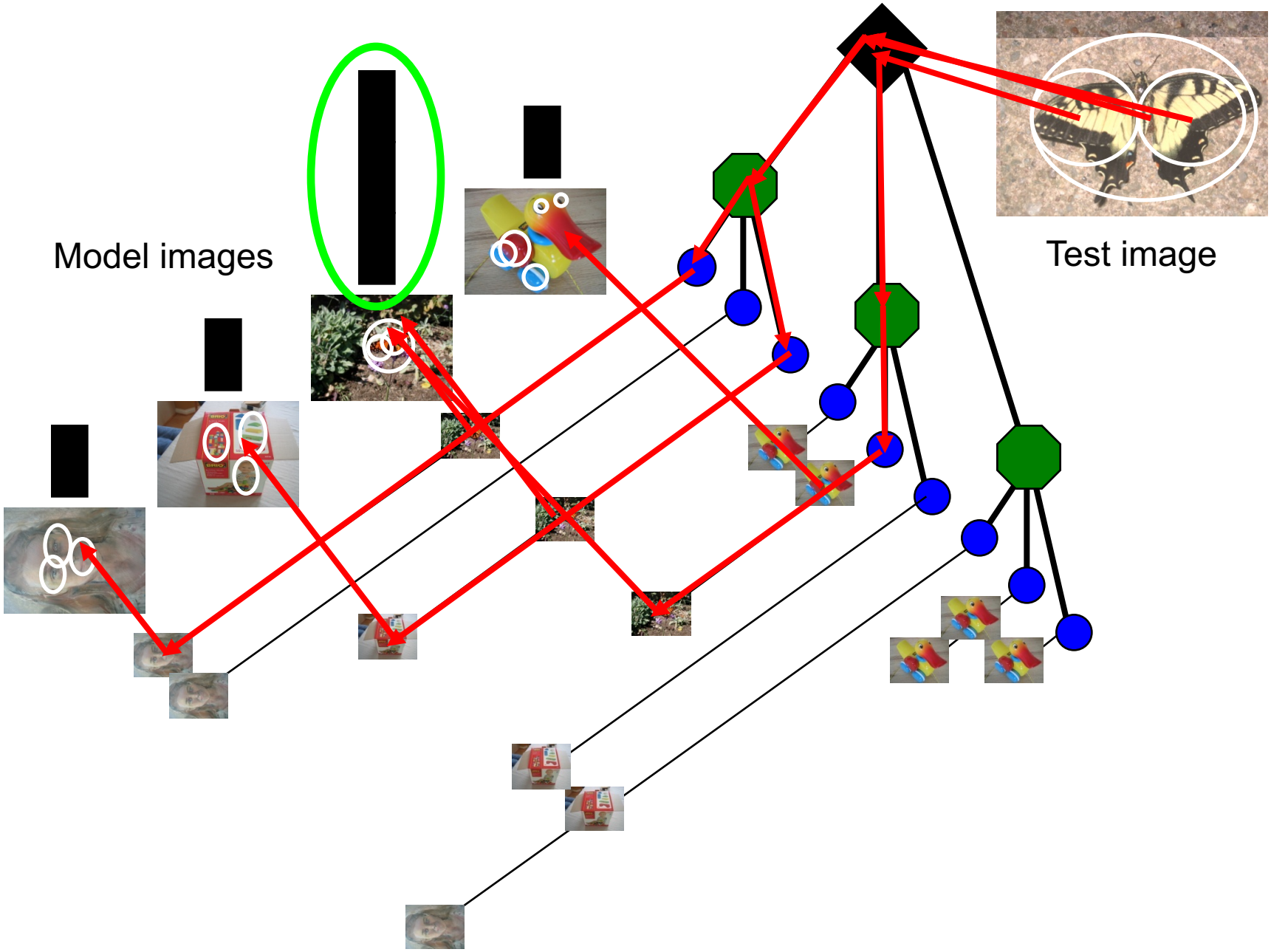
Populating the vocabulary tree/inverted index

Slide credit: D. Nister



Populating the vocabulary tree/inverted index

Slide credit: D. Nister



Looking up a test image

Slide credit: D. Nister

Many schemas for object/image recognition

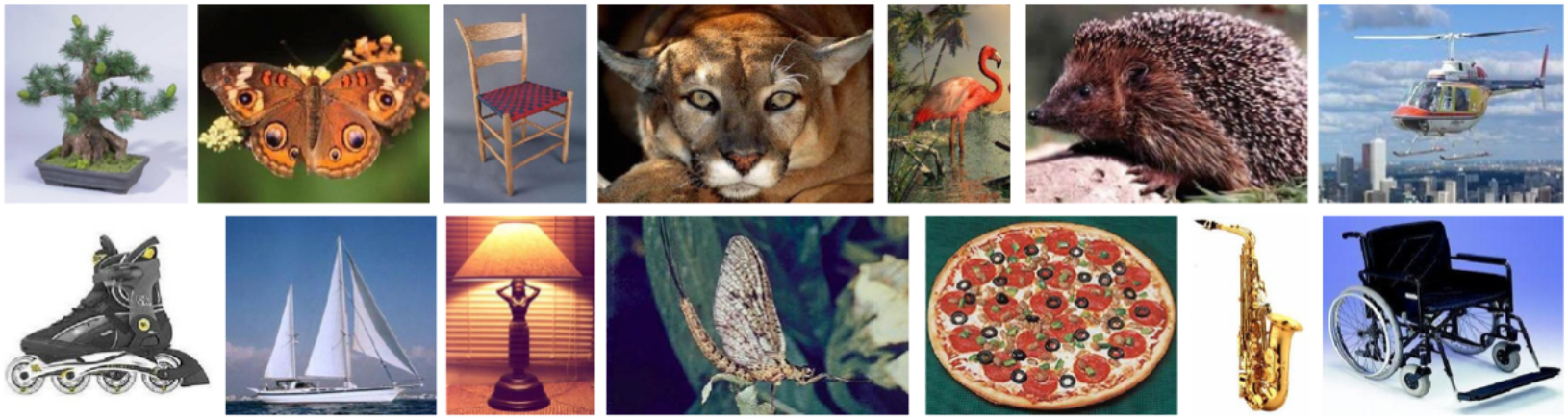
1. Manually gather training images
   - extract the visual words (and of course the vocabulary)
     - the vectors of ‘faces’ are labeled this way
2. Learn visual category model
   - could be a decision tree or a SMV
3. Collect test images
4. Evaluate classifier/detector

Examples

Multi-class classification results (100 training images per class)

	Weak features (vocabulary size: 16)		Strong features (vocabulary size: 200)	
Level	Single-level	Pyramid	Single-level	Pyramid
0 (1 × 1)	45.3 ±0.5		72.2 ±0.6	
1 (2 × 2)	53.6 ±0.3	56.2 ±0.6	77.9 ±0.6	79.0 ±0.5
2 (4 × 4)	61.7 ±0.6	64.7 ±0.7	79.4 ±0.3	**81.1** ±0.3
3 (8 × 8)	63.3 ±0.8	**66.8** ±0.6	77.2 ±0.4	80.7 ±0.3

Examples

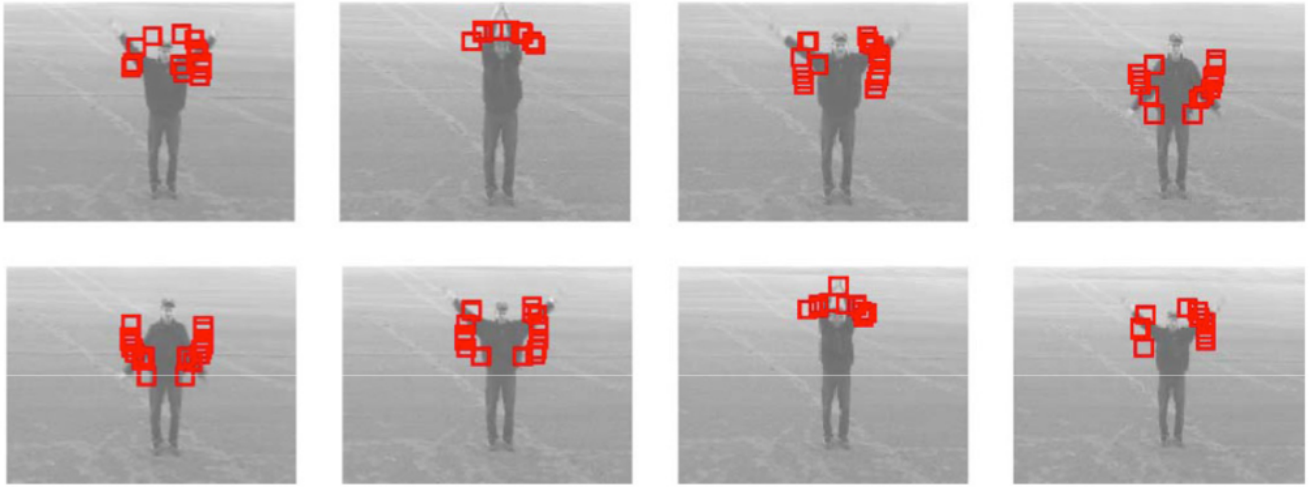


Multi-class classification results (30 training images per class)

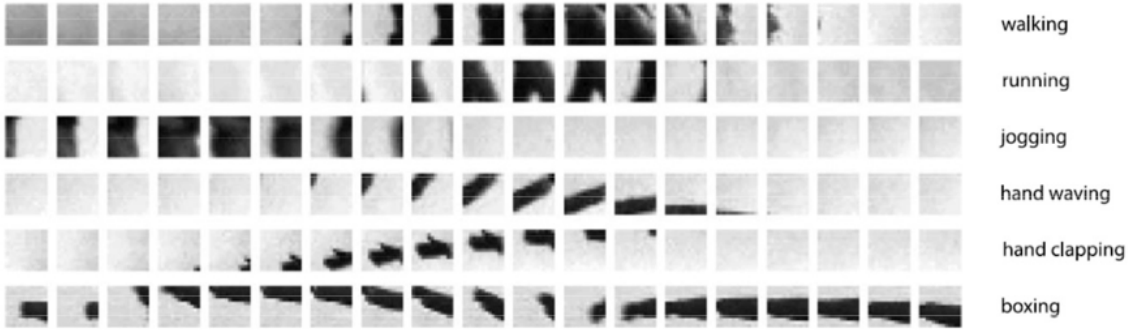
	Weak features (16)		Strong features (200)	
Level	Single-level	Pyramid	Single-level	Pyramid
0	15.5 ±0.9		41.2 ±1.2	
1	31.4 ±1.2	32.8 ±1.3	55.9 ±0.9	57.0 ±0.8
2	47.2 ±1.1	49.3 ±1.4	63.6 ±0.9	**64.6** ±0.8
3	52.2 ±0.8	**54.0** ±1.1	60.3 ±0.9	64.6 ±0.7

Examples

Space-time interest points

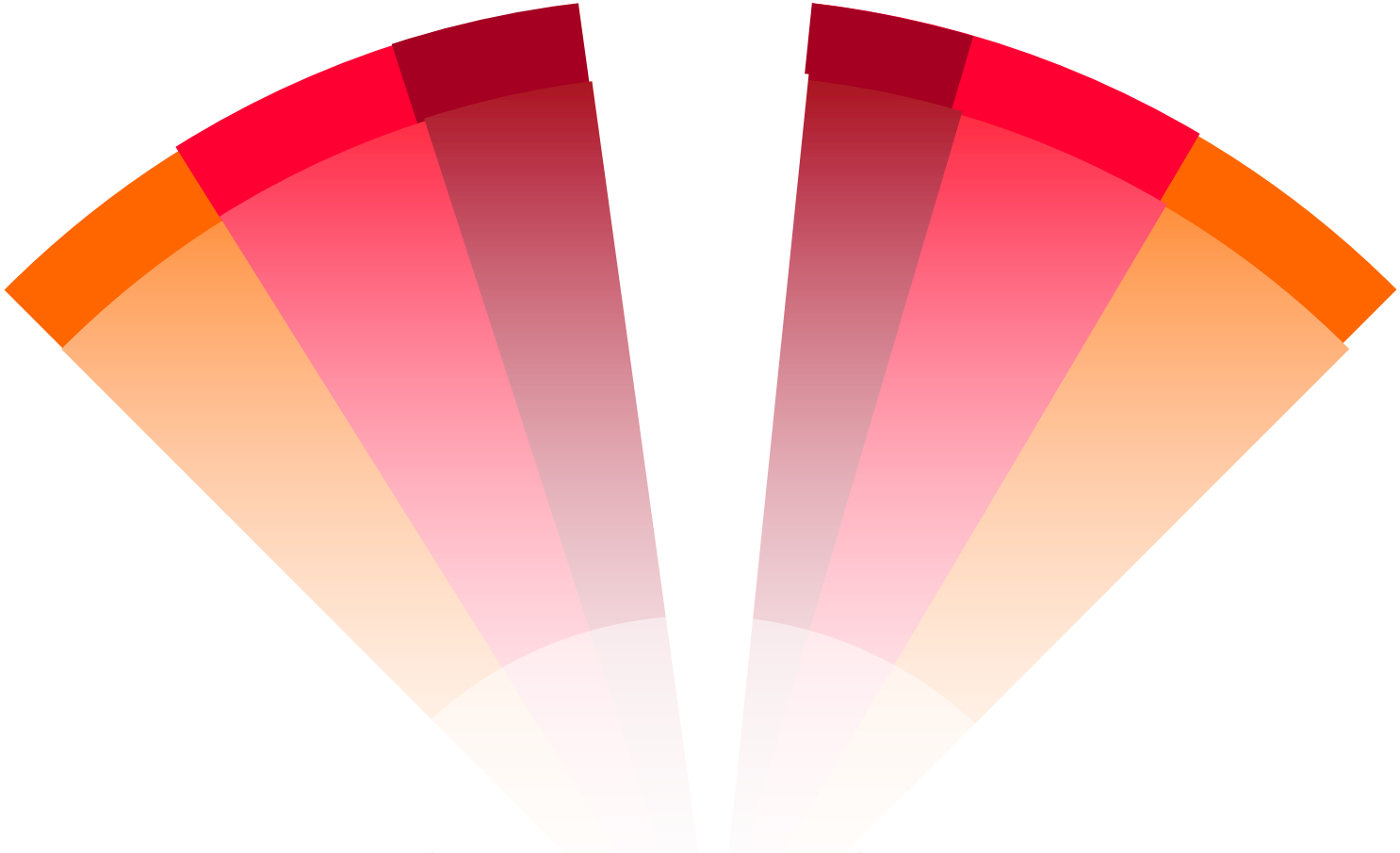


(Niebles, Wang and Fei-Fei, 2008)



Example

- Navigate and map a priori unknown environments.
- Navigate in self-similar environments:
  - Recover from gross errors.
  - BoW model to selected best frames to reconstruct a position.
  - Provide robust wide-baseline correspondences between pairs of frames (hierarchical vocabulary matching).
- Mapping trajectories in challenging dynamic outdoors environments.
- No other sensor input
- Example
  - http://hilandtom.com/tombotterill/



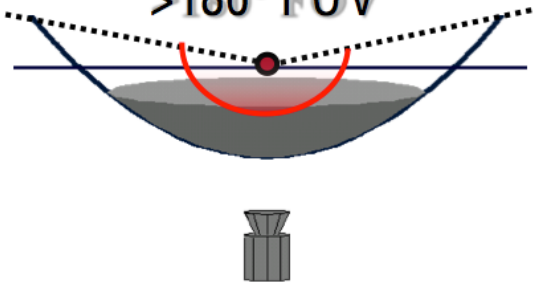
Omni-directional cameras

Omni-directional cameras

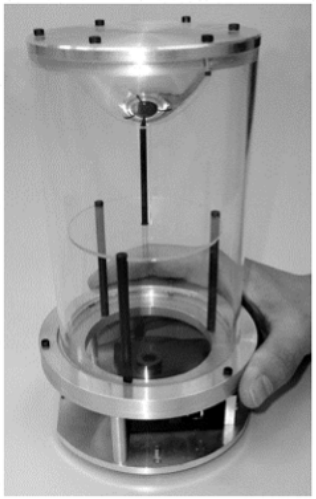
~180° FOV

wide FOV **dioptric** camera (e.g. fisheye)

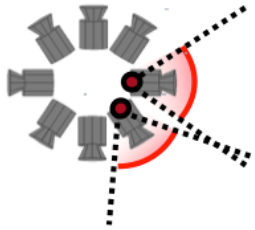
>180° FOV



catadioptric camera (camera + shaped mirror)



~360° FOV



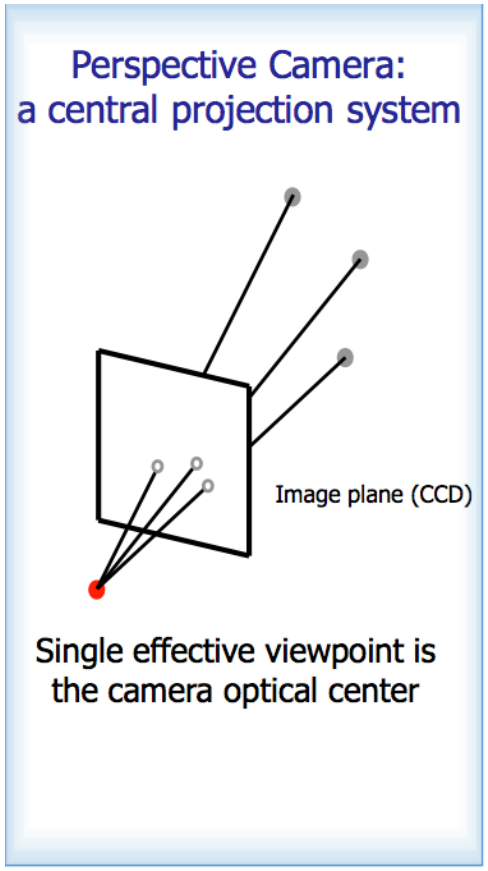
polydioptric camera (multiple cameras with overlapping FOV)

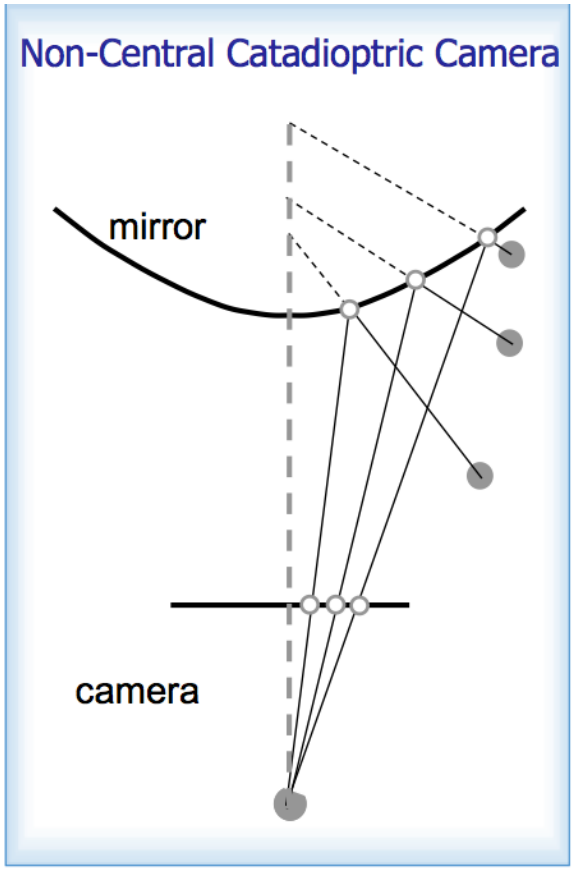
Omni-directional images

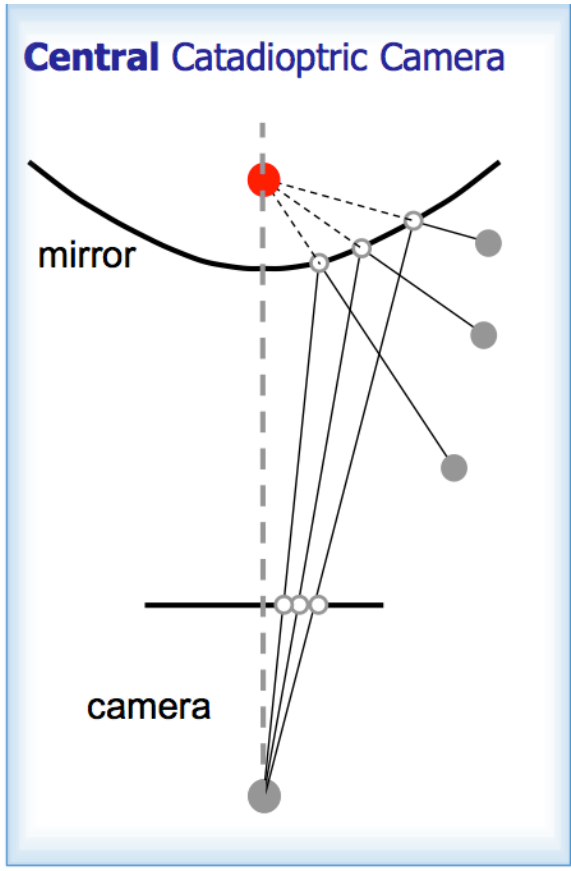
- Very popular in the robotics community
  Omni-vision offers:
  Tracking of landmarks in all directions
- Tracking of landmarks for longer periods of time better accuracy in motion estimation and
- map building
- Google Street View uses omni-directional snapshots to build photo- realistic 3D maps

Central Cameras

- A vision system is **central** when all optical rays intersect to a single point







Central Cameras

- Central catadioptric cameras can be built by appropriate choice of
  - Mirror shape, and
  - Distance of camera mirror
- Why is **a single effective point** so desirable?
  - Every pixel in the captured image, measures the irradiance of light in one particular direction
  - Allows the generation of geometrically correct perspective images

Summary

- Bag of visual words models inspired in texture and text document analysis.
- Has proven to be very robust and effective in image/object recognition:
- Despite of spatial information is lost.
  Image/object recognition have may application fields:
- Image and video retrieval.
  Human action recognition.
  Visual recognition for robotic navigation.
- Extension of the model for further reasoning:
  Latent topic models (pLSA, LDA, ...)

THANK YOU