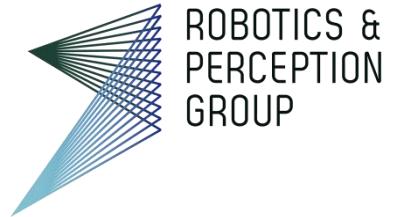




University of
Zurich^{UZH}

ETH zürich

Institute of Informatics – Institute of Neuroinformatics



ROBOTICS &
PERCEPTION
GROUP

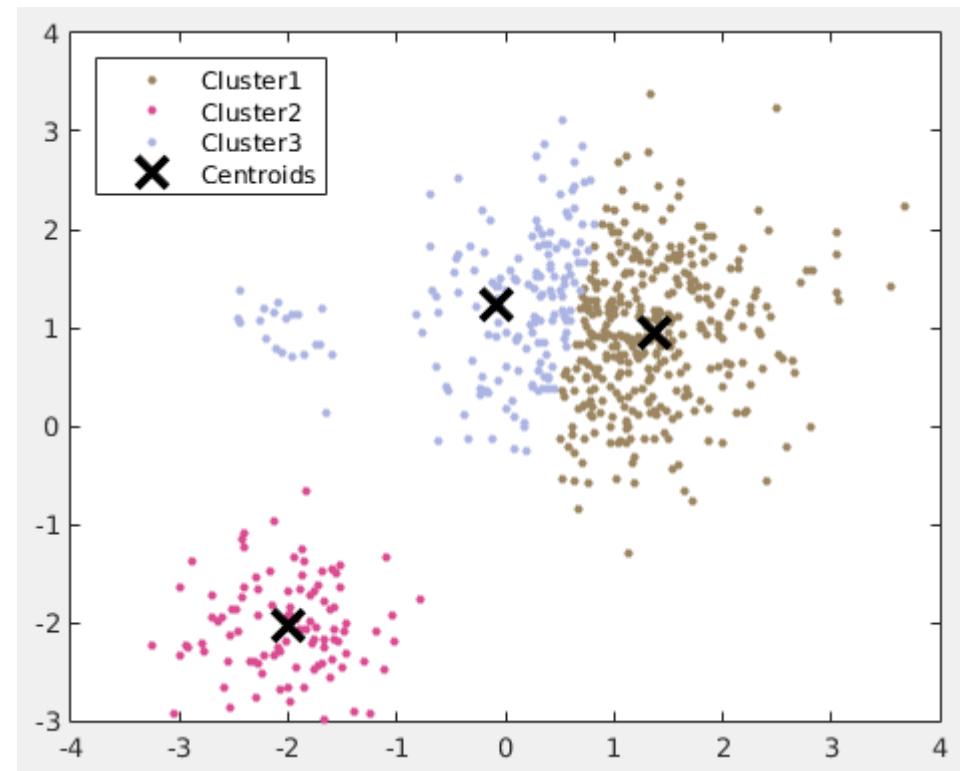
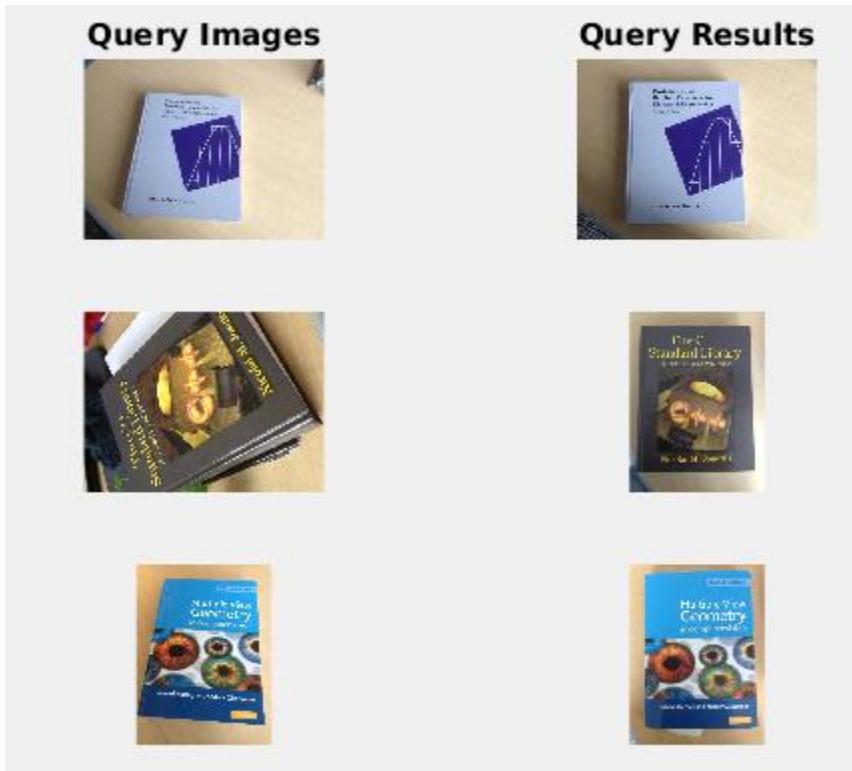
Lecture 12

Recognition

Davide Scaramuzza

Lab exercise today replaced by Deep Learning Tutorial

- Room ETH HG E 1.1 from 13:15 to 15:00
- Optional lab exercise is online: K-means clustering and Bag of Words place recognition

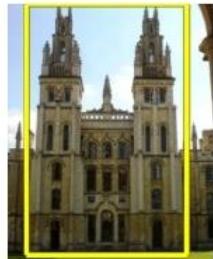


Outline

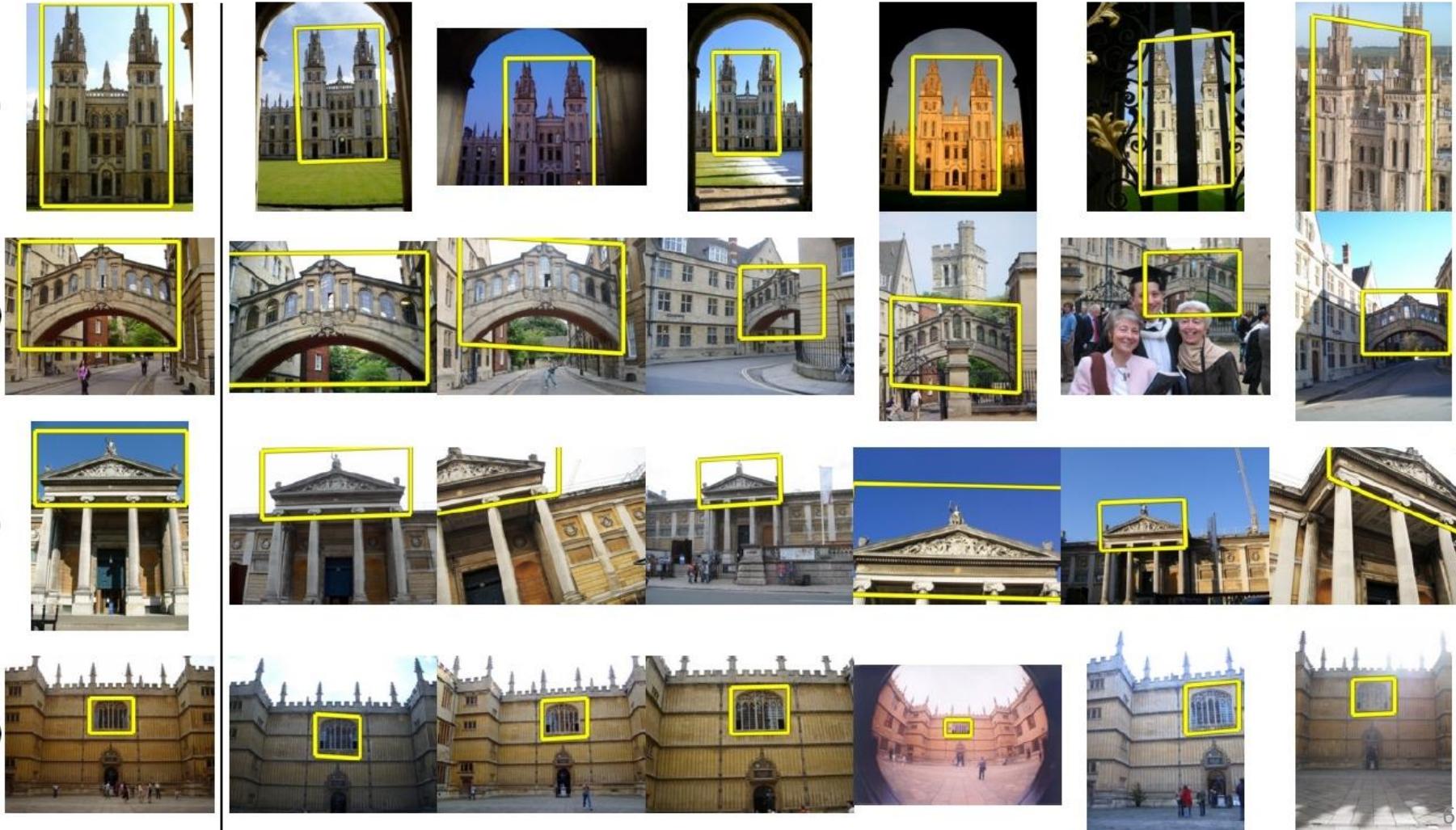
- Recognition applications and challenges
- Recognition approaches
- Classifiers
- K-means clustering
- Bag of words

Application: large-scale image retrieval

Query image



Closest results from a database of 100 million images



Application: recognition for mobile phones



- Smartphone:
 - Lincoln Microsoft Research
 - Point & Find, Nokia
 - SnapTell.com (Amazon)
 - Google Goggles

Application: Face recognition

See iPhoto, Google Photos, Facebook



Detection

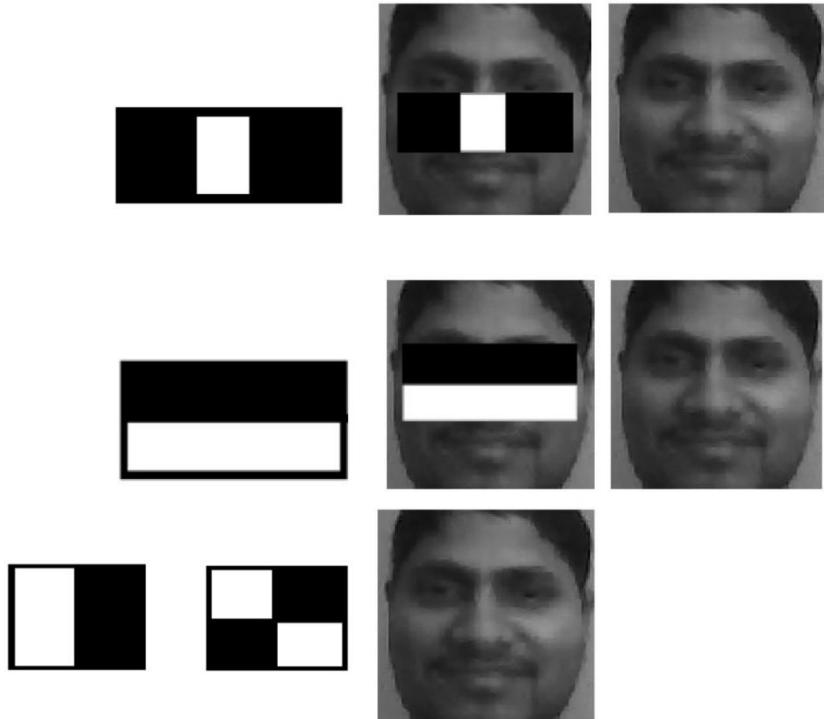


Recognition

“Sally”

Application: Face recognition

- Detection works by using four basic types of feature detectors
 - The white areas are subtracted from the black ones.
 - A special representation of the sample called the **integral image** makes feature extraction faster.



Application: Optical character recognition (OCR)

Technology to convert scanned docs to text

- If you have a scanner, it probably came with OCR software



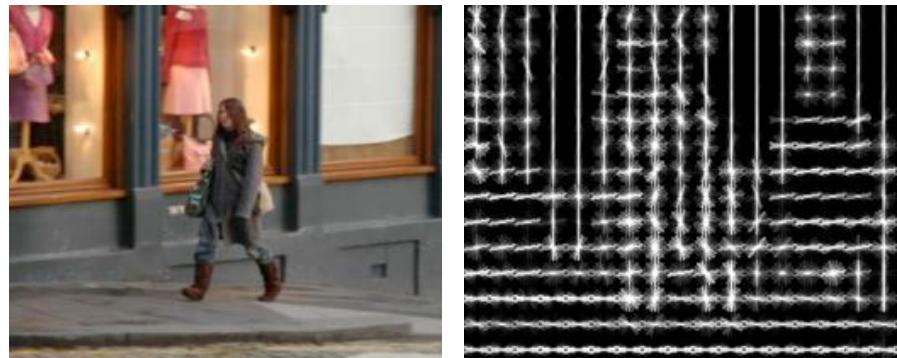
Digit recognition, AT&T labs, using CNN,
by Yann LeCun (1993)
<http://yann.lecun.com/>



License plate readers
http://en.wikipedia.org/wiki/Automatic_number_plate_recognition

Application: pedestrian recognition

- Detector: Histograms of oriented gradients (HOG)



Credit: Van Gool's lab, ETH Zurich

Challenges: object intra-class variations

- How to recognize ANY car

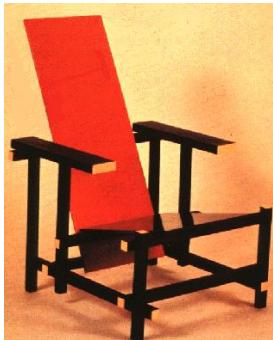


- How to recognize ANY cow



Challenges: object intra-class variations

- How to recognize ANY chair



Challenges: context and human experience



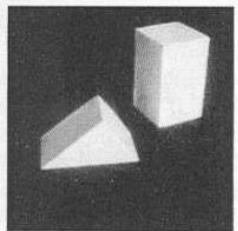
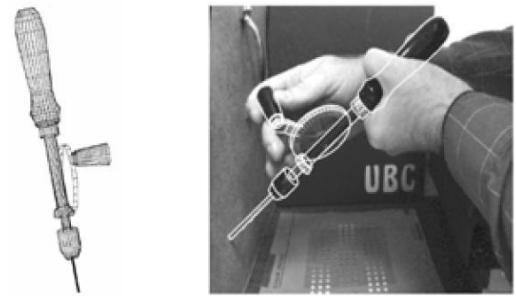
Outline

- Recognition applications and challenges
- Recognition approaches
- Classifiers
- K-means clustering
- Bag of words

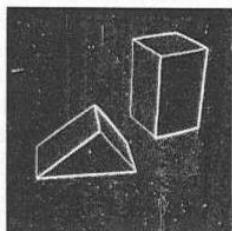
Research progress in recognition

1960-1990

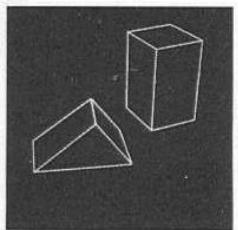
Polygonal objects



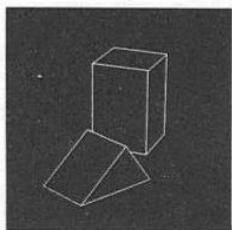
b)



c)



d)



e)

1990-2000

Faces, characters,
planar objects



7 5 9 2 6 5
1 2 2 2 2 3
0 2 3 8 0 7



2000-today

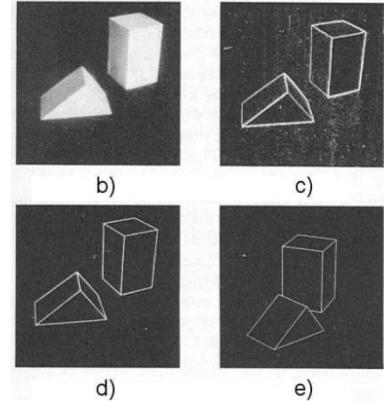
Any kind of object



Two schools of approaches

- **Model based**

- Tries to fit a model (2D or 3D) using a set of corresponding features (lines, point features)
 - Example: SIFT matching and RANSAC for model validation



- **Appearance based**

- The model is defined by a set of images representing the object
 - Example: template matching can be thought as a simple object recognition algorithm (the template is the object to recognize)



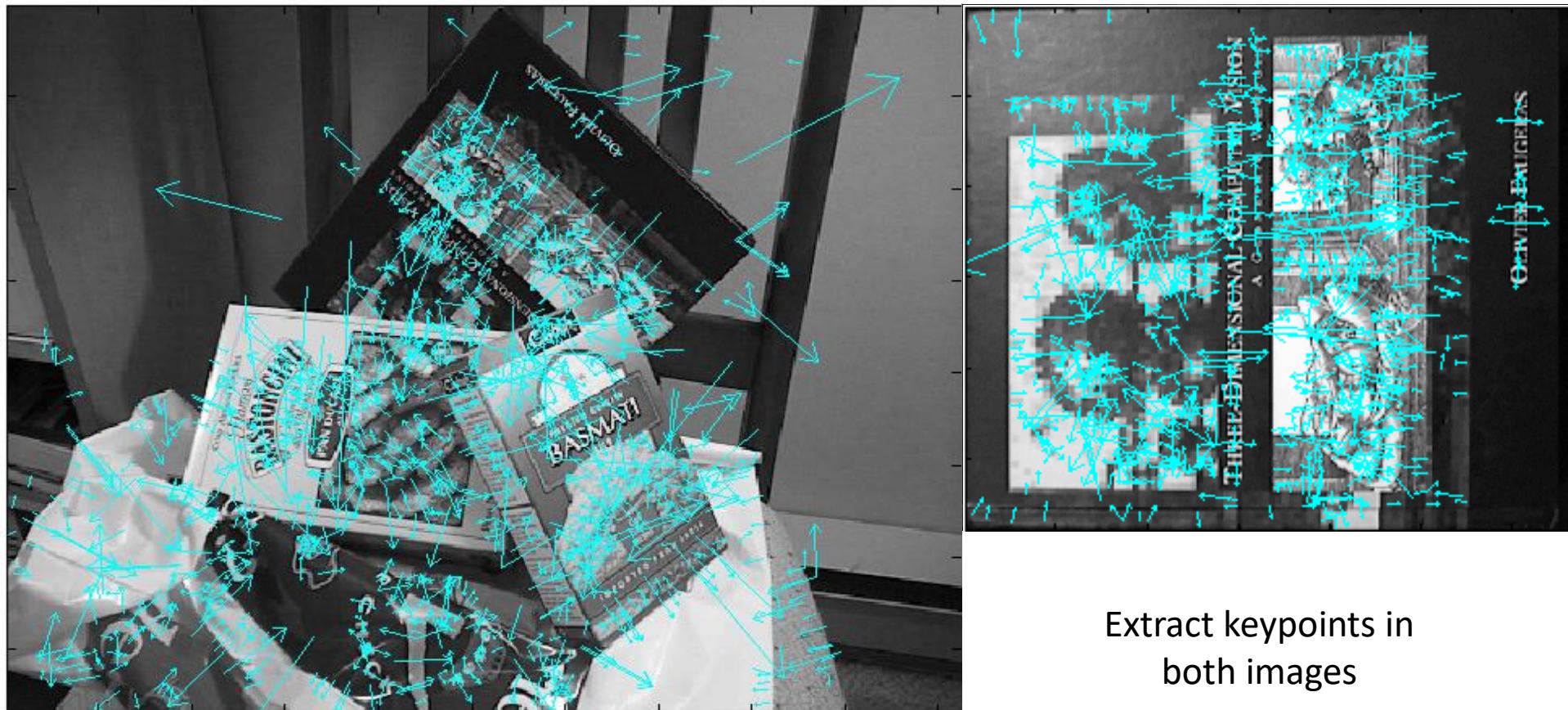
Example of 2D model-based approach

Q: Is this Book present in the Scene?



Example of 2D model-based approach

Q: Is this Book present in the Scene?



Extract keypoints in
both images

Example of 2D model-based approach

Q: Is this Book present in the Scene?



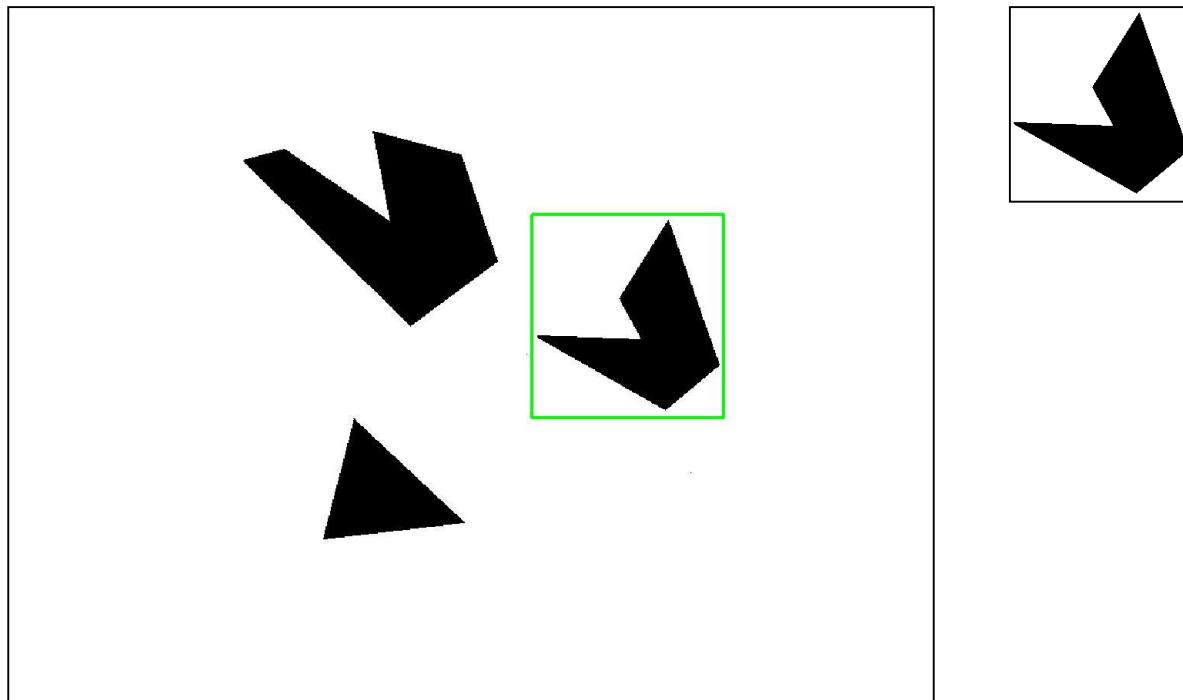
Look for corresponding
matches

Most of the Book's keypoints are present in the Scene

⇒ A: The Book is present in the Scene

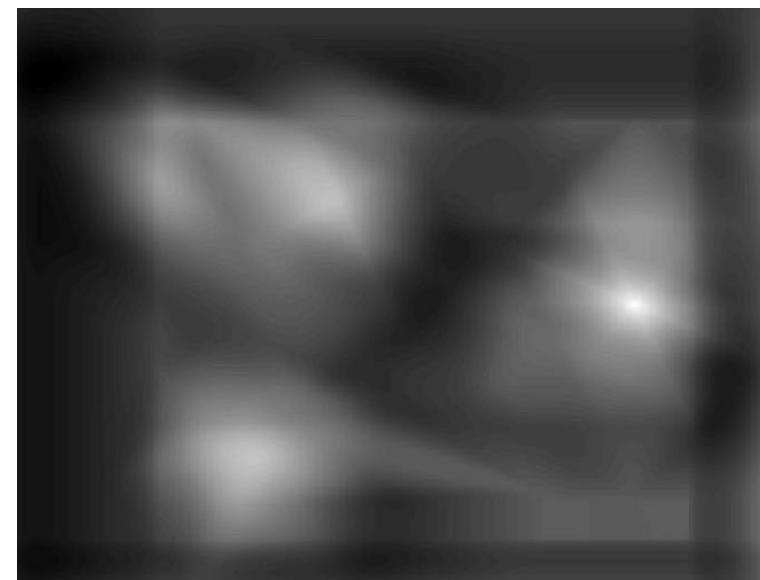
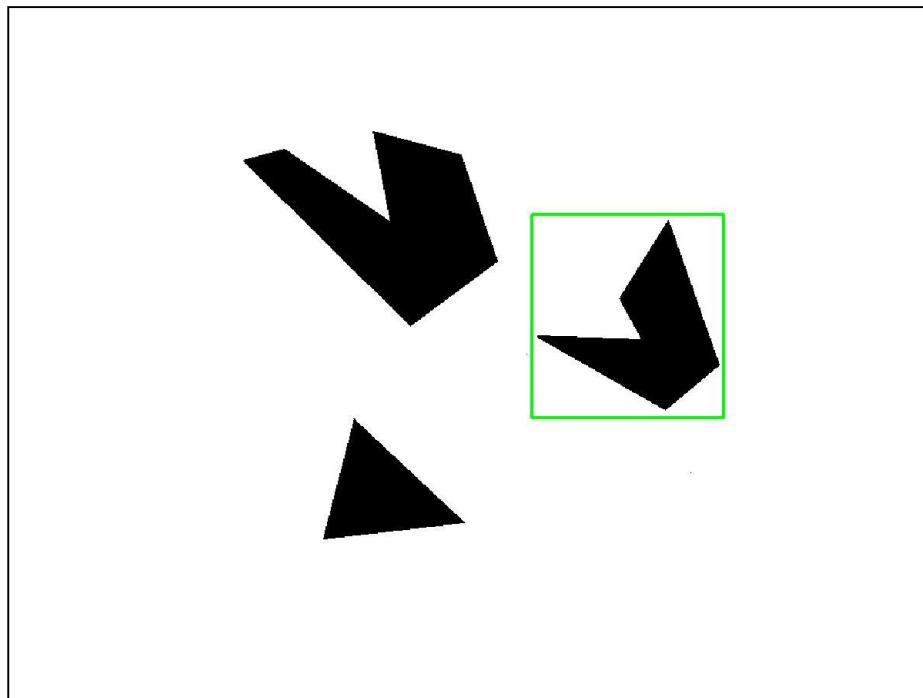
Example of appearance-based approach: Simple 2D template matching

- The model of the object is simply an image
- A simple example: Template matching
 - Shift the template over the image and compare (e.g. NCC or SSD)
 - Problem: works only if template and object are identical



Example of appearance-based approach: Simple 2D template matching

- The model of the object is simply an image
- A simple example: Template matching
 - Shift the template over the image and compare (e.g. NCC or SSD)
 - Problem: works only if template and object are identical



Outline

- Recognition applications and challenges
 - Recognition approaches
 - Classifiers
-
- K-means clustering
 - Bag of words

What is the goal of object recognition?

Goal: **classify!**

- **Binary classifier**
 - say yes/no as to whether an object is present in an image
- **Multi-class classifier**
 - categorize an object: determine what class it belongs to (e.g., car, apple, etc.)

How to display the result to the user

- Bounding box on object
- Full segmentation



Is this or is this not a car?



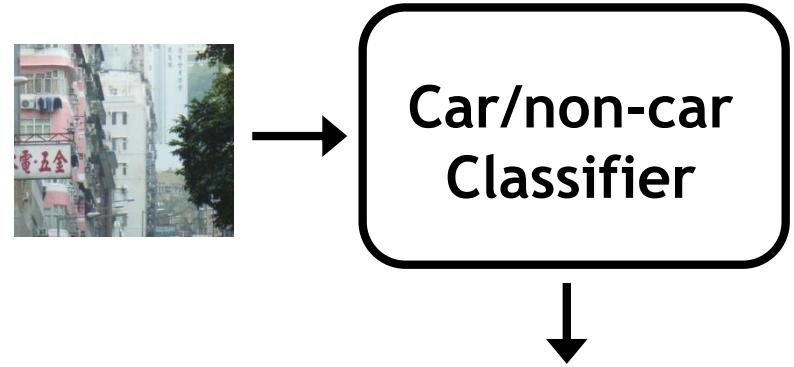
Bounding box on object



Full segmentation

Detection via classification: Main idea

Basic component: a **binary** classifier

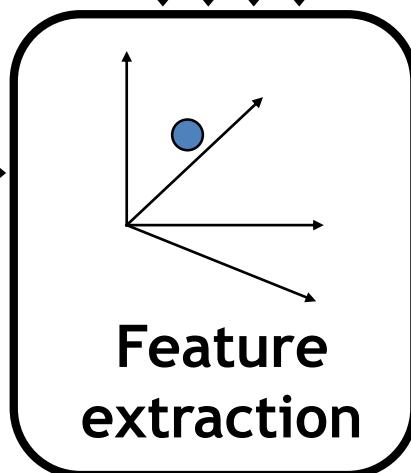
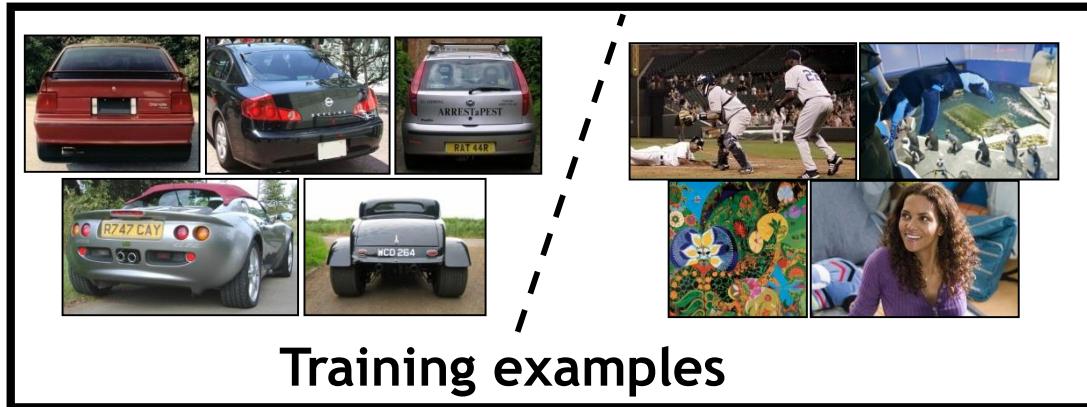


No, ~~yes~~ not car.

Detection via classification: Main idea

More in detail, we need to:

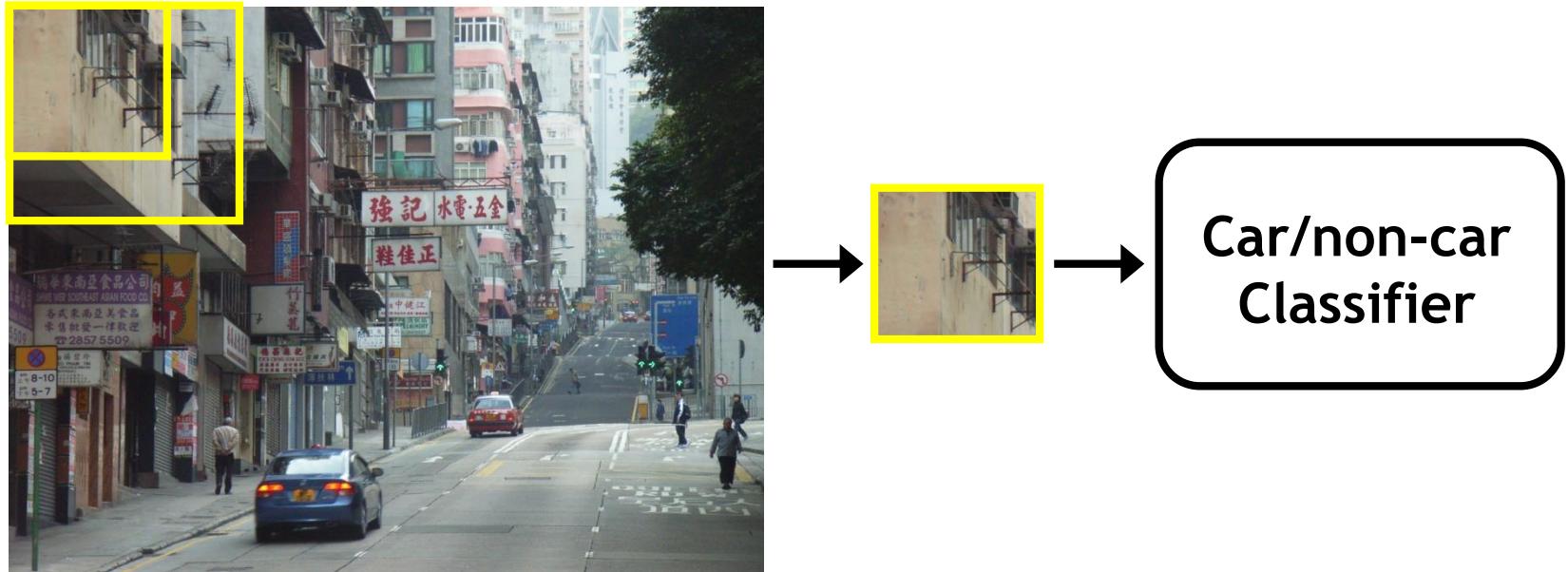
1. Obtain training data
2. Define features
3. Define classifier



Car/non-car
Classifier

Detection via classification: Main idea

- Consider all subwindows in an image
 - Sample at multiple scales and positions
- Make a decision per window:
 - “Does this contain object X or not?”



Generalization: the machine learning approach



Generalization: the machine learning approach

- Apply a prediction function to a feature representation of the image to get the desired output:

$$f(\text{apple}) = \text{"apple"}$$
$$f(\text{tomato}) = \text{"tomato"}$$
$$f(\text{cow}) = \text{"cow"}$$

The machine learning framework

$$y = f(x)$$

The diagram illustrates the components of the machine learning framework. It shows the equation $y = f(x)$ in blue. Three red arrows point upwards from labels below to the corresponding parts of the equation: one arrow points to the variable y from the label "output", another points to the function symbol f from the label "prediction function", and a third points to the variable x from the label "Input: Image features".

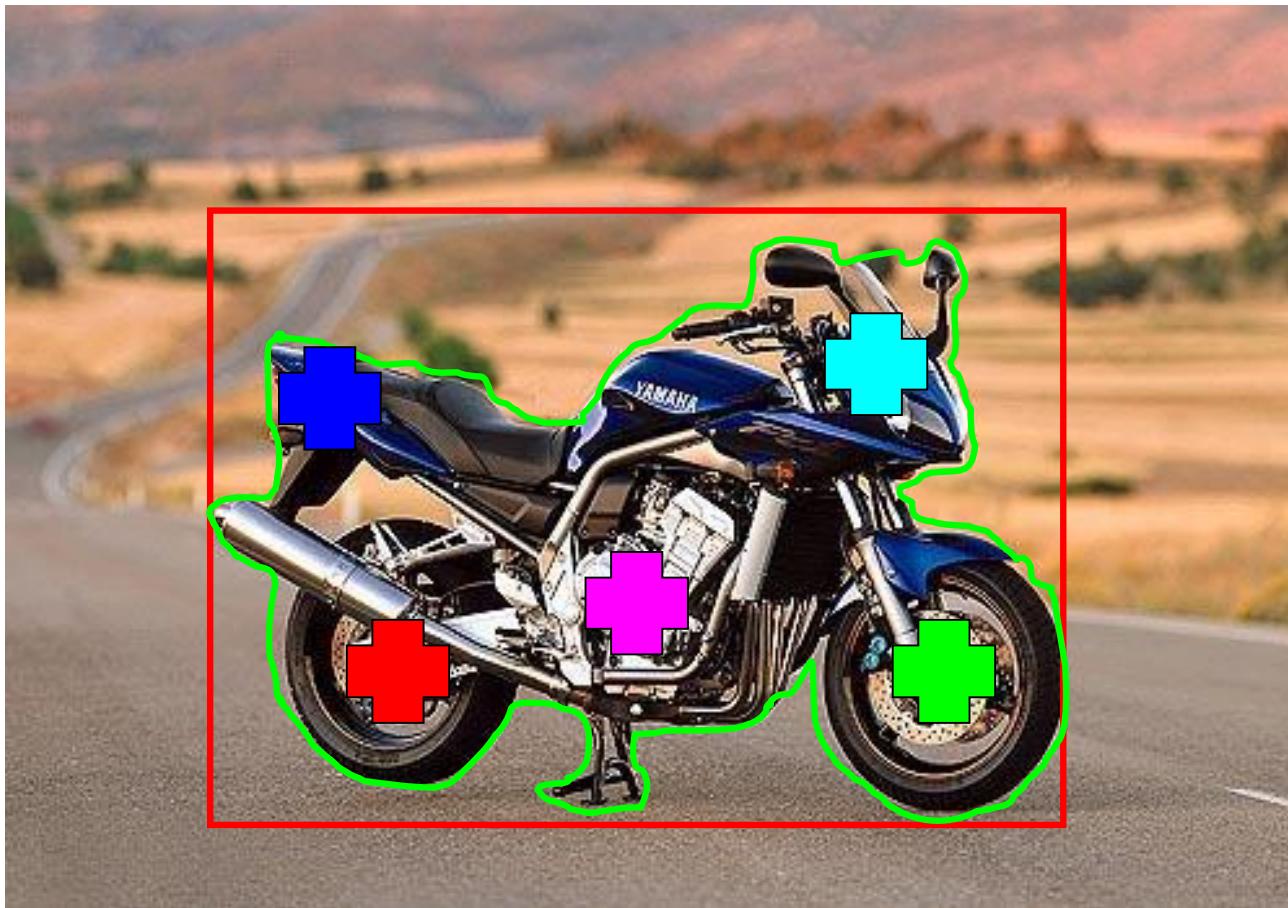
output prediction function Input: Image features

- **Training:** given a *training set* of labeled examples $\{(x_1, y_1), \dots, (x_N, y_N)\}$, estimate the prediction function f by minimizing the prediction error ($loss = y - f(x)$) on the training set
- **Testing:** apply f to a *never-before-seen test example* x and output the predicted value $y = f(x)$

Recognition task and supervision

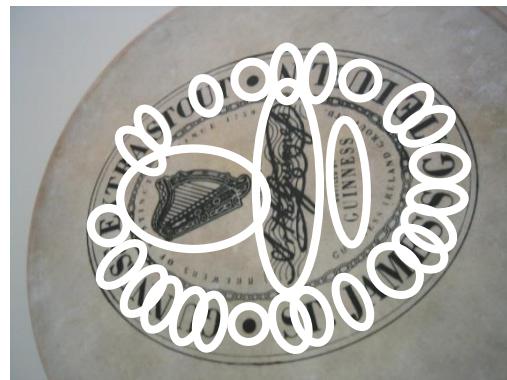
- Images in the training set must be *labeled* with the “correct answer” that the model is expected to produce

Contains a motorbike

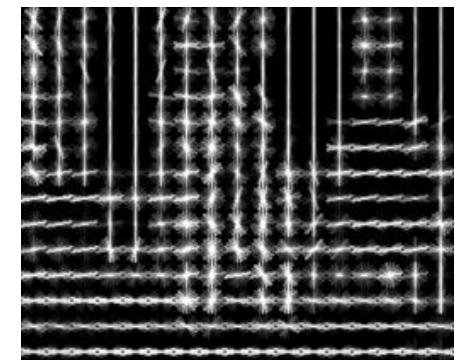


Examples of possible features

- Blob features



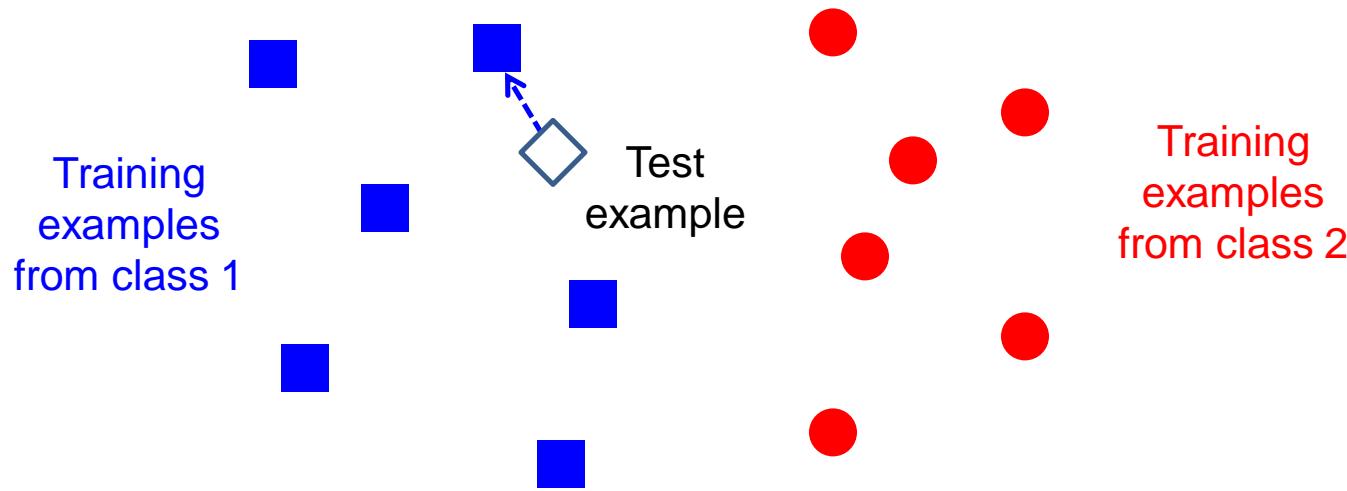
- Image Histograms
- Histograms of oriented gradients (HOG)



Classifiers: Nearest neighbor

Features are represented in the descriptor space

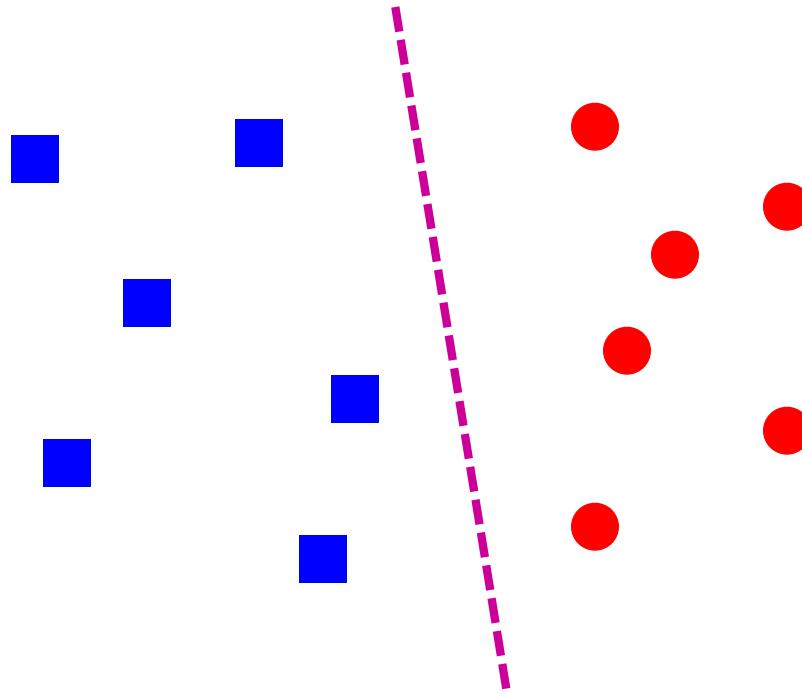
(Ex. What is the dimensionality of the descriptor space for SIFT features?)



$$f(\mathbf{x}) = \text{label of the training example nearest to } \mathbf{x}$$

- **No training required!**
- All we need is a distance function for our inputs
- Problem: need to compute distances to all training examples! (what if you have 1 million training images and 1 thousand features per image?)

Classifiers: Linear

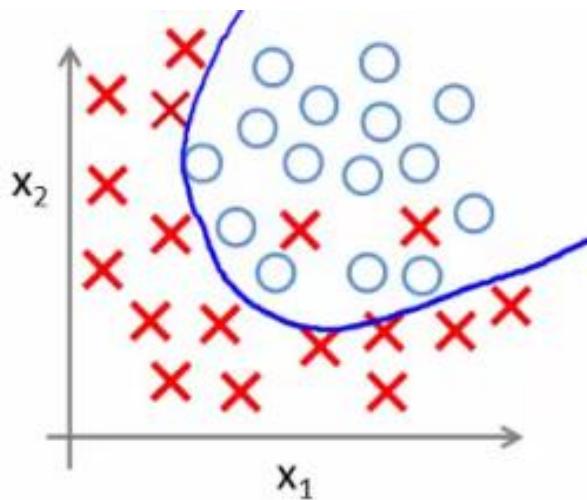


- Find a *linear function* to separate the classes:

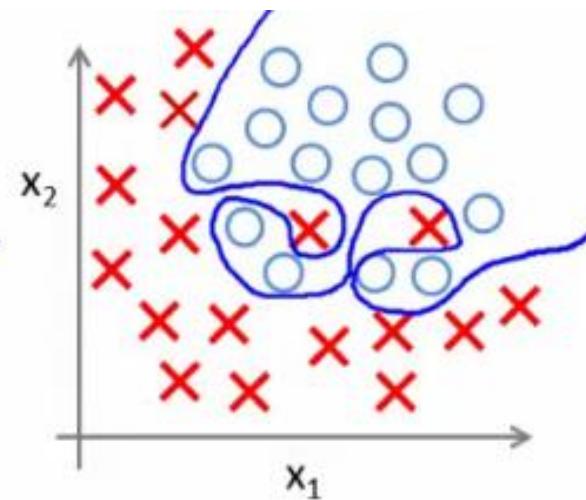
$$f(\mathbf{x}) = \text{sgn}(\mathbf{w} \cdot \mathbf{x} + b)$$

Classifiers: non-linear

Good classifier



Bad classifier (over fitting)

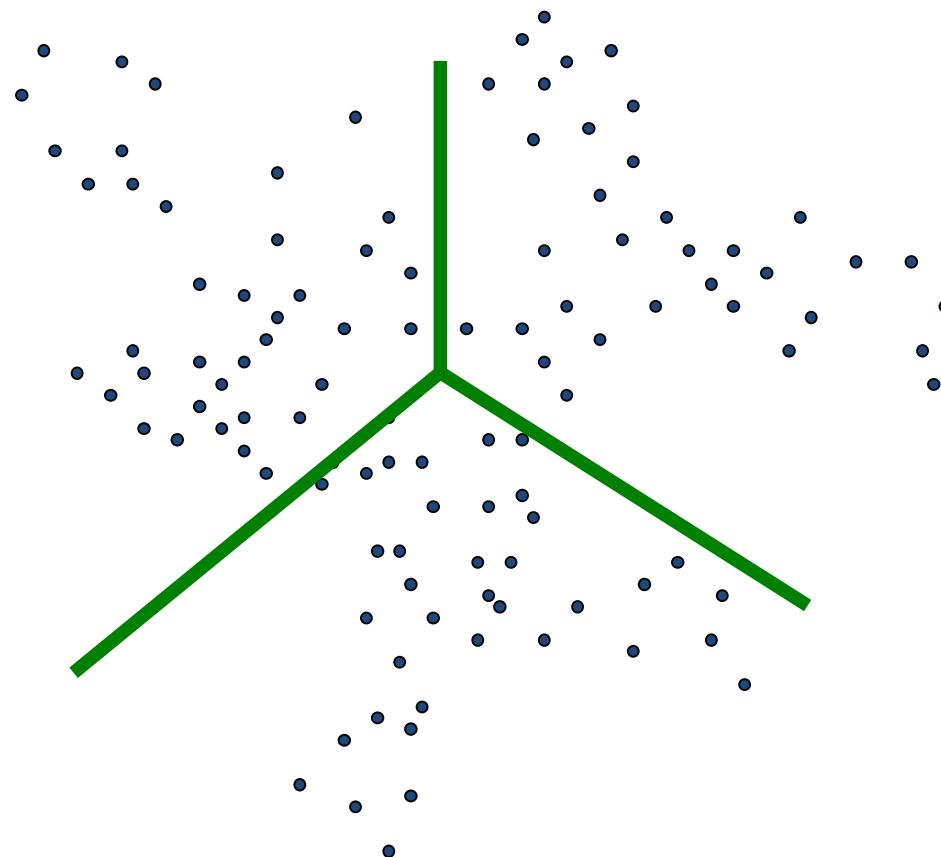


Outline

- Recognition applications and challenges
- Recognition approaches
- Classifiers
- K-means clustering
- Bag of words

How do we define a classifier?

- We first need to **cluster** the training data
- Then, we need a distance function to determine to which cluster the query image belongs to



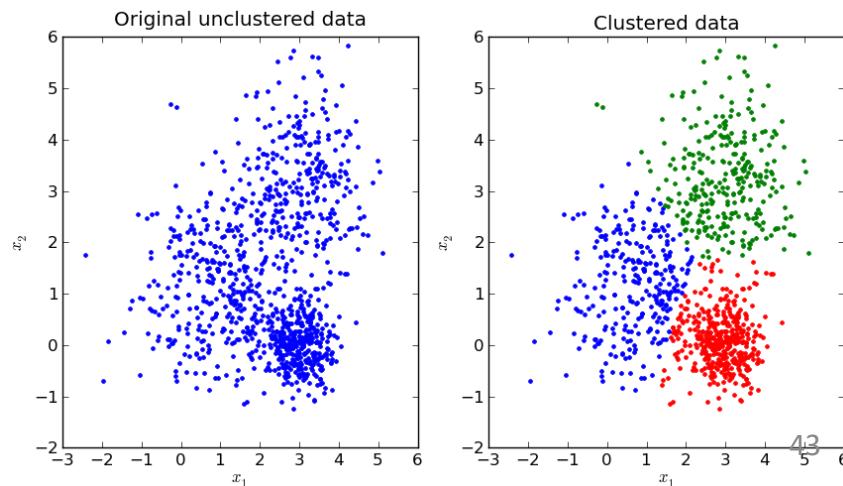
K-means clustering

- *k-means clustering* is an algorithm to partition n observations into k clusters in which each observation x belongs to the cluster S_i with center \mathbf{m}_i
- It minimizes the sum of squared Euclidean distances between points x and their nearest cluster centers \mathbf{m}_i

$$D(X, M) = \sum_{i=1}^k \sum_{x \in S_i} (x - m_i)^2$$

Algorithm:

- Randomly initialize k cluster centers
- Iterate until convergence:
 - Assign each data point x_j to the nearest center \mathbf{m}_i
 - Recompute each cluster center as the mean of all points assigned to it



K-means demo



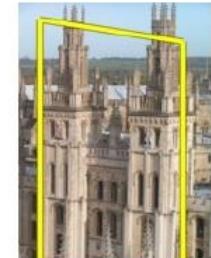
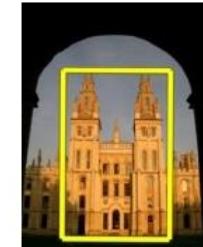
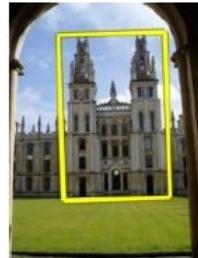
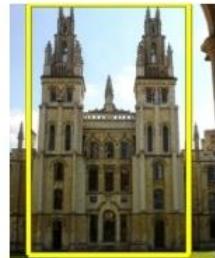
Source: <http://shabal.in/visuals/kmeans/1.html>

Outline

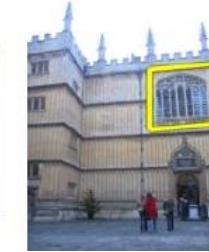
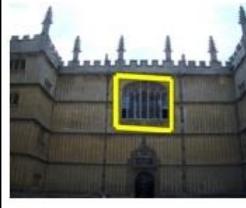
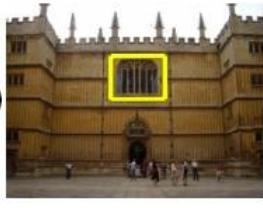
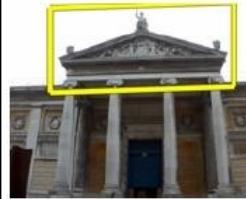
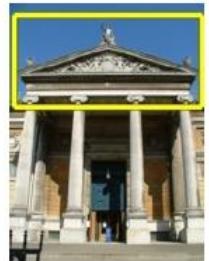
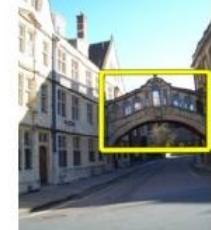
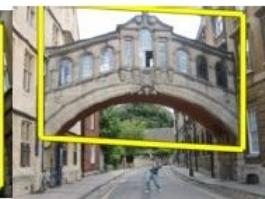
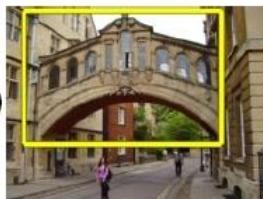
- Recognition applications and challenges
- Recognition approaches
- Classifiers
- K-means clustering
- Bag of words

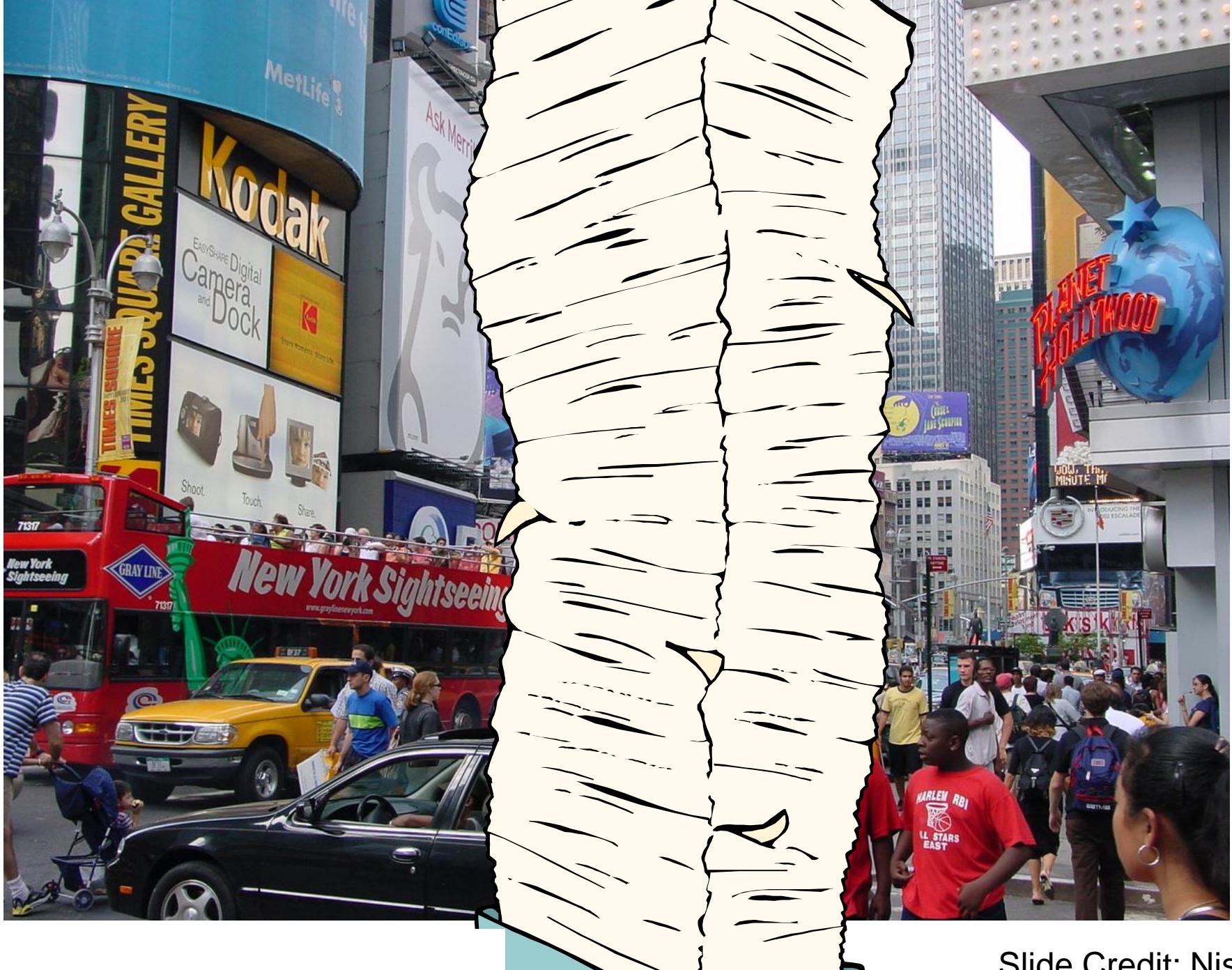
Application: large-scale image retrieval

Query image



Results on a database of 100 million images





Slide Credit: Nister



Slide Credit: Nister



(C) L.W. Wildervanck

Slide Credit: Nister

Fast visual search

- Query of 1 image in a database of 100 million images in 6 seconds



“Video Google”, Sivic and Zisserman, ICCV 2003

“Scalable Recognition with a Vocabulary Tree”, Nister and Stewenius, CVPR 2006.

Bag of Words

- Extension to scene/place recognition:
 - Is this image in my database?
 - Robot: Have I been to this place before?

A screenshot of a Google Images search results page. The search query "matterhorn mountain" is entered in the search bar, with a "JPG" filter applied. The results are categorized under "Images". Below the search bar, it says "About 624 results (1.07 seconds)". The first result is a thumbnail of the Matterhorn peak, with its image size listed as 1501 x 999. A link below the thumbnail reads "Find other sizes of this image: All sizes - Medium - Large". The "Best guess for this image" is "matterhorn mountain". Below this, there are two search results: "Matterhorn - Wikipedia, the free encyclopedia" and "The Matterhorn Mountain in Switzerland, Zermatt". Underneath these results is a section titled "Visually similar images" which displays a grid of eight smaller images of the Matterhorn peak.

Visual Place Recognition

- **Goal:** find the most similar images of a **query** image in a database of N **images**
- **Complexity:** $\frac{N^2 \cdot M^2}{2}$ feature comparisons (assumes each image has M features)
 - Each image must be compared with all other images!
 - N is the number of all images collected by a robot
 - Example: assume your Roomba robot takes 1 image every meter to cover a 100 m² house; assume 100 SIFT features per image → $M = 100$, $N = 100$ → $N^2 M^2 / 2 = \sim 50 \text{ Million}$ feature comparisons!

Solution: Use an inverted file index!
Complexity reduces to $N \cdot M$

[“Video Google”, Sivic & Zisserman, ICCV’03]
[“Scalable Recognition with a Vocabulary Tree”, Nister & Stewenius, CVPR’06]
See also FABMAP and Galvez-Lopez’12’s (DBoW2)]

Indexing local features: inverted file text

- For text documents, an efficient way to find all *pages* in which a *word* occurs is to use an index
- We want to find all *images* in which a *feature* occurs
- How many distinct SIFT or BRISK features exist?
 - SIFT → Infinite
 - BRISK-128 → $2^{128} = 3.4 \cdot 10^{38}$
- Since the number of image features may be *infinite*, before we build our visual vocabulary we need to map our features to “*visual words*”
- Using analogies from text retrieval, we should:
 - Define a “Visual Word”
 - Define a “vocabulary” of Visual Words
 - This approach is known as “Bag of Words” (BOW)

Index	
"Along I-75," From Detroit to Florida; <i>Inside back cover</i>	Butterfly Center, McGuire; 134
"Drive I-95," From Boston to Florida; <i>Inside back cover</i>	CAA (see AAA)
1929 Spanish Trail Roadway;	CCC, The; 111,113,115,135,142
101-102,104	Ca d'Zan; 147
511 Traffic Information; 83	Caloosahatchee River; 152
A1A (Barrier Isl) - I-95 Access; 86	Name; 150
AAA (and CAA); 83	Canaveral Natnl Seashore; 173
AAA National Office; 88	Cannon Creek Airpark; 130
Abbreviations,	Canopy Road; 106,169
Colored 25 mile Maps; cover	Cape Canaveral; 174
Exit Services; 196	Castillo San Marcos; 169
Travelogue; 85	Cave Diving; 131
Africa; 177	Cayo Costa, Name; 150
Agricultural Inspection Stns; 126	Celebration; 93
Ah-Tah-Thi-Ki Museum; 160	Charlotte County; 149
Air Conditioning, First; 112	Charlotte Harbor; 150
Alabama; 124	Chautauqua; 116
Alachua; 132	Chipley; 114
County; 131	Name; 115
Alafia River; 143	Choctawhatchee, Name; 115
Alapaha, Name; 126	Circus Museum, Ringling; 147
Alfred B Maclay Gardens; 106	Citrus; 88,97,130,136,140,180
Alligator Alley; 154-155	CityPlace, W Palm Beach; 180
Alligator Farm, St Augustine; 169	City Maps,
Alligator Hole (definition); 157	Ft Lauderdale Expwy; 194-195
Alligator, Buddy; 155	Jacksonville; 163
Alligators; 100,135,138,147,156	Kissimmee Expwy; 192-193
Anastasia Island; 170	Miami Expressways; 194-195
Anhaila; 108-109,146	Orlando Expressways; 192-193
Apalachicola River; 112	Pensacola; 26
Appleton Mus of Art; 136	Tallahassee; 191
Aquifer; 102	Tampa-St. Petersburg; 63
Arabian Nights; 94	St. Augustine; 191
Art Museum, Ringling; 147	Civil War; 100,108,127,138,141
Aruba Beach Cafe; 183	Cleanwater Marine Aquarium; 187
Aucilla River Project; 106	Collier County; 154
Babcock-Web WMA; 151	Collier, Barron; 152
Bahia Mar Marina; 184	Colonial Spanish Quarters; 168
Baker County; 99	Columbia County; 101,128
Barefoot Mallmen; 182	Coquina Building Material; 165
Barge Canal; 137	Corkscrew Swamp, Name; 154
Bee Line Expy; 80	Cowboys; 95
Belz Outlet Mall; 89	Crab Trap II; 144
Bernard Castro; 136	Cracker, Florida; 88,95,132
Big "I"; 165	Crosstown Expy; 11,35,98,143
Big Cypress; 155,158	Cuban Bread; 184
Big Foot Monster; 105	Dade Battlefield; 140
Billie Swamp Safari; 160	Dade, Maj. Francis; 139-140,161
Blackwater River SP; 117	Dania Beach Hurricane; 184
Blue Angels	Daniel Boone, Florida Walk; 117
A4-C Skyhawk; 117	Daytona Beach; 172-173
Atrium; 121	De Land; 87
Blue Springs SP; 87	De Soto, Hernando,
Blue Star Memorial Highway; 125	Anhaila; 108-109,146
Boca Ciega; 189	County; 149
Boca Grande; 150	Explorer; 146
	Landing; 146
	Napitaca; 103

Building the Visual Vocabulary

Image Collection

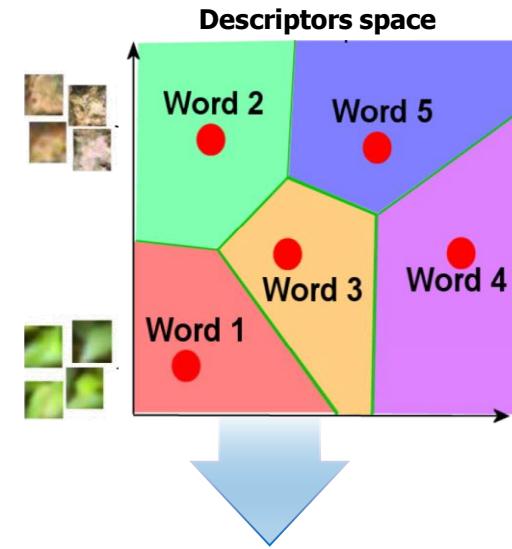


Extract Features

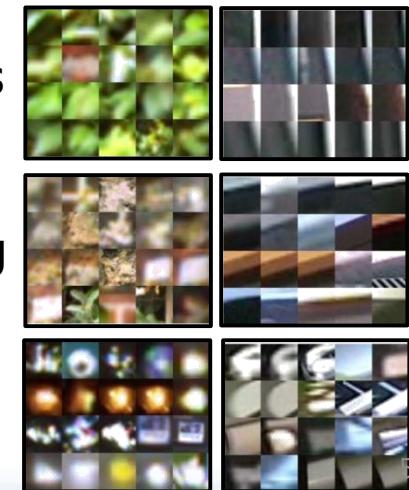


What is a visual word?
A visual word is the
centroid of a cluster!

Cluster Descriptors

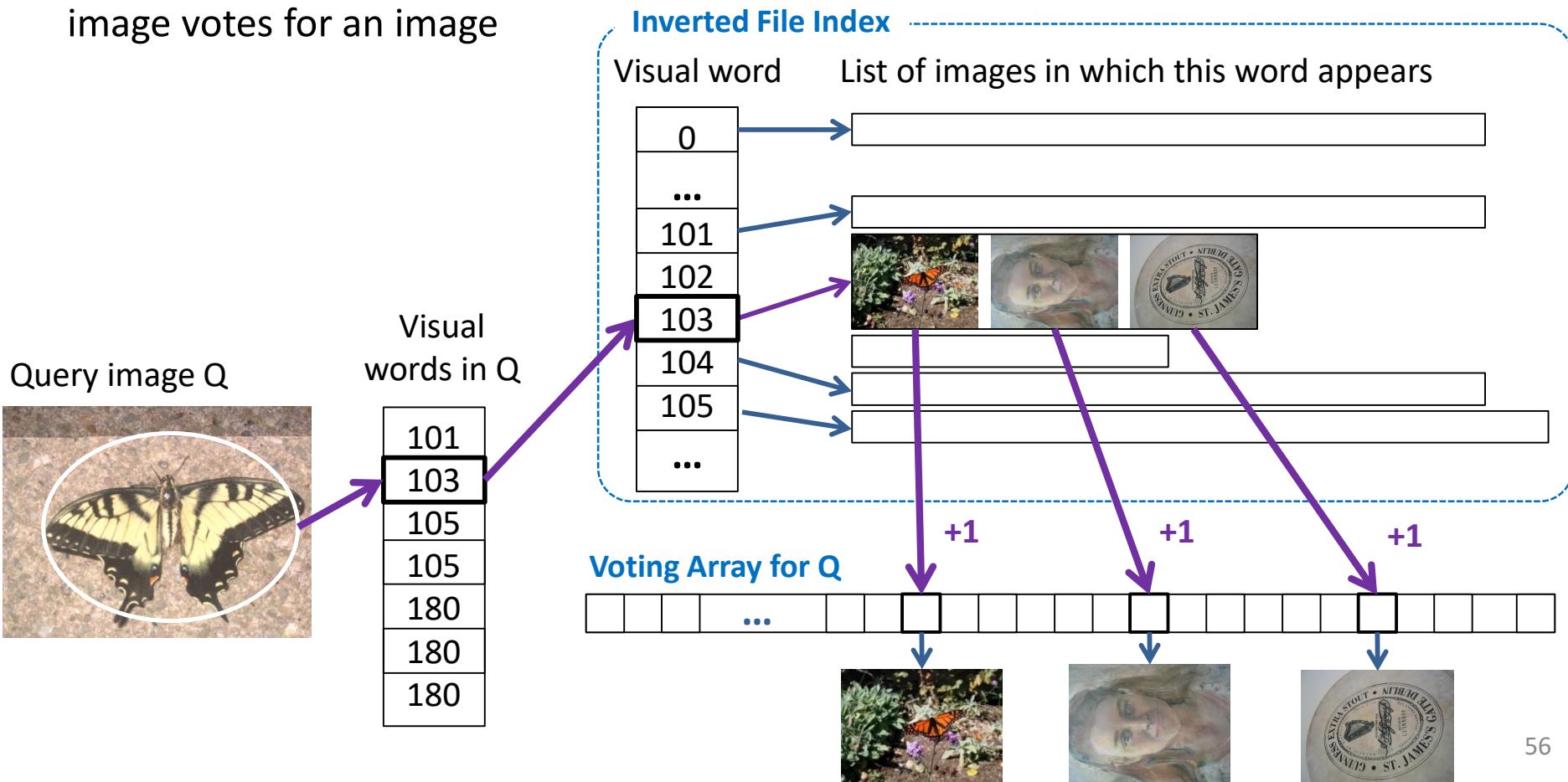


Examples
of
Features
belonging
to the
same
clusters

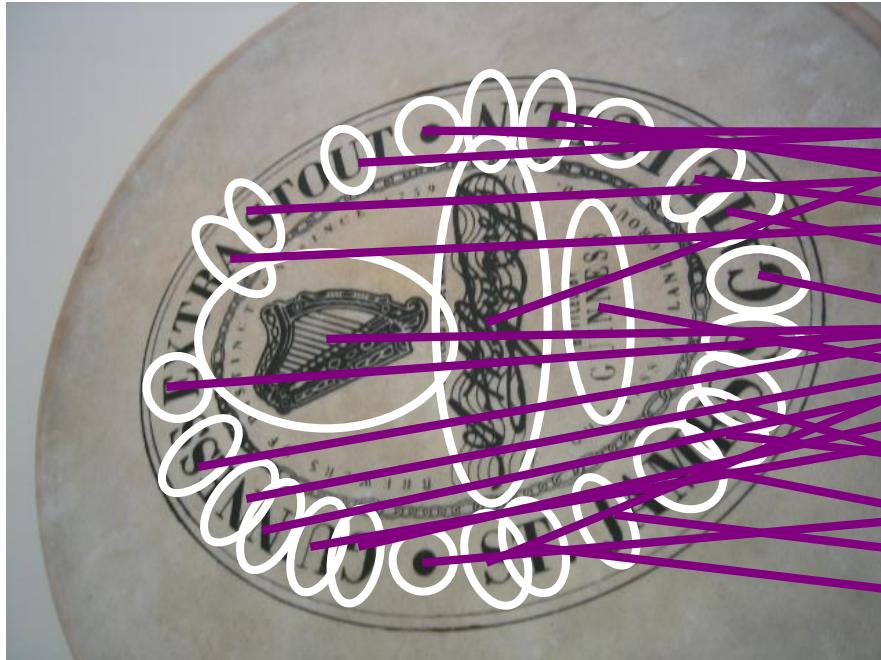


Inverted File index

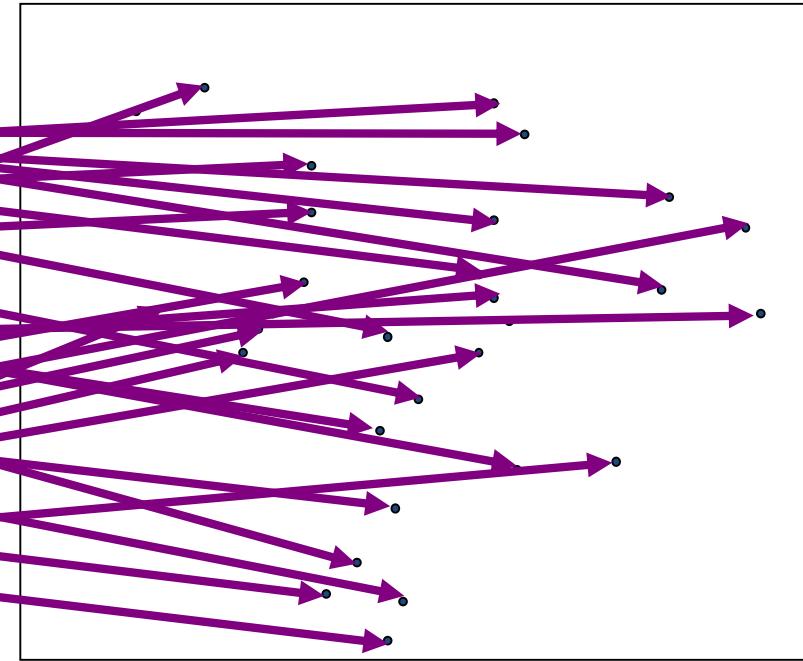
- **Inverted File Index** lists all visual words in the vocabulary (extracted at training time)
- Each word points to a **list of images**, from the all image Data Base (DB), in which that word appears. The DB grows as the robot navigates and collects new images.
- **Voting array**: has as many cells as the images in the DB. Each word in the query image votes for an image



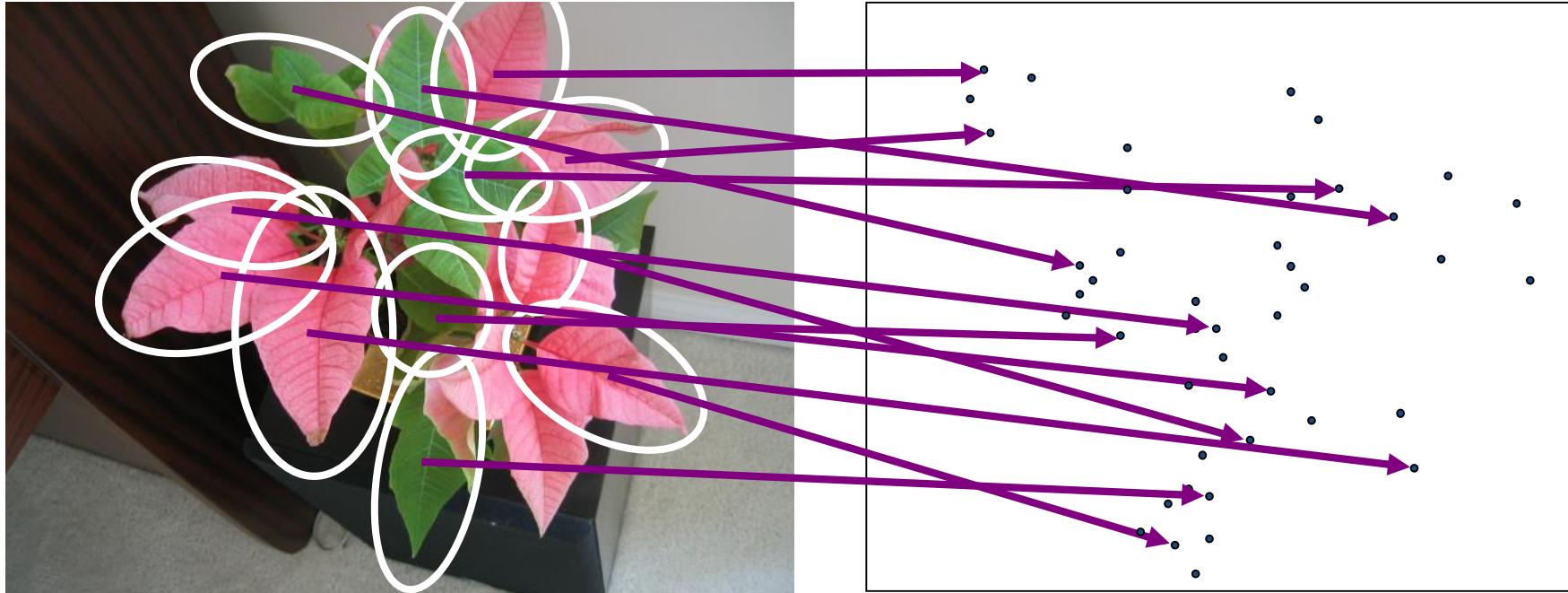
Populating the vocabulary



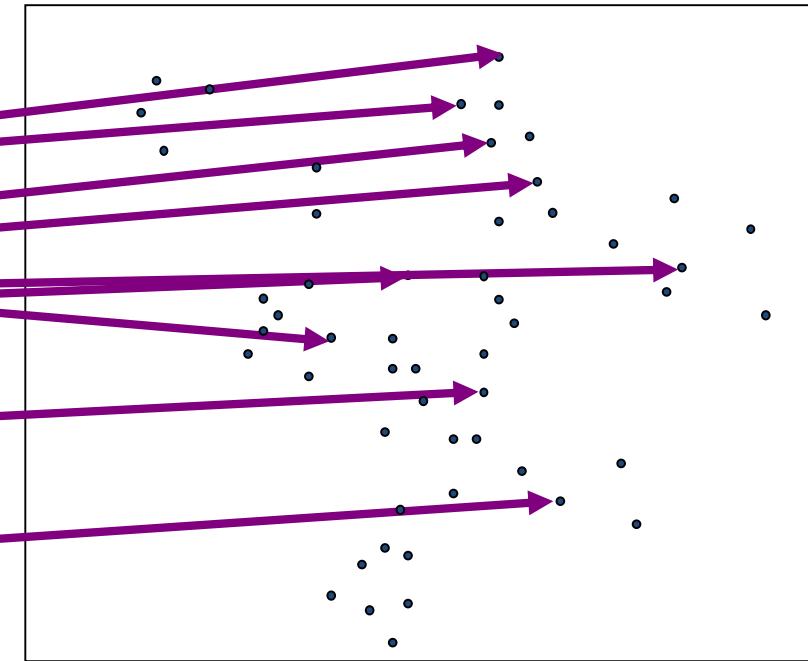
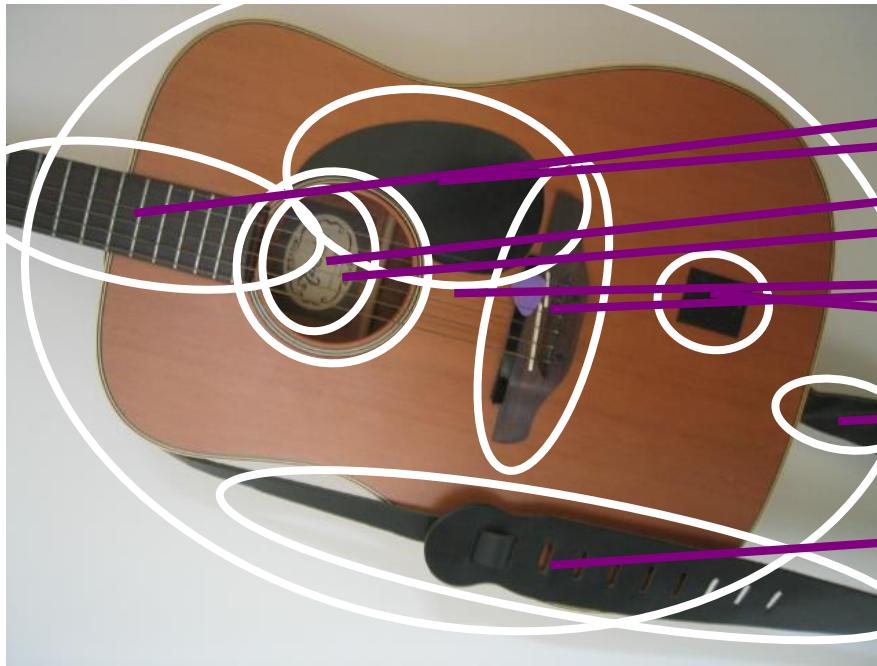
Feature descriptor space



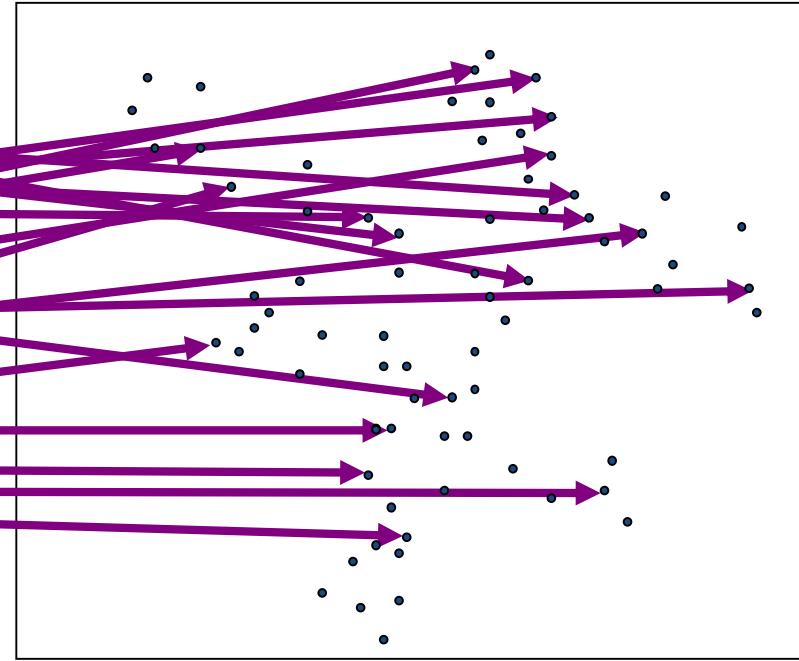
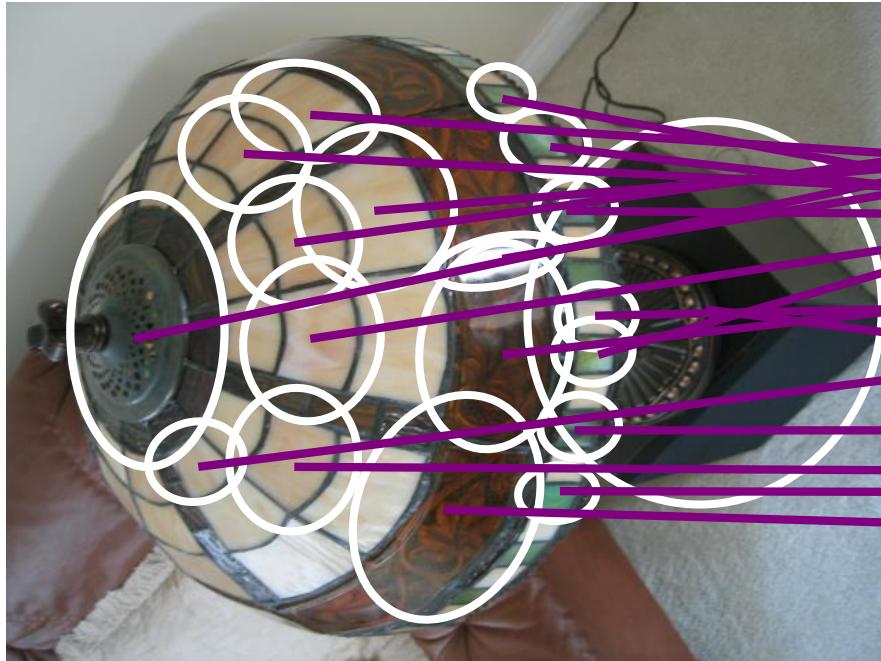
Populating the vocabulary



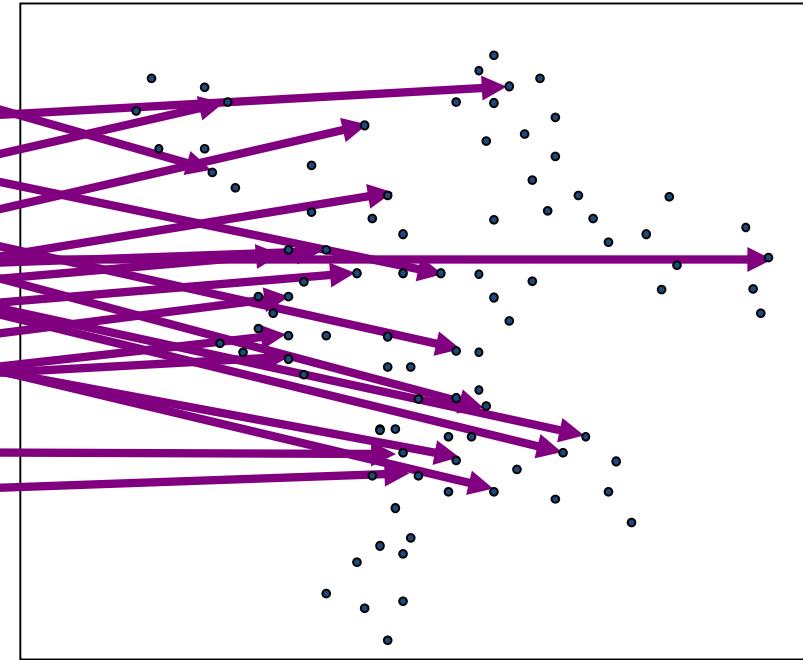
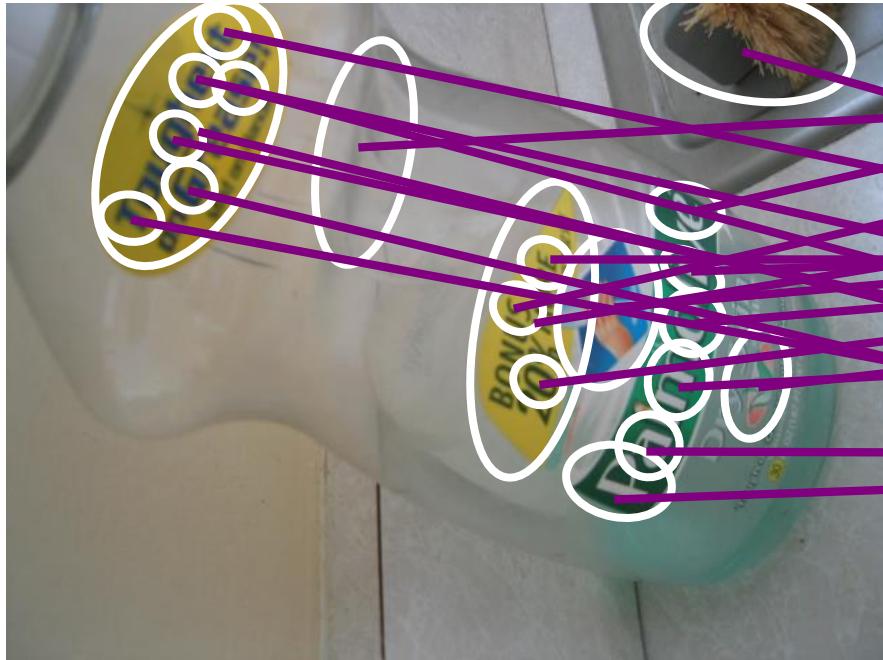
Populating the vocabulary



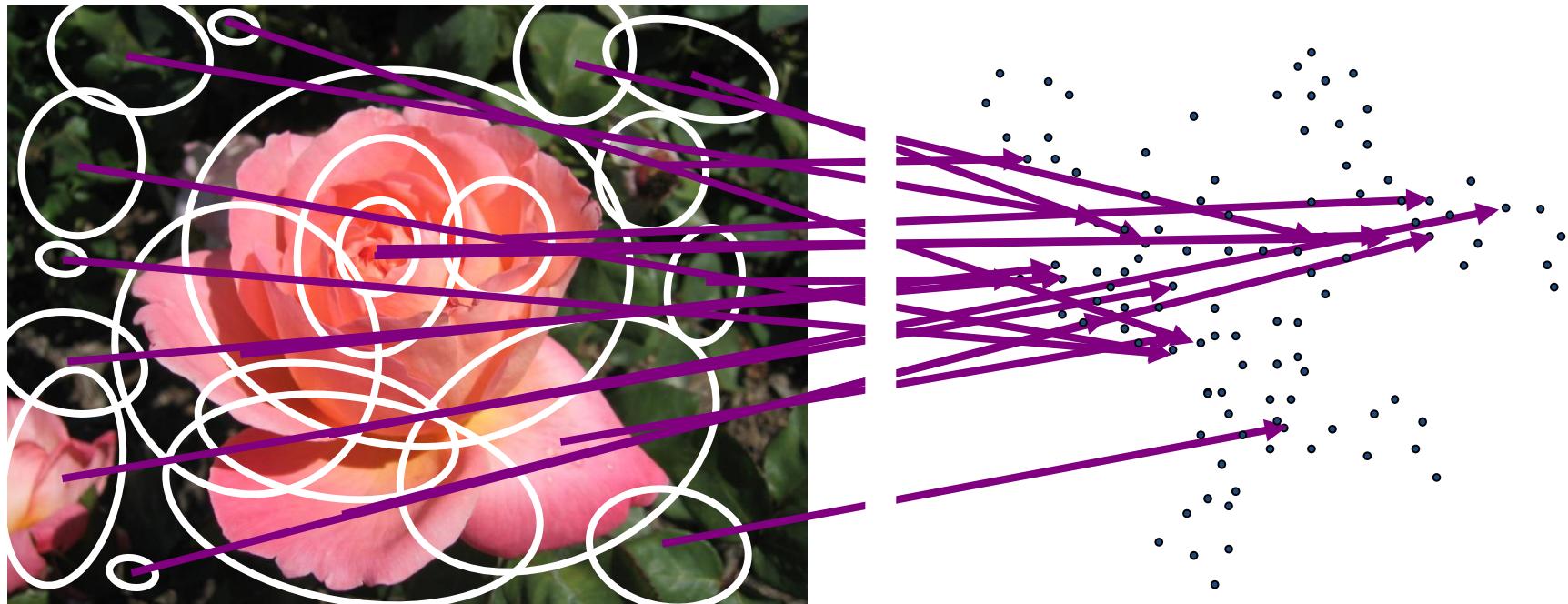
Populating the vocabulary

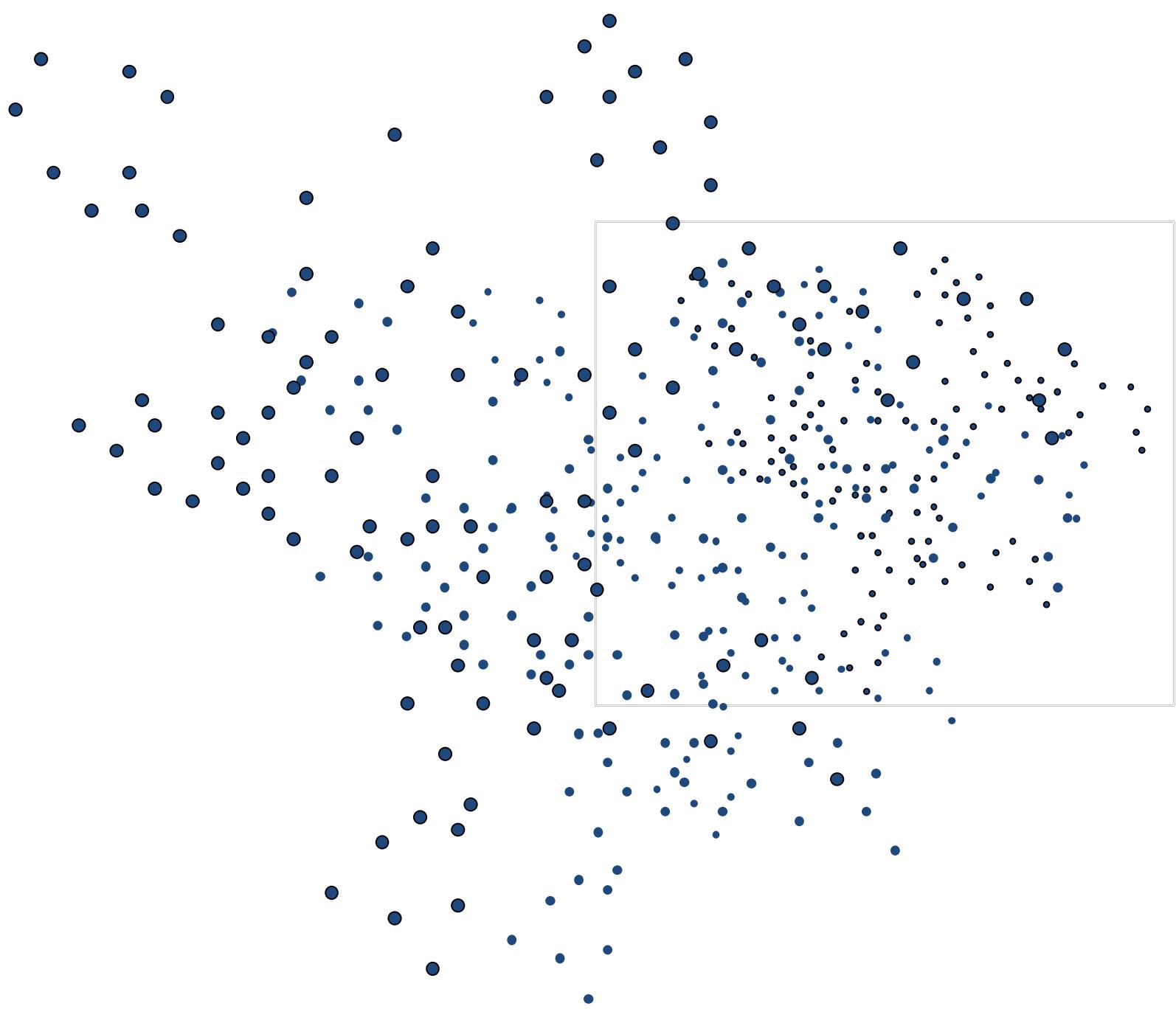


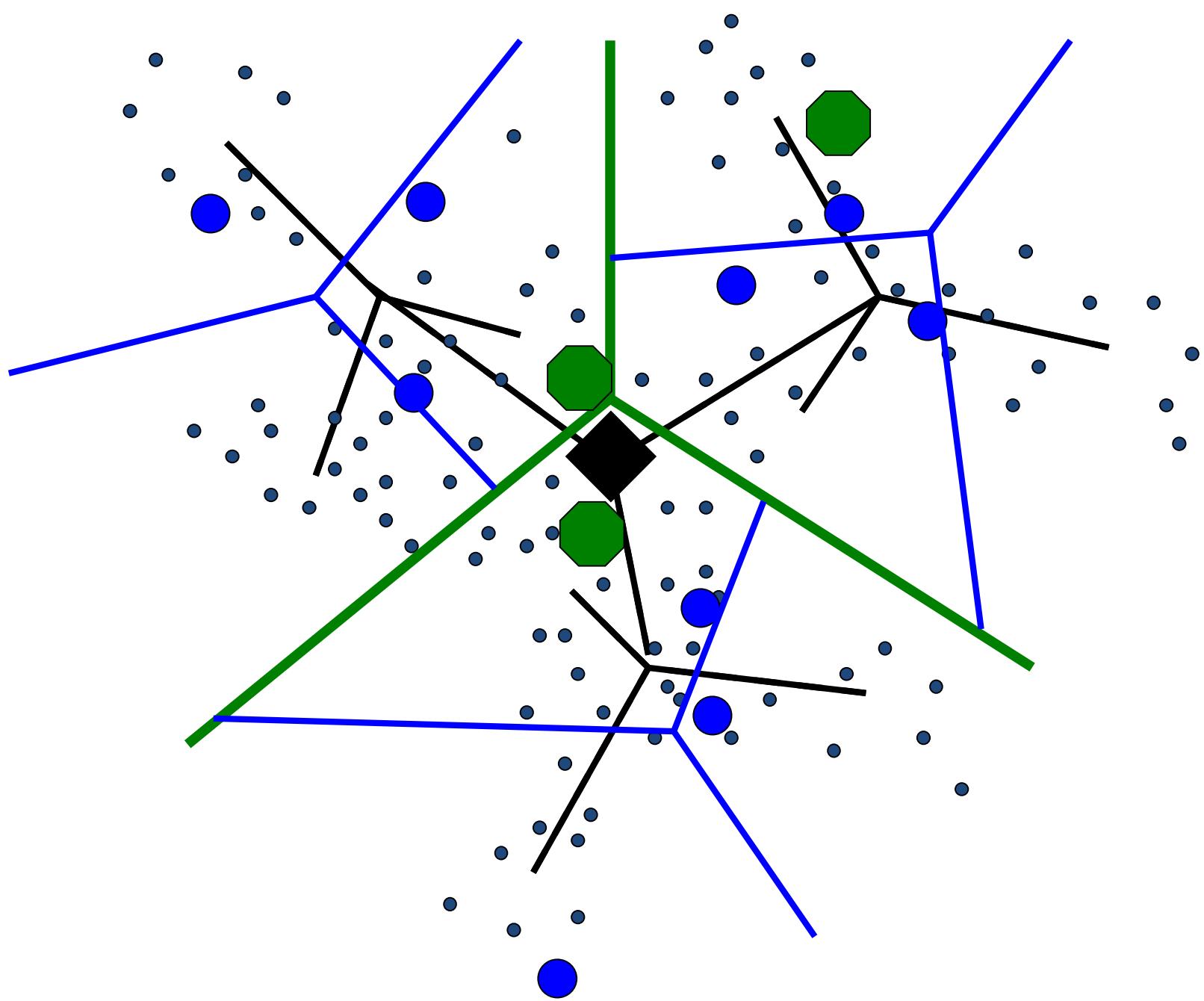
Populating the vocabulary

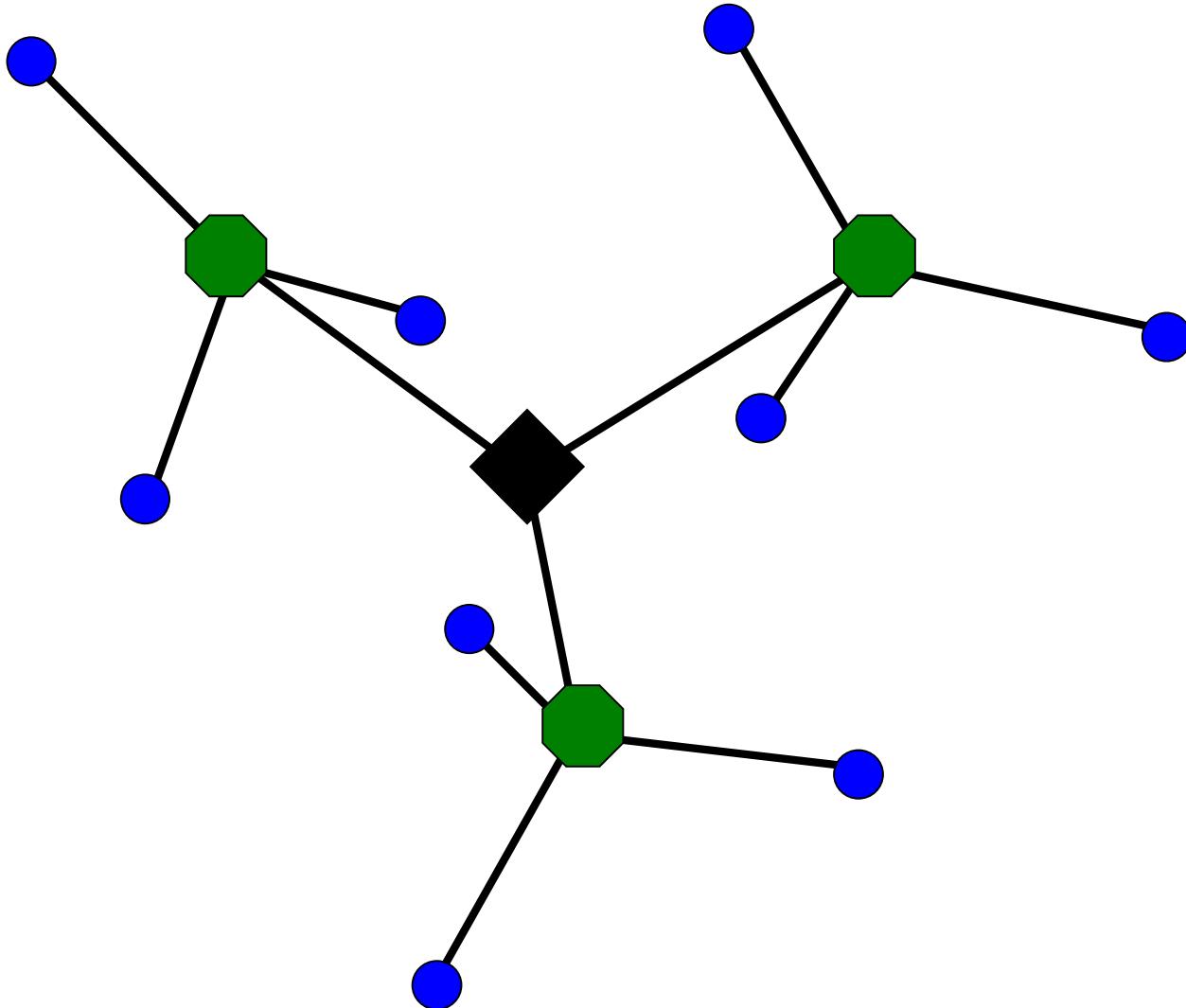


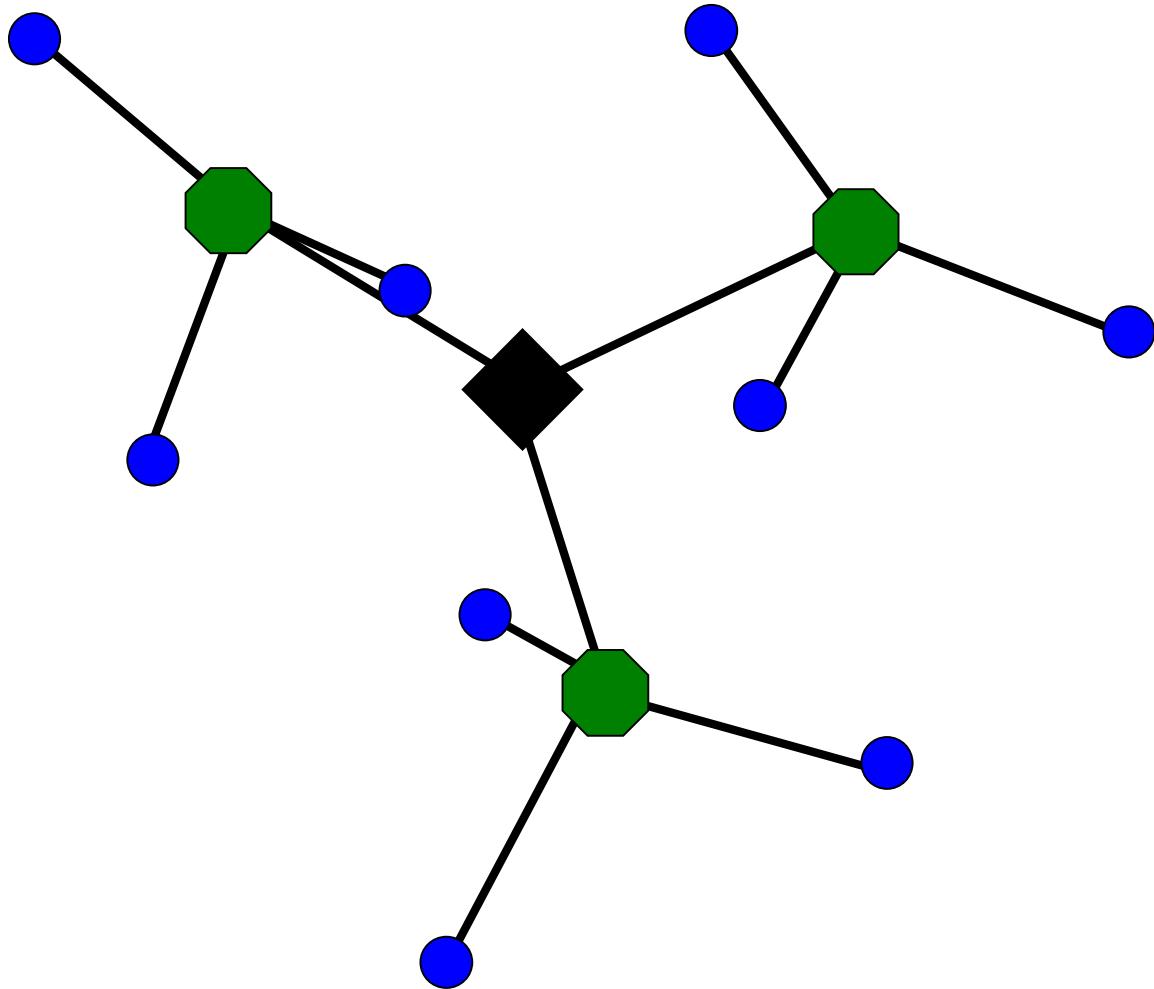
Populating the vocabulary

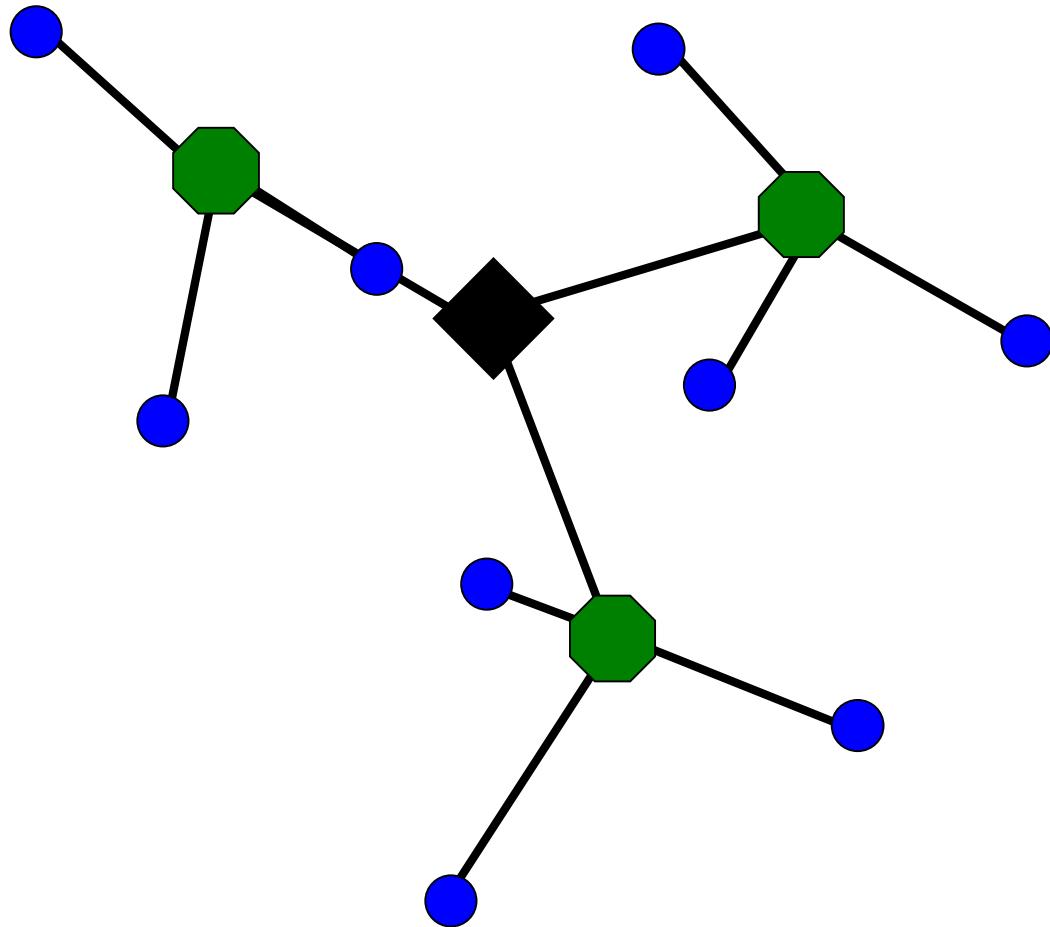


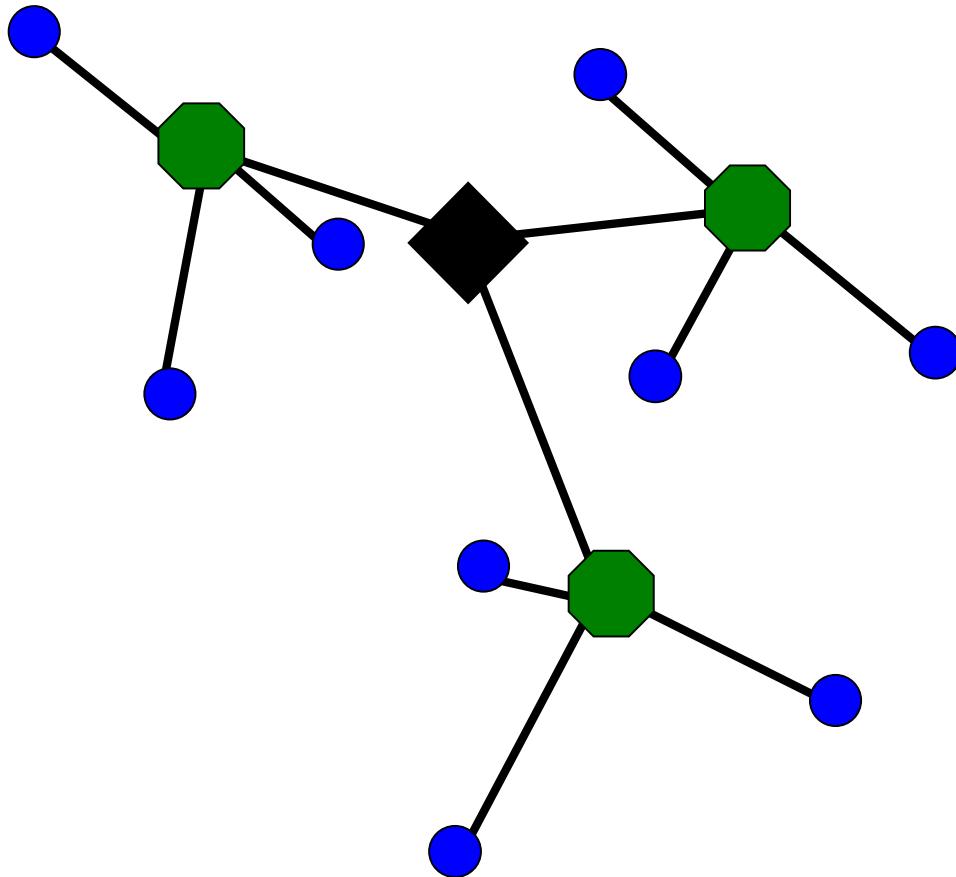


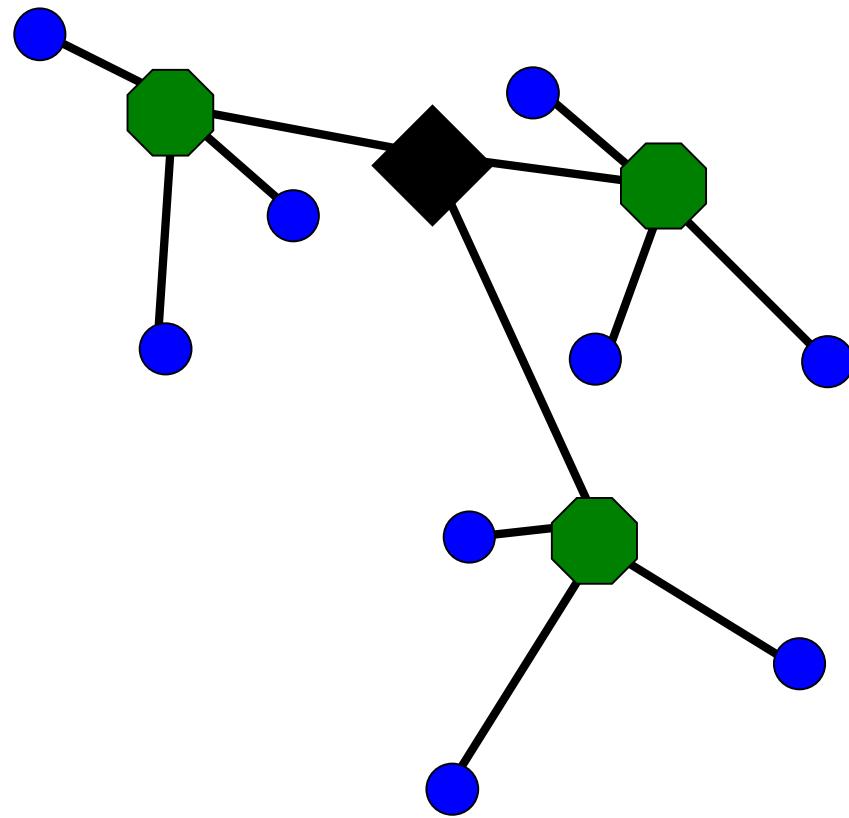


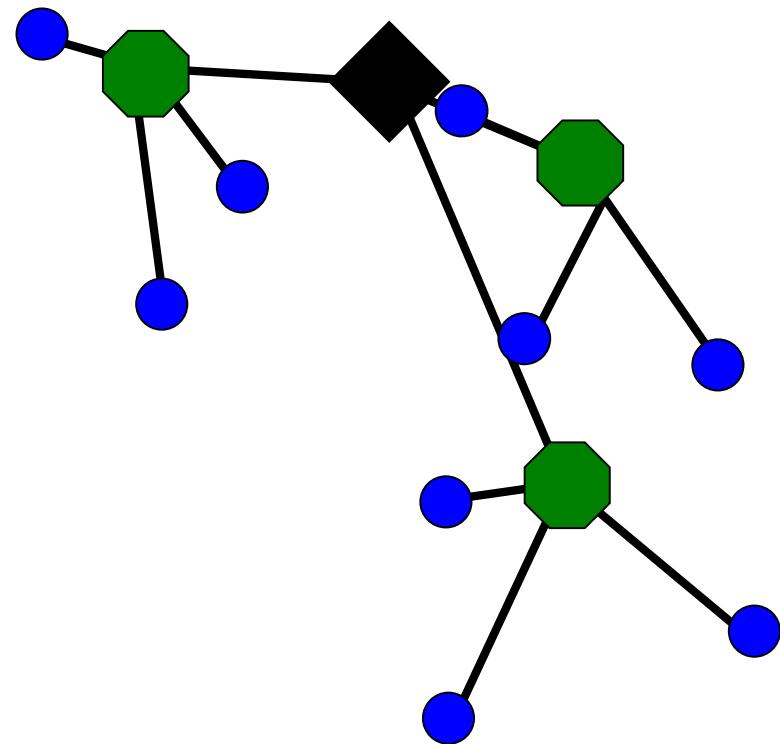


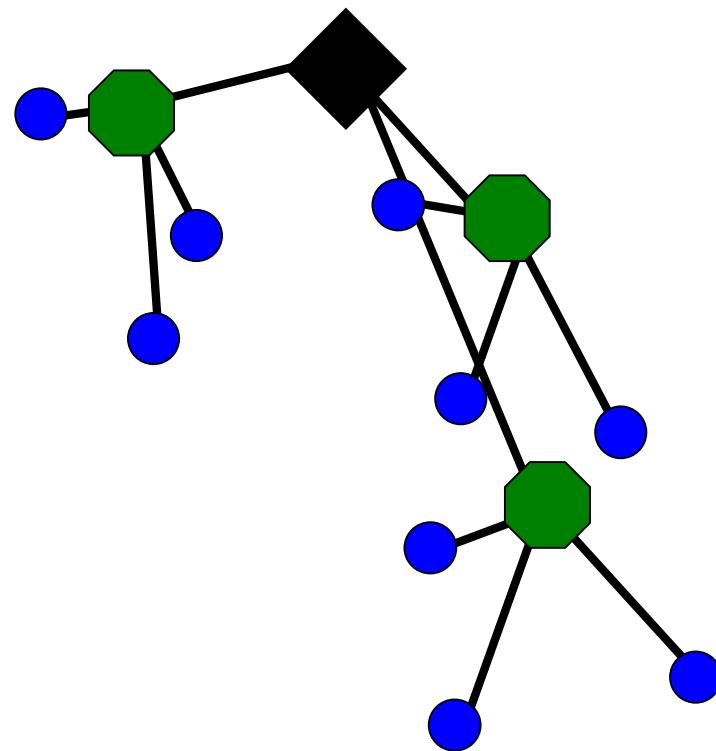


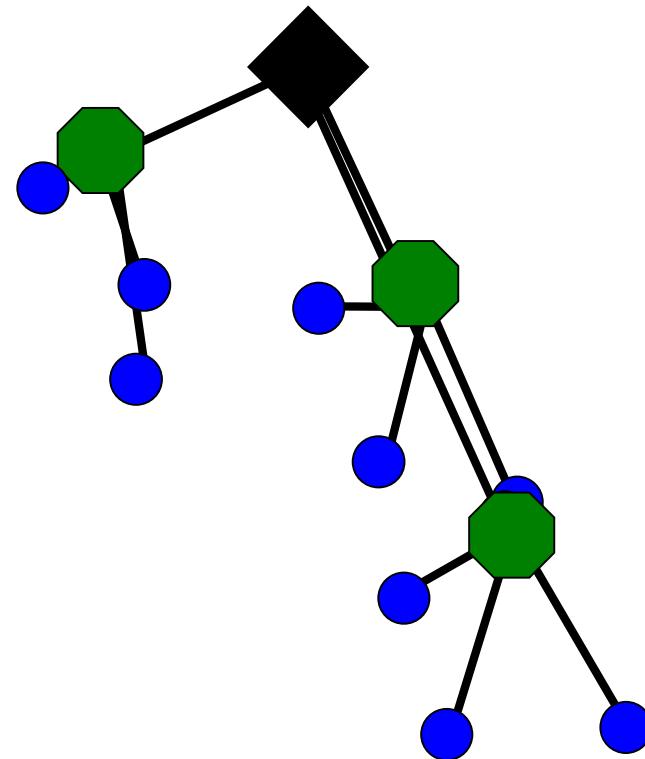


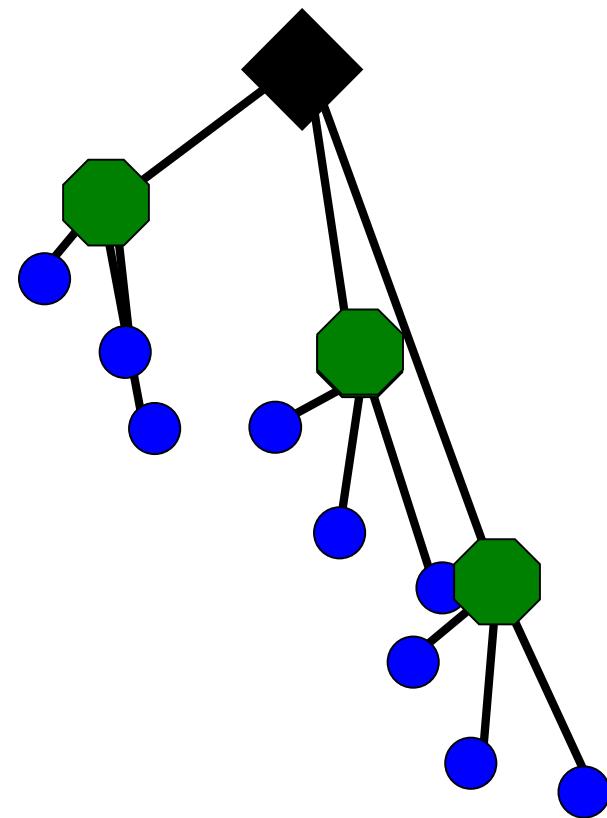


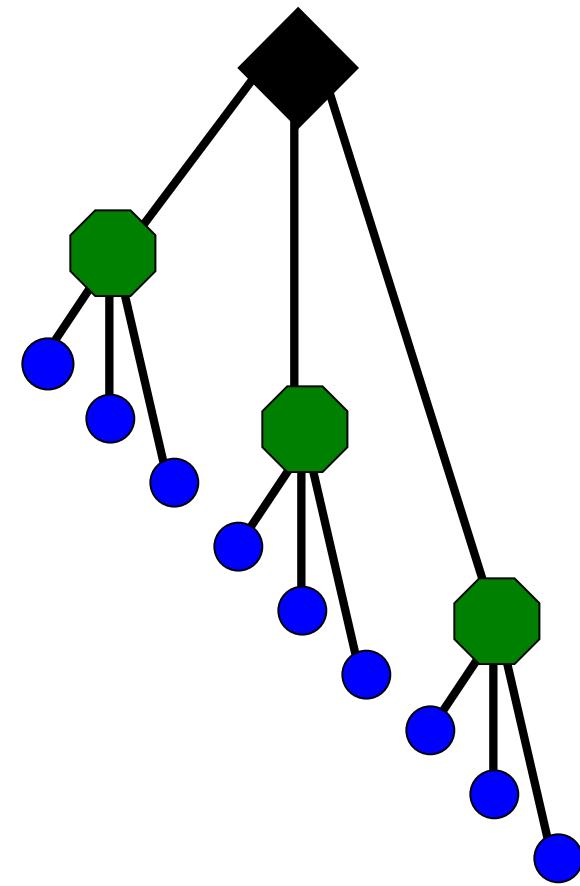




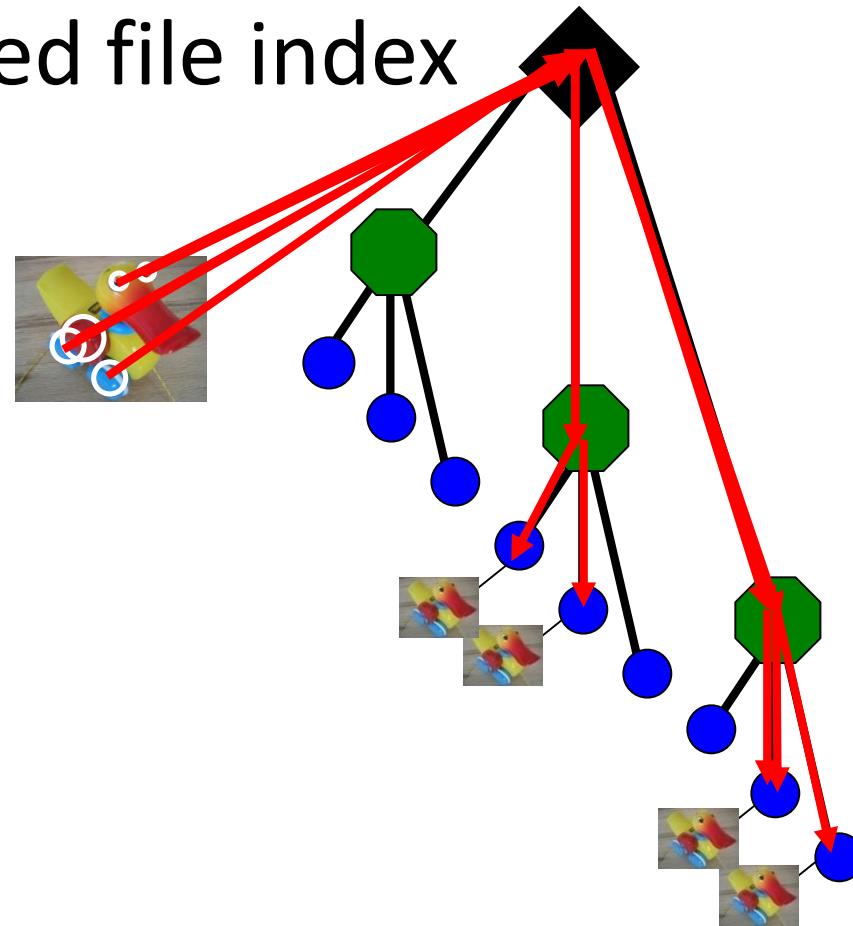




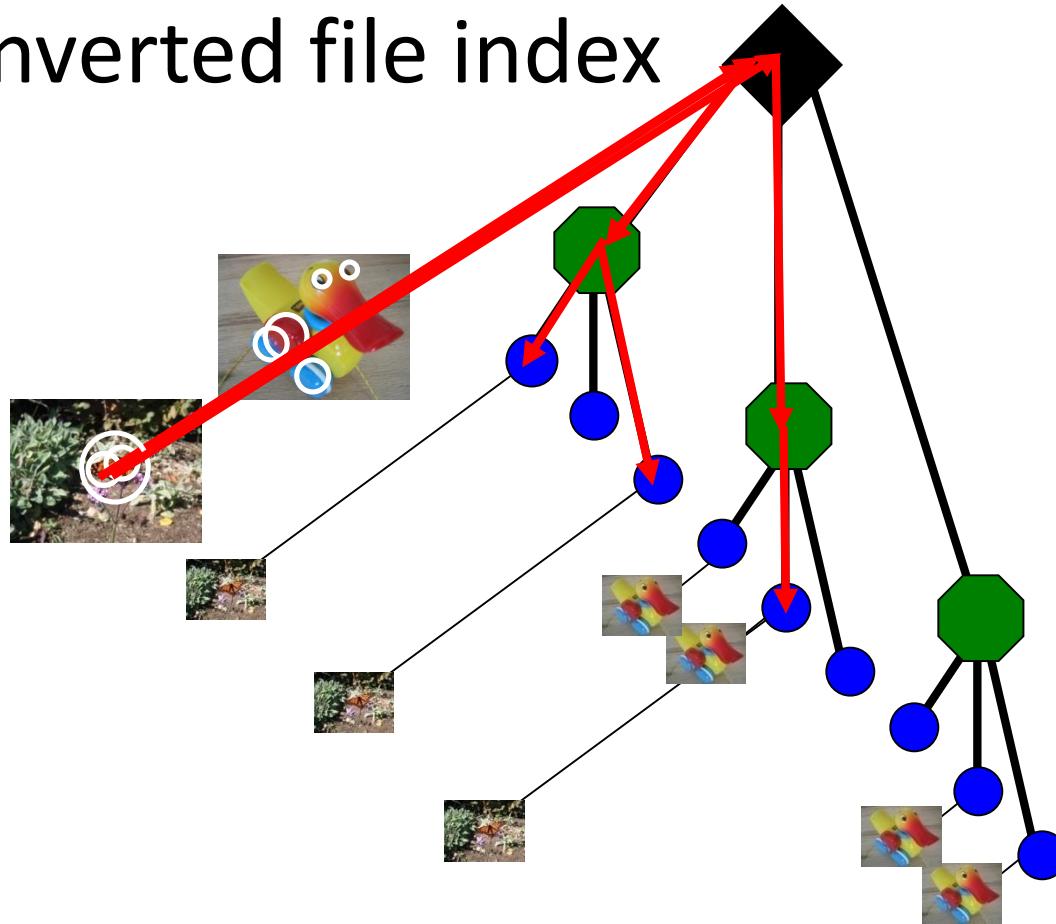




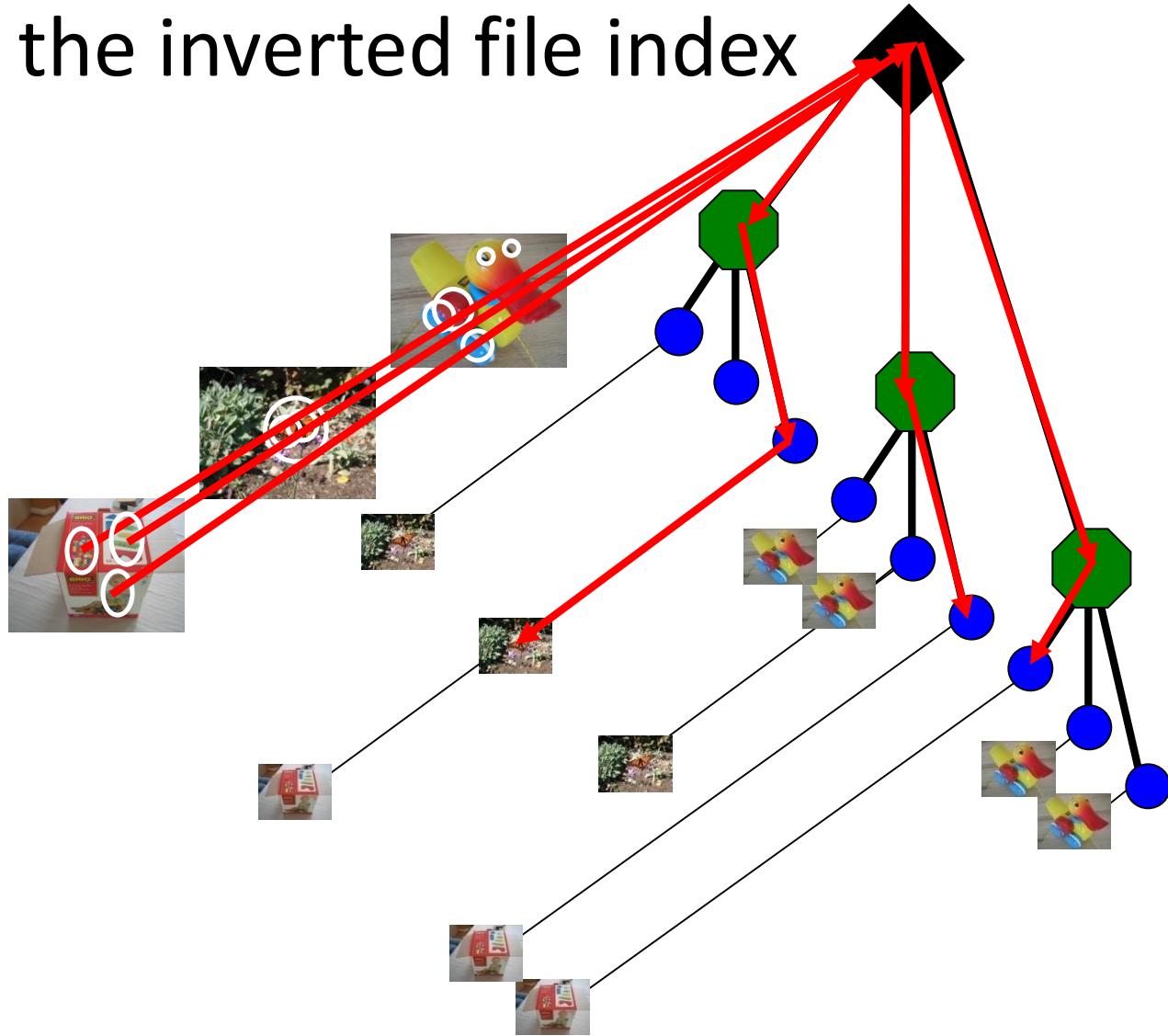
Building the inverted file index



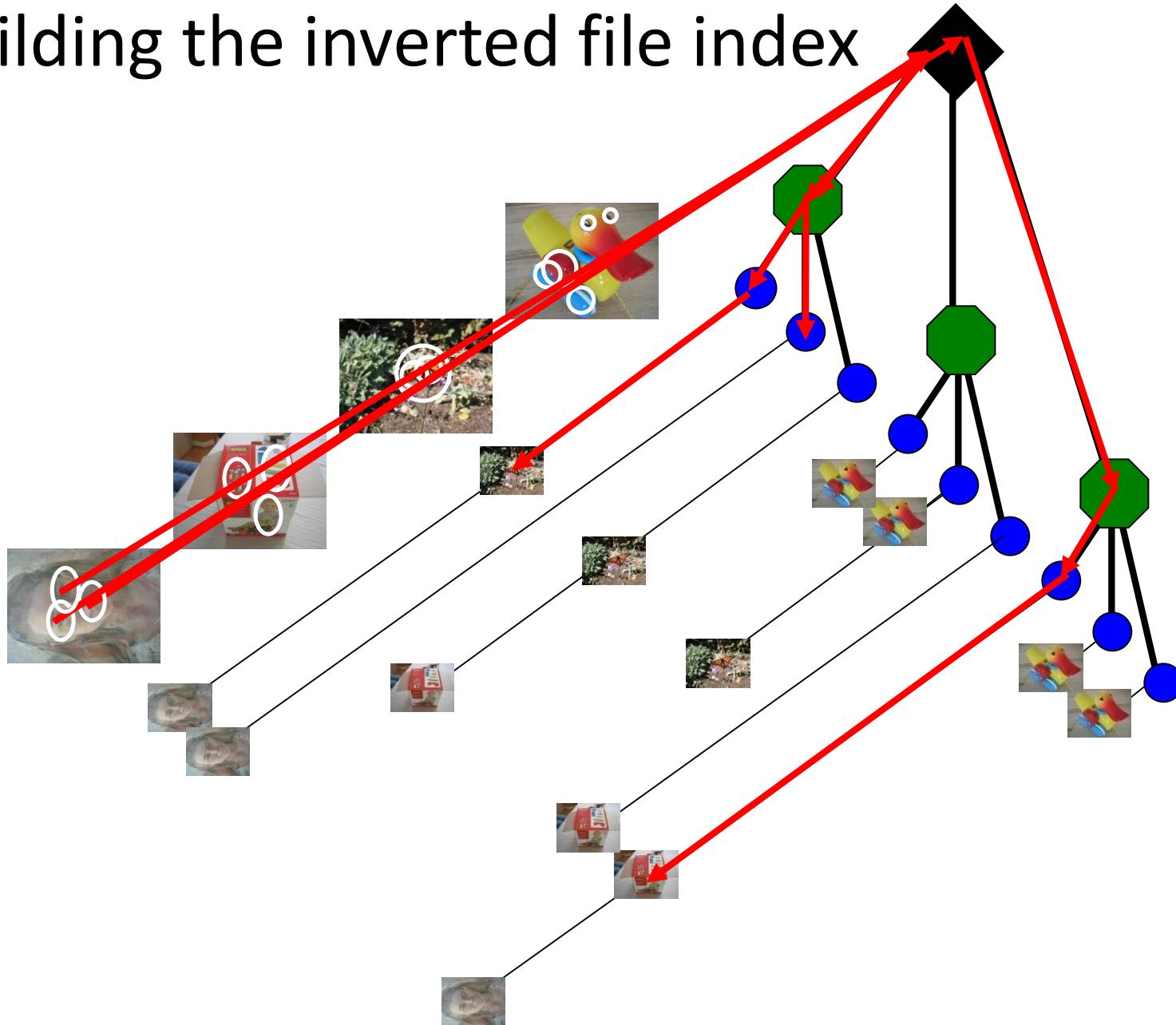
Building the inverted file index



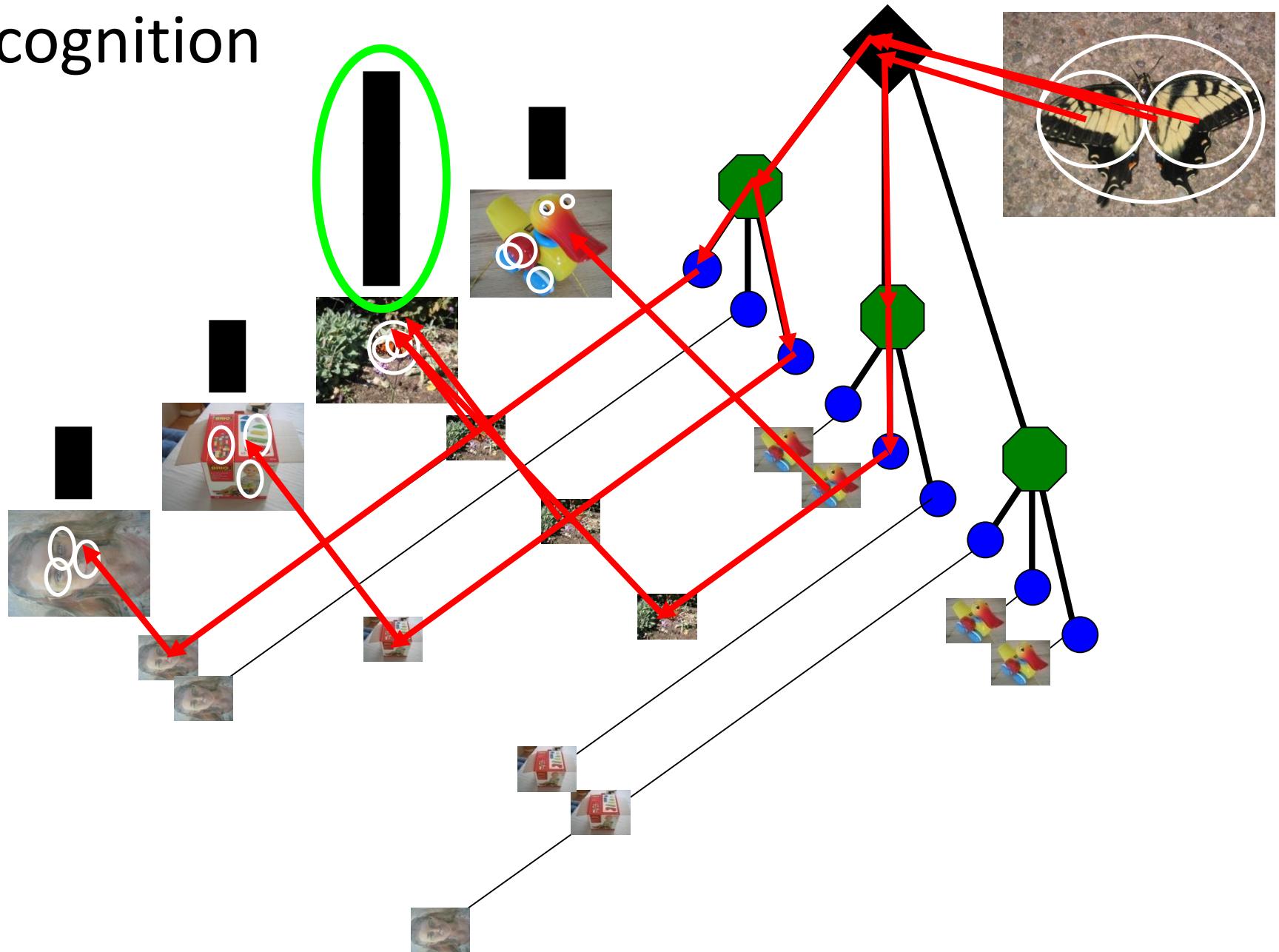
Building the inverted file index



Building the inverted file index



Recognition



Robust object/scene recognition

- Visual Vocabulary discards the spatial relationships between features
 - Two images with the same features *shuffled around* will return a 100% match when using only appearance information.
- This can be overcome using **geometric verification**
 - Test the h most similar images to the query image for geometric consistency (e.g. using 5- or 8-point RANSAC) and retain the image with the smallest reprojection error and largest number of inliers
 - Further reading (out of scope of this course):
 - [Cummins and Newman, IJRR 2011]
 - [Stewénius et al, ECCV 2012]

Video Google System

1. Collect all words within query region
2. Inverted file index to find relevant frames
3. Compare word counts
4. Spatial verification

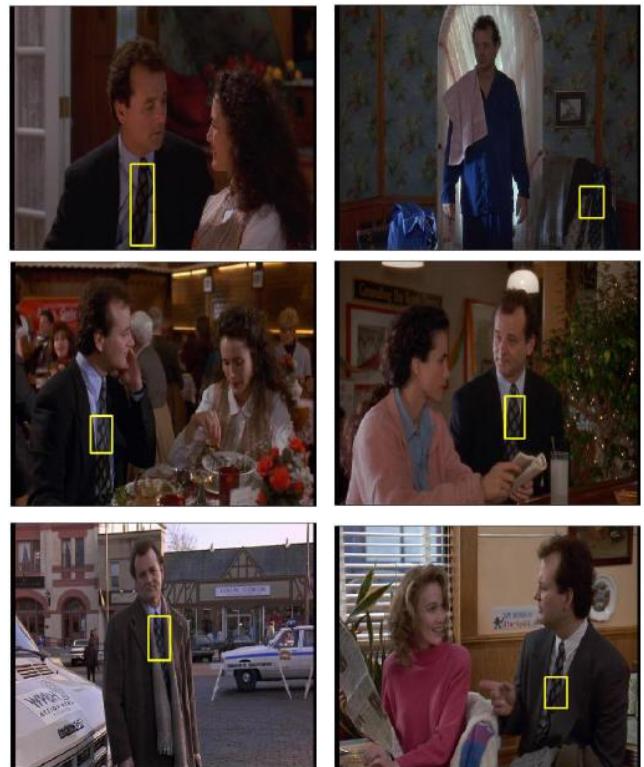
Sivic & Zisserman, ICCV 2003

- Demo online at :
<http://www.robots.ox.ac.uk/~vgg/research/vgoogle/>

Query region



Retrieved frames



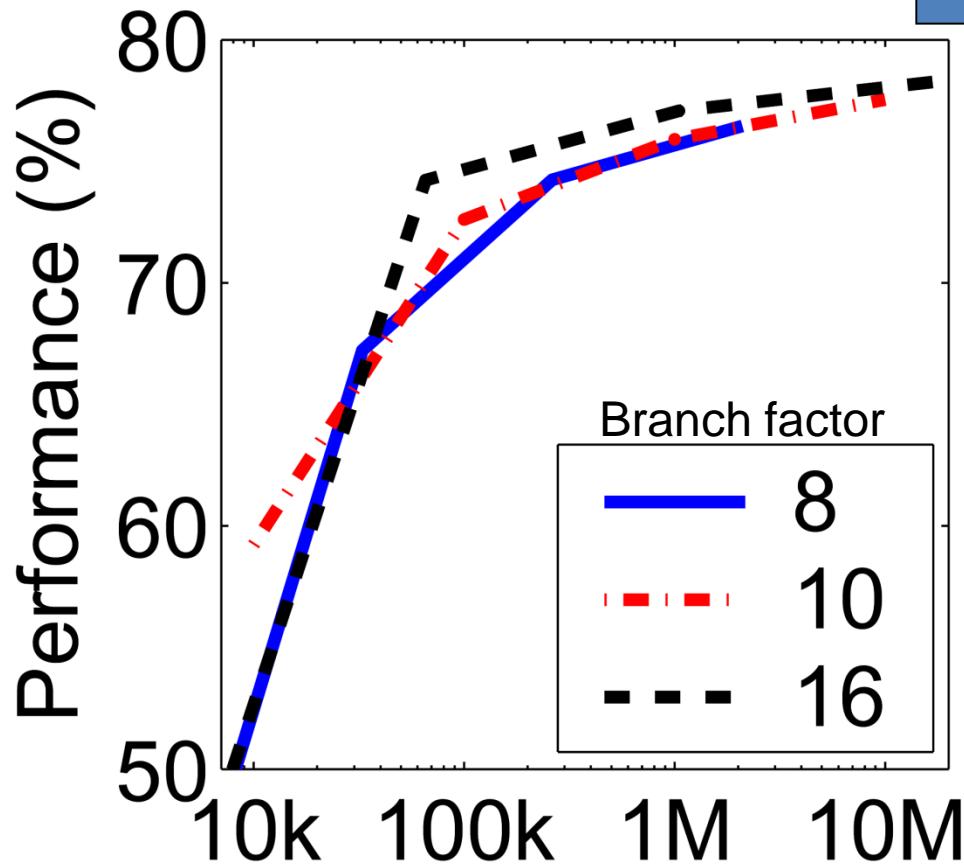
More words is better



Improves
Retrieval



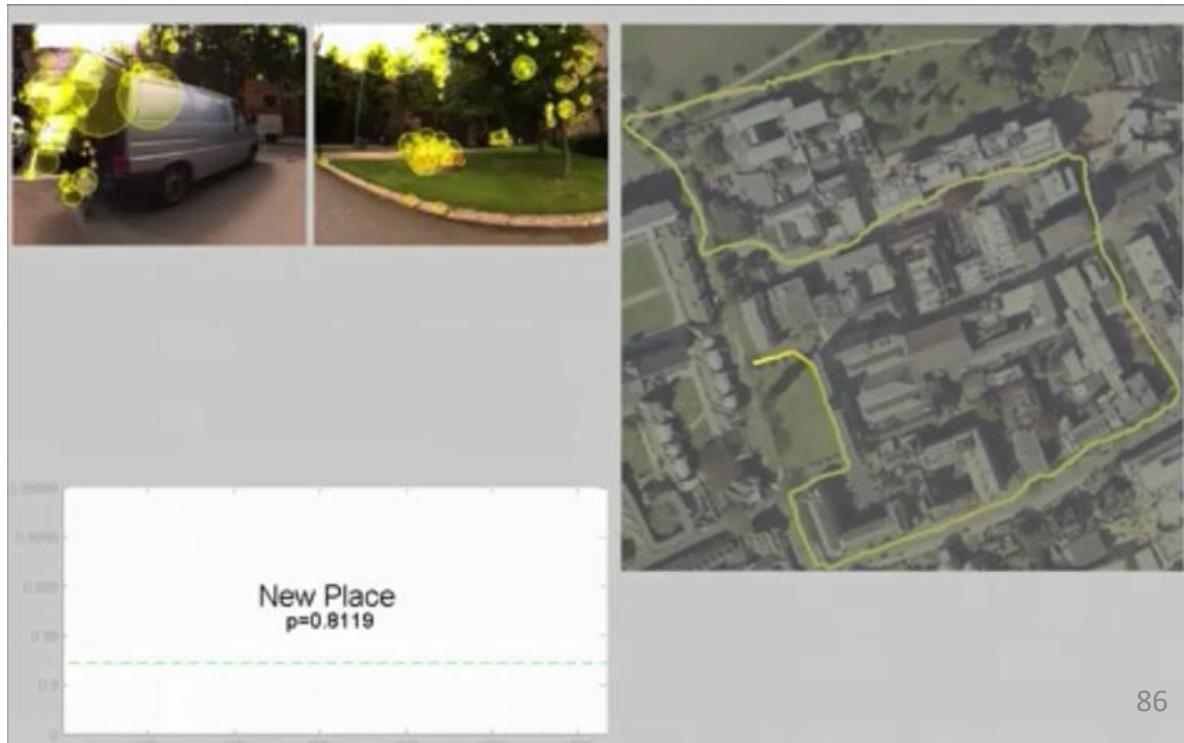
Improves
Speed



FABMAP

[Cummins and Newman IJRR 2011]

- Place recognition for robot localization
- Uses training images to build the BoW database
- **Captures the spatial dependencies of visual words** to distinguish the most characteristic structure of each scene
- Probabilistic model of the world. At a new frame, compute:
 - $P(\text{being at a known place})$
 - $P(\text{being at a new place})$
- Very high performance
- Binaries available [online](#)
- [Open FABMAP](#)



Things to remember

- Appearance-based object recognition
 - Classifiers
 - K-means clustering
- Bag of Words approach
 - What is visual word
 - Inverted file index
 - How it works
- **Chapter 14 of the Szeliski's book**

