ROBOTICS & PERCEPTION GROUP

Universität Zürich[UZH]

Lecture 12
Recognition

Davide Scaramuzza

Oral exam dates

- UZH
  - January 19-20

- ETH
  - 30.01 to 9.02 2017 (schedule handled by ETH)

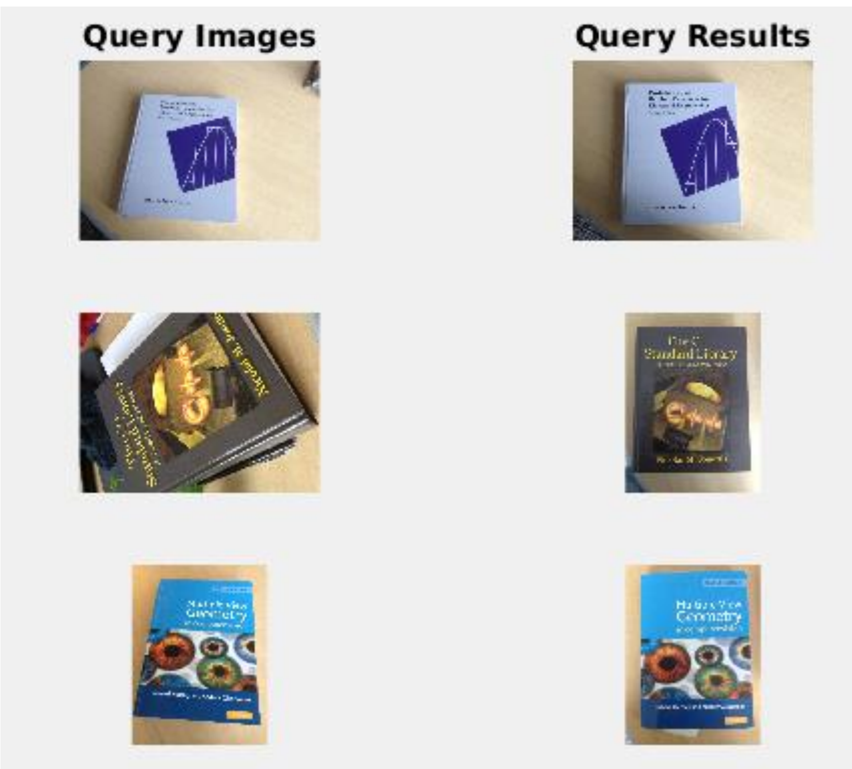
- Exam location
  - Davide Scaramuzza’s office:
    - Andreasstrasse 15, 2.10, 8050 Zurich

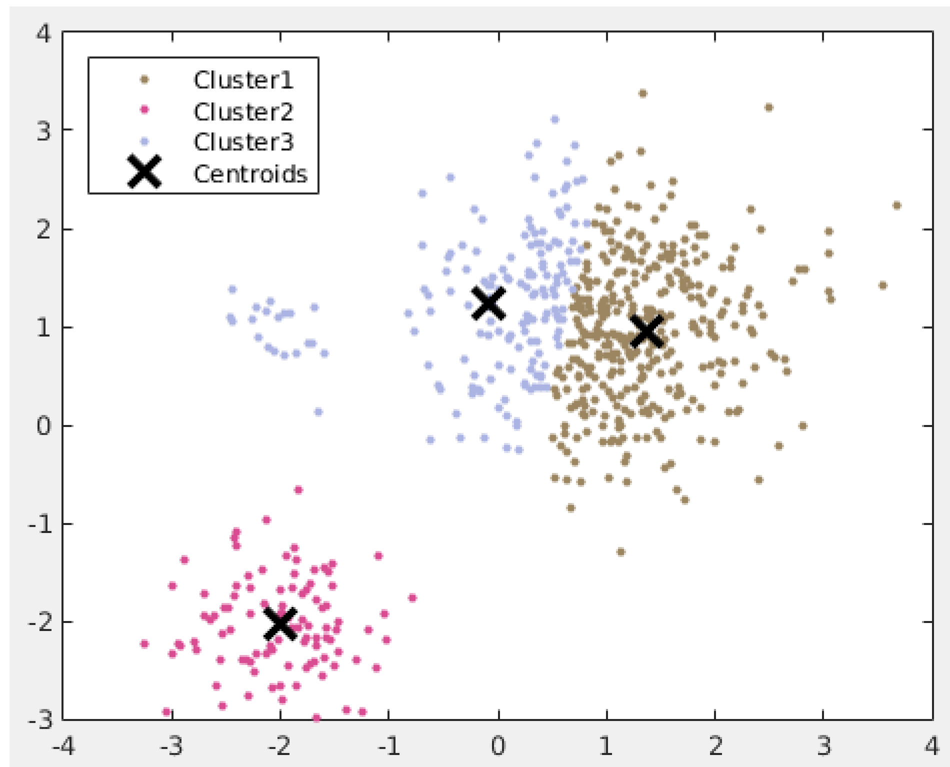
Course Evaluation

- Please fill the evaluation form you received by email!
- Provide feedback on
  - Exercises: good and bad
  - Course: good and bad
  - How to improve

Lab Exercise 6 - Today

- Room ETH HG E 33.1 from 14:15 to 16:00
- Work description: K-means clustering and Bag of Words place recognition



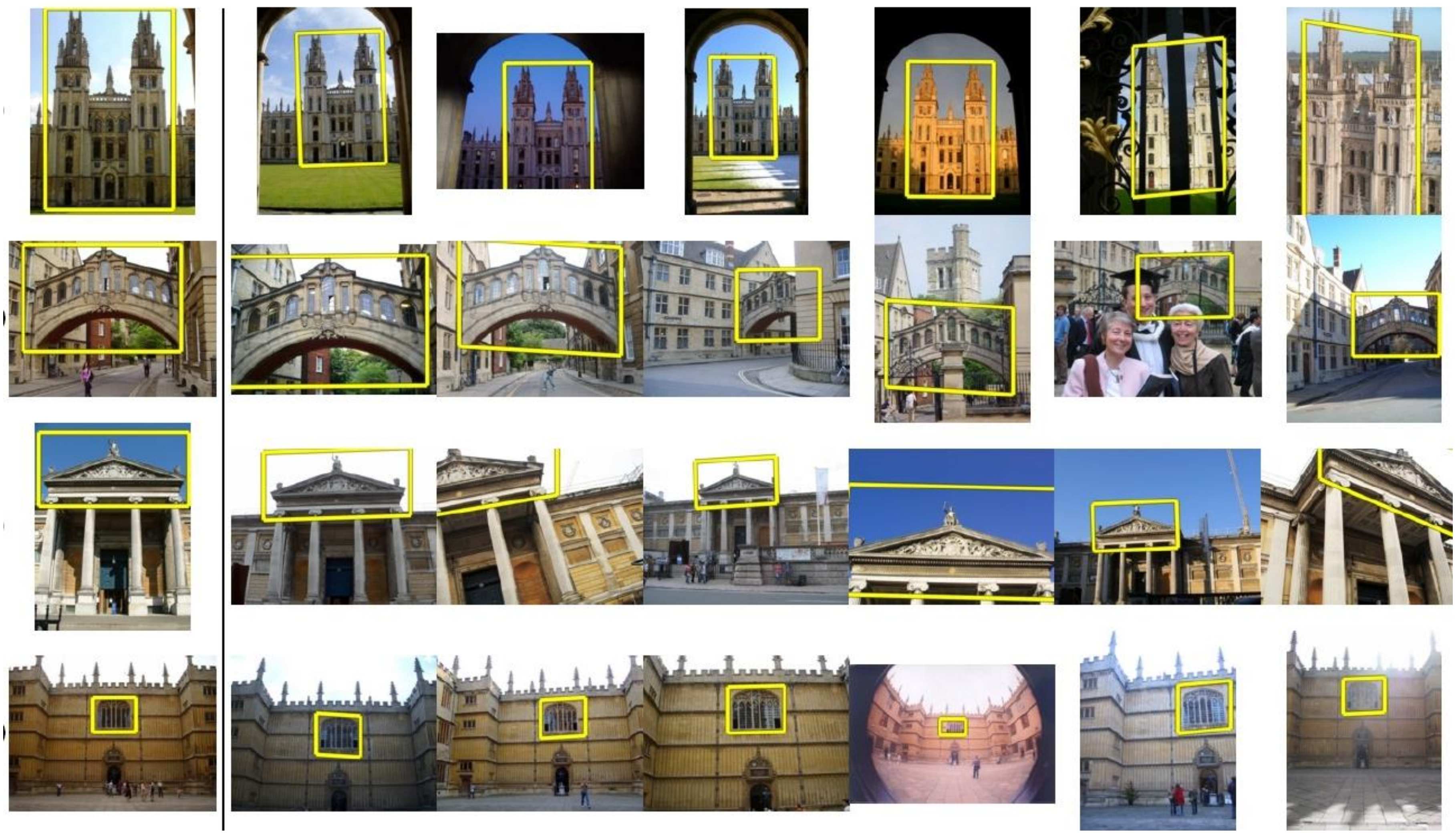


Outline

- Recognition applications and challenges
- Recognition approaches
- Classifiers
- K-means clustering
- Bag of words
- Oral Exam – Instructions and Example questions

Application: large-scale retrieval

Query image Results on a database of 100 Million images



Application: recognition for mobile phones

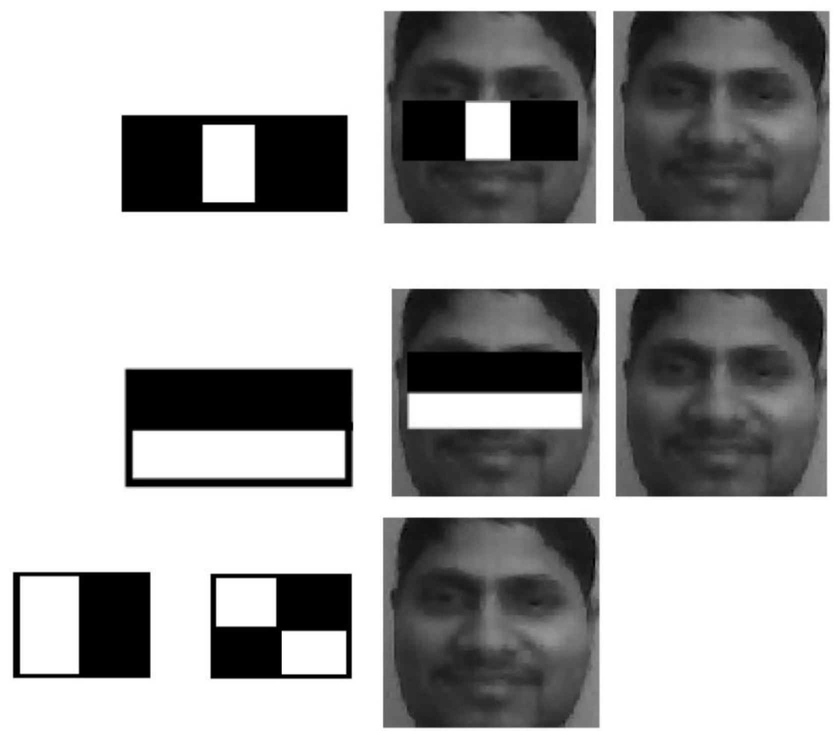
- Smartphone:
  - Lincoln Microsoft Research
  - Point & Find, Nokia
  - SnapTell.com (Amazon)
  - Google Goggles

Application: Face recognition

See iPhoto, Google Photos, Facebook

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.

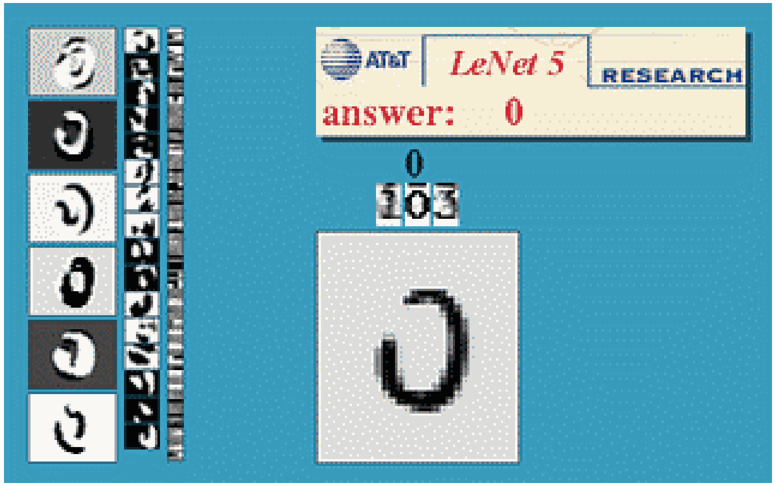


P. Viola, M. Jones: Robust Real-time Object Detection, Int. Journal of Computer Vision 2001

Application: Optical character recognition (OCR)

Technology to convert scanned docs to text

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

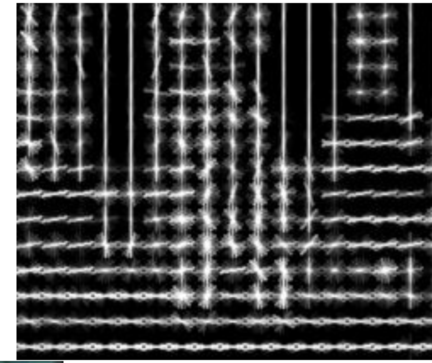


Digit recognition, AT&T labs, using CNN, by Yann LeCun (1993)
http://yann.lecun.com/

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

Application: pedestrian recognition

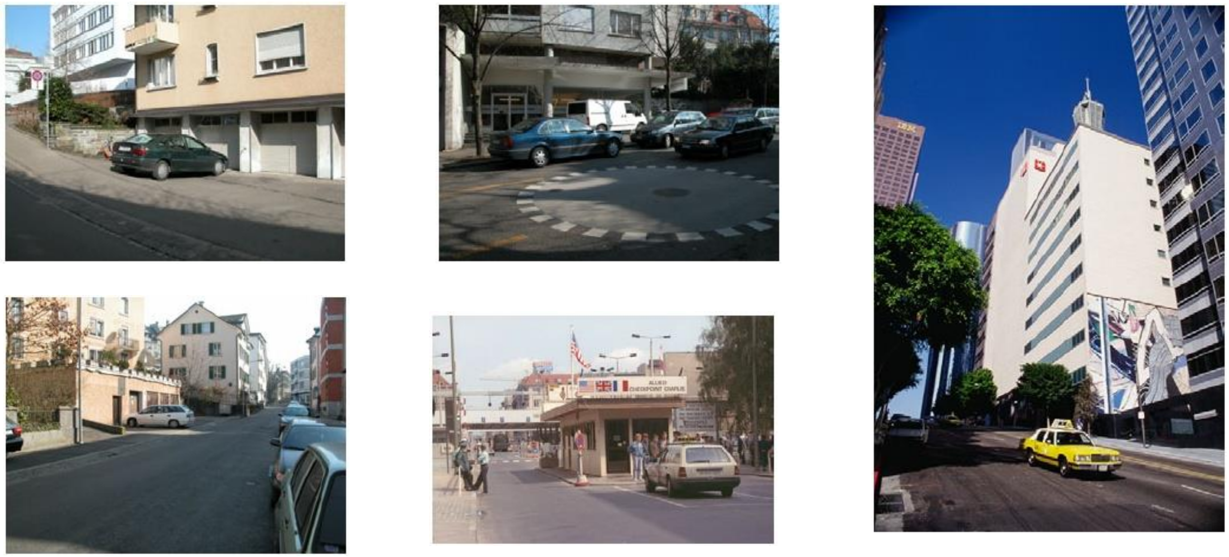
- Detector: Histograms of oriented gradients (HOG)



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





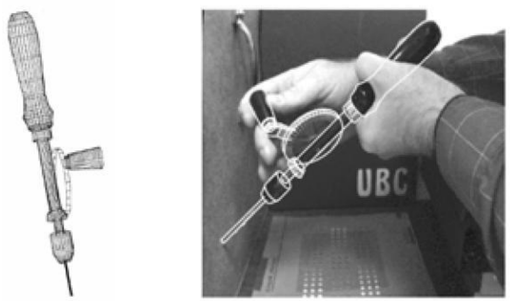

Challenges: context and human experience

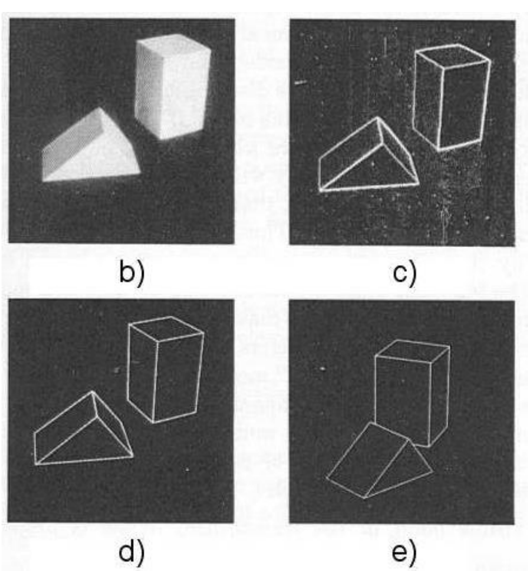
Outline

- Recognition challenges
- Recognition approaches
- Classifiers
- K-means clustering
- Bag of words
- Review of the course
- Evaluation of the course

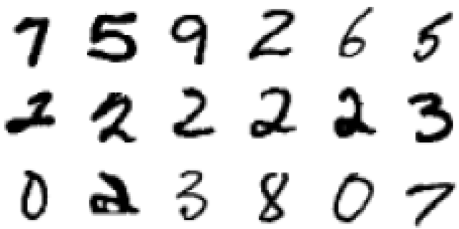
Research progress in recognition

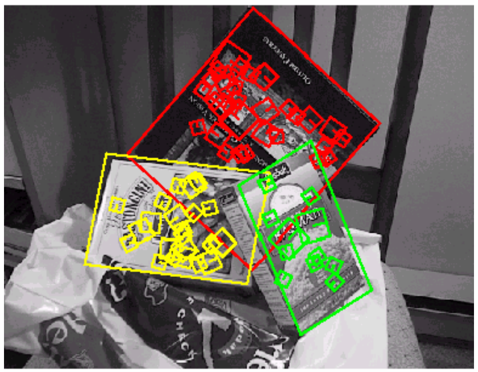
1960-1990
Polygonal objects





1990-2000
Faces, characters, planar objects

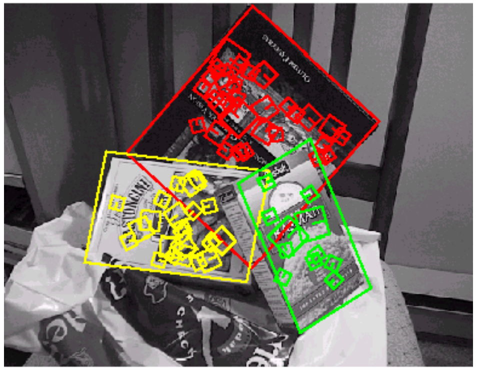


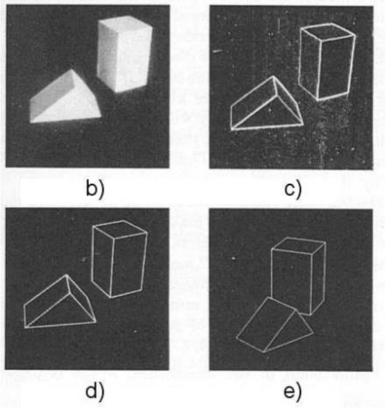


2000-today
Any kind of object

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

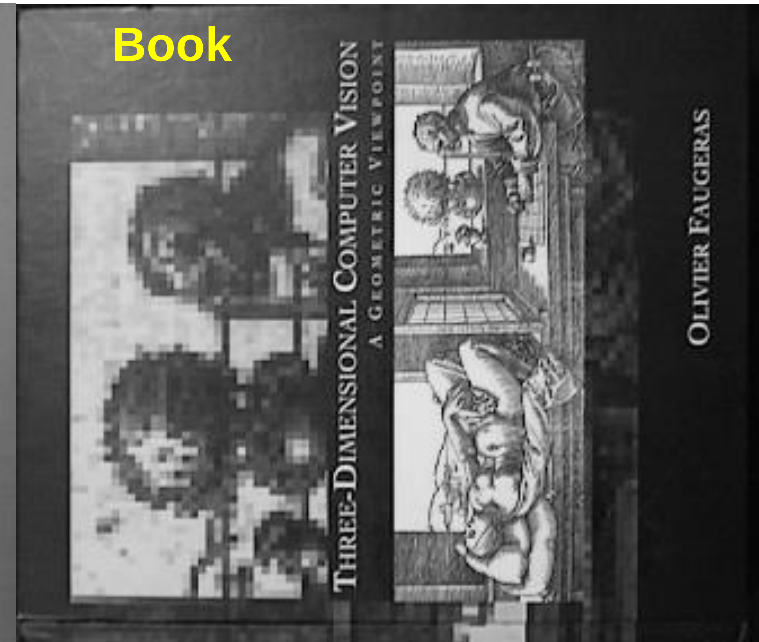




- Appearance based
  - The model is defined by a set of images representing the object
    - Example: template matching can be thought as a simple object recognition algorithm (the template is the object to recognize); disadvantage of template matching: it works only when the image matches exactly the query

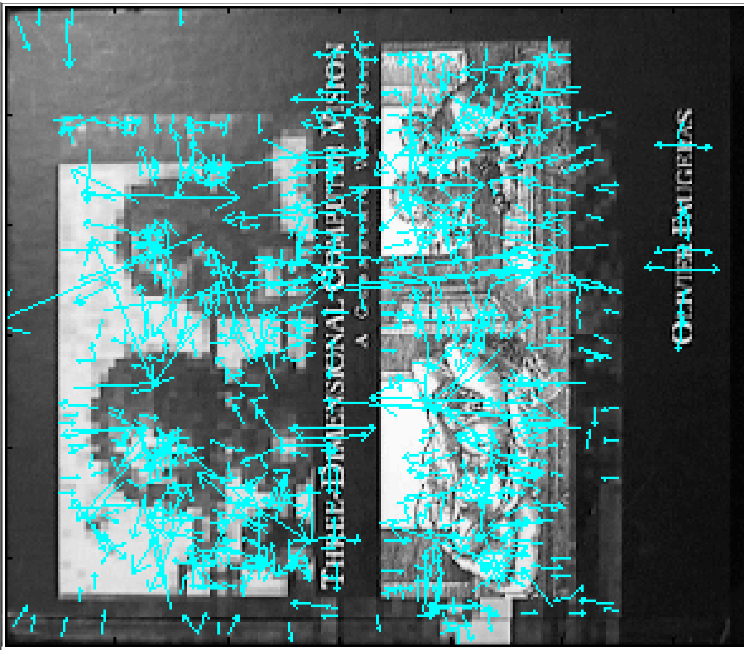
Example of 2D model-based approach

Q: Is this Book present in the Scene?



Example of 2D model-based approach

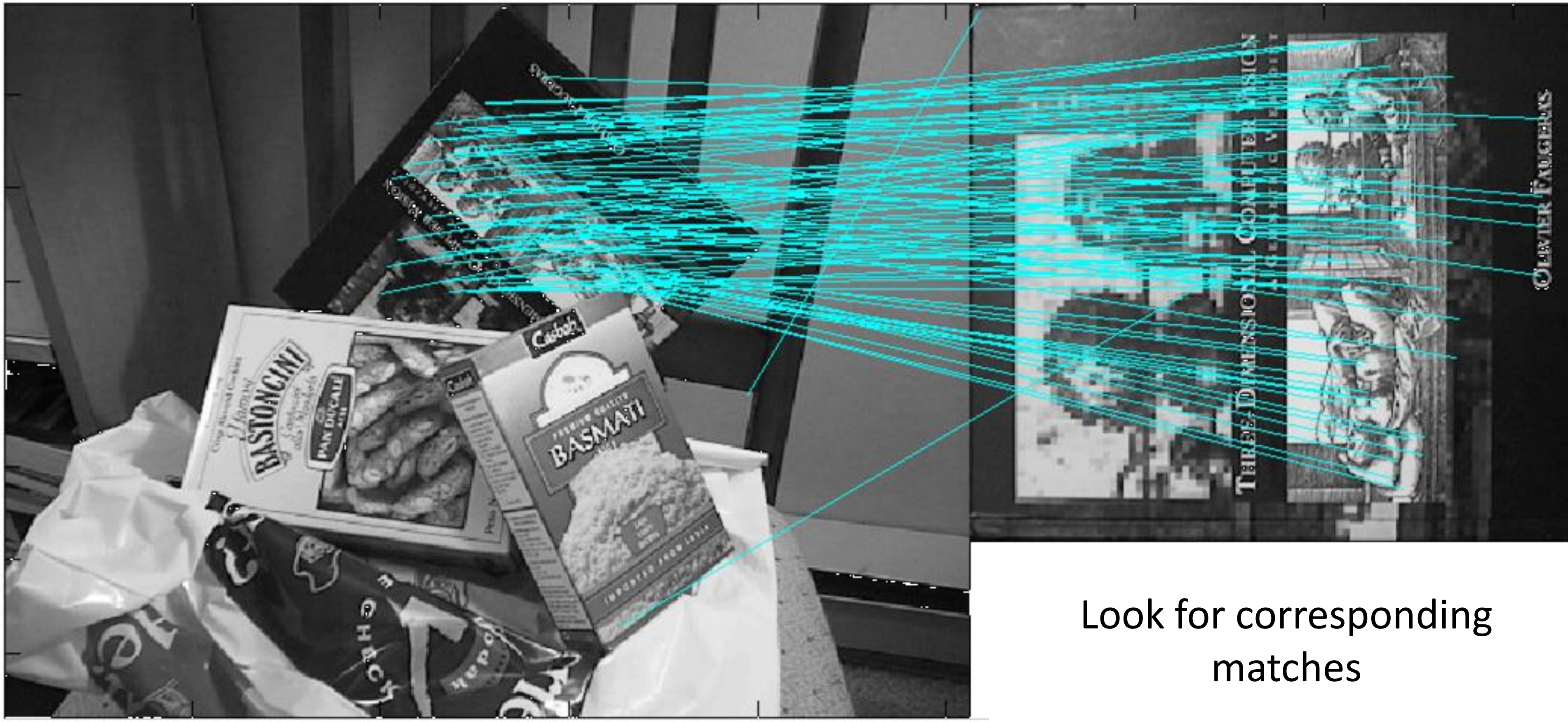
Q: Is this Book present in the Scene?



Extract keypoints in both images

Example of 2D model-based approach

Q: Is this Book present in the Scene?



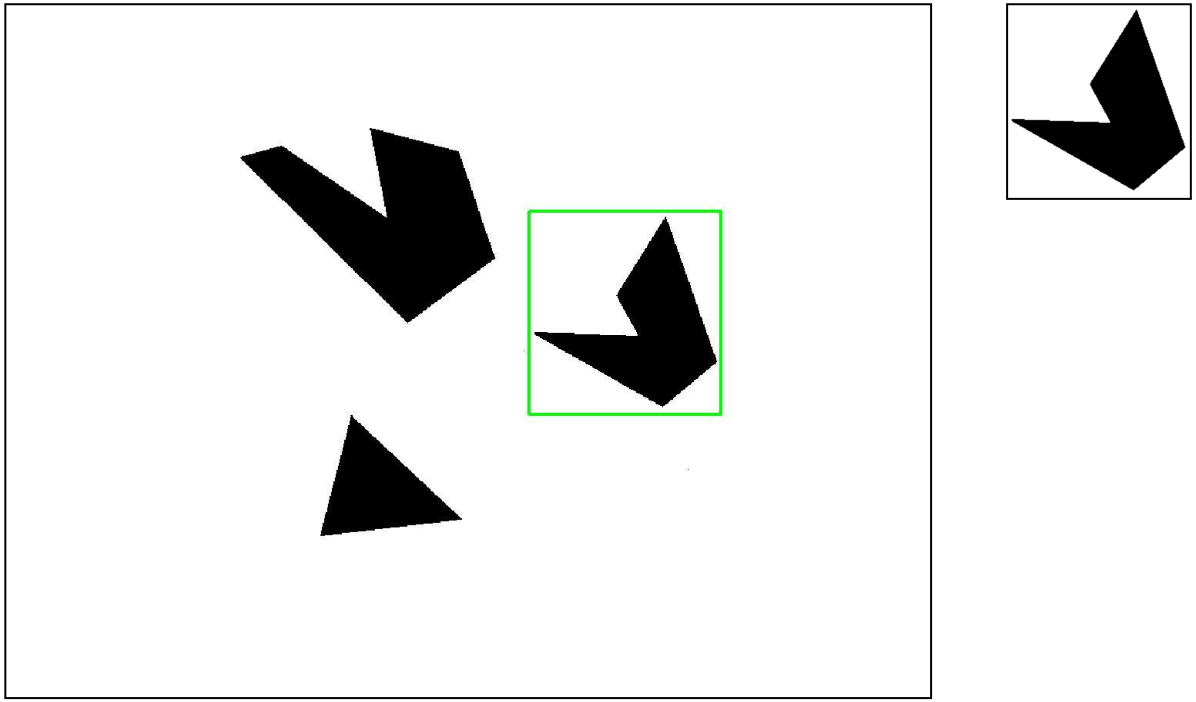
Look for corresponding matches

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

⇨ A: The Book is present in the Scene

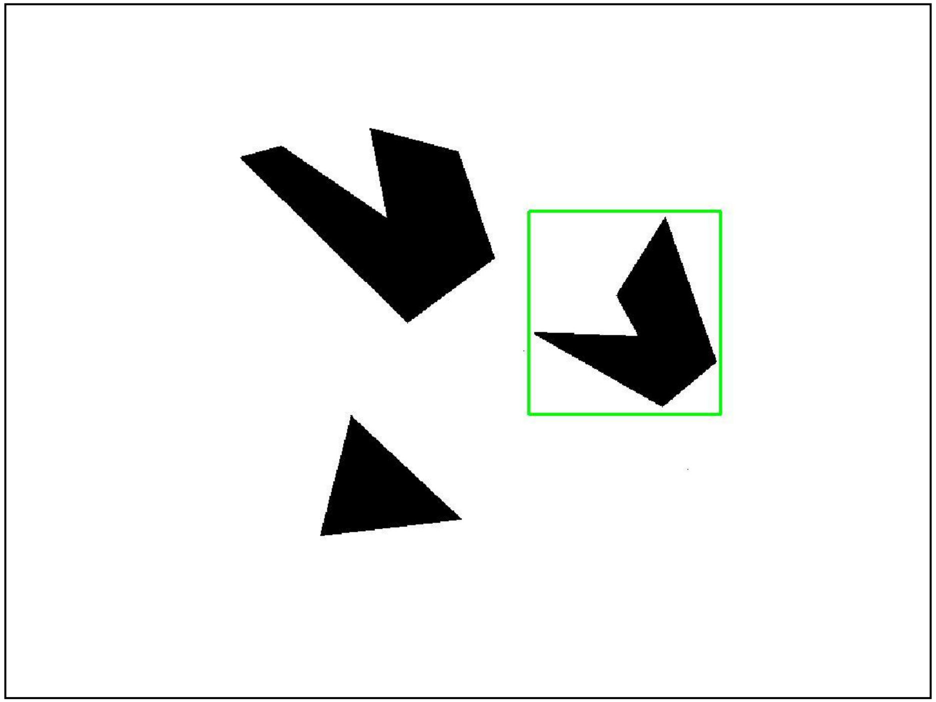
Example of appearance-based approach: Simple 2D template matching

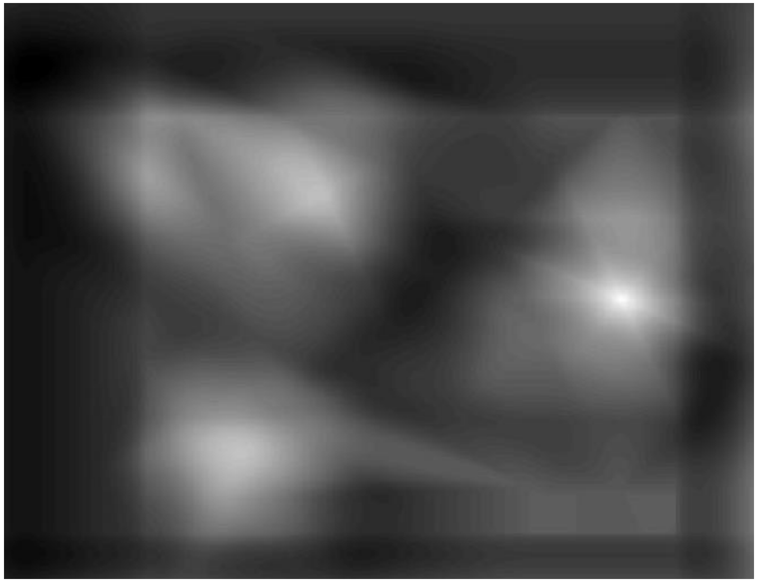
- The model of the object is simply an image
- A simple example: Template matching
  - Shift the template over the image and compare (e.g. NCC or SSD)
  - Problem: works only if template and object are identical



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 challenges
- Recognition approaches
- Classifiers
- K-means clustering
- Bag of words
- Review of the course
- Evaluation of the course

What is the goal of object recognition?

Goal: **classify**!

- Either
  - say yes/no as to whether an object is present in an image
- Or
  - 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 it or is it not a car?

Bounding box on object

Full segmentation

Detection via classification: Main idea

Basic component: a **binary** classifier

Car/non-car Classifier

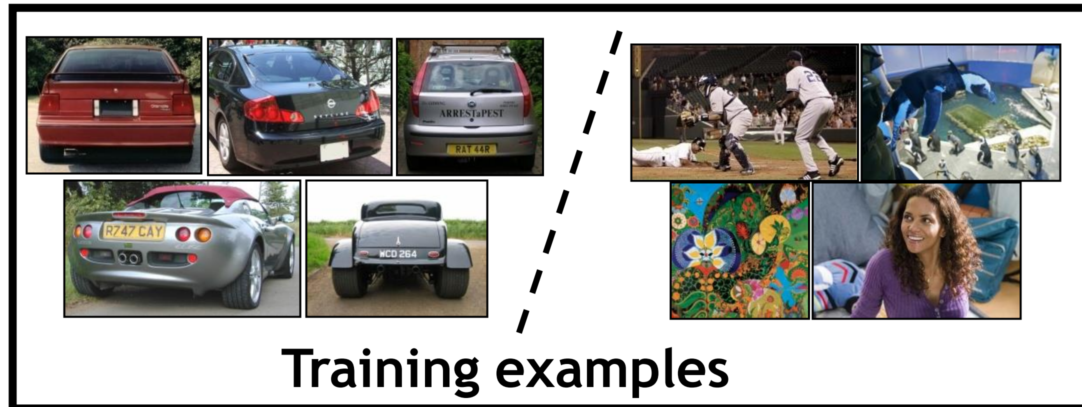
No, not a car.

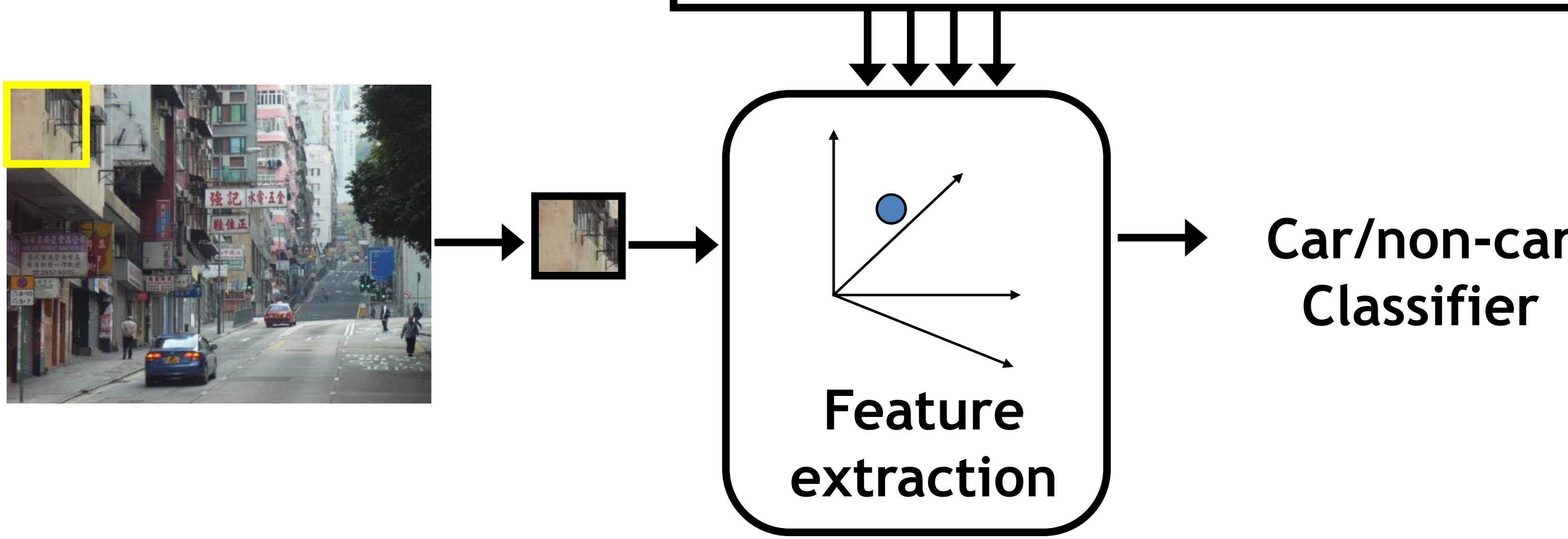
Yes, car.

Detection via classification: Main idea

More in detail, we need to:

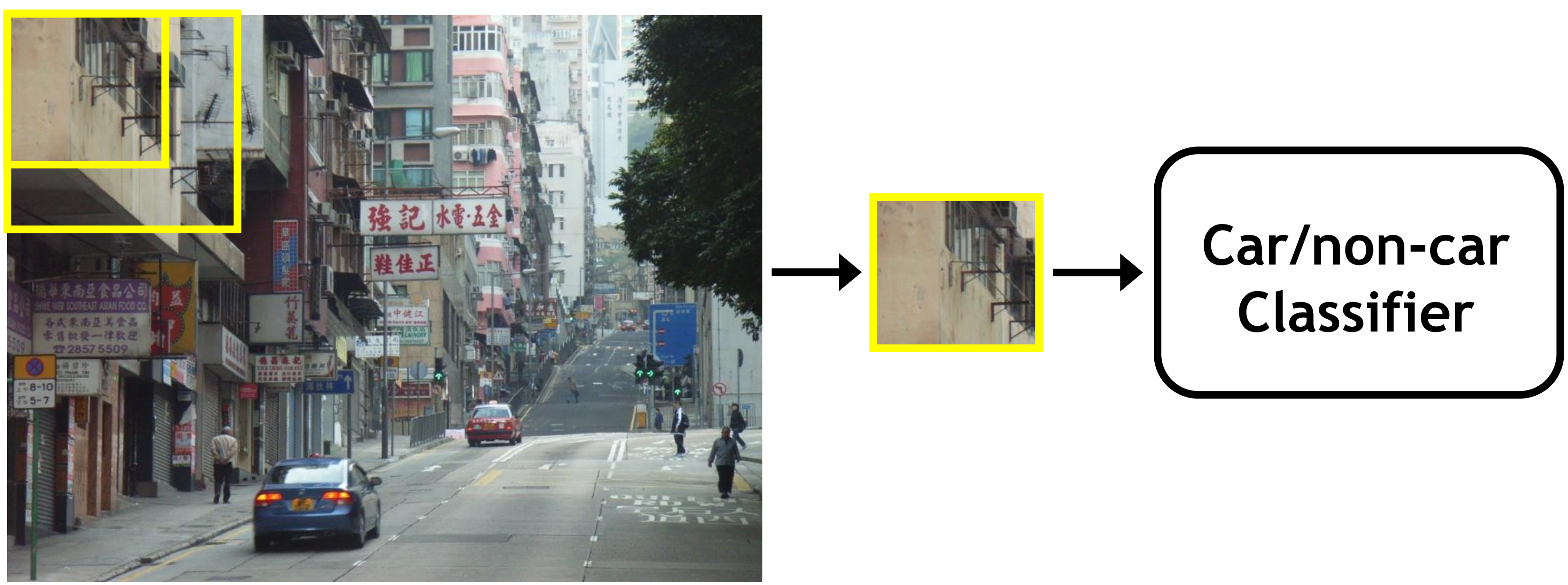
1. Obtain training data
2. Define features
3. Define classifier





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() = “apple”

f() = “tomato”

f() = “cow”

The machine learning framework

$$y = f(\boldsymbol{x})$$

output

prediction function

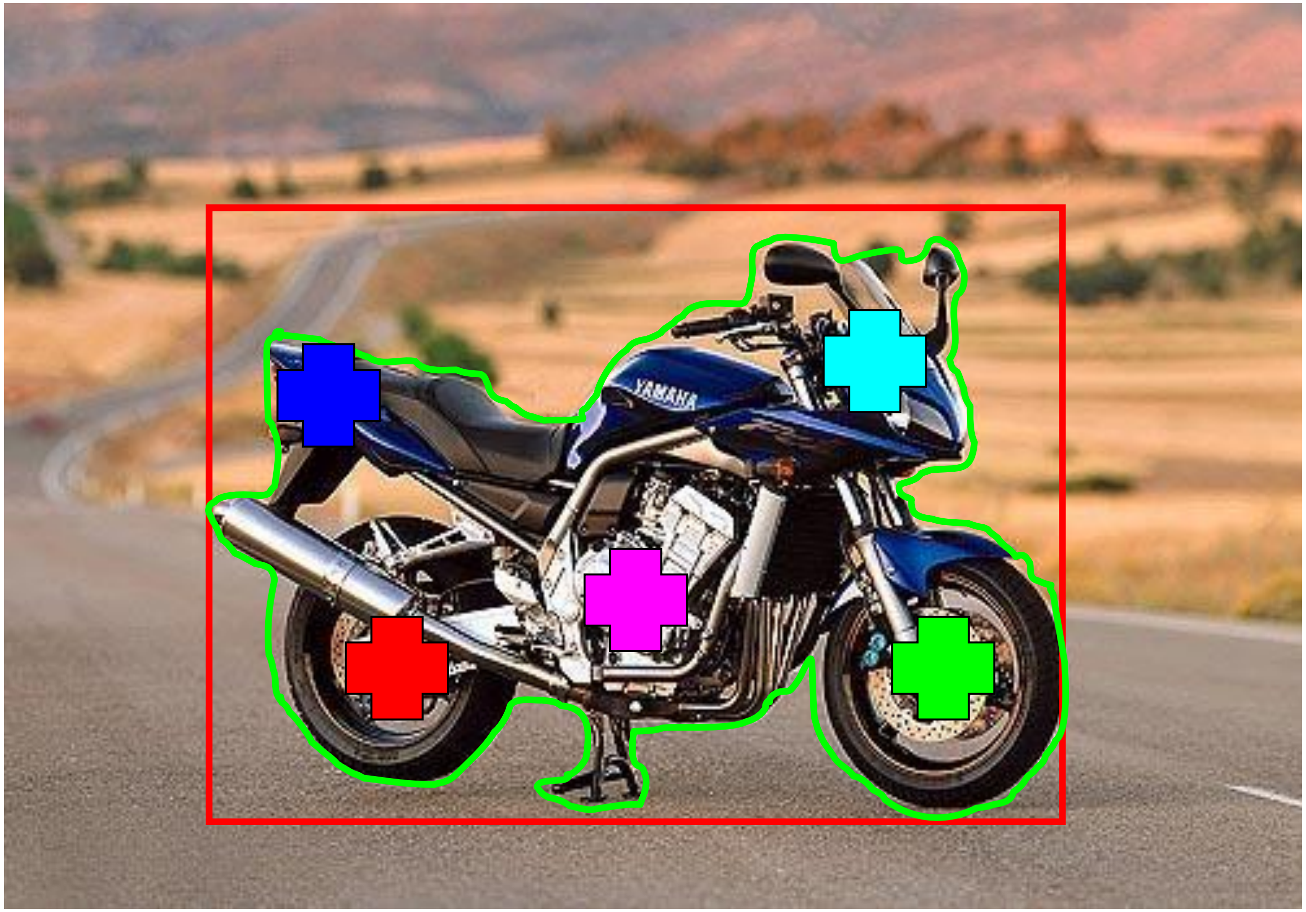
Image feature

- **Training:** given a *training set* of labeled examples $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, yN)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a *never-before-seen test example* $\boldsymbol{x}$ and output the predicted value $y = f(\boldsymbol{x})$

Recognition task and supervision

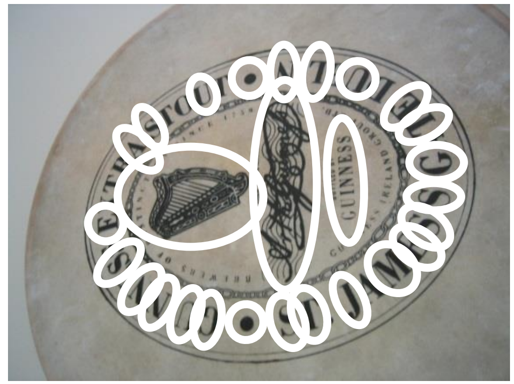
- Images in the training set must be *annotated* with the "correct answer" that the model is expected to produce

Contains a motorbike

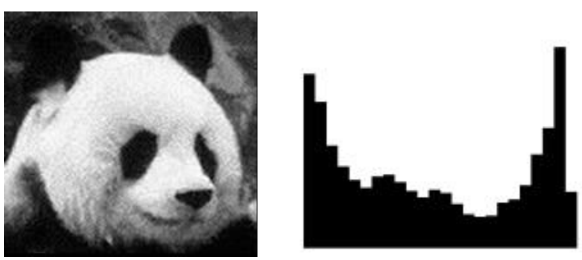


Examples of possible features

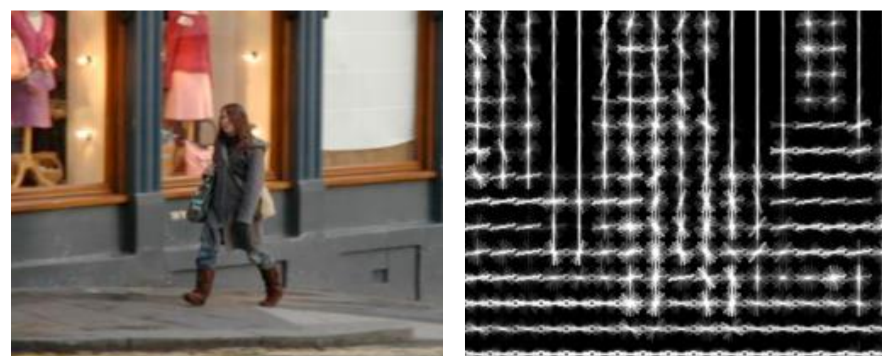
- Blob features



- Image Histograms



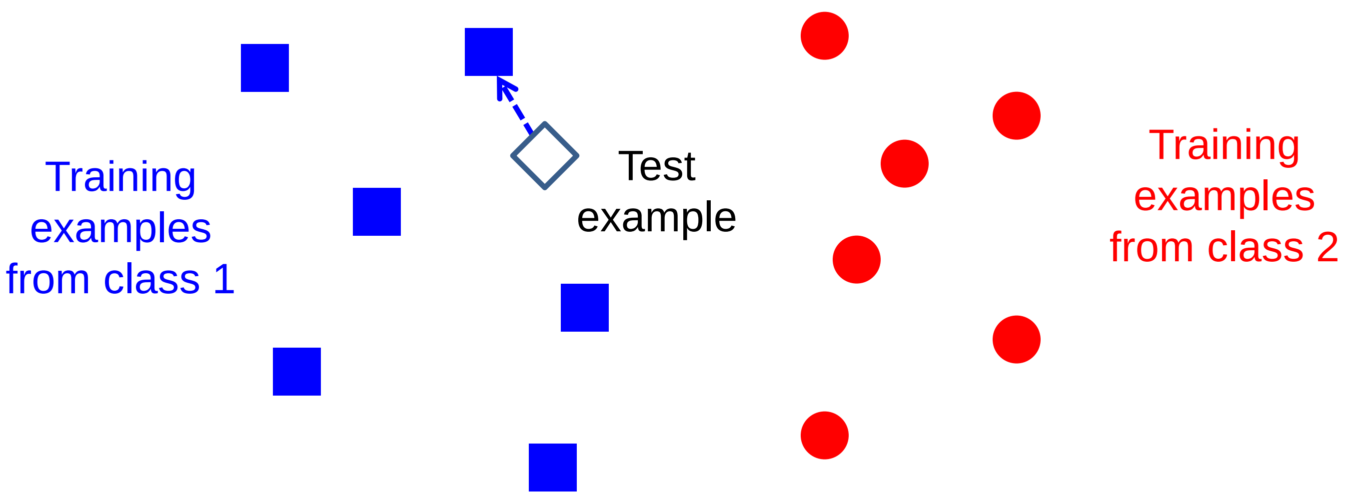
- Histograms of oriented gradients (HOG)



Classifiers: Nearest neighbor

Features are represented in the descriptor space

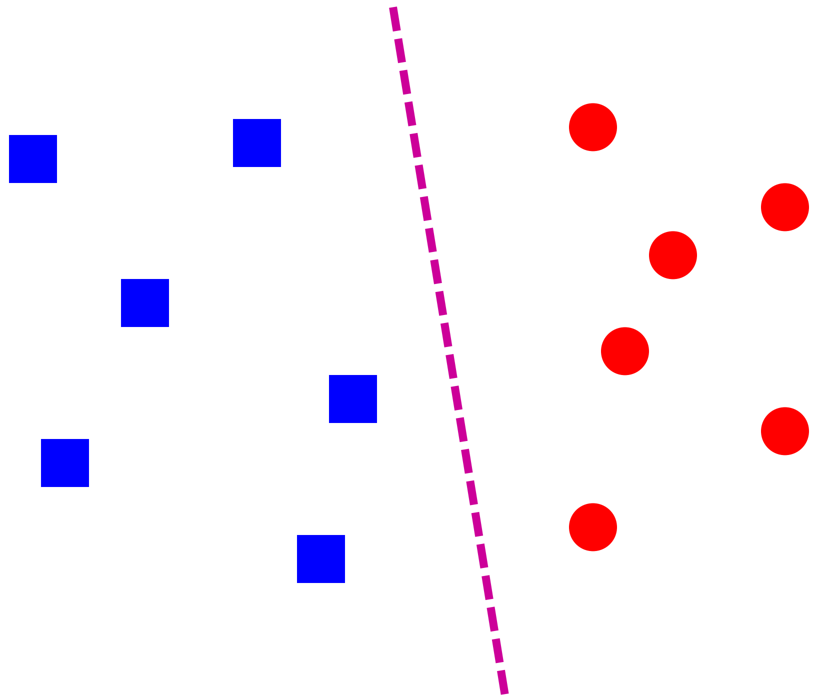
(Ex. What is the dimensionality of the descriptor space for SIFT features?)



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

- **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

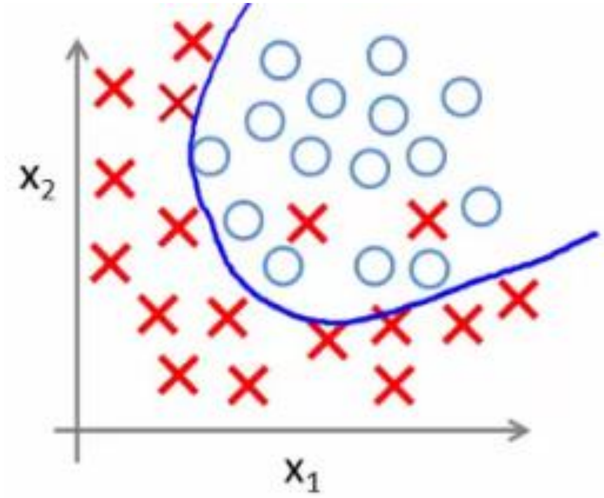


- Find a *linear function* to separate the classes:

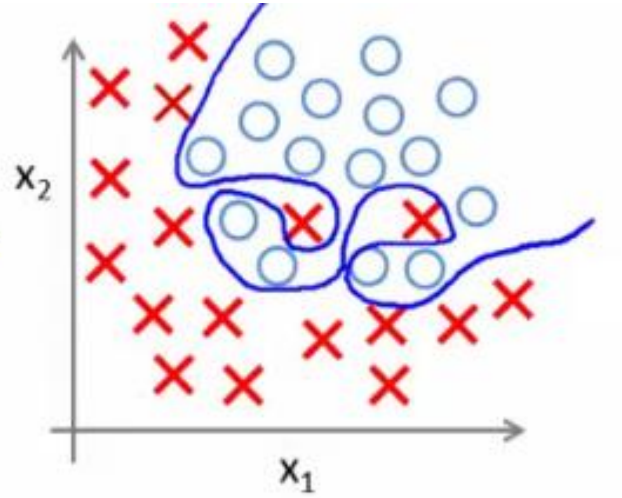
$$f(\boldsymbol{x}) = sgn(\boldsymbol{w} \cdot \boldsymbol{x} + b)$$

Classifiers: non-linear

Good classifier



Bad classifier (over fitting)

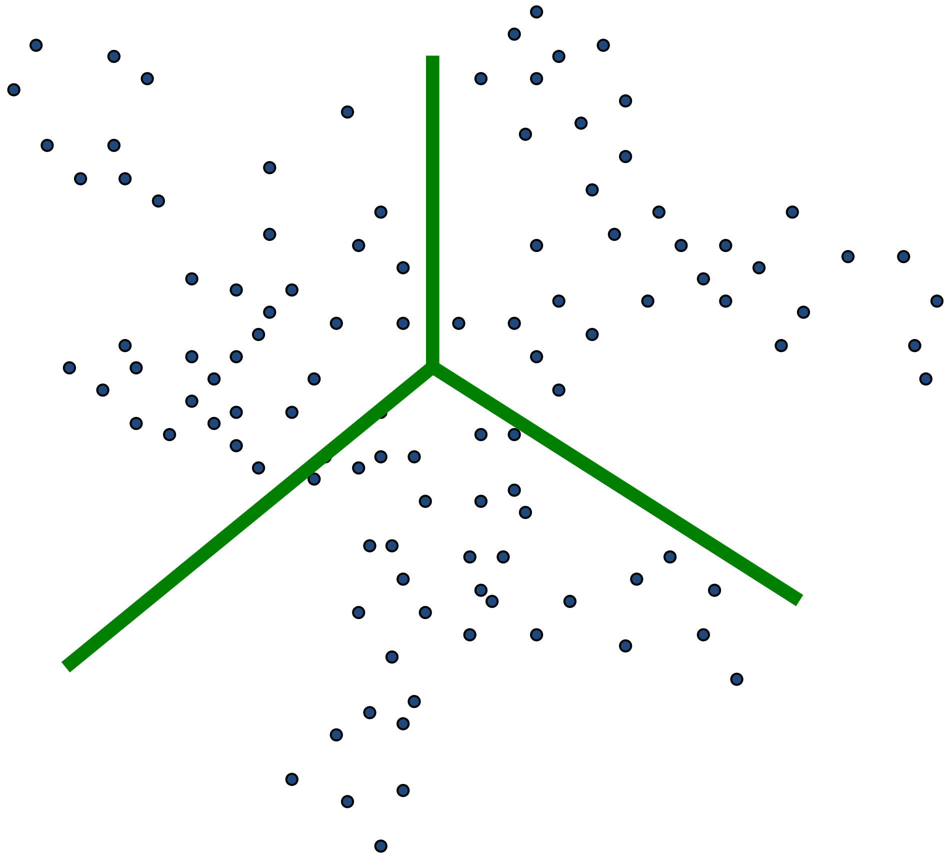


Outline

- Recognition challenges
- Recognition approaches
- Classifiers
- K-means clustering
- Bag of words
- Review of the course
- Evaluation of the course

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



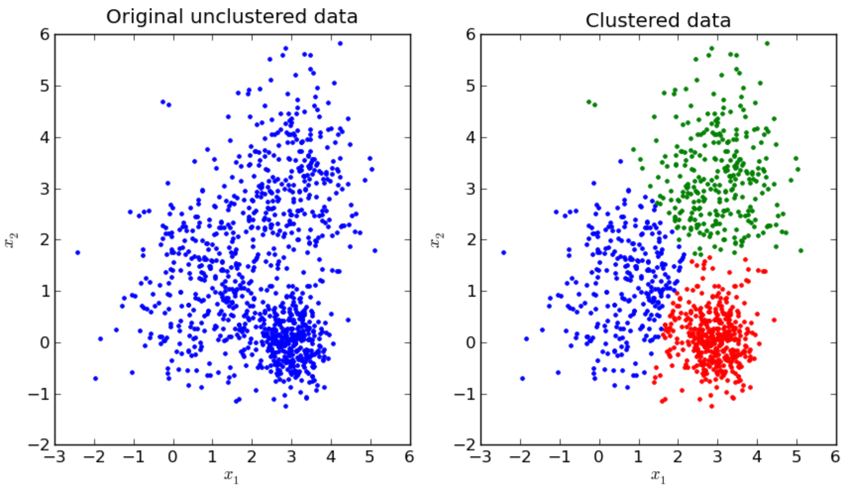
K-means clustering

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

$$D(X,M) = \sum_{i=1}^{k} \sum_{j=1}^{n} (x_j - m_i)^2$$

Algorithm:

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



K-means demo

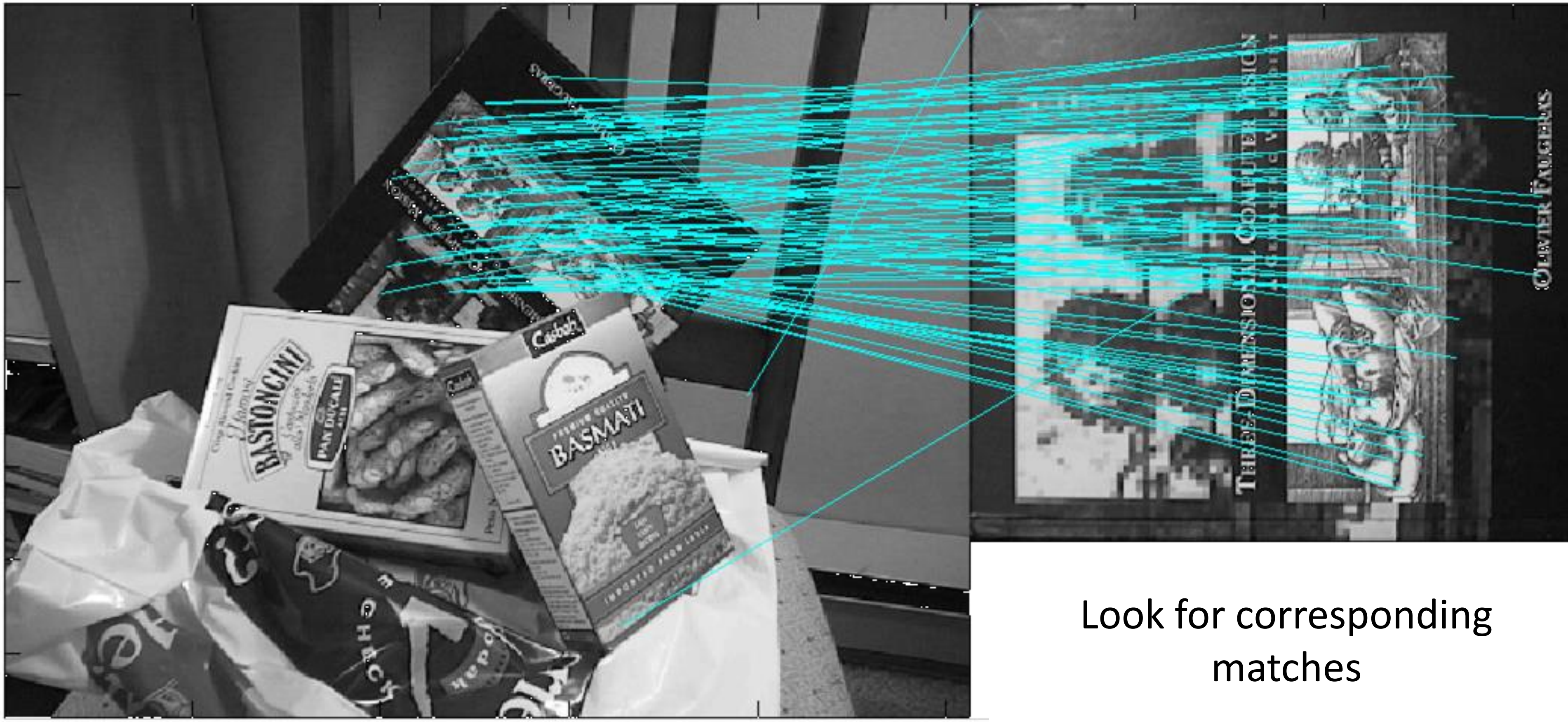
Source: http://shabal.in/visuals/kmeans/1.html

Outline

- Recognition challenges
- Recognition approaches
- Classifiers
- K-means clustering
- Bag of words
- Review of the course
- Evaluation of the course

Review: Feature-based object recognition

Q: Is this Book present in the Scene?



Look for corresponding matches

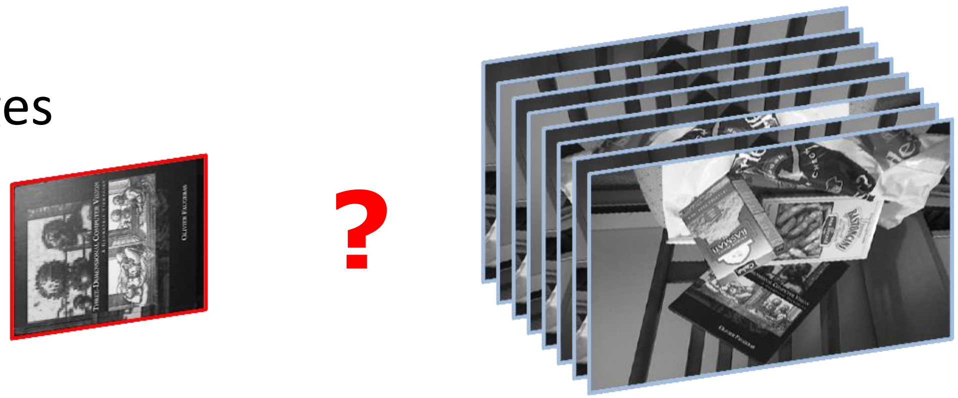
Most of the Book’s keypoints are present in the Scene

⇨ A: The Book is present in the Scene

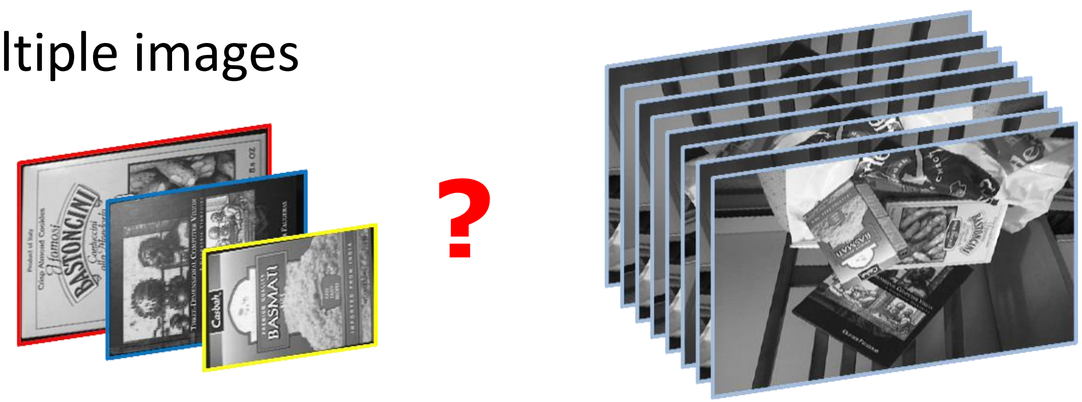
Taking this a step further…

- Find an object in an image

- Find an object in multiple images



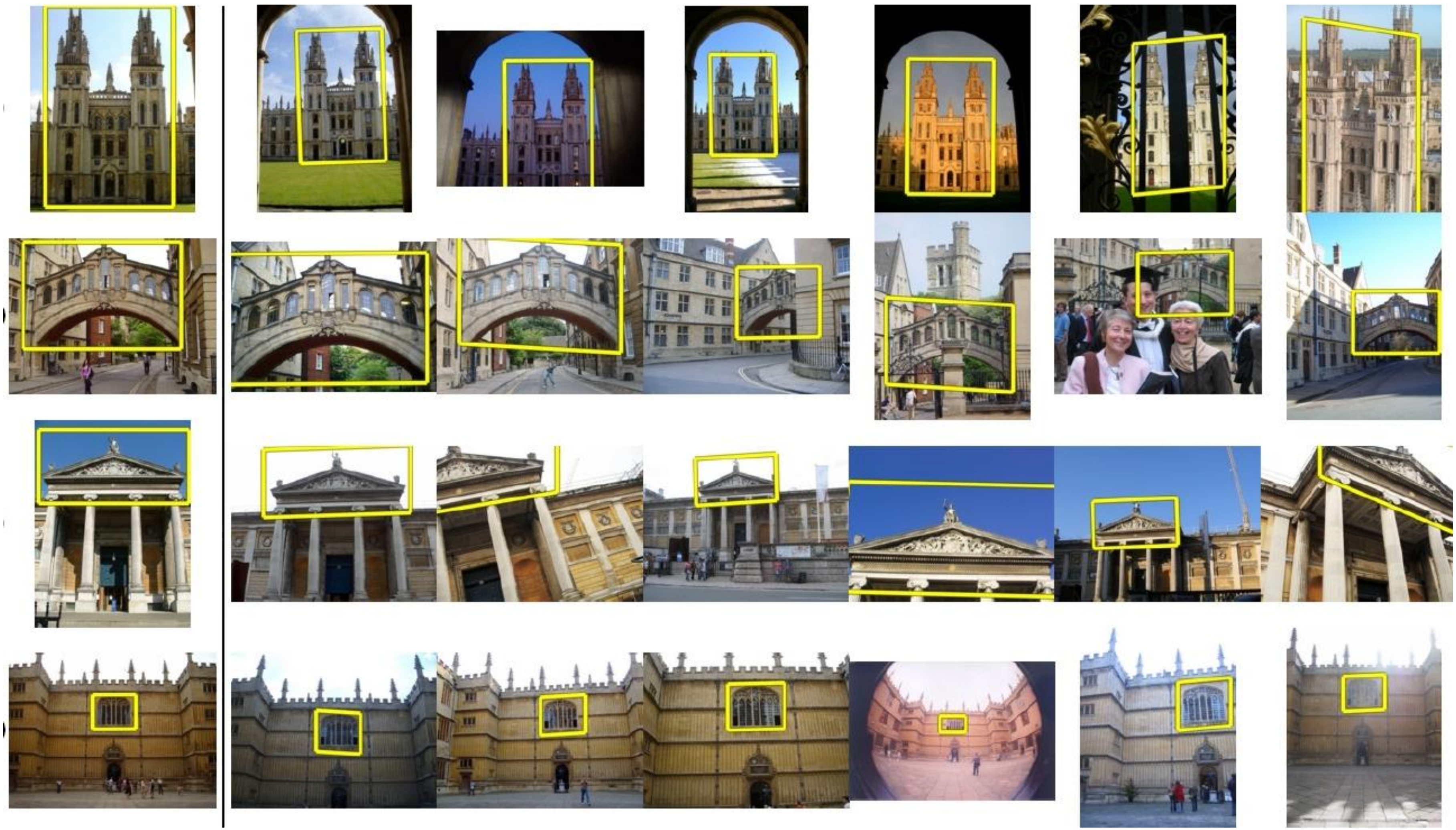
- Find multiple objects in multiple images

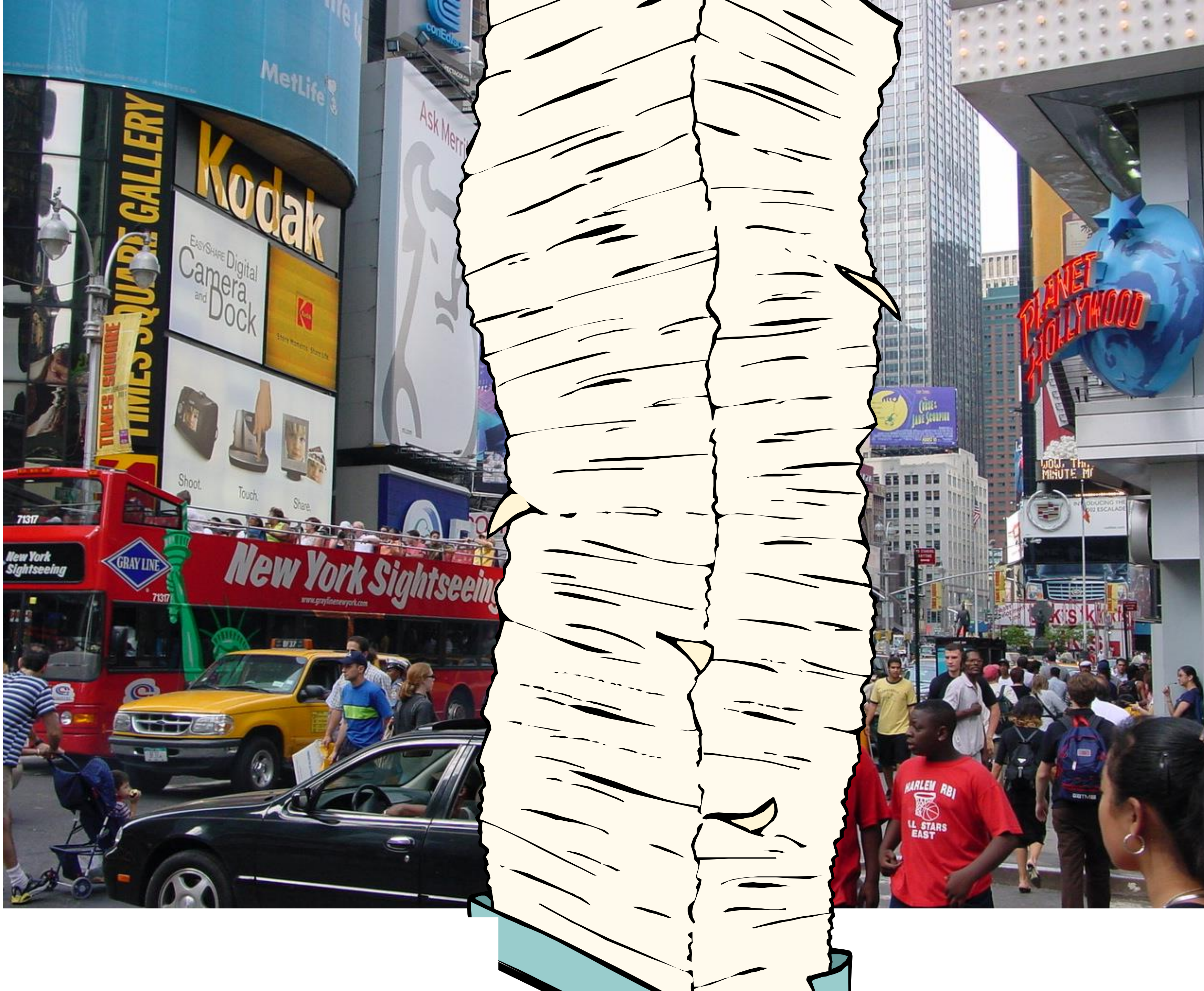


As the number of images increases, feature-based object recognition becomes computationaly more and more expensive

Application: large-scale image retrieval

Query image Results on a database of 100 million images




MetLife
Kodak
TIMES SQUARE GALLERY
EasyShare Digital Camera and Dock
Shoot.
Touch.
Share.
New York Sightseeing
GRAY LINE
New York Sightseeing
PLANET HOLLYWOOD
HARLEM RBI
ALL STARS
EAST

Slide Credit: Nister

Fast visual search

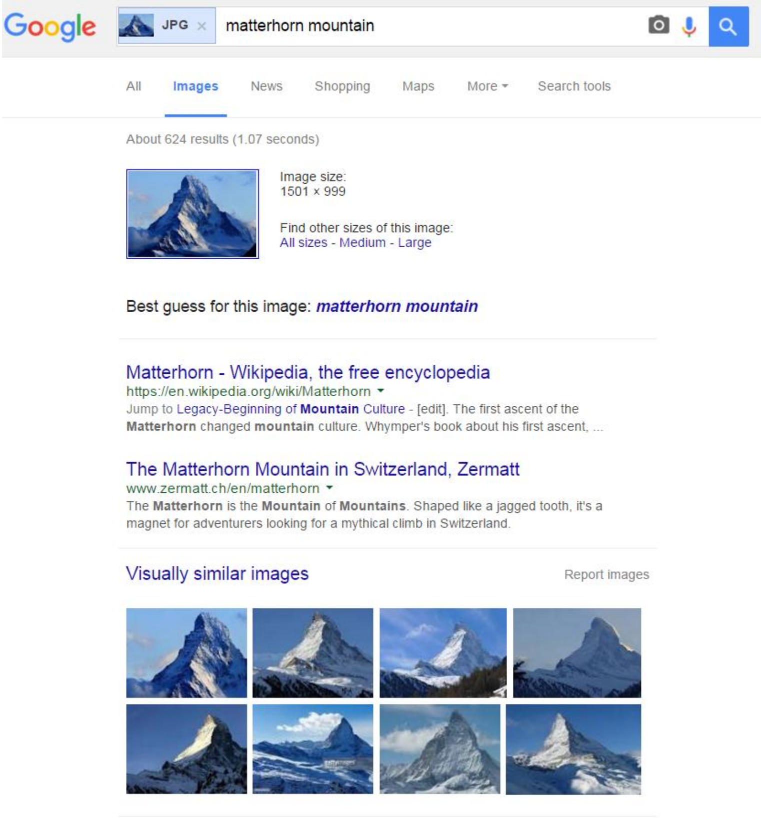
- Query in a database of 100 million images in 6 seconds

"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?



Visual Place Recognition

- **Goal**: find the most similar images of a **query** image in a database of $\boldsymbol{N}$ **images**
- **Complexity:** $\frac{N^2 \cdot M^2}{2}$ feature comparisons (*worst-case* scenario)
  - Each image must be compared with all other images!
  - N is the number of all images collected by a robot
    - Example: 1 image per meter of travelled distance over a $100m^2$ house with one robot and 100 feature per image → $\mathrm{M} = 100,\ N = 100 \rightarrow N^2M^2/2$= $\sim 50\ Million$ feature comparisons!

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

[“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

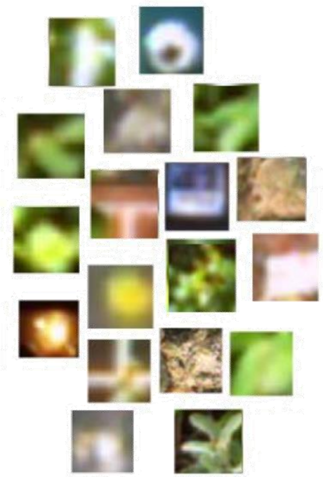
- For text documents, an efficient way to find all *pages* on which a *word* occurs is to use an index
- We want to find all *images* in which a *feature* occurs
- How many distinct SIFT or BRISK features exist?
  - SIFT → Infinite
  - BRISK-128 → $2^{128} = 3.4 \cdot 10^{38}$
- Since the number of image 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

Building the Visual Vocabulary

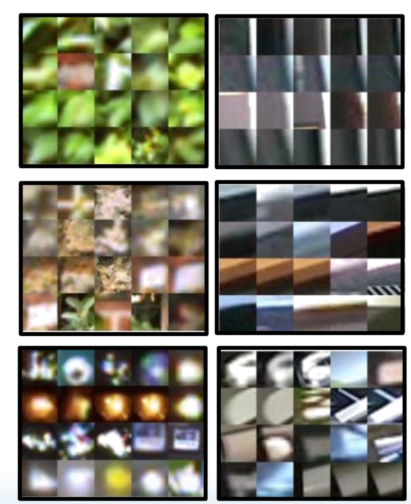
Image Collection

Extract Features



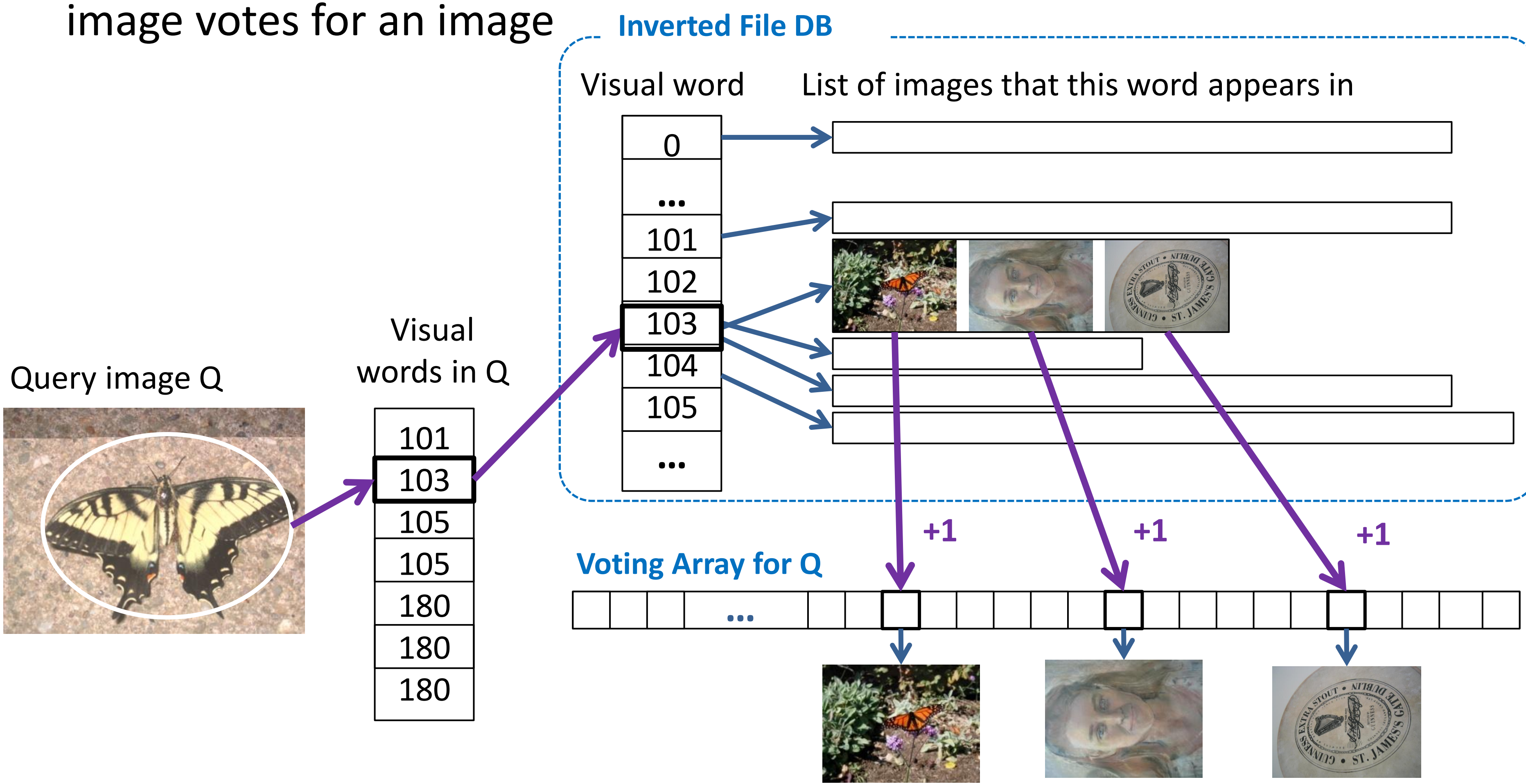
Cluster Descriptors

Examples of Visual Words:

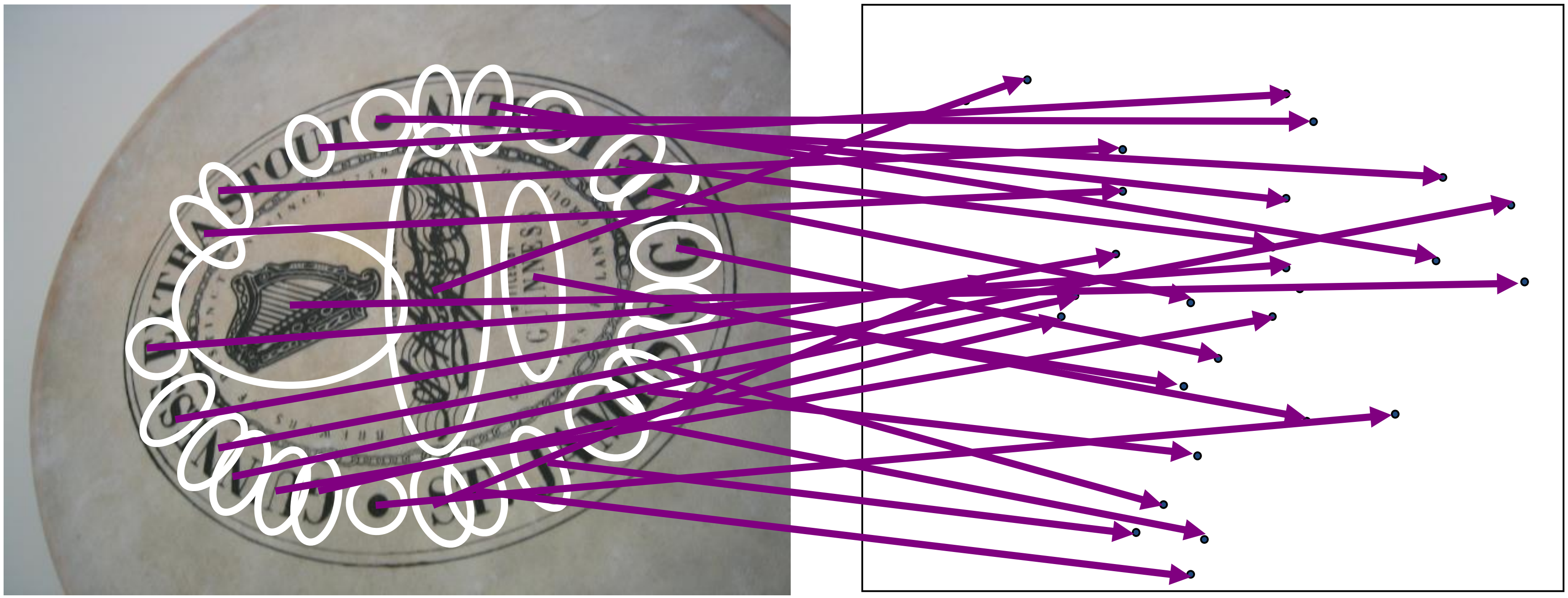


Inverted File index

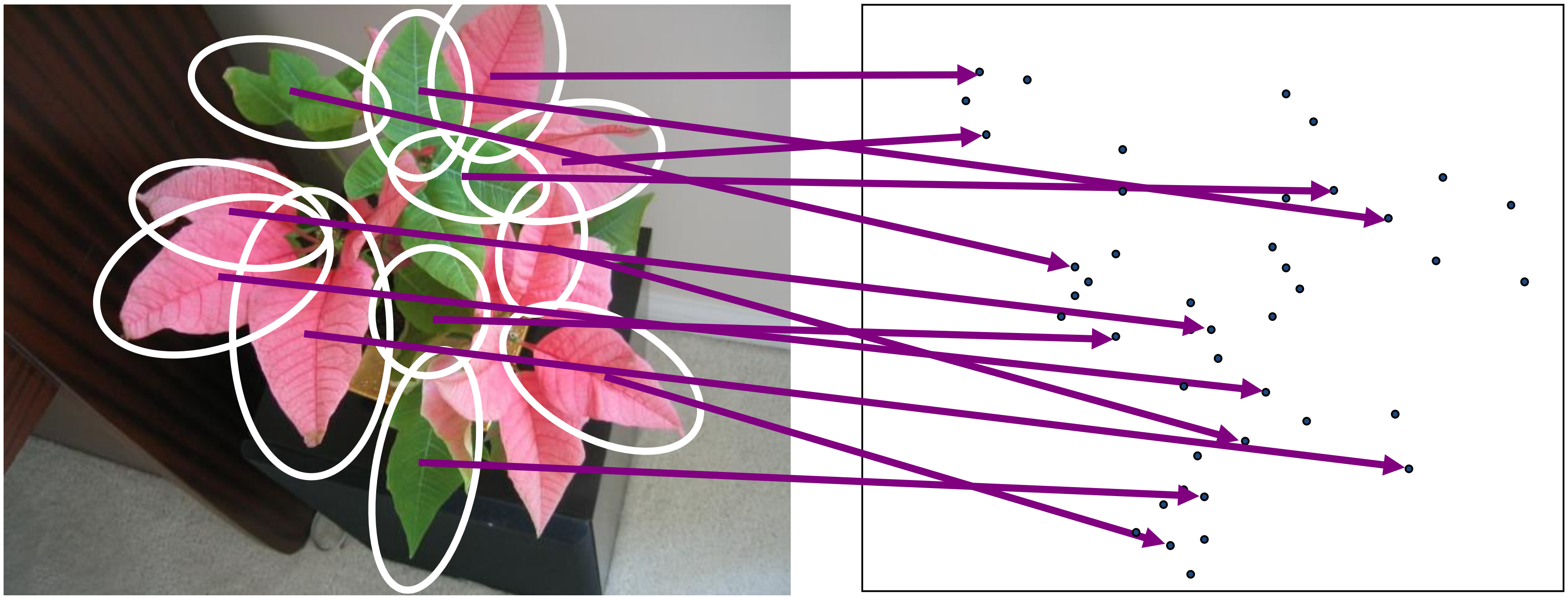
- Inverted File Data Base (DB) lists all possible visual words
- Each word points to a list of images where this word occurs
- Voting array: has as many cells as images in the DB – each word in query image votes for an image



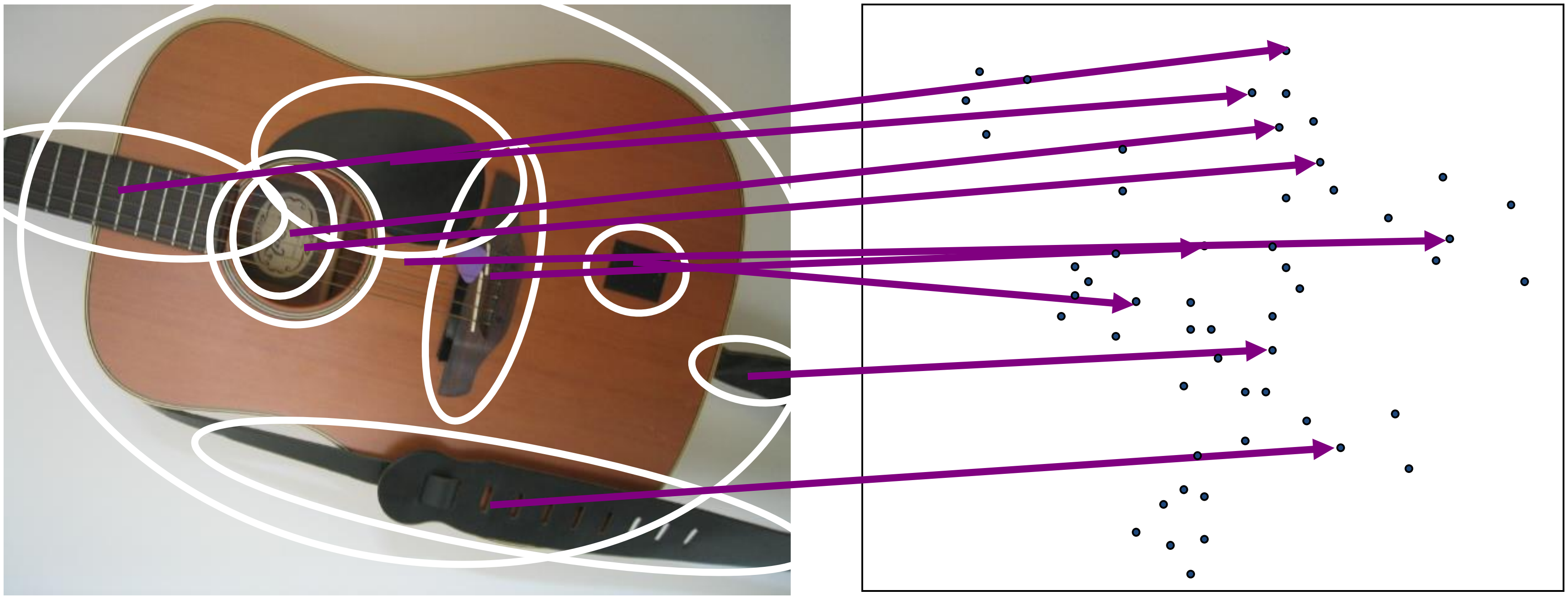
Populating the vocabulary



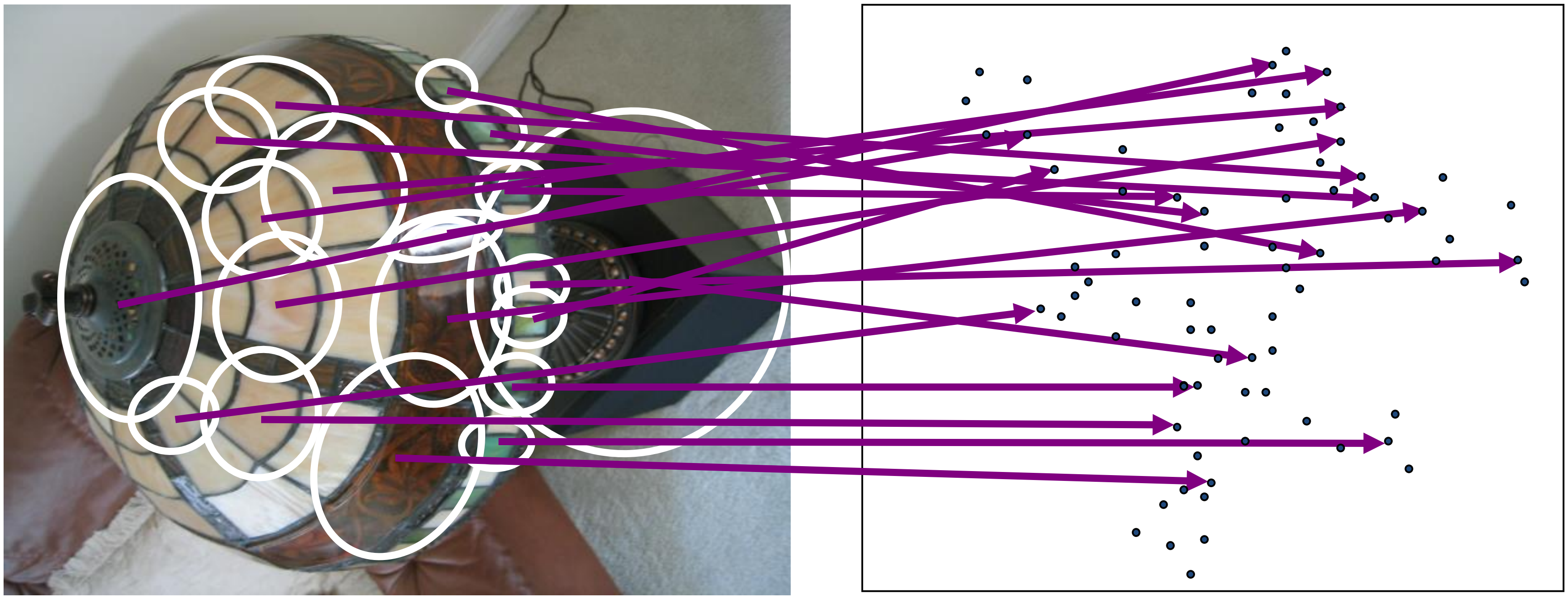
Populating the vocabulary



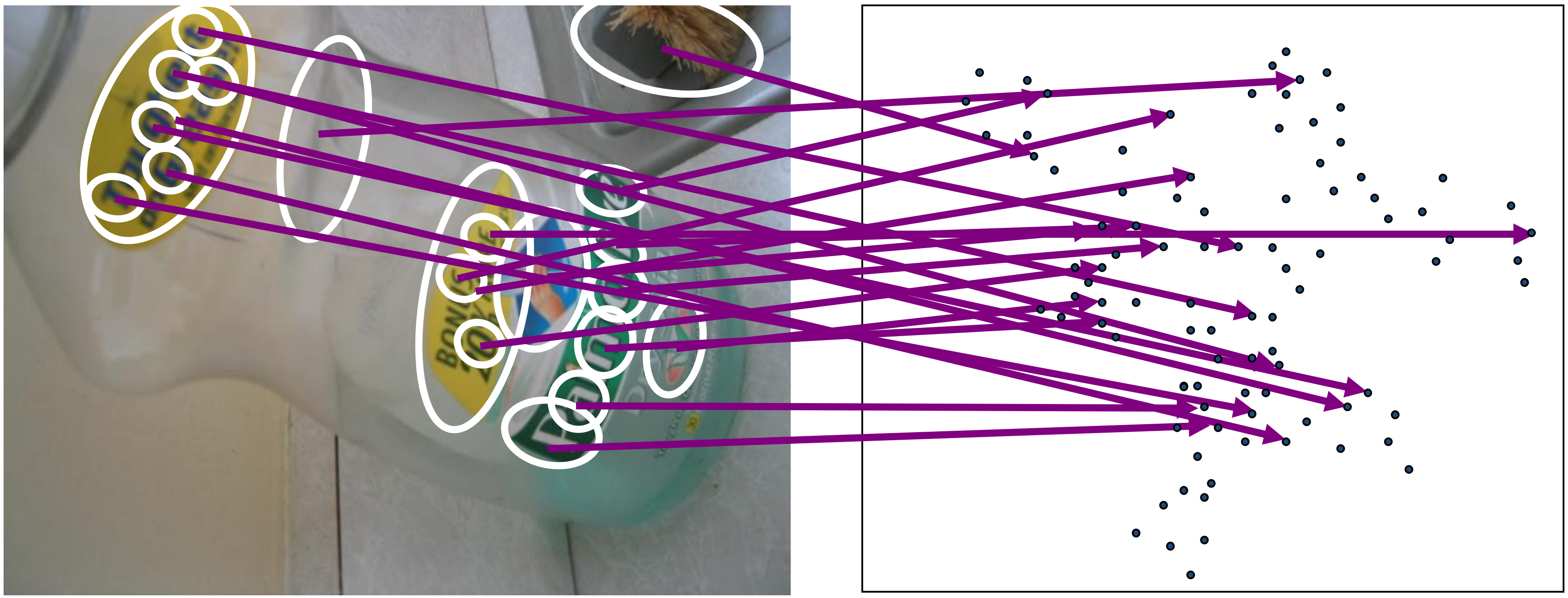
Populating the vocabulary



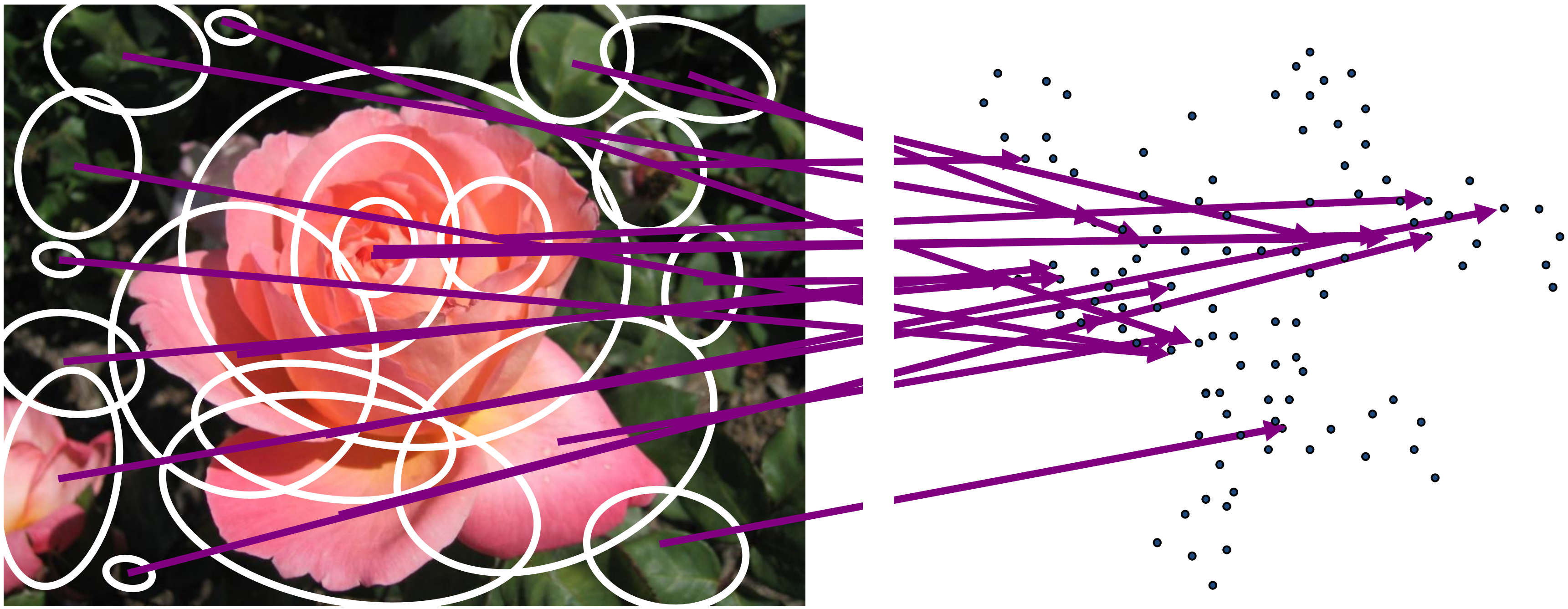
Populating the vocabulary

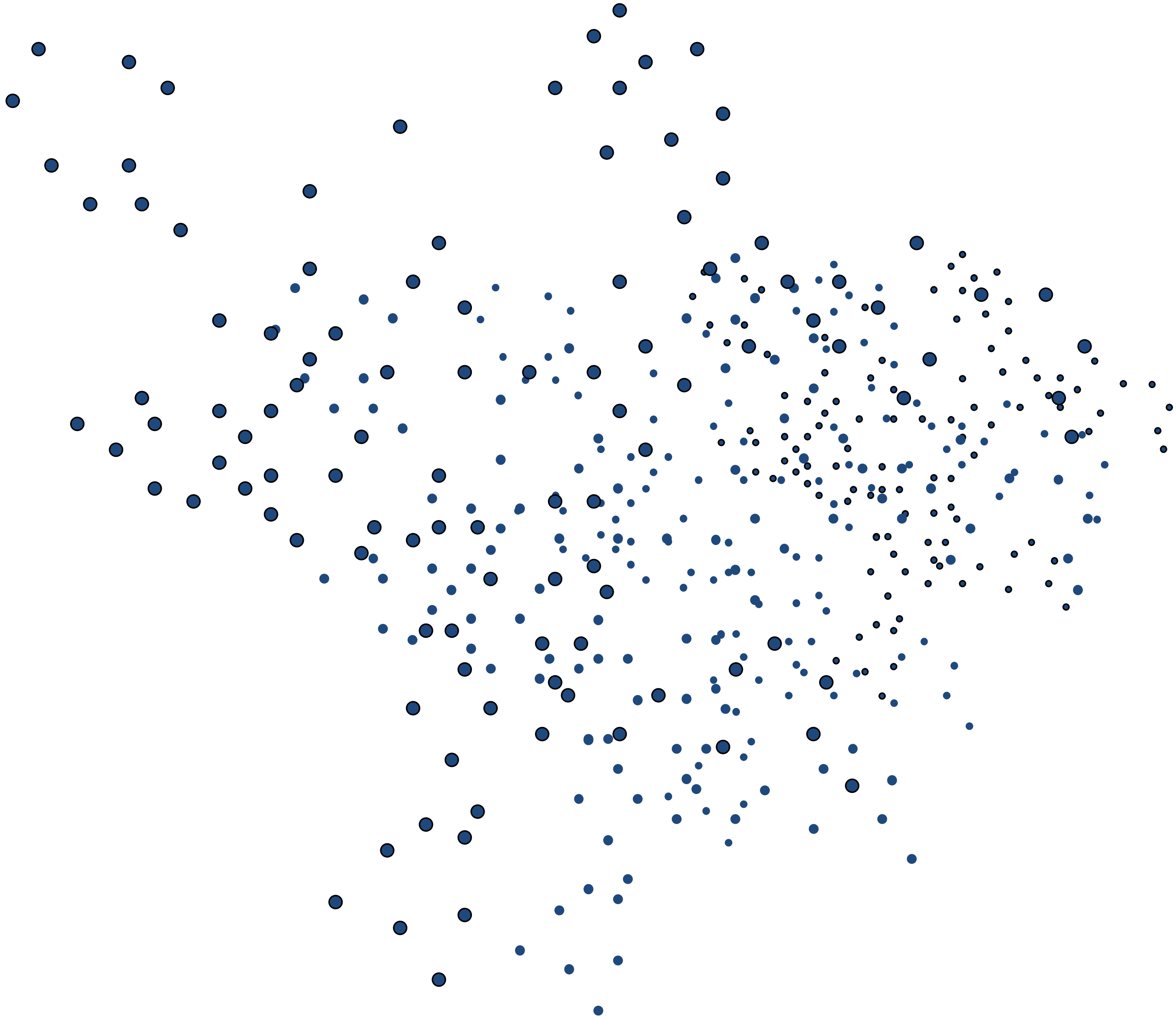


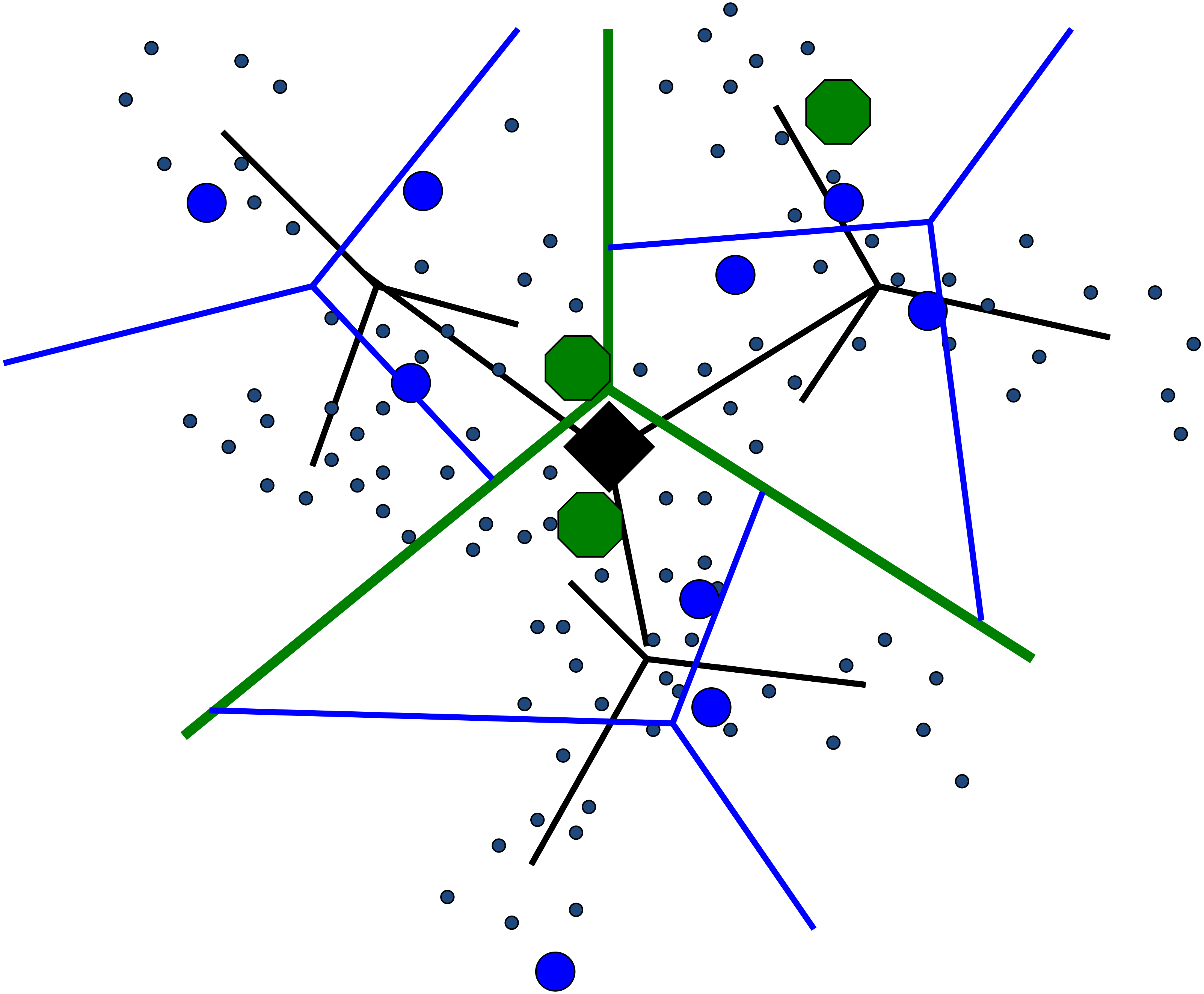
Populating the vocabulary

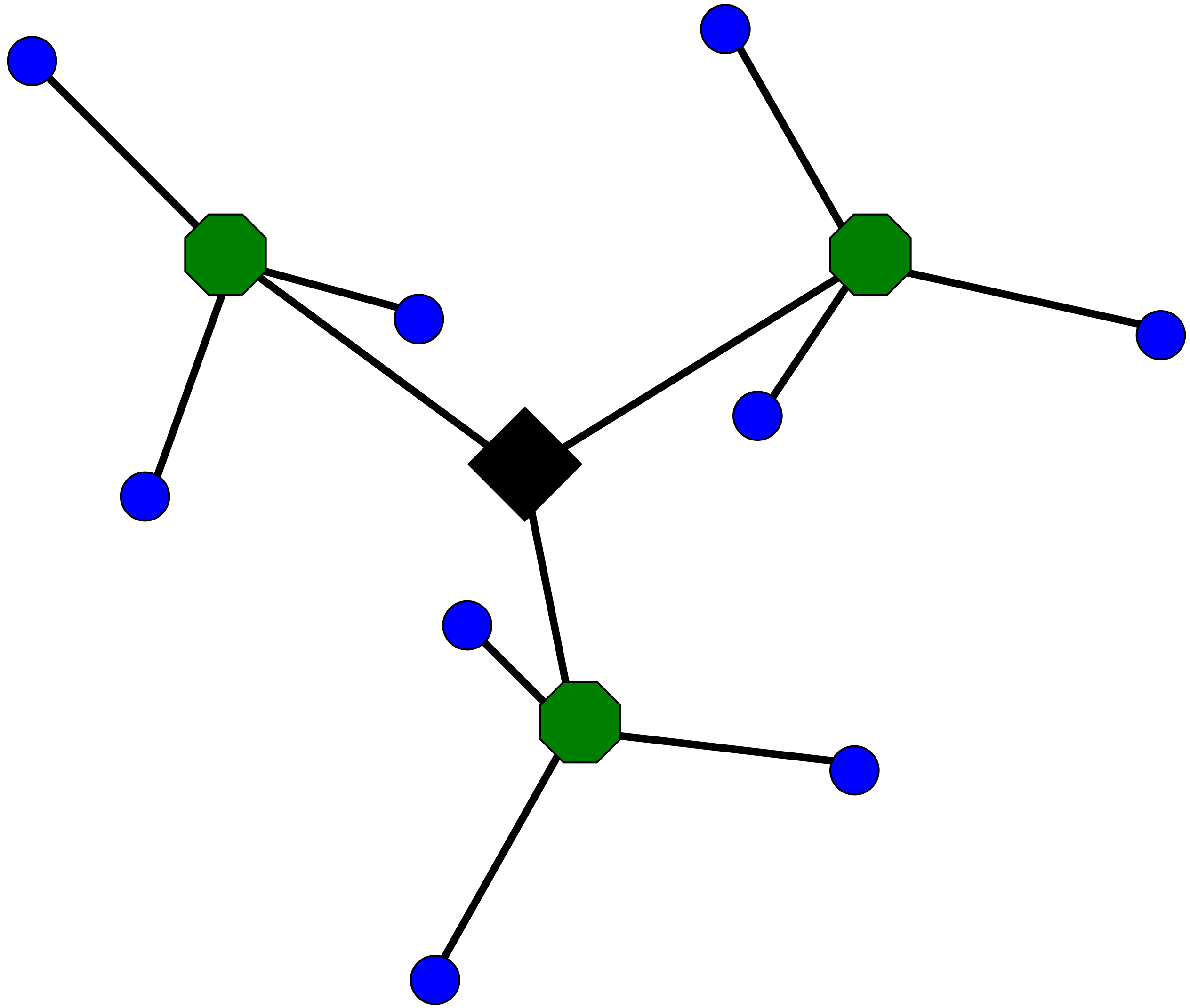


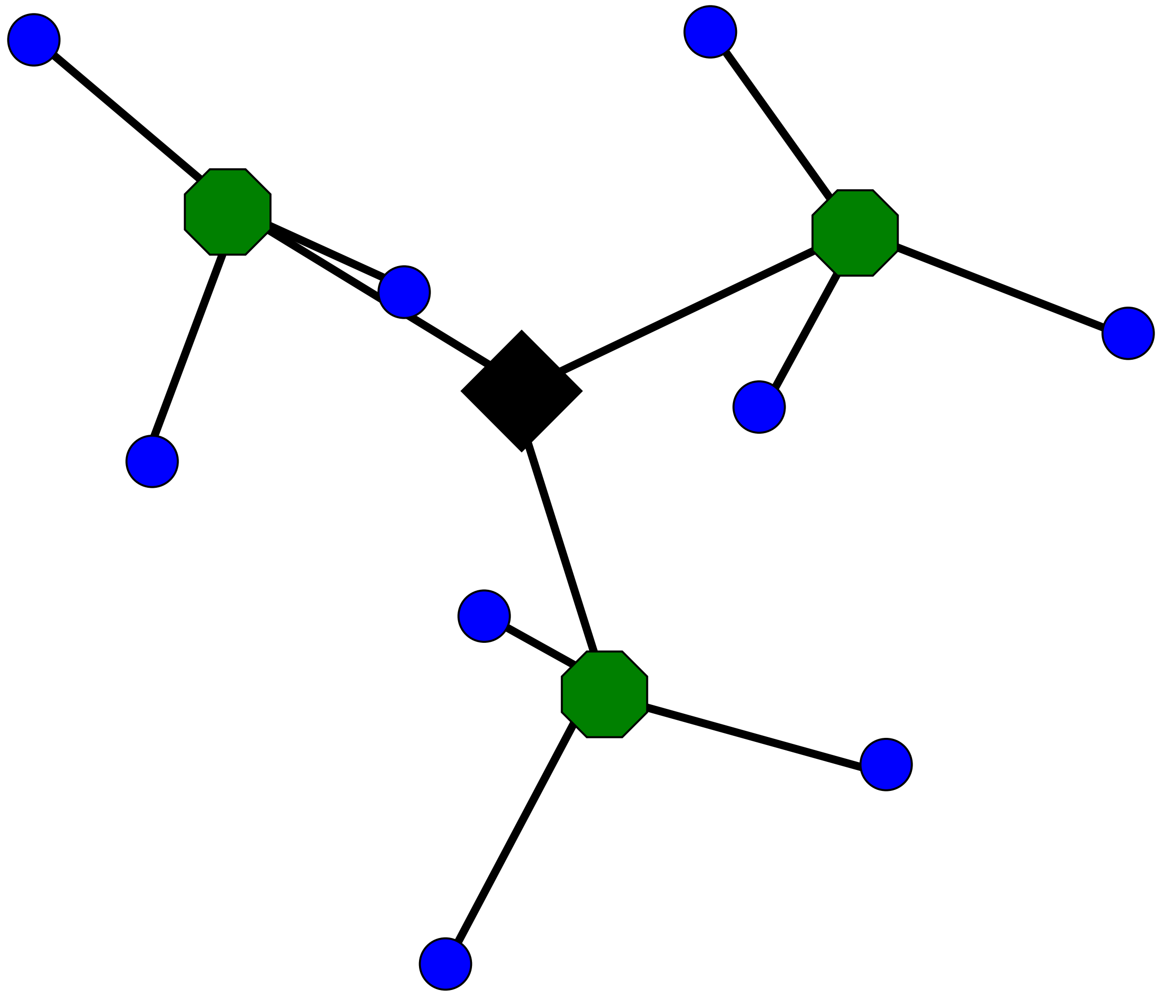
Populating the vocabulary

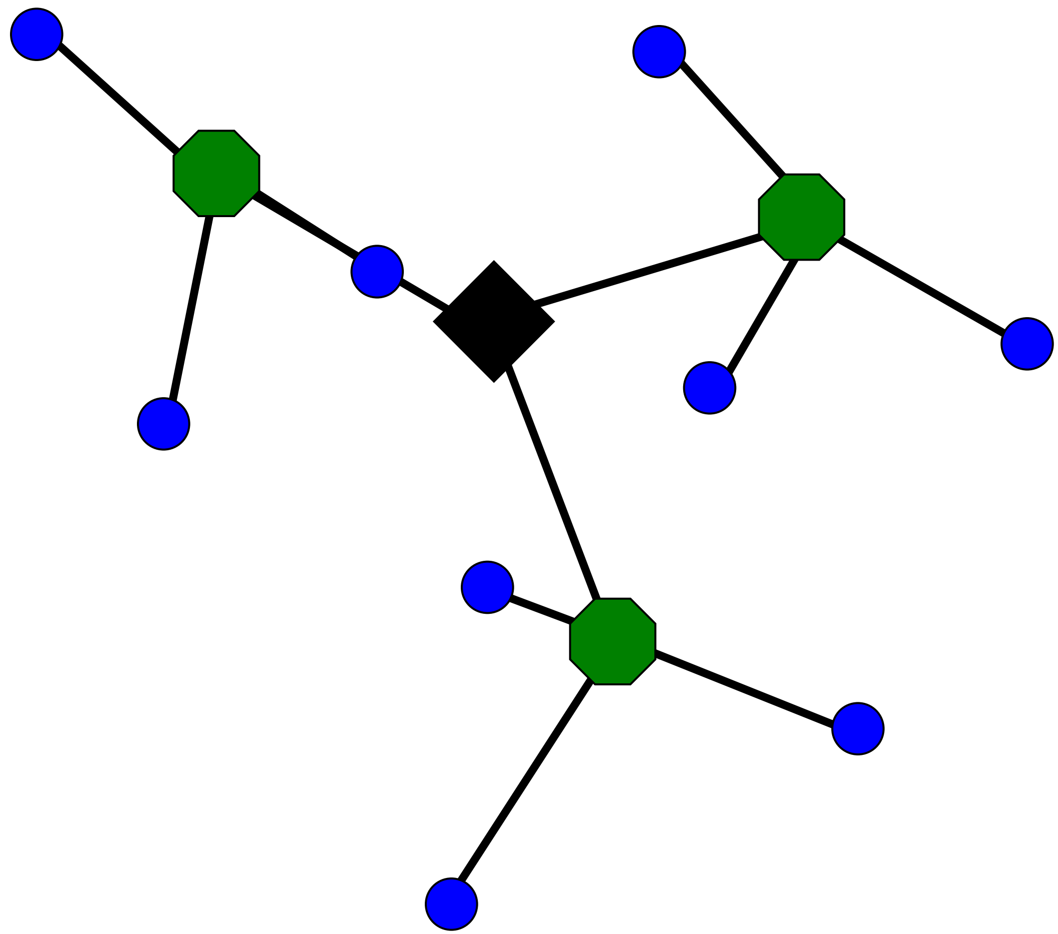


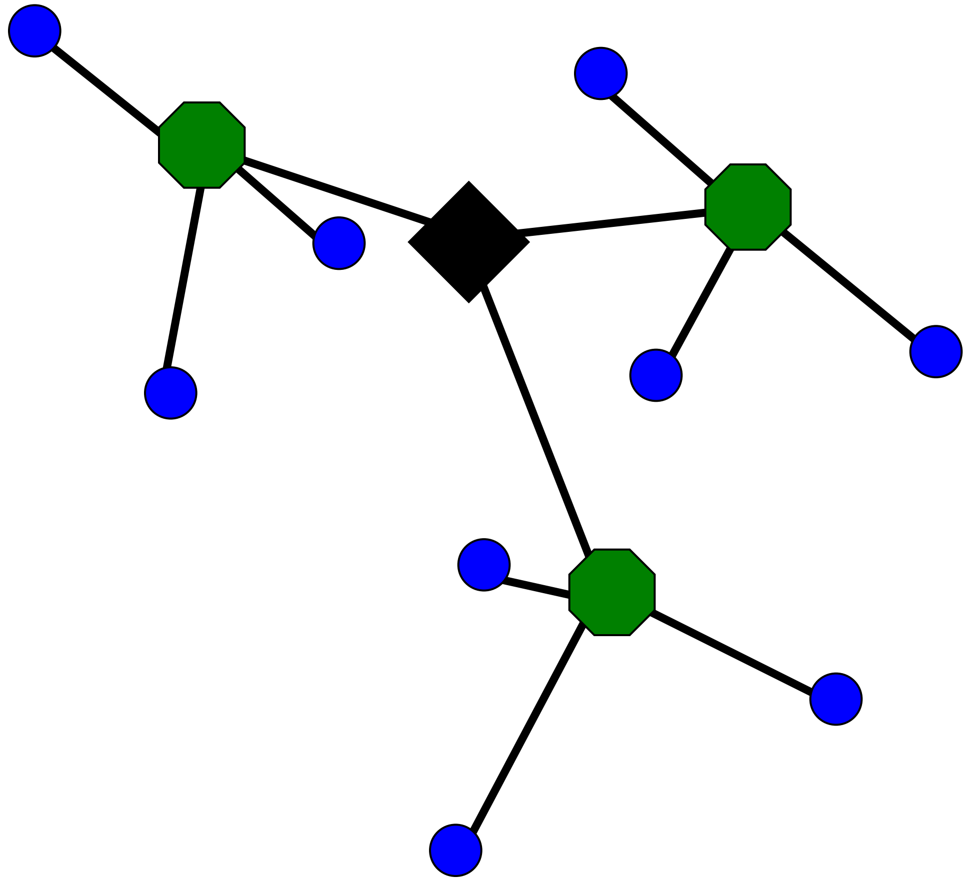


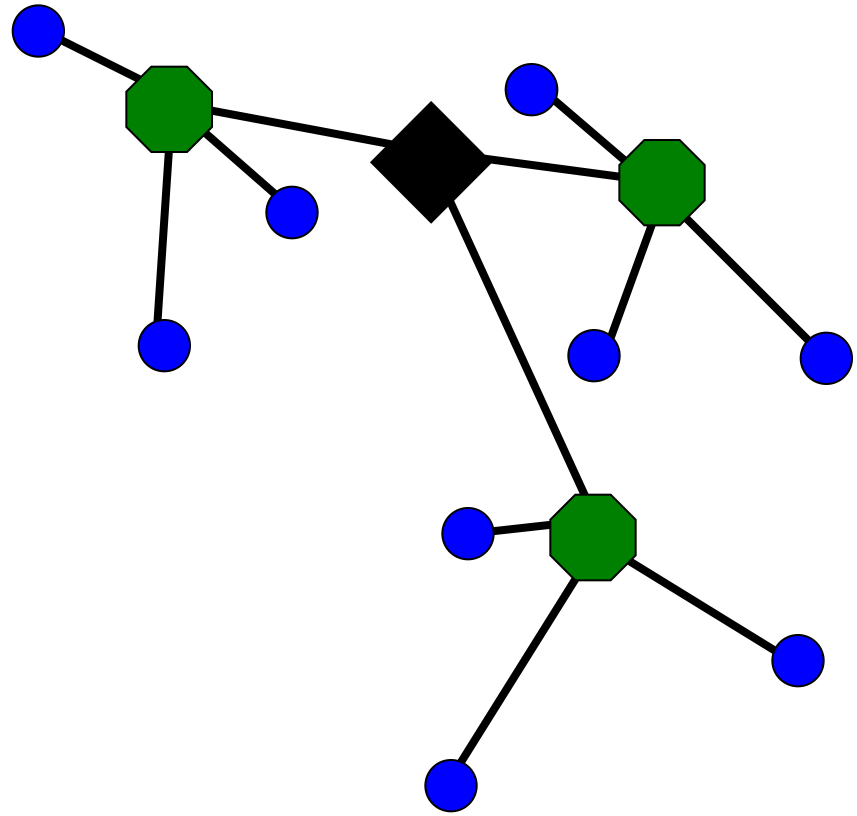


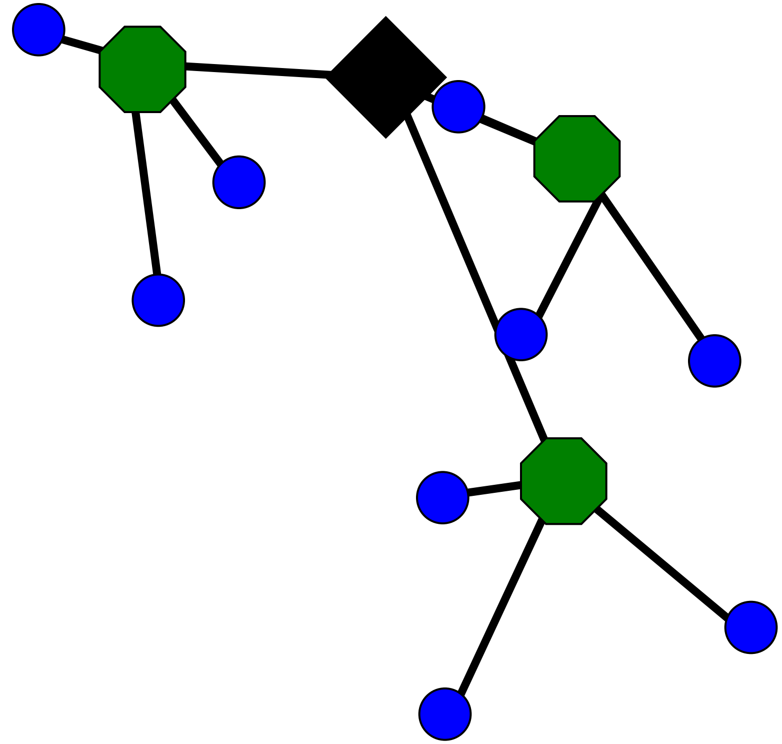


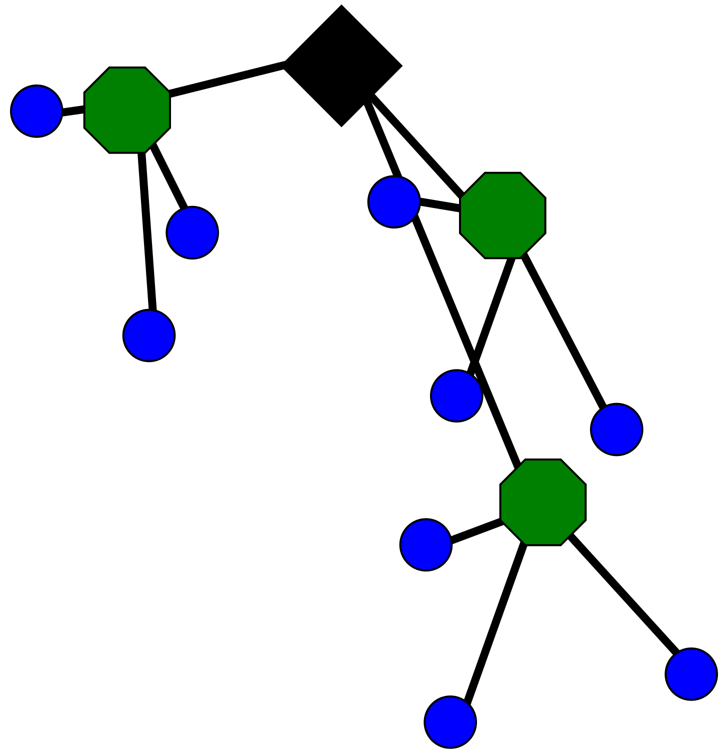


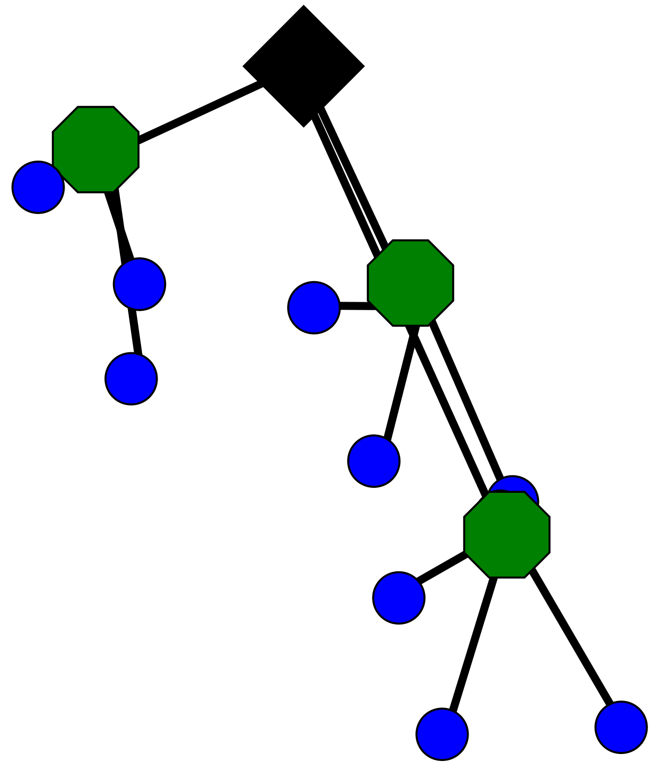


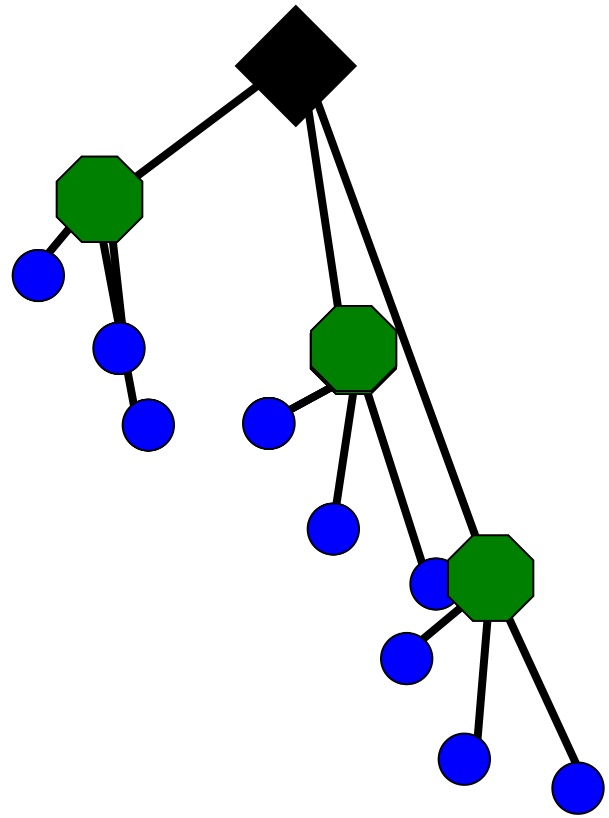


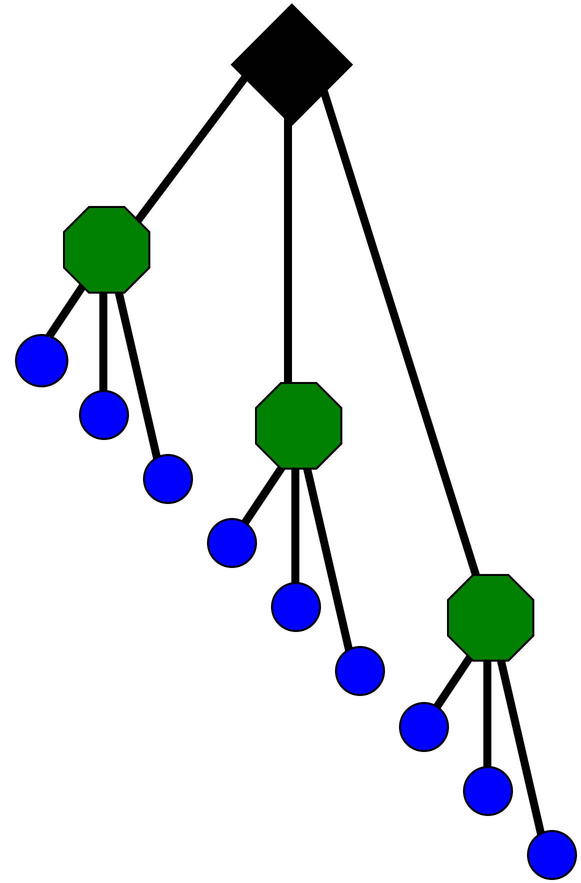




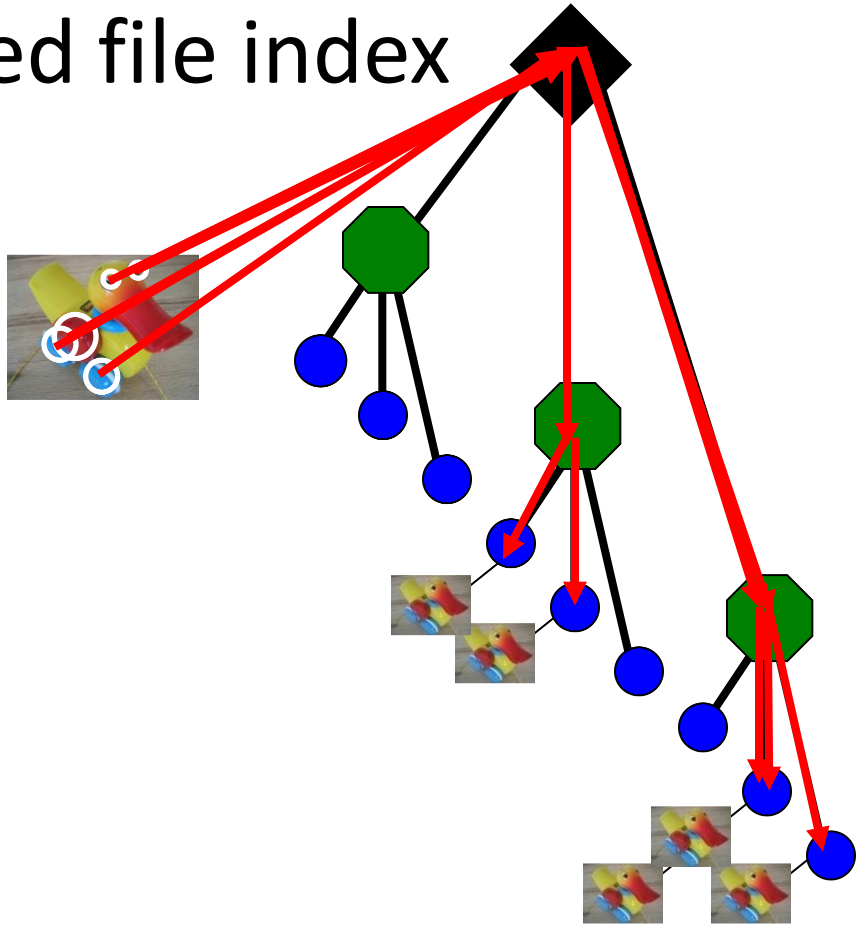




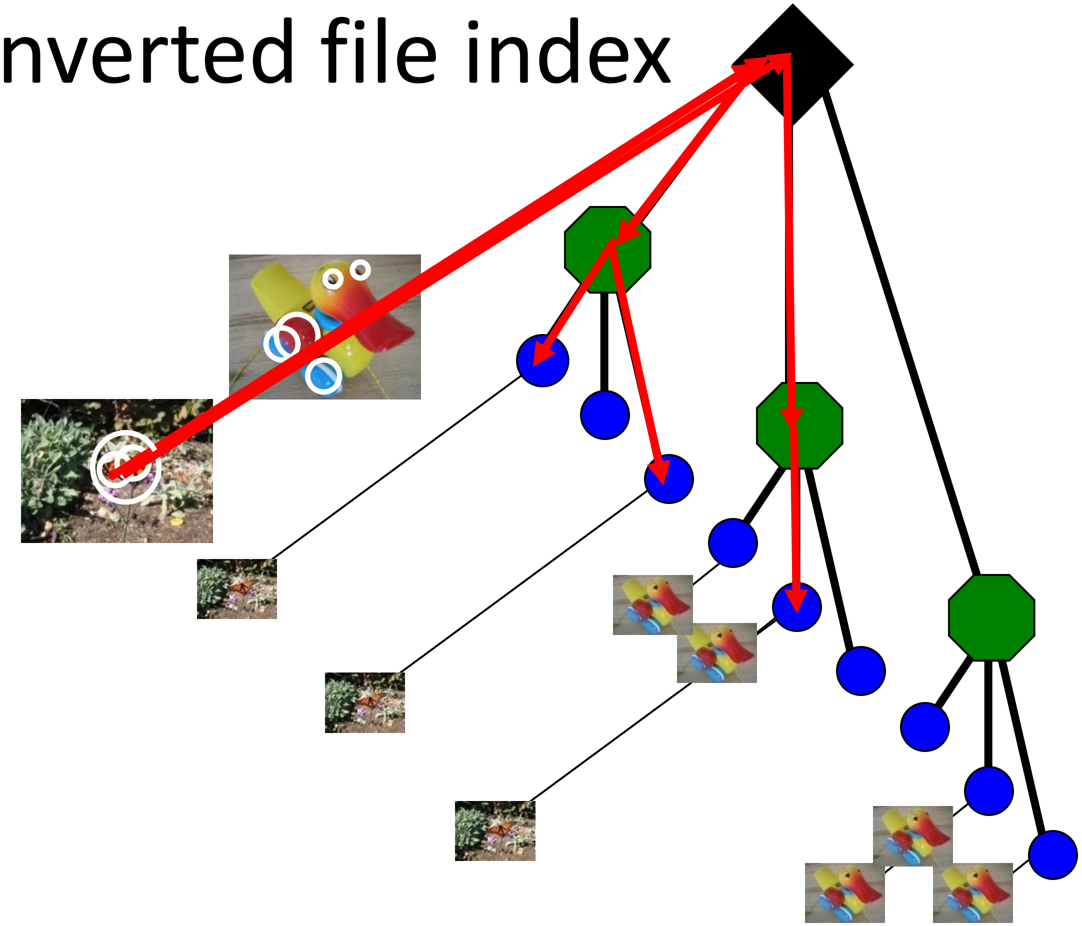




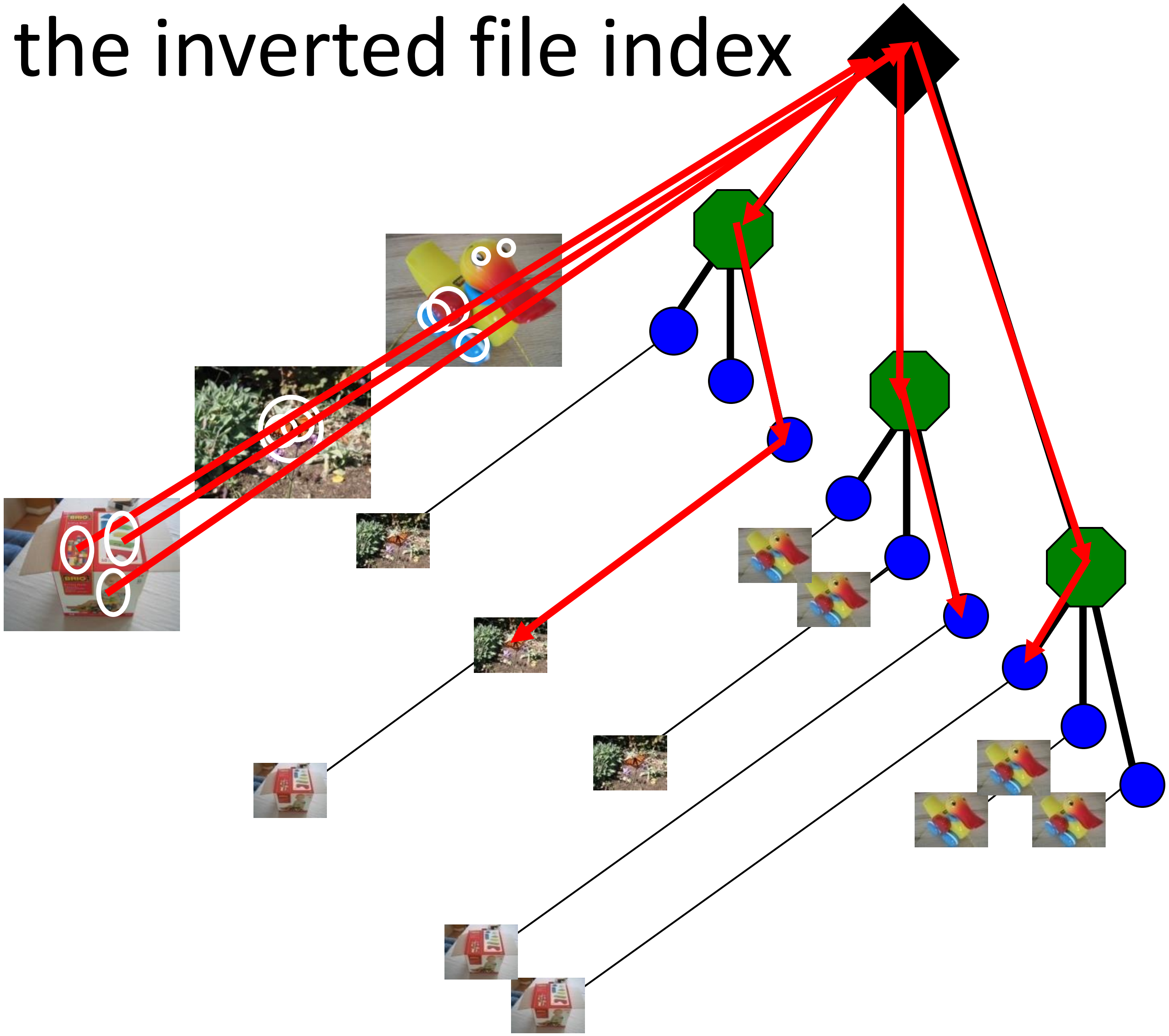
Building the inverted file index



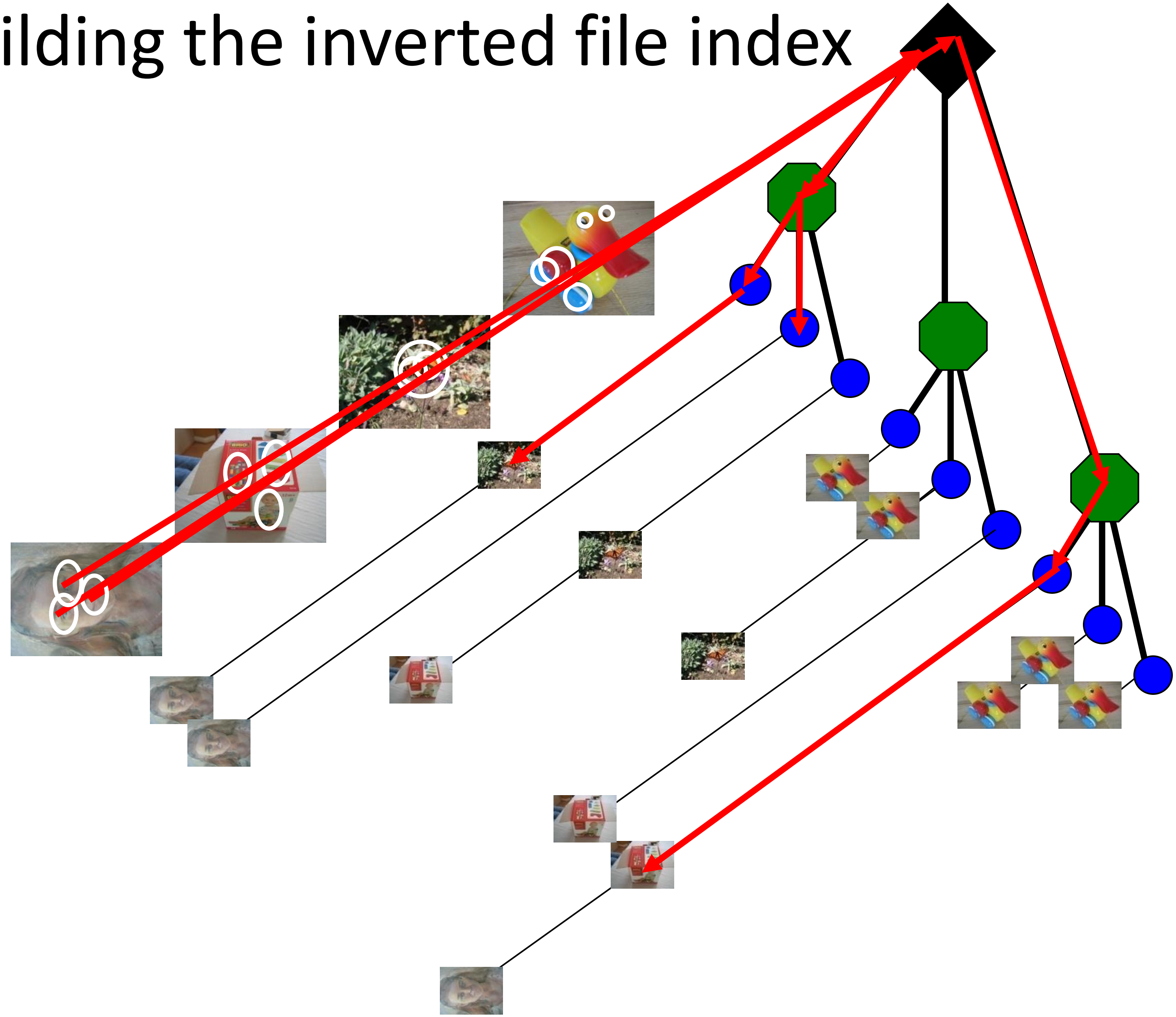
Building the inverted file index



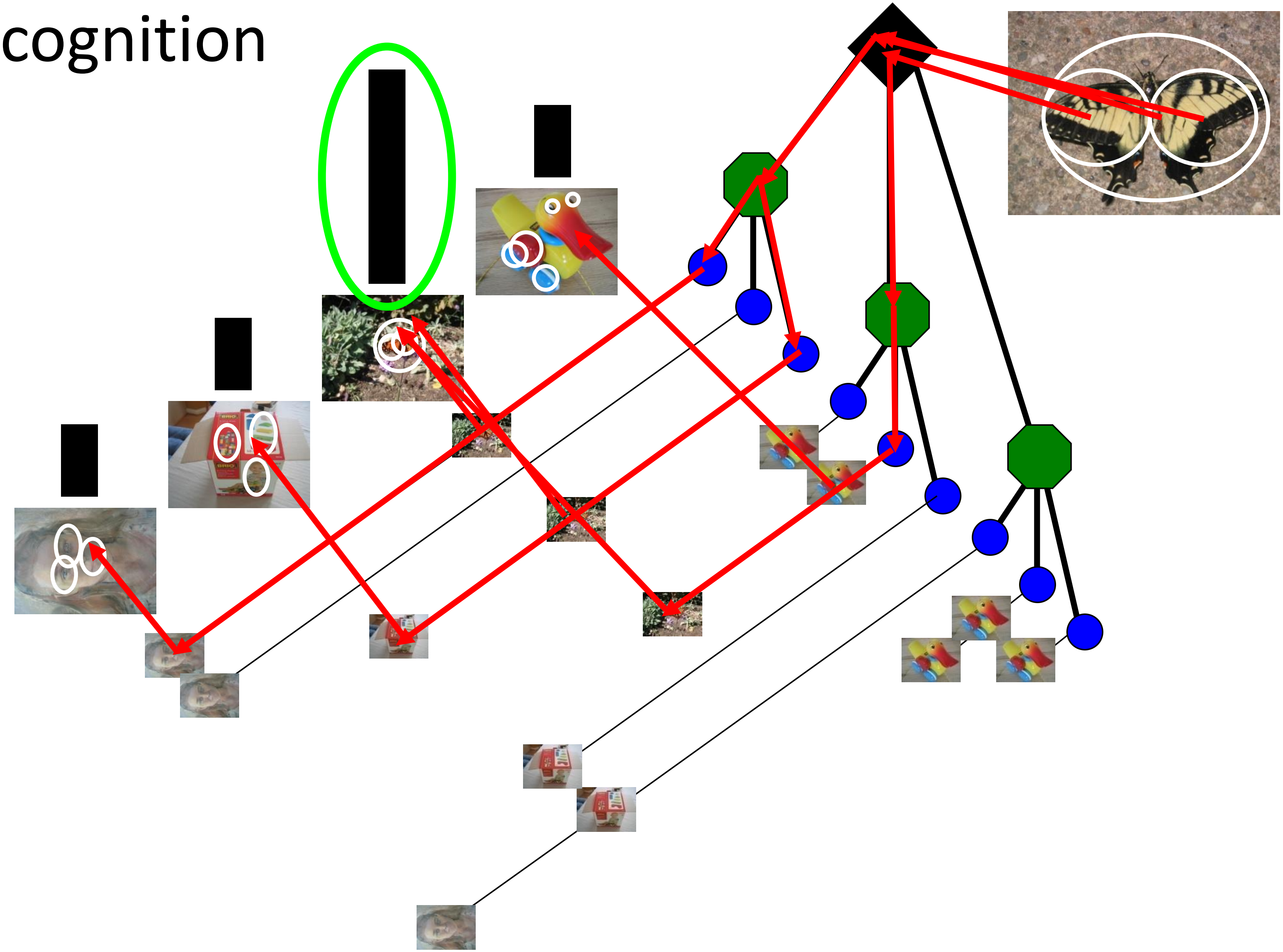
Building the inverted file index



Building the inverted file index



Recognition



Robust object/scene recognition

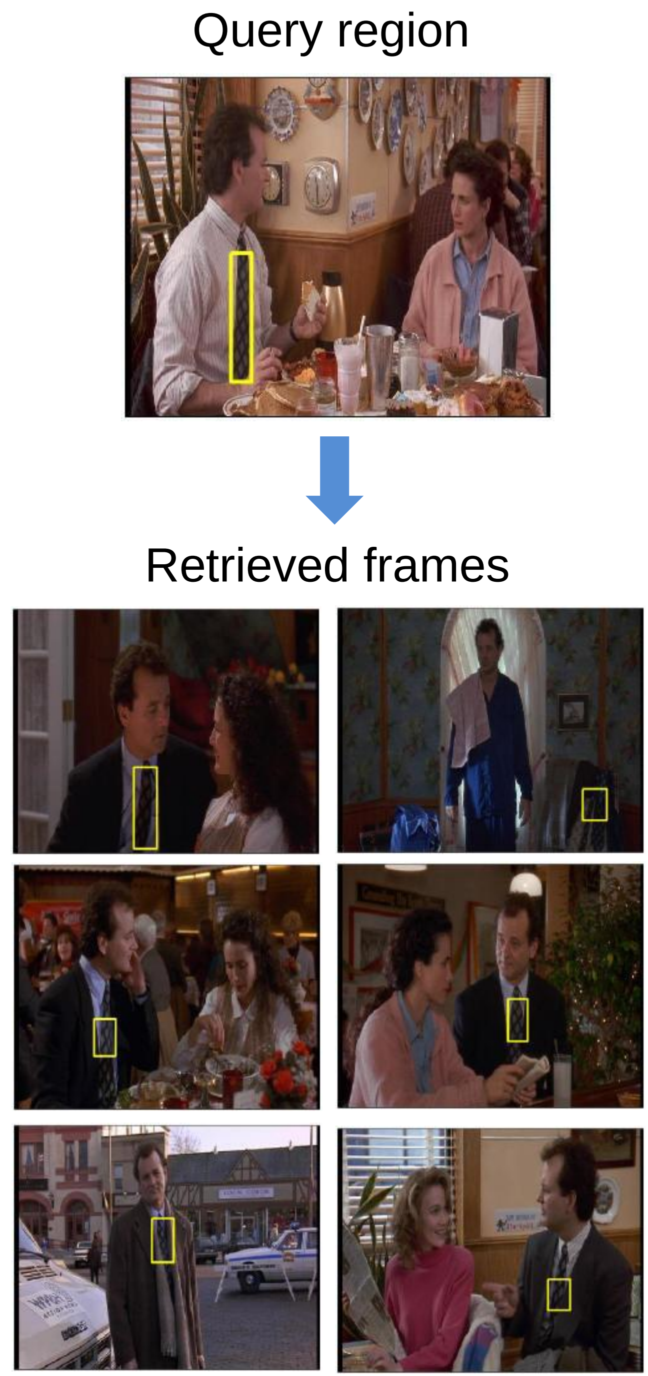
- Visual Vocabulary discards the spatial relationships between features
  - Two images with the same features *shuffled around* will return a 100% match when using only appearance information.
- This can be overcome using **geometric verification**
  - Test the h most similar images to the query image 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/



FABMAP [Cummins and Newman IJRR 2011]

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