

Lecture 12

Recognition

Davide Scaramuzza

Oral exam dates

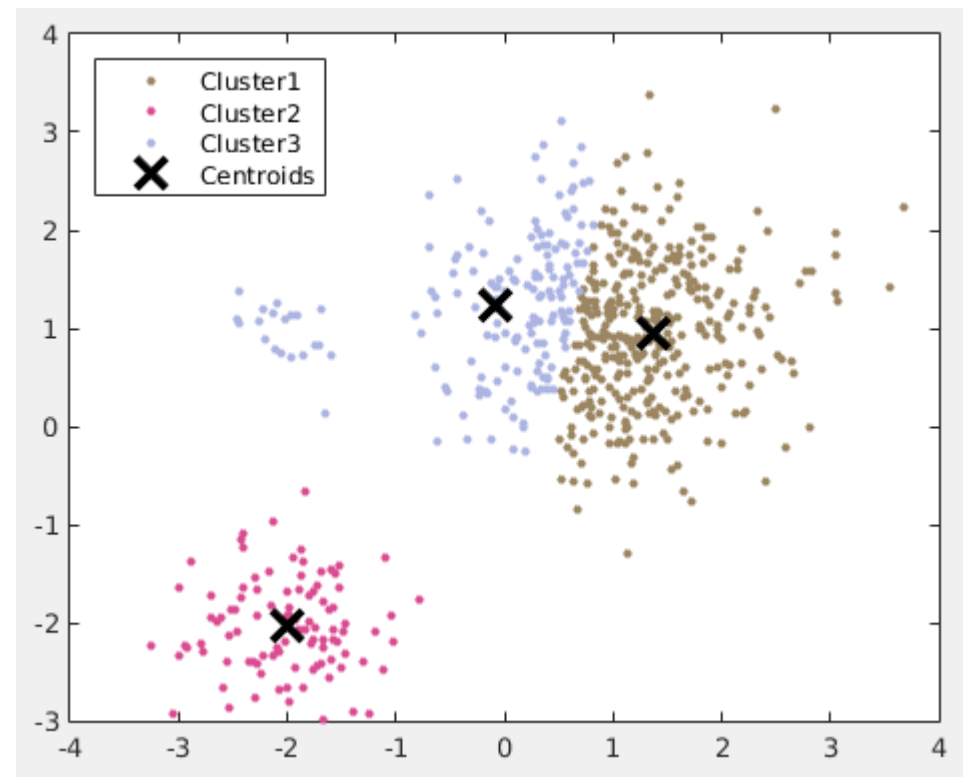
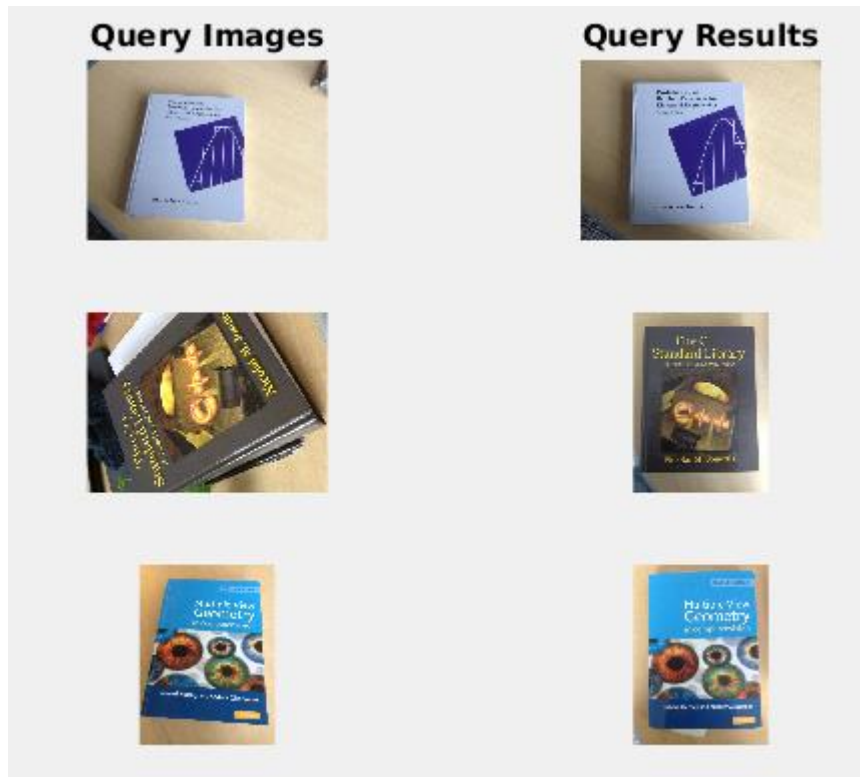
- UZH
 - January 19-20
- ETH
 - 30.01 to 9.02 2017 (schedule handled by ETH)
- Exam location
 - Davide Scaramuzza's office:
 - Andreasstrasse 15, 2.10, 8050 Zurich

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



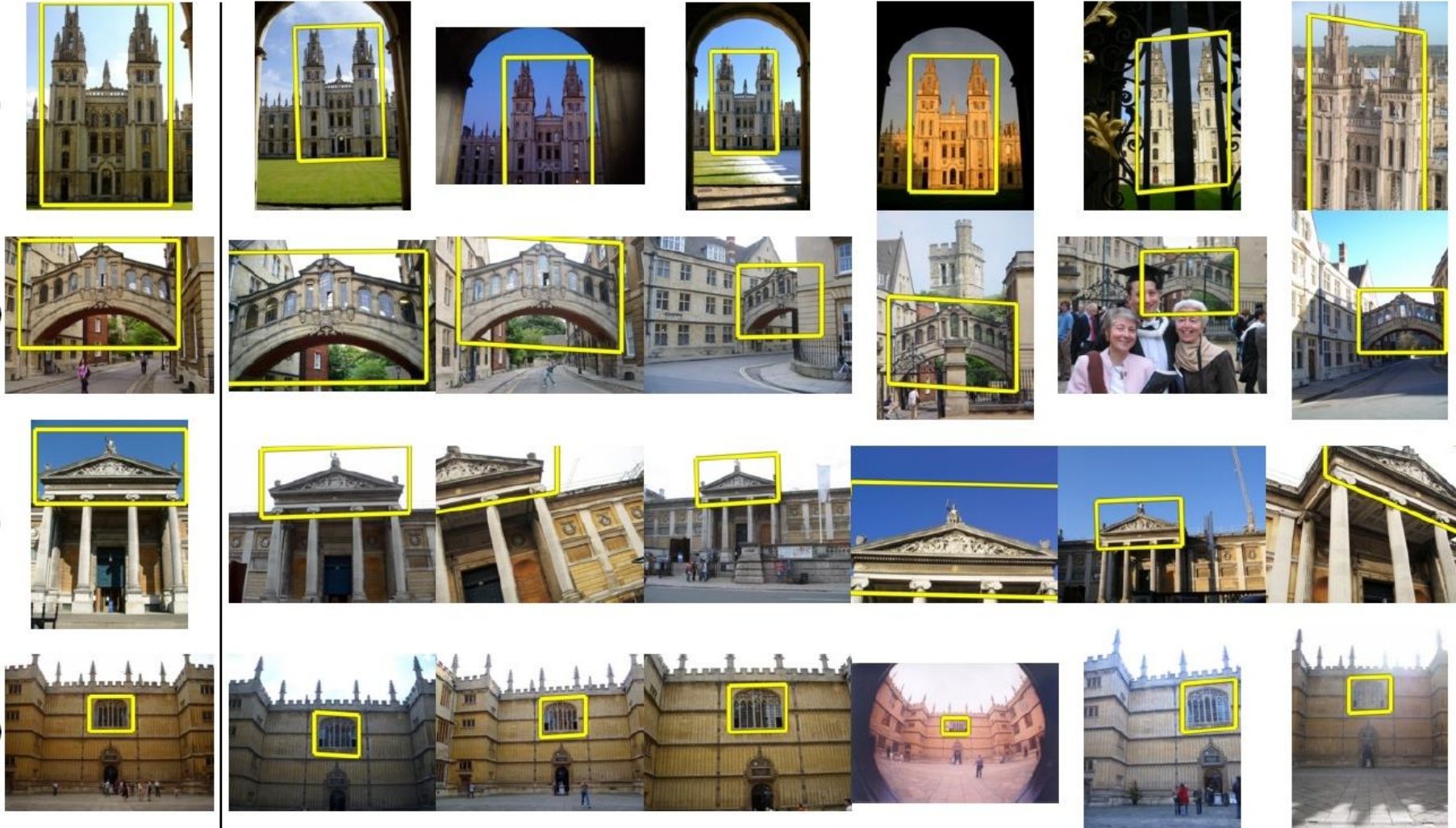
Outline

- 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



Detection

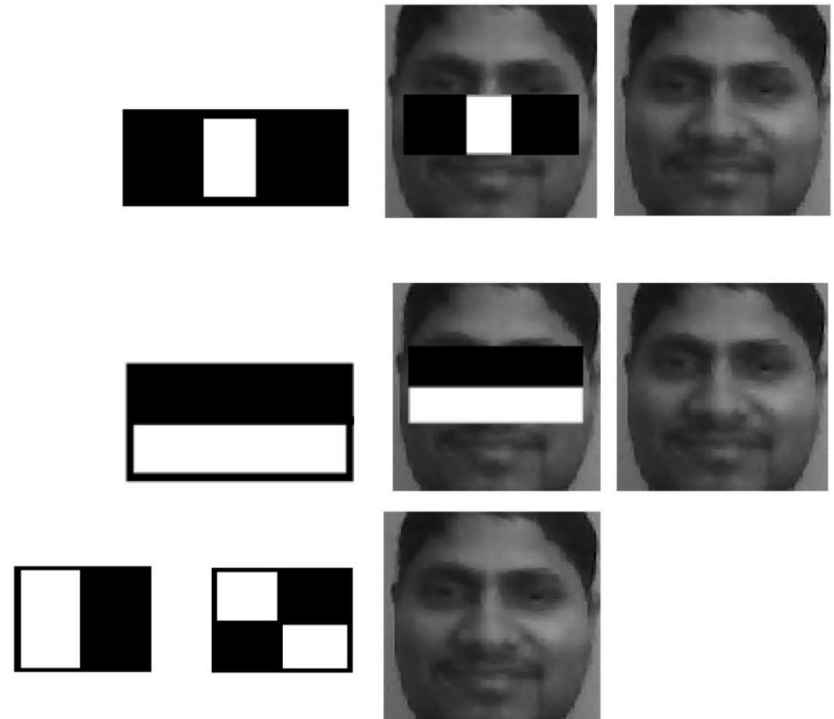


Recognition

“Sally”

Application: Face recognition

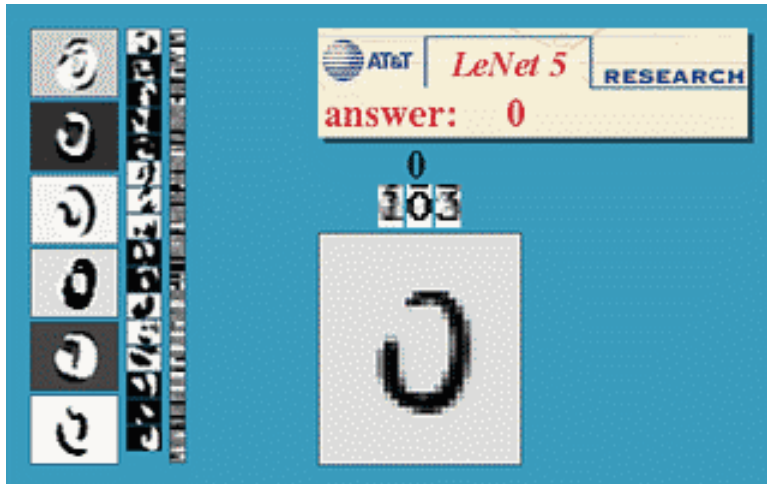
- Detection works by using four basic types of feature detectors
 - The white areas are subtracted from the black ones.
 - A special representation of the sample called the **integral image** makes feature extraction faster.



Application: Optical character recognition (OCR)

Technology to convert scanned docs to text

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



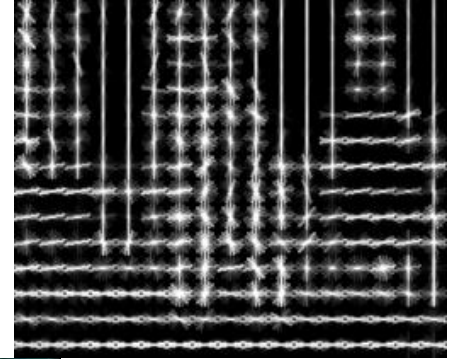
Digit recognition, AT&T labs, using CNN,
by Yann LeCun (1993)
<http://yann.lecun.com/>



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

Application: pedestrian recognition

- Detector: Histograms of oriented gradients (HOG)



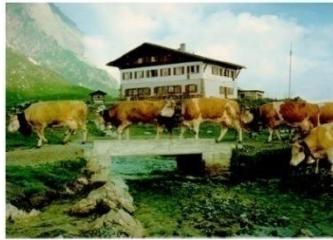
Credit: Van Gool's lab, ETH Zurich

Challenges: object intra-class variations

- How to recognize ANY car

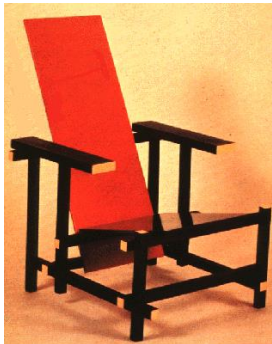


- How to recognize ANY cow

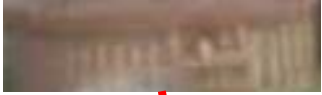


Challenges: object intra-class variations

- How to recognize ANY chair



Challenges: context and human experience

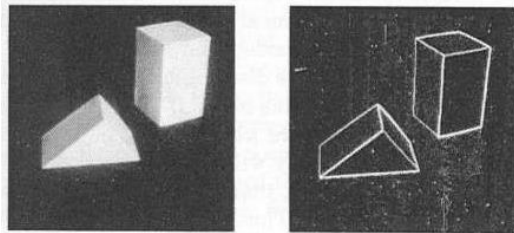
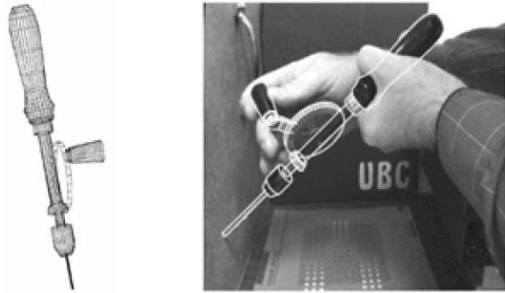


Outline

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

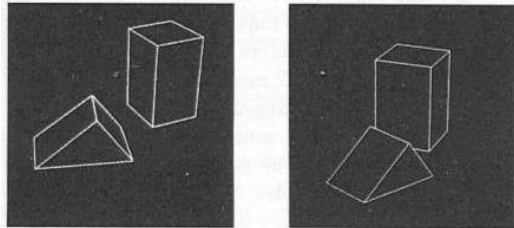
Research progress in recognition

1960-1990
Polygonal objects



b)

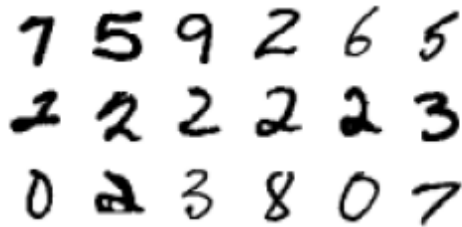
c)



d)

e)

1990-2000
Faces, characters,
planar objects

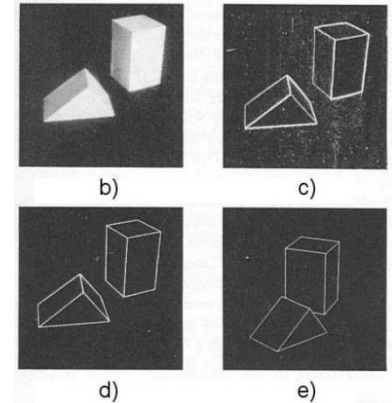
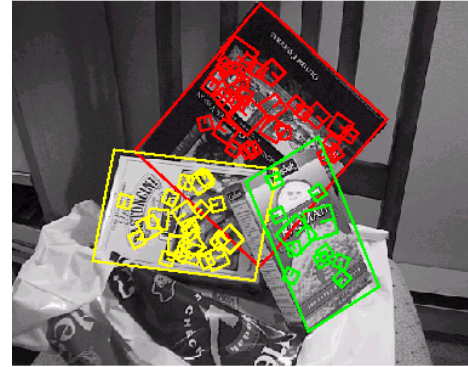


2000-today
Any kind of object



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



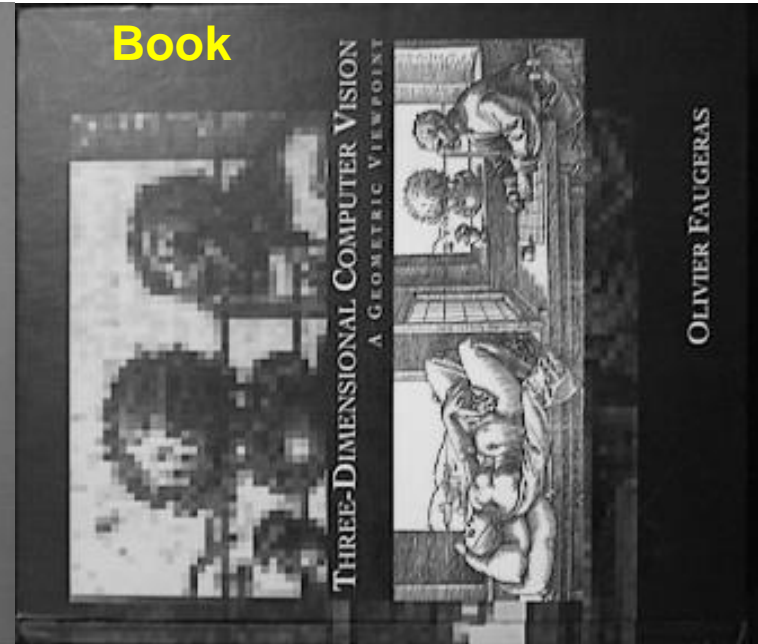
Example of 2D model-based approach

Q: Is this Book present in the Scene?

Scene

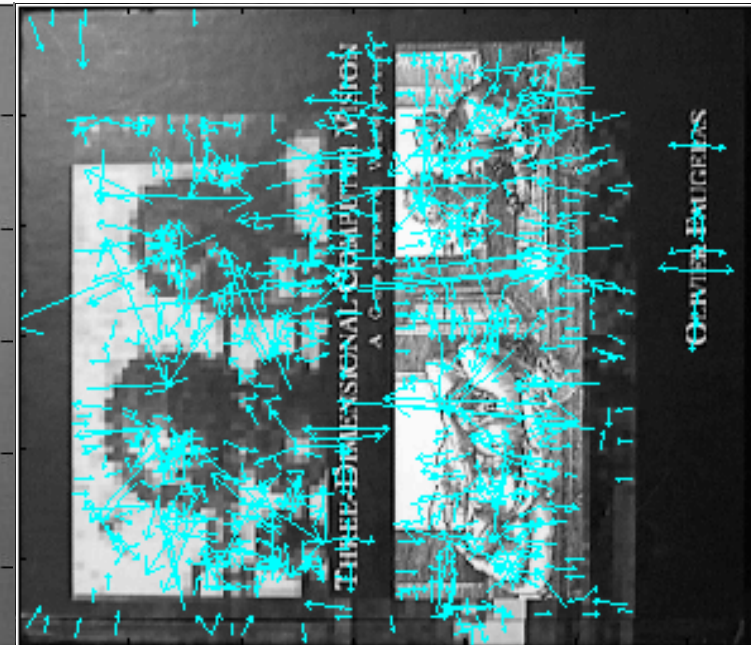


Book



Example of 2D model-based approach

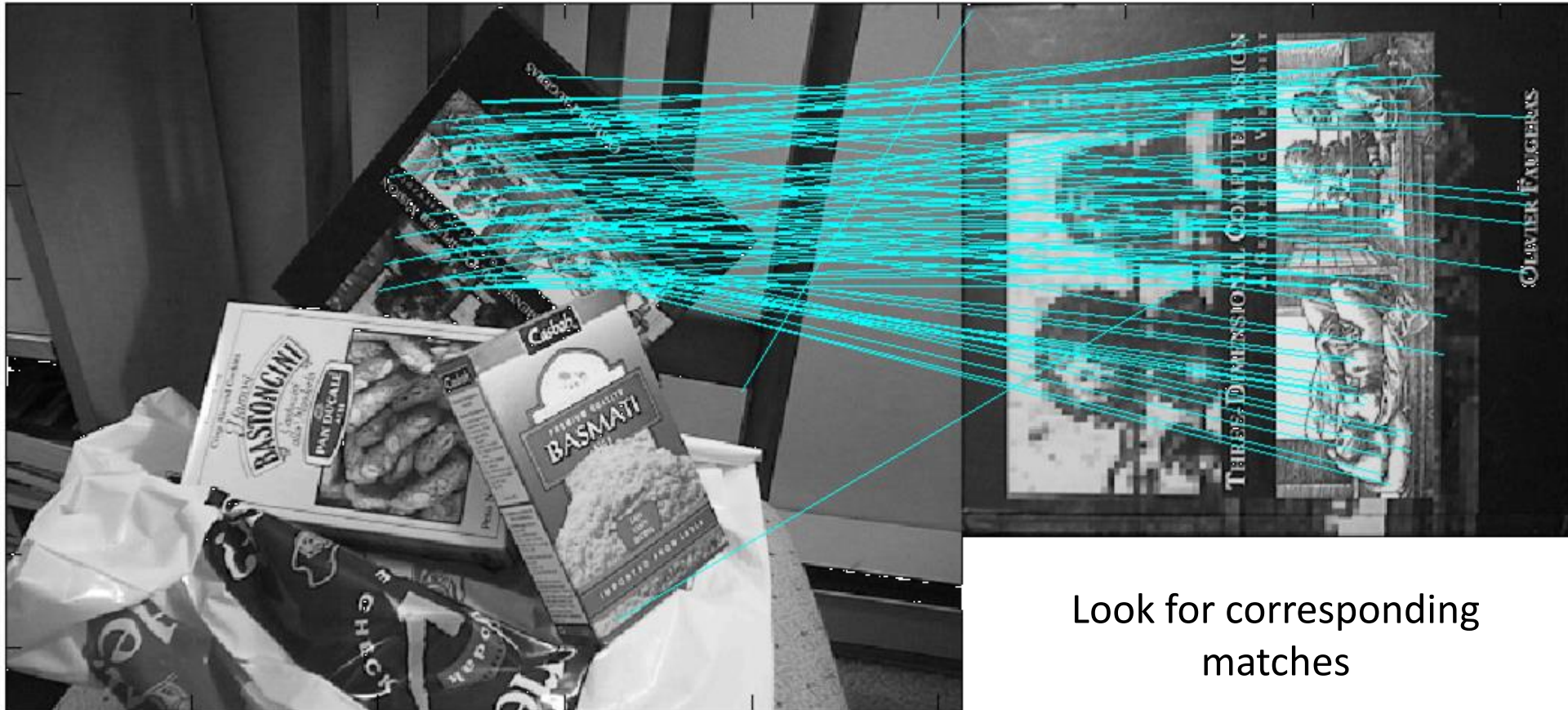
Q: Is this Book present in the Scene?



Extract keypoints in
both images

Example of 2D model-based approach

Q: Is this Book present in the Scene?



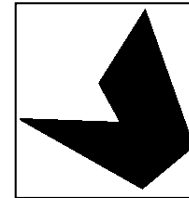
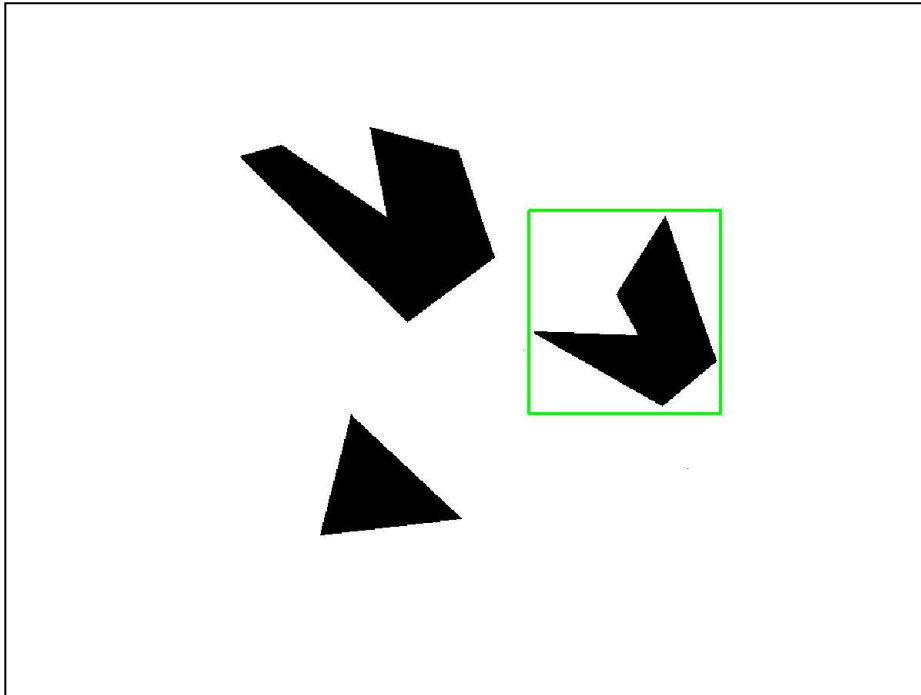
Look for corresponding matches

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

⇒ A: The Book is present in the Scene

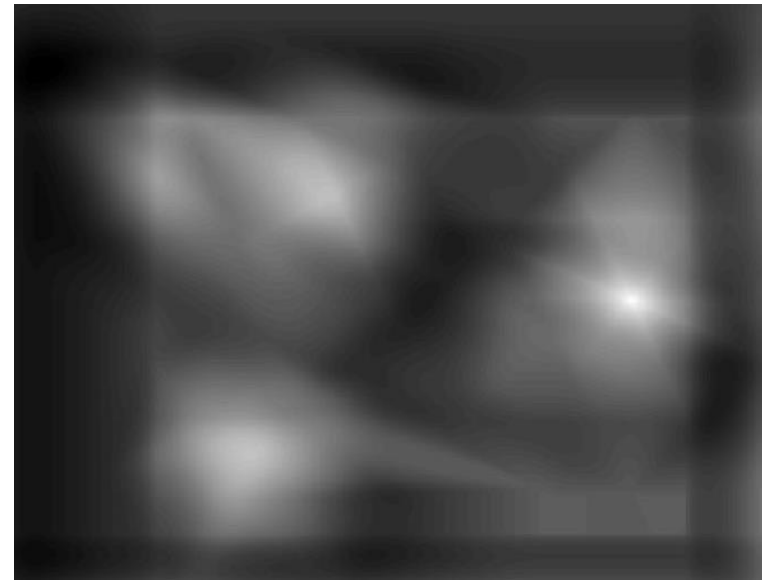
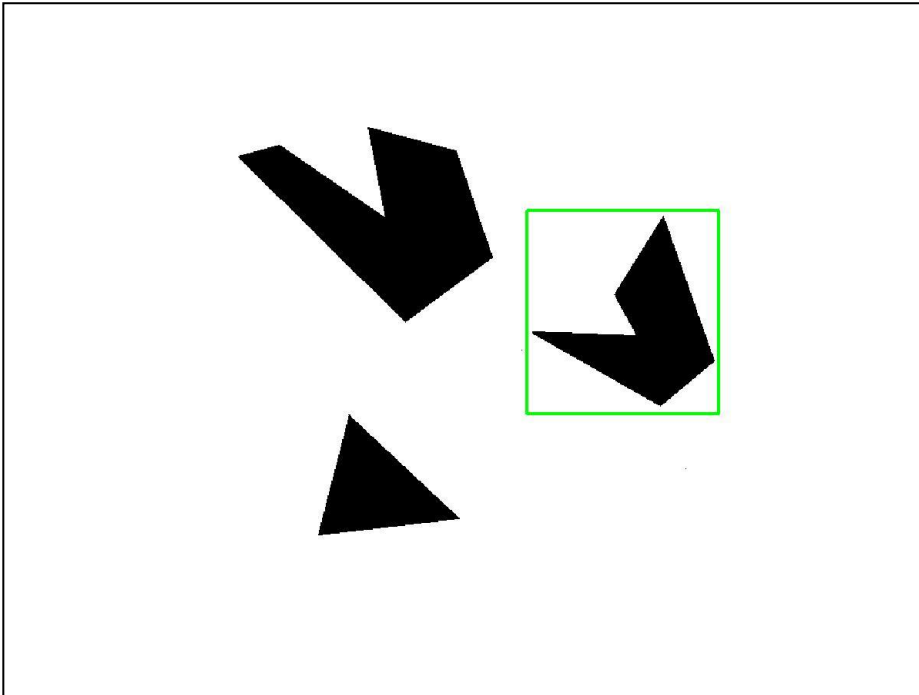
Example of appearance-based approach: Simple 2D template matching

- 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?



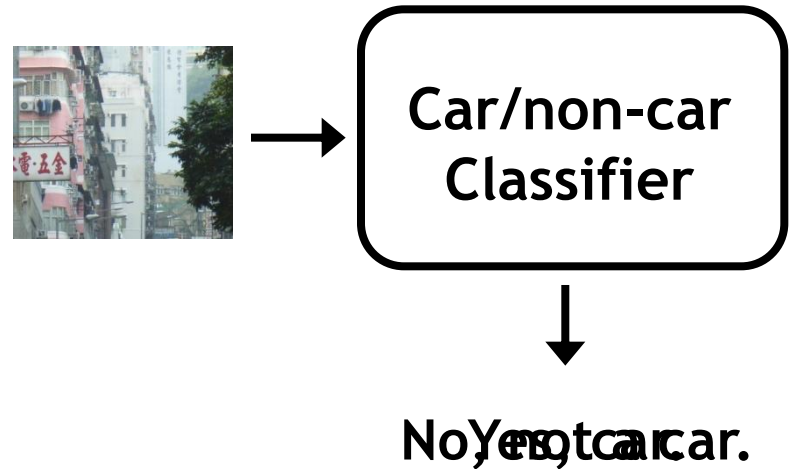
Bounding box on object



Full segmentation

Detection via classification: Main idea

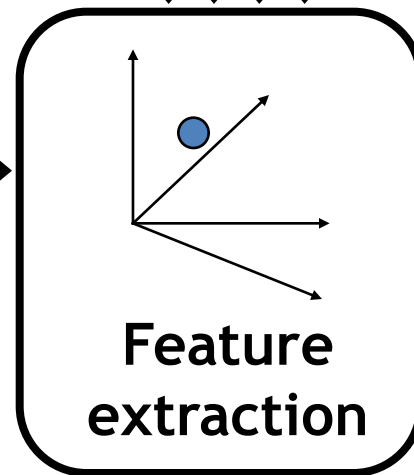
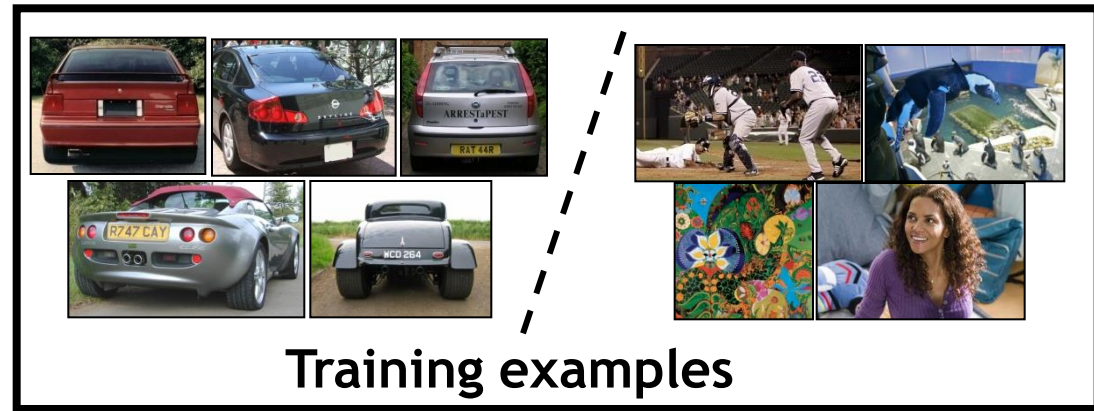
Basic component: a **binary** classifier



Detection via classification: Main idea

More in detail, we need to:

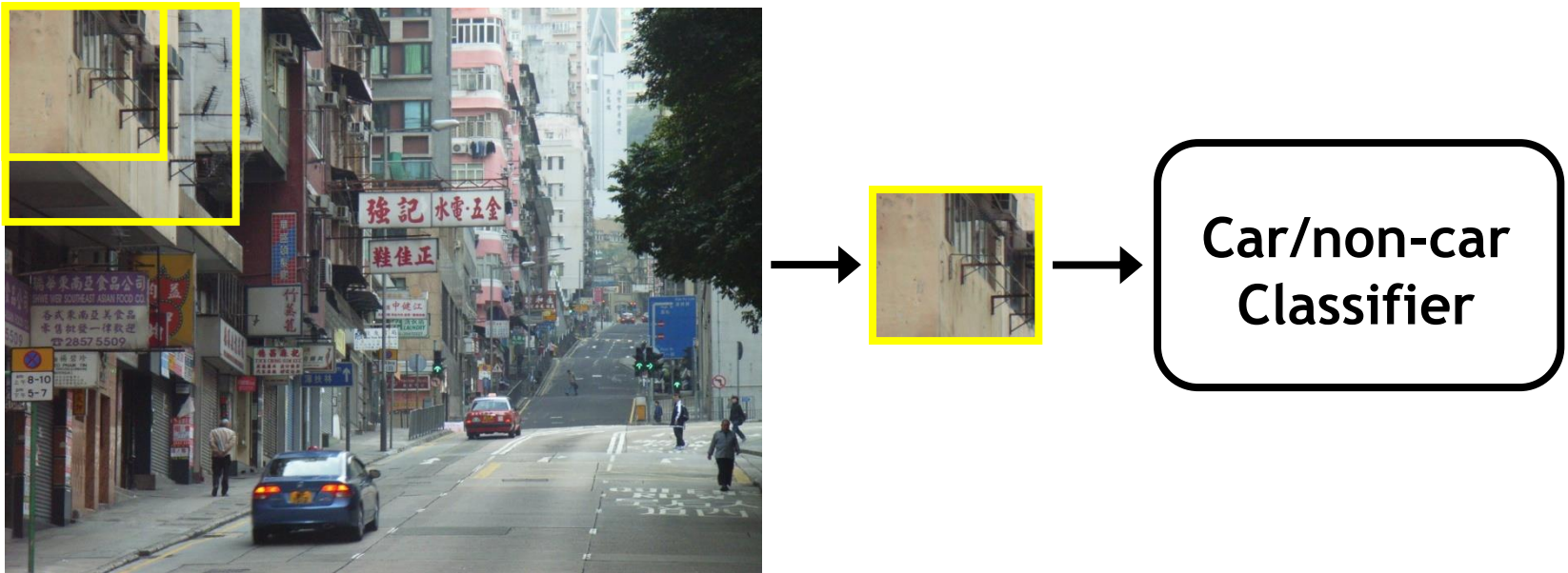
1. Obtain training data
2. Define features
3. Define classifier



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:

$f(\text{apple image}) = \text{"apple"}$

$f(\text{tomato image}) = \text{"tomato"}$

$f(\text{cow image}) = \text{"cow"}$

The machine learning framework

$$y = f(x)$$

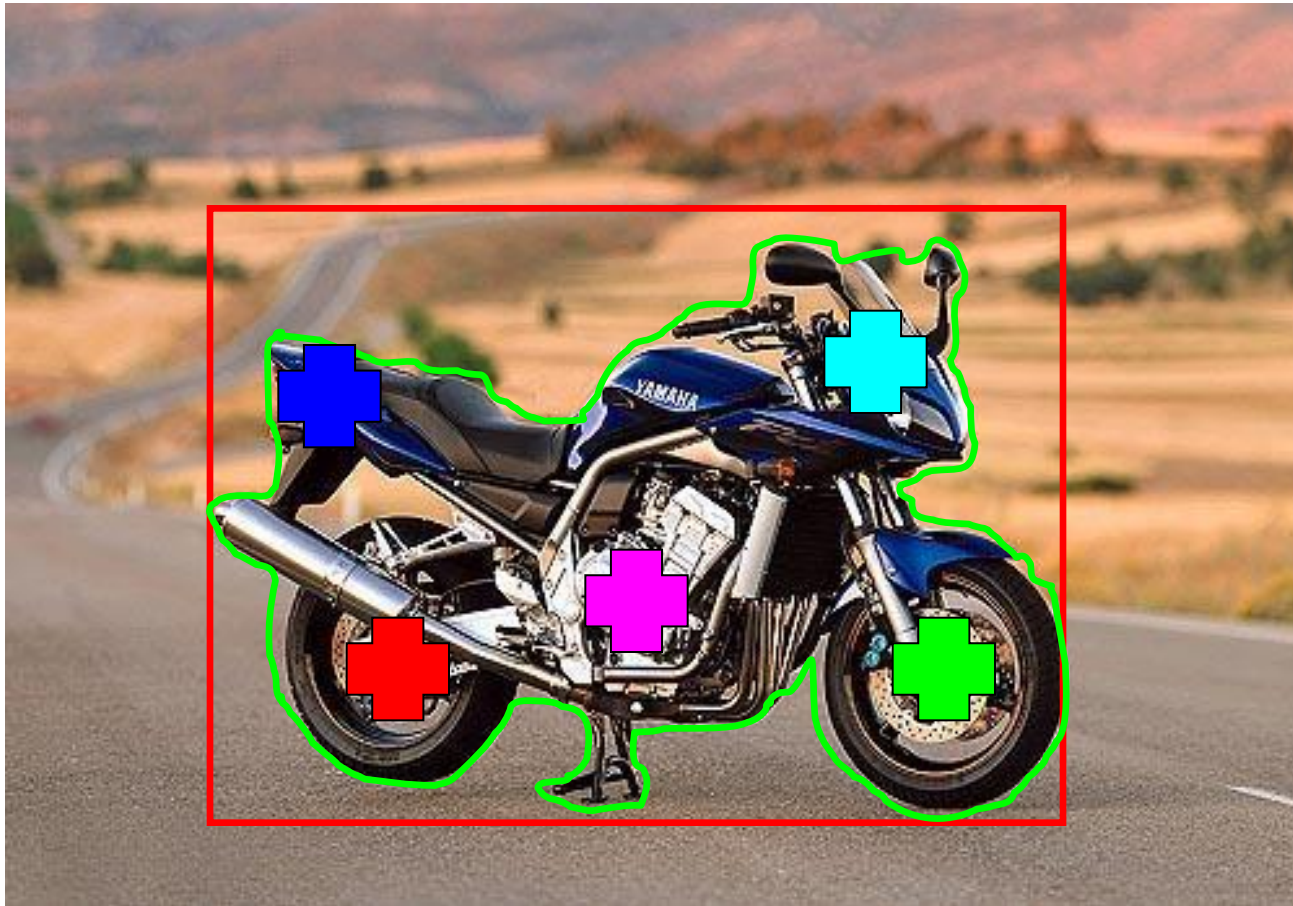
output prediction function Image feature

- **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 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

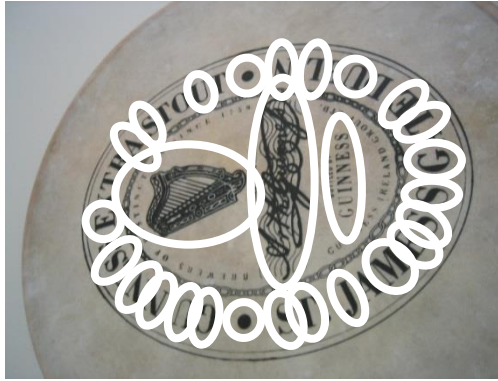
- Images in the training set must be *annotated* with the “correct answer” that the model is expected to produce

Contains a motorbike



Examples of possible features

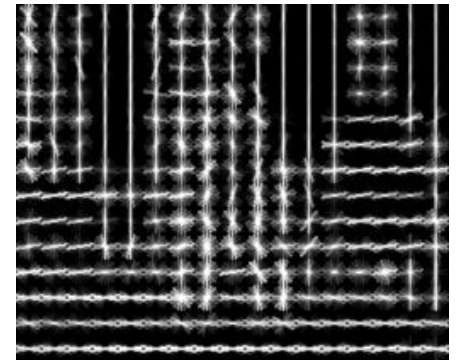
- Blob features



- Image Histograms



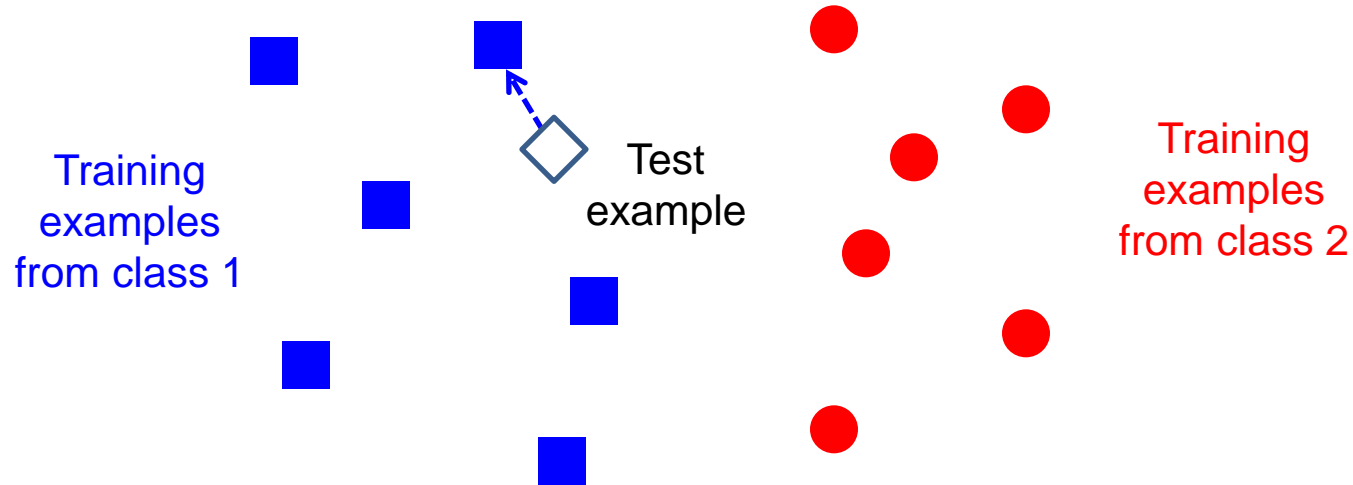
- Histograms of oriented gradients (HOG)



Classifiers: Nearest neighbor

Features are represented in the descriptor space

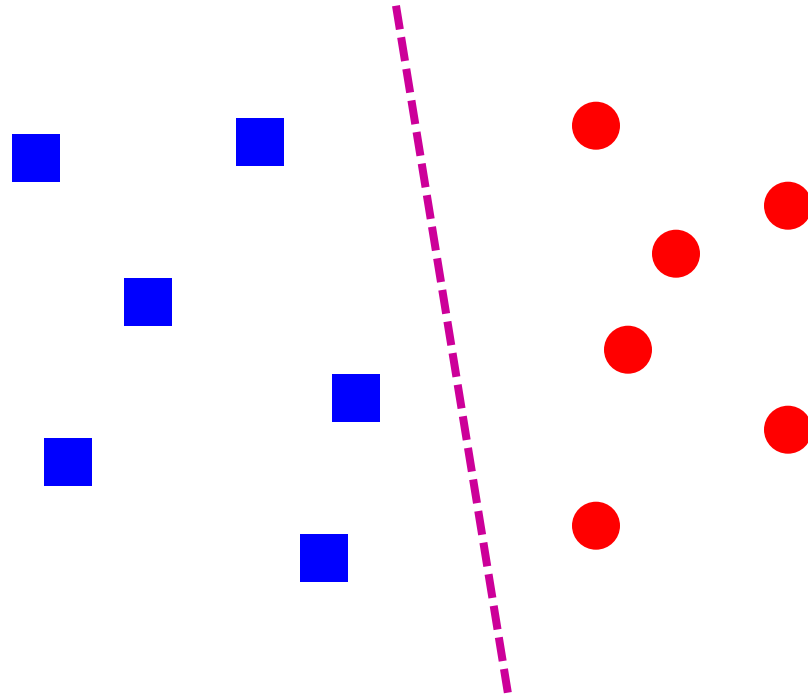
(Ex. What is the dimensionality of the descriptor space for SIFT features?)



$f(x)$ = label of the training example nearest to 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

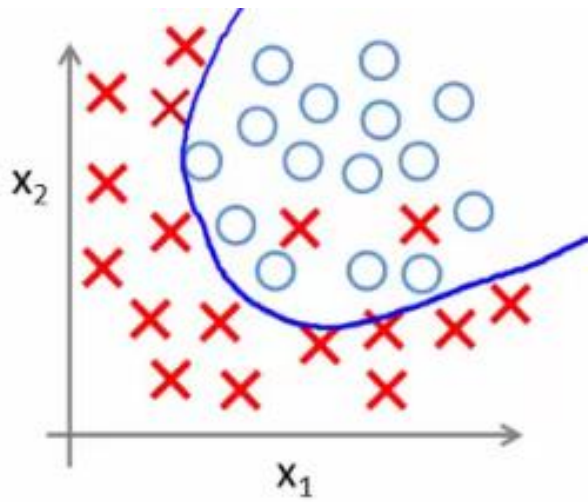


- Find a *linear function* to separate the classes:

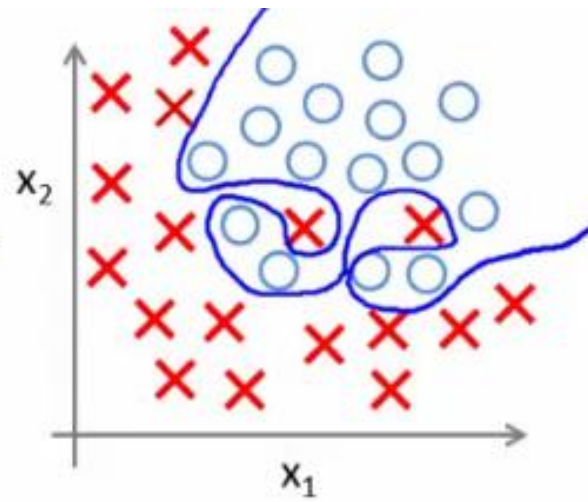
$$f(\mathbf{x}) = \text{sgn}(\mathbf{w} \cdot \mathbf{x} + b)$$

Classifiers: non-linear

Good classifier



Bad classifier (over fitting)

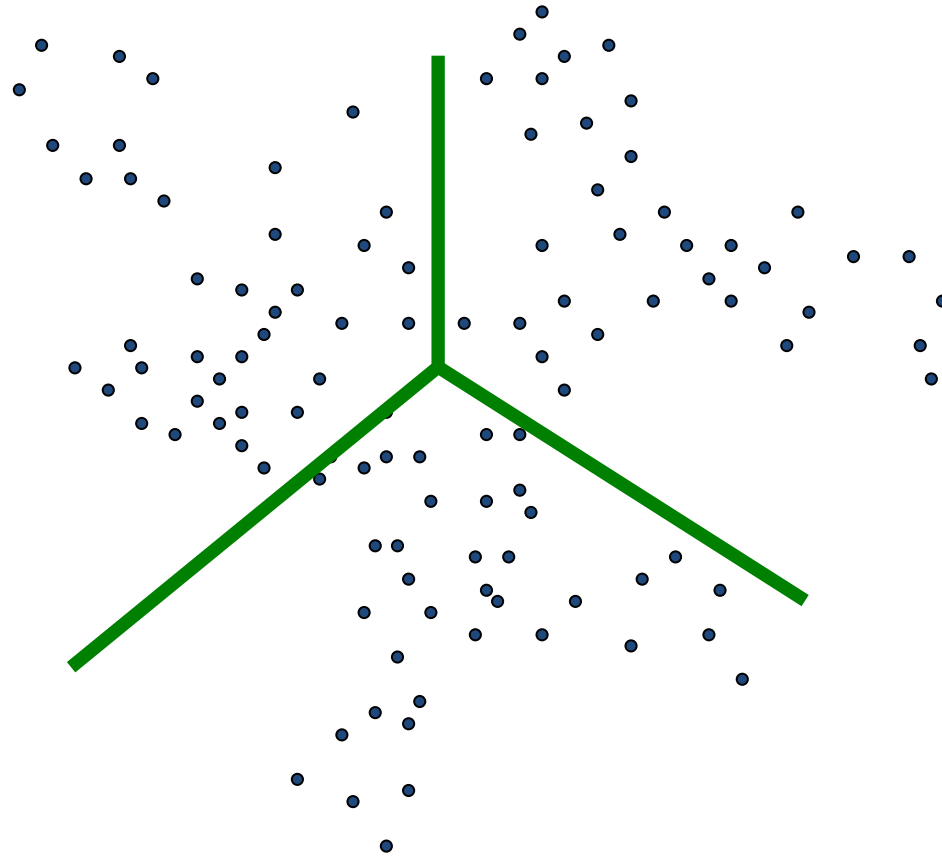


Outline

- Recognition 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



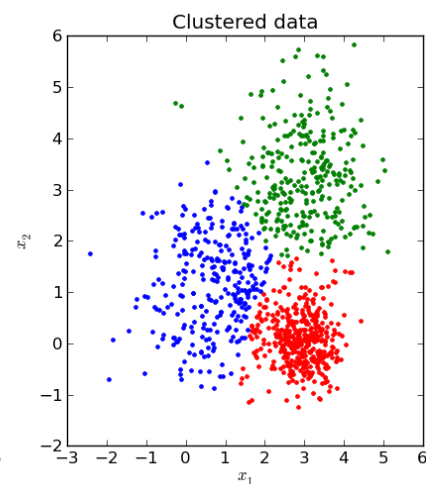
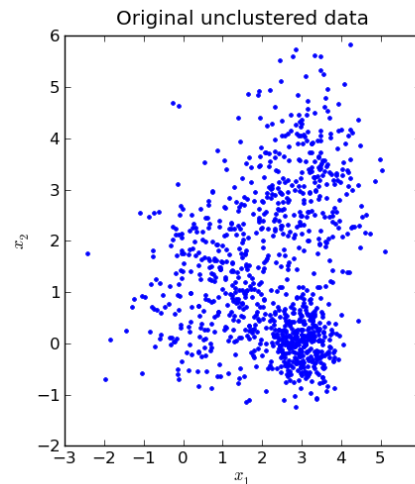
K-means clustering

- *k-means clustering* is an algorithm to partition n observations into k clusters in which each observation x_j belongs to the cluster with center m_i
- It minimizes the sum of squared Euclidean distances between points x_j and their nearest cluster centers 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 x_j to the nearest center m_i
 - Recompute each cluster center as the mean of all points assigned to it



K-means demo



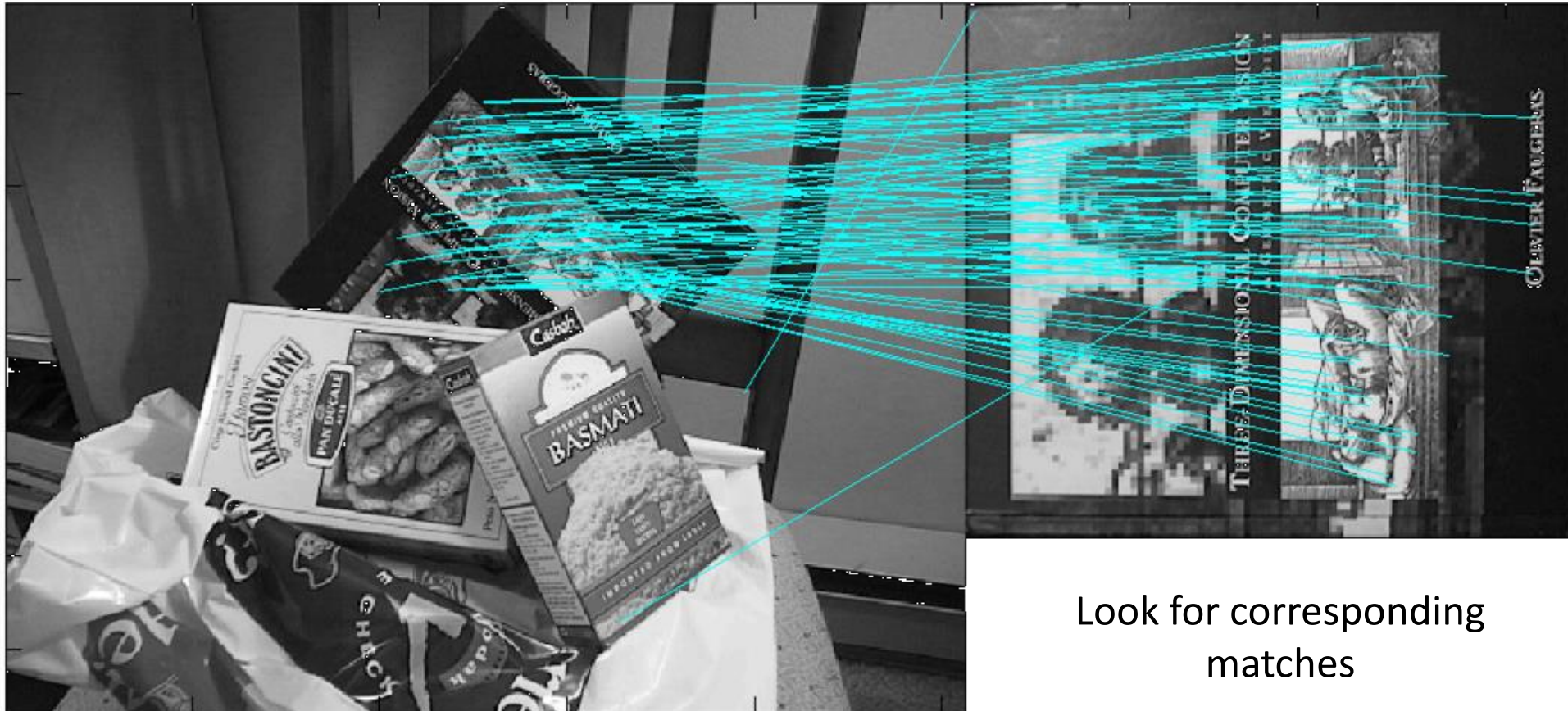
Source: <http://shabal.in/visuals/kmeans/1.html>

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

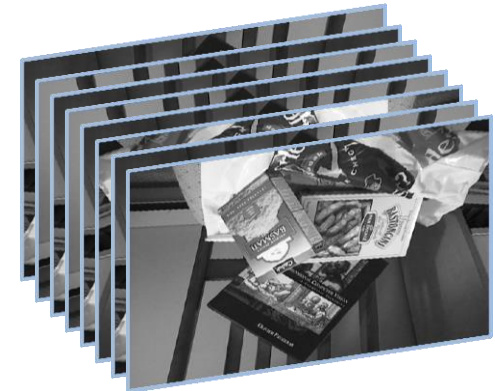
⇒ A: The Book is present in the Scene

Taking this a step further...

- Find an object in an image



- Find an object in multiple images



- Find multiple objects in multiple images

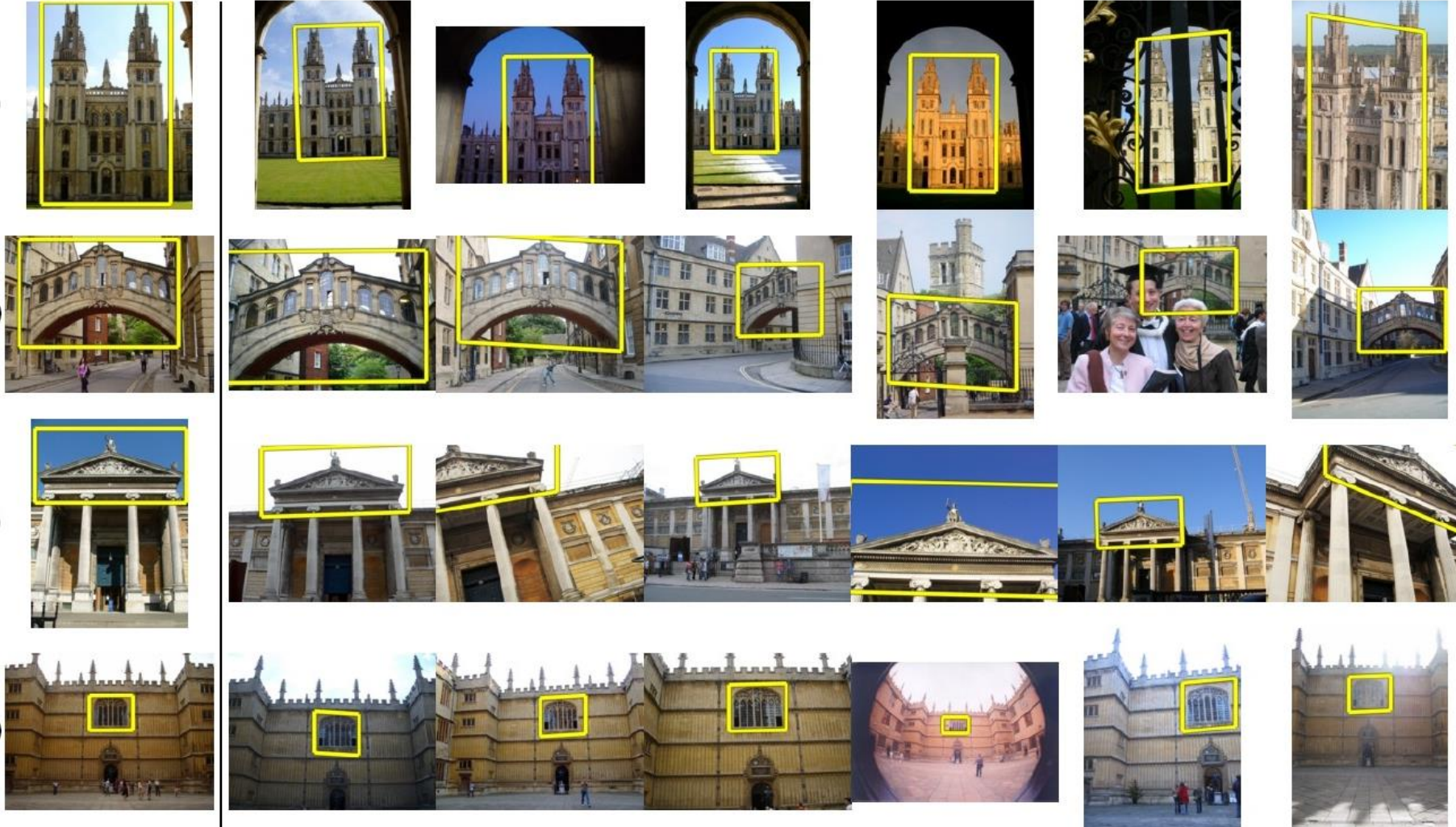
As the number of images increases,
feature-based object recognition
becomes computationally more and
more expensive



Application: large-scale image retrieval

Query image

Results on a database of 100 million images





TIMES SQUARE GALLERY

Kodak

EasyShare Digital Camera and Dock



Shoot. Touch. Share.

Ask Merri



COME JOIN THE PARTY



WOLFE THE MINUTE TM

RE-IMAGINING THE 1930 ESCALADE

WOLFE THE MINUTE TM

WOLFE THE MINUTE TM

WOLFE THE MINUTE TM

WOLFE THE MINUTE TM

WOLFE THE MINUTE TM

WOLFE THE MINUTE TM

WOLFE THE MINUTE TM

WOLFE THE MINUTE TM

WOLFE THE MINUTE TM

WOLFE THE MINUTE TM

WOLFE THE MINUTE TM

WOLFE THE MINUTE TM

WOLFE THE MINUTE TM

HARLEN RBI
U.S. STARS
EAST



Slide Credit: Nister



Fast visual search

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

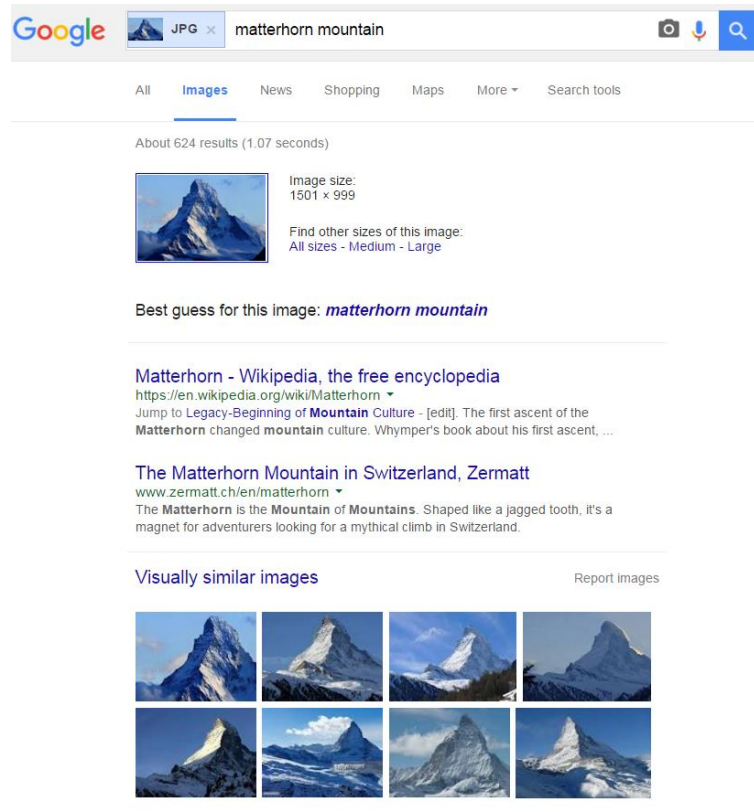


“Video Google”, Sivic and Zisserman, ICCV 2003

“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?




The screenshot shows a Google search interface. The search bar contains the text "matterhorn mountain" and a small image icon. Below the search bar, there are navigation tabs for "All", "Images", "News", "Shopping", "Maps", "More", and "Search tools". The "Images" tab is selected. Below the tabs, it says "About 624 results (1.07 seconds)". A large image of the Matterhorn mountain is displayed, with a small thumbnail to its left. To the right of the image, there is text: "Image size: 1501 x 999" and "Find other sizes of this image: All sizes - Medium - Large". Below the image, it says "Best guess for this image: *matterhorn mountain*".

Matterhorn - Wikipedia, the free encyclopedia
<https://en.wikipedia.org/wiki/Matterhorn>
Jump to Legacy-Beginning of **Mountain** Culture - [edit]. The first ascent of the **Matterhorn** changed **mountain** culture. Whymper's book about his first ascent, ...

The Matterhorn Mountain in Switzerland, Zermatt
www.zermatt.ch/en/matterhorn
The **Matterhorn** is the **Mountain of Mountains**. Shaped like a jagged tooth, it's a magnet for adventurers looking for a mythical climb in Switzerland.

Visually similar images Report images



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 (*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 $\rightarrow M = 100, N = 100 \rightarrow N^2 M^2 / 2 = \sim 50$ 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)]

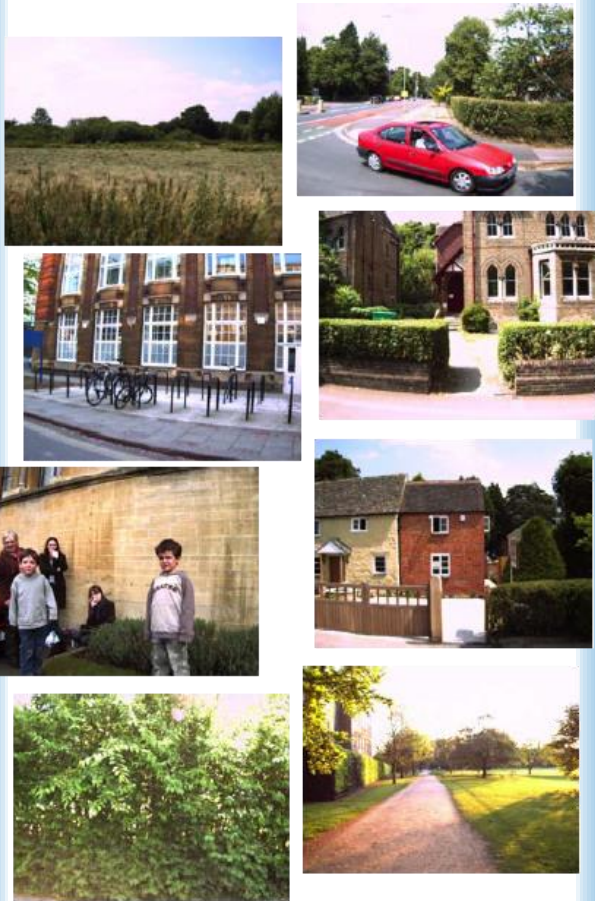
Indexing local features: inverted file text

- For text documents, an efficient way to find all *pages* 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)

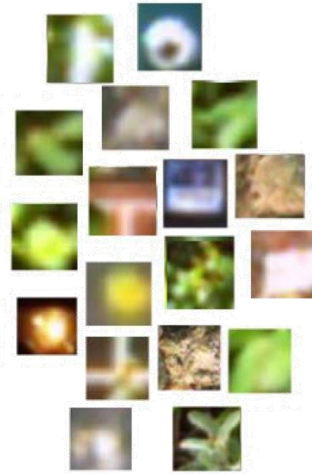
Index		
"Along I-75," From Detroit to Florida; <i>inside back cover</i>	Butterfly Center, McGuire; 134	Driving Lanes; 85
"Drive I-95," From Boston to Florida; <i>inside back cover</i>	CAA (see AAA)	Duval County; 163
1929 Spanish Trail Roadway; 101-102,104	CCC, The; 111,113,115,135,142	Eau Gallie; 175
511 Traffic Information; 83	Ca d'Zan; 147	Edison, Thomas; 152
A1A (Barrier Isl) - I-95 Access; 86	Caloosahatchee River; 152	Eglin AFB; 116-118
AAA (and CAA); 83	Name; 150	Eight Reale; 176
AAA National Office; 88	Canaveral Natnl Seashore; 173	Ellenton; 144-145
Abbreviations,	Cannon Creek Airpark; 130	Emanuel Point Wreck; 120
Colored 25 mile Maps; cover	Canopy Road; 106,169	Emergency Callboxes; 63
Exit Services; 196	Cape Canaveral; 174	Epiphytes; 142,148,157,159
Travelogue; 85	Castillo San Marcos; 169	Escambia Bay; 119
Africa; 177	Cave Diving; 131	County; 120
Agricultural Inspection Stns; 126	Cayo Costa, Name; 150	Esteros; 153
Ah-Tah-Thi-Ki Museum; 160	Celebration; 93	Everglade,90,95,139-140,154-160
Air Conditioning, First; 112	Charlotte County; 149	Draining of; 156,181
Alabama; 124	Charlotte Harbor; 150	Wildlife MA; 160
Alachua; 132	Chautauqua; 116	Wonder Gardens; 154
County; 131	ChIPLEY; 114	Falling Waters SP; 115
Alafia River; 143	Name; 115	Fantasy of Flight; 95
Alapaha, Name; 126	Choctawatchee, Name; 115	Fayer Dykes SP; 171
Alfred B Maclay Gardens; 106	Circus Museum, Ringling; 147	Fires, Forest; 168
Alligator Alley; 154-155	Citrus; 88,97,130,136,140,180	Fires, Prescribed ; 148
Alligator Farm, St Augustine; 169	CityPlace, W Palm Beach; 180	Fisherman's Village; 151
Alligator Hole (definition); 157	City Maps,	Flagler County; 171
Alligator, Buddy; 155	Ft Lauderdale Expwys; 194-195	Flagler, Henry; 97,165,167,171
Alligators; 100,135,138,147,156	Jacksonville; 163	Florida Aquarium; 186
Anastasia Island; 170	Kissimmee Expwys; 192-193	Florida,
Anhaica; 108-109,146	Miami Expressways; 194-195	12,000 years ago; 187
Apalachicola River; 112	Orlando Expressways; 192-193	Cavern SP; 114
Appleton Mus of Art; 136	Pensacola; 26	Map of all Expressways; 2-3
Aquifer; 102	Tallahassee; 191	Mus of Natural History; 134
Arabian Nights; 94	Tampa-St. Petersburg; 63	National Cemetery ; 141
Art Museum, Ringling; 147	St. Augustine; 191	Part of Africa; 177
Aruba Beach Cafe; 183	Civil War; 100,108,127,138,141	Platform; 187
Aucilla River Project; 106	Clearwater Marine Aquarium; 187	Sheriff's Boys Camp; 126
Babcock-Web WMA; 151	Collier County; 154	Sports Hall of Fame; 130
Bahia Mar Marina; 184	Collier, Barron; 152	Sun 'n Fun Museum; 97
Baker County; 99	Colonial Spanish Quarters; 168	Supreme Court; 107
Barefoot Mailmen; 182	Columbia County; 101,128	Florida's Turnpike (FTP), 178,189
Barge Canal; 137	Coquina Building Material; 165	25 mile Strip Maps; 66
Bee Line Expy; 80	Corkscrew Swamp, Name; 154	Administration; 189
Belz Outlet Mall; 89	Cowboys; 95	Coin System; 190
Bernard Castro; 136	Crab Trap II; 144	Exit Services; 189
Big 'I'; 165	Cracker, Florida; 88,95,132	HEFT; 76,161,190
Big Cypress; 155,158	Crostown Expy; 11,35,98,143	History; 189
Big Foot Monster; 105	Cuban Bread; 184	Names; 189
Billie Swamp Safari; 160	Dade Battlefield; 140	Service Plazas; 190
Blackwater River SP; 117	Dade, Maj. Francis; 139-140,161	Spur SR91; 76
Blue Angels	Dania Beach Hurricane; 184	Ticket System; 190
A4-C Skyhawk; 117	Daniel Boone, Florida Walk; 117	Toll Plazas; 190
Atrium; 121	Daytona Beach; 172-173	Ford, Henry; 152
Blue Springs SP; 87	De Land; 87	Fort Barrancas; 122
Blue Star Memorial Highway; 125	De Soto, Hernando,	Buried Alive; 123
Boca Ciega; 189	Anhaica; 108-109,146	Fort Caroline; 164
Boca Grande; 150	County; 149	Fort Clinch SP; 161
	Explorer; 146	Fort De Soto & Egmont Key; 188
	Landing; 146	Fort Lauderdale; 161,182-184
	Napitaca; 103	

Building the Visual Vocabulary

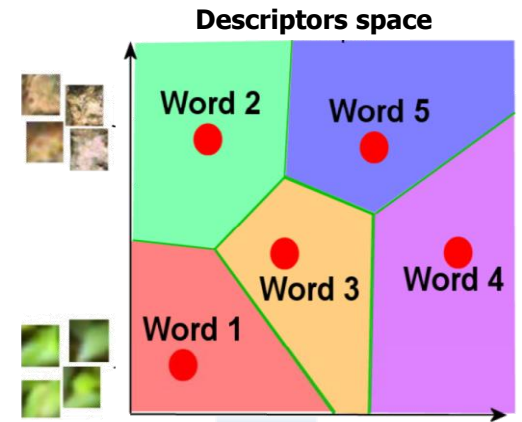
Image Collection



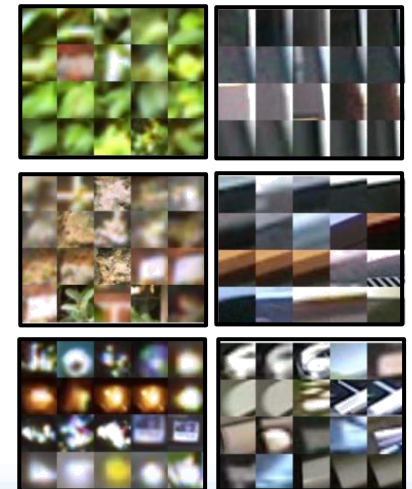
Extract Features



Cluster Descriptors

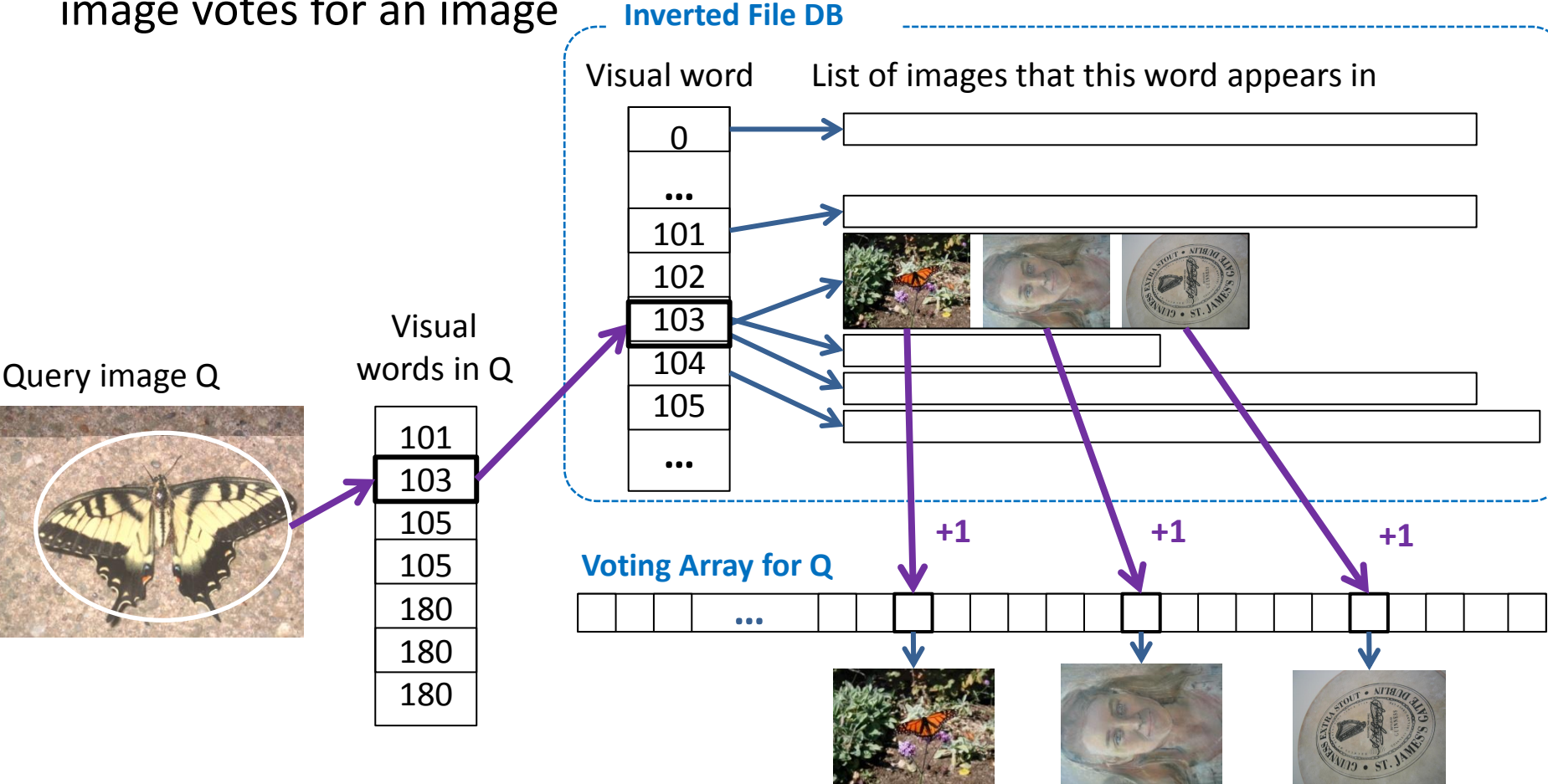


Examples
of
Visual
Words:

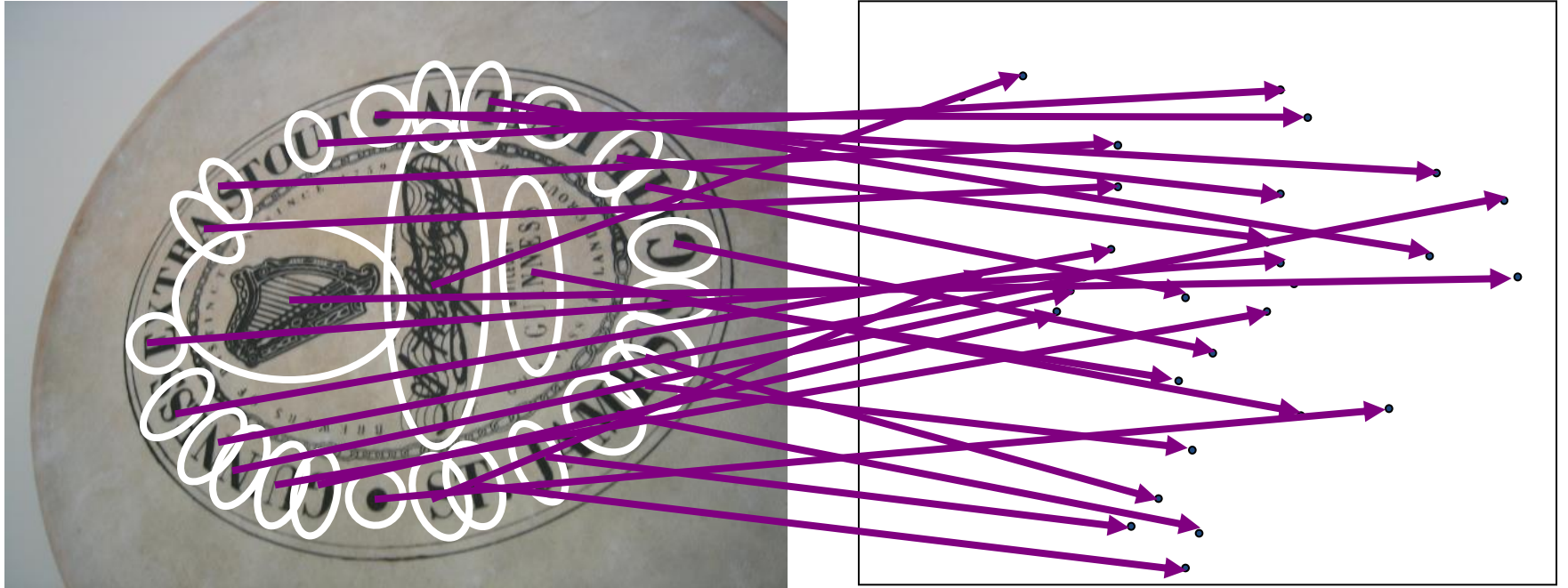


Inverted File index

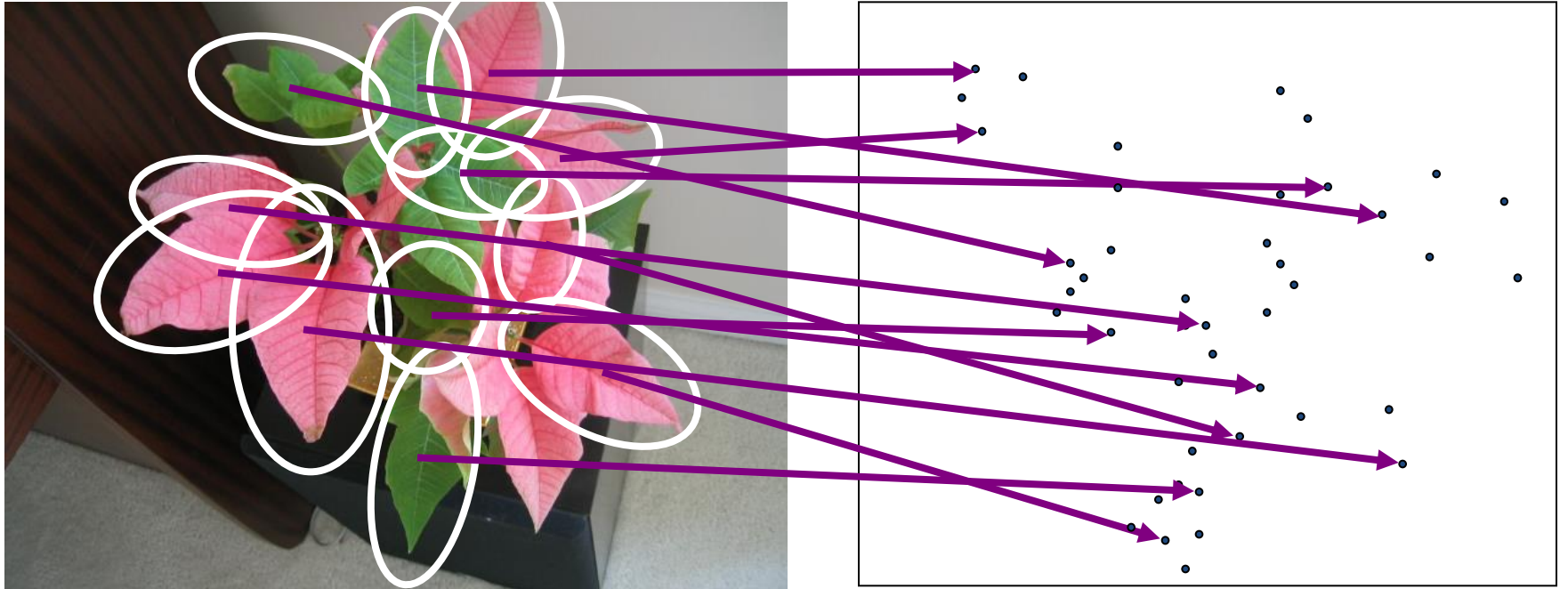
- Inverted File Data Base (DB) lists all possible visual words
- Each word points to a list of images where this word occurs
- Voting array: has as many cells as images in the DB – each word in query image votes for an image



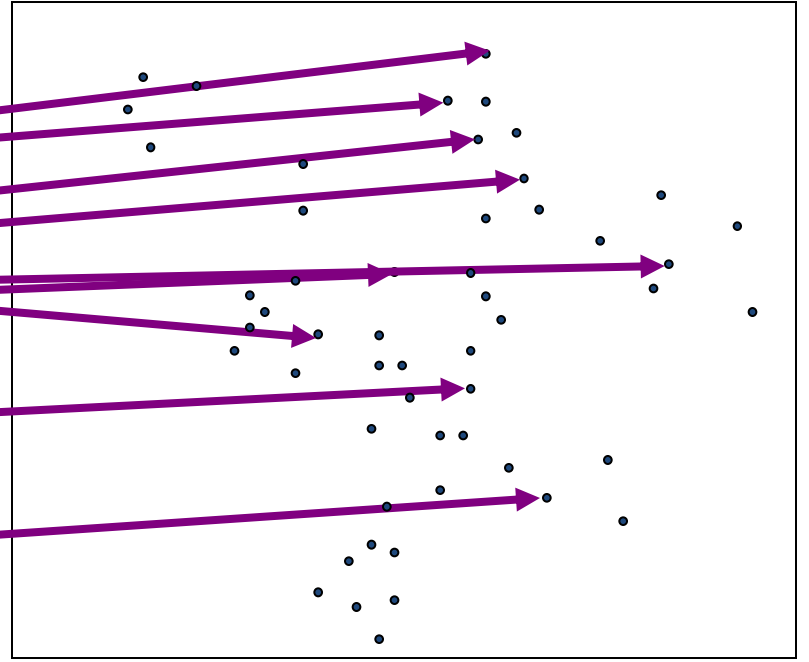
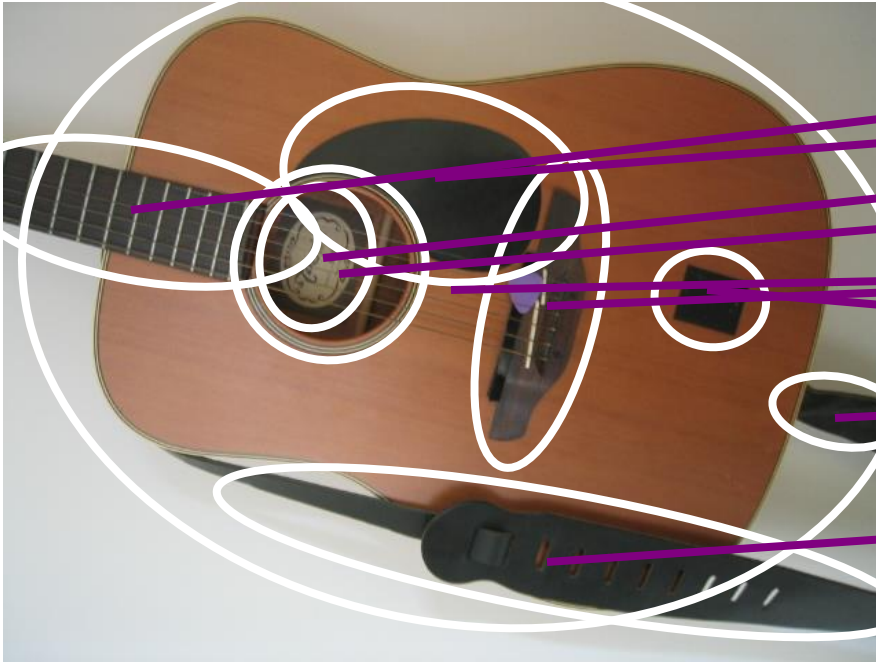
Populating the vocabulary



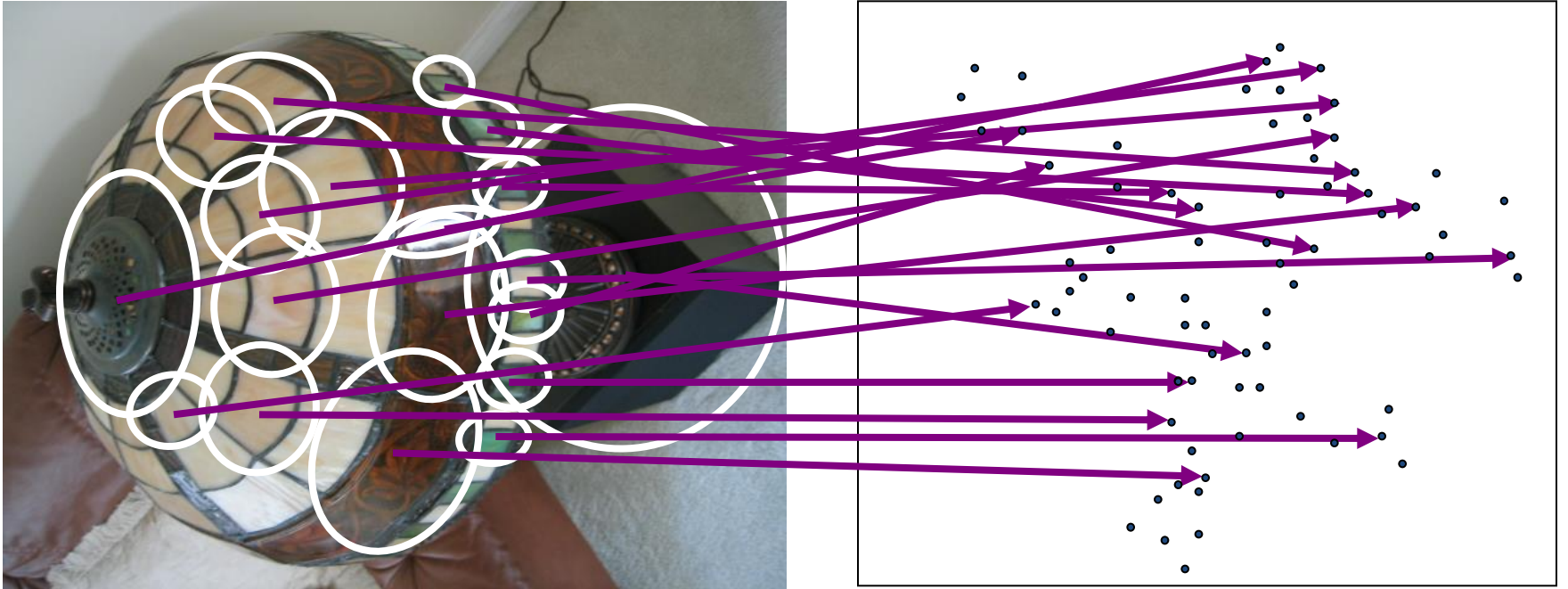
Populating the vocabulary



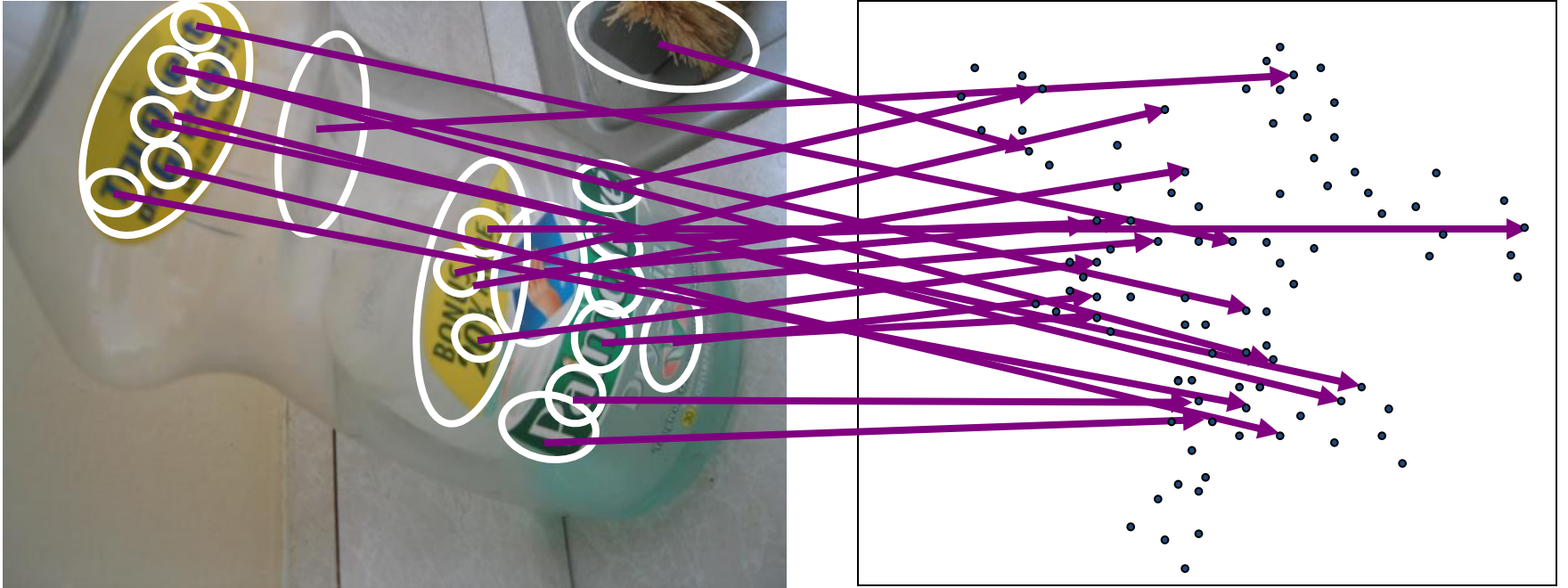
Populating the vocabulary



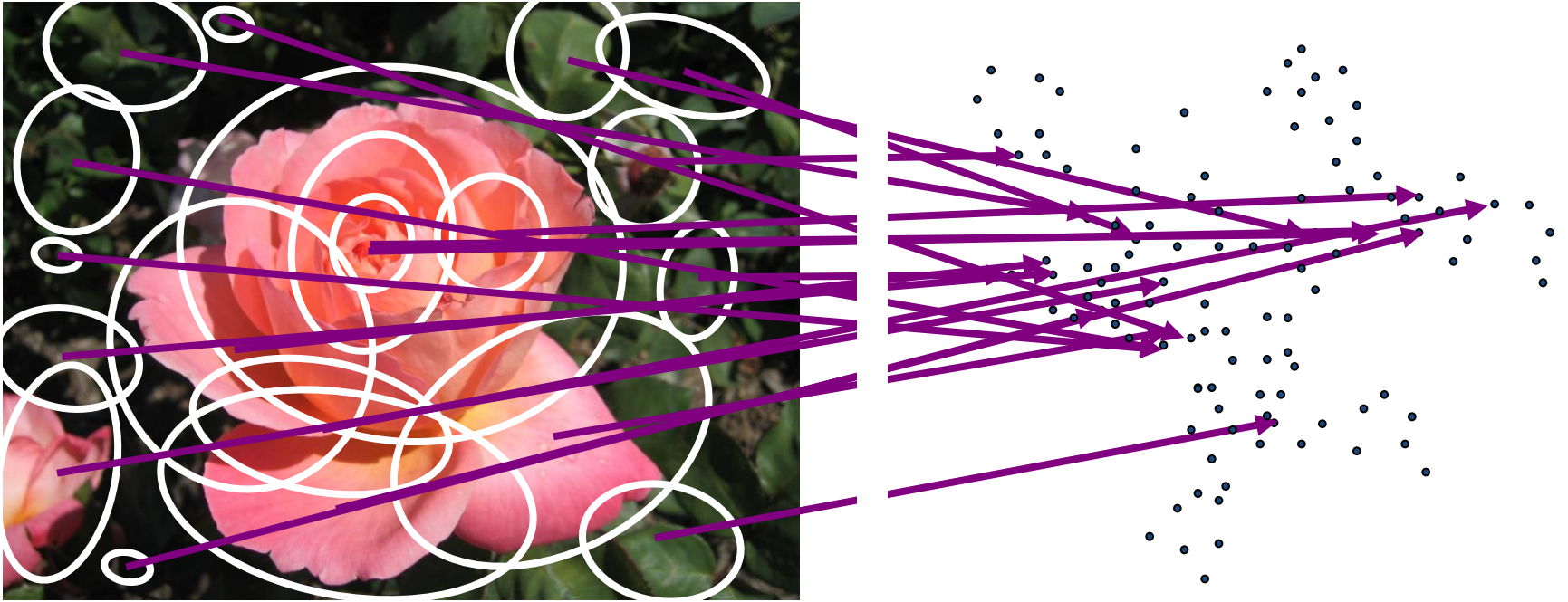
Populating the vocabulary

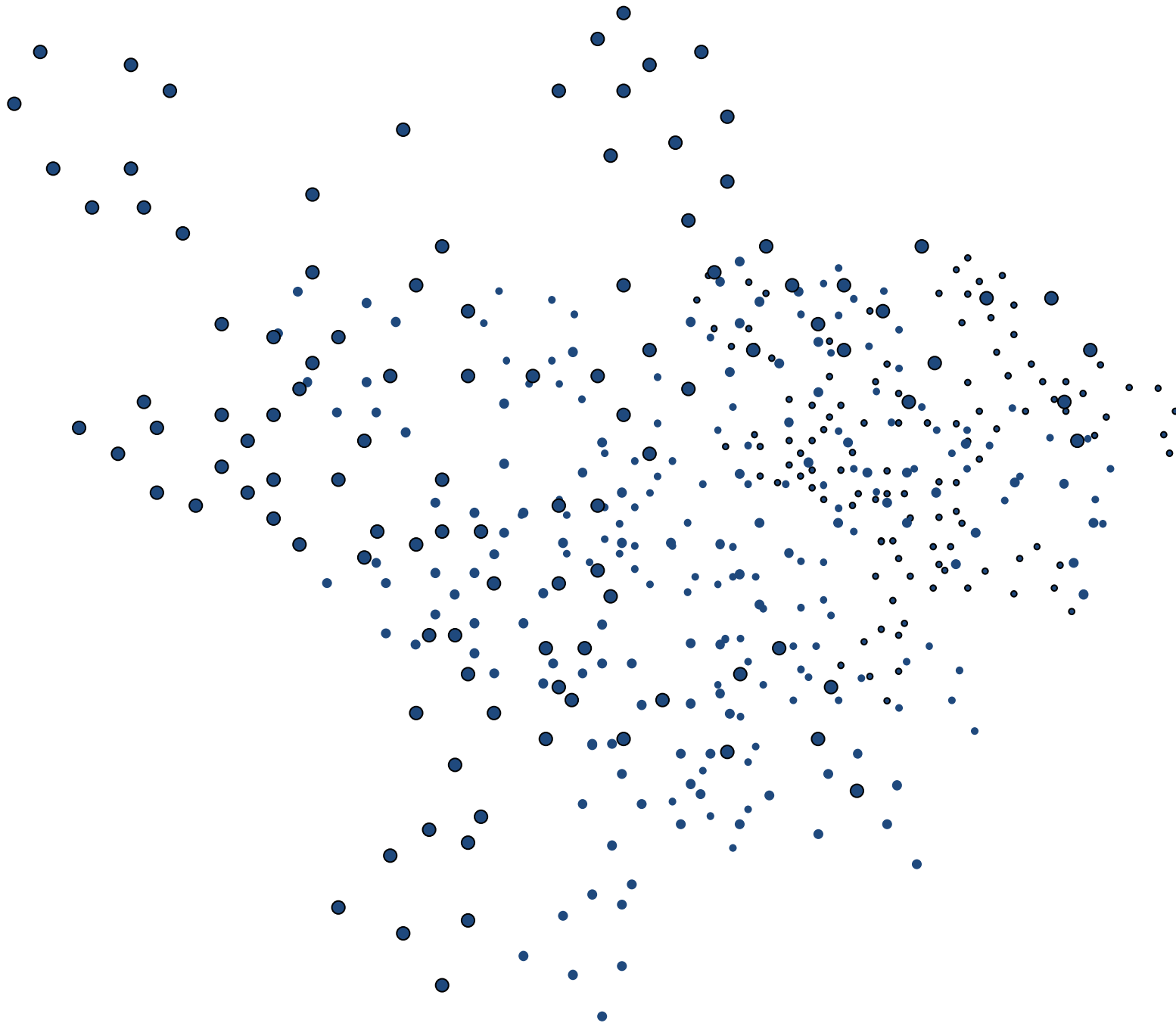


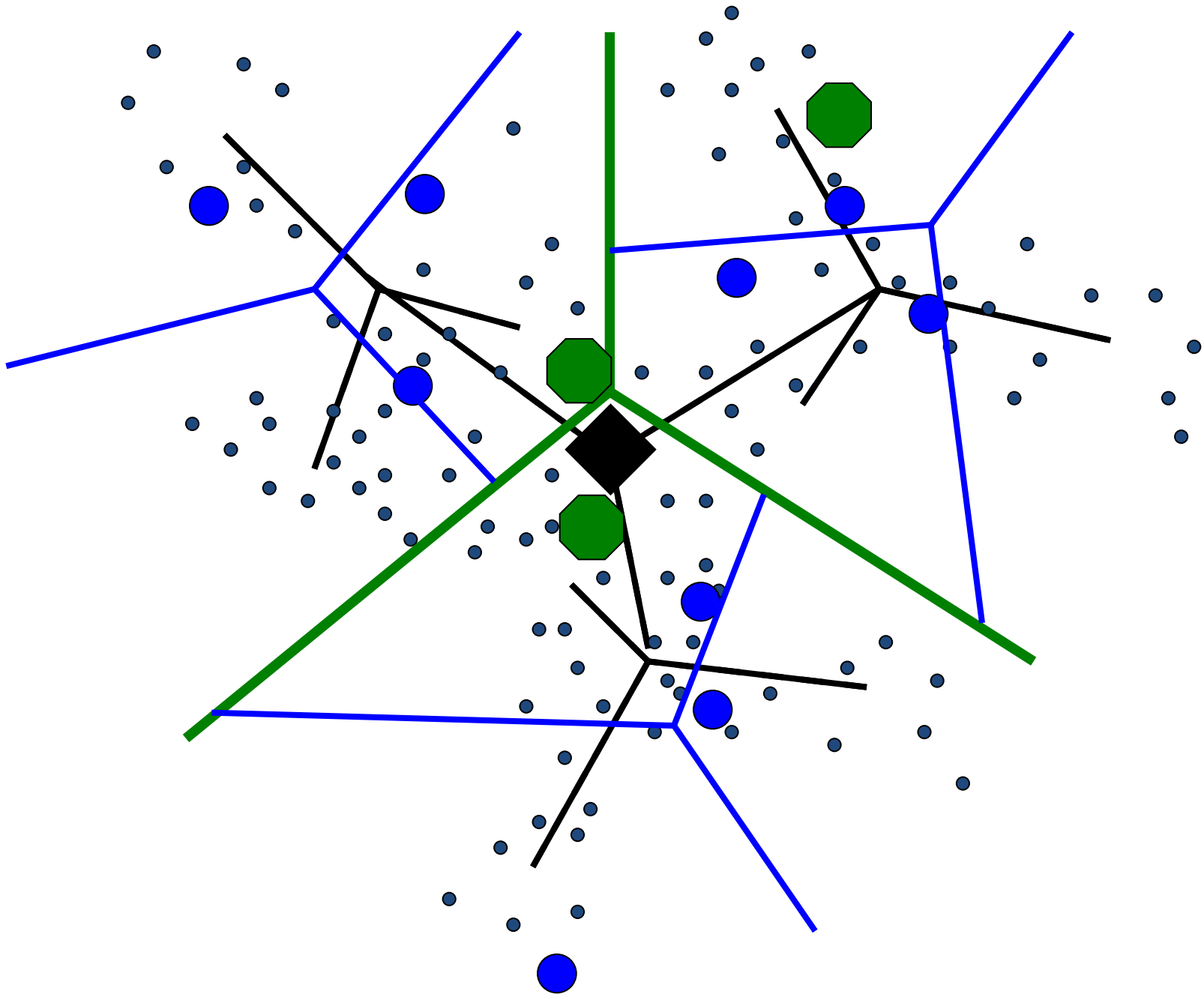
Populating the vocabulary

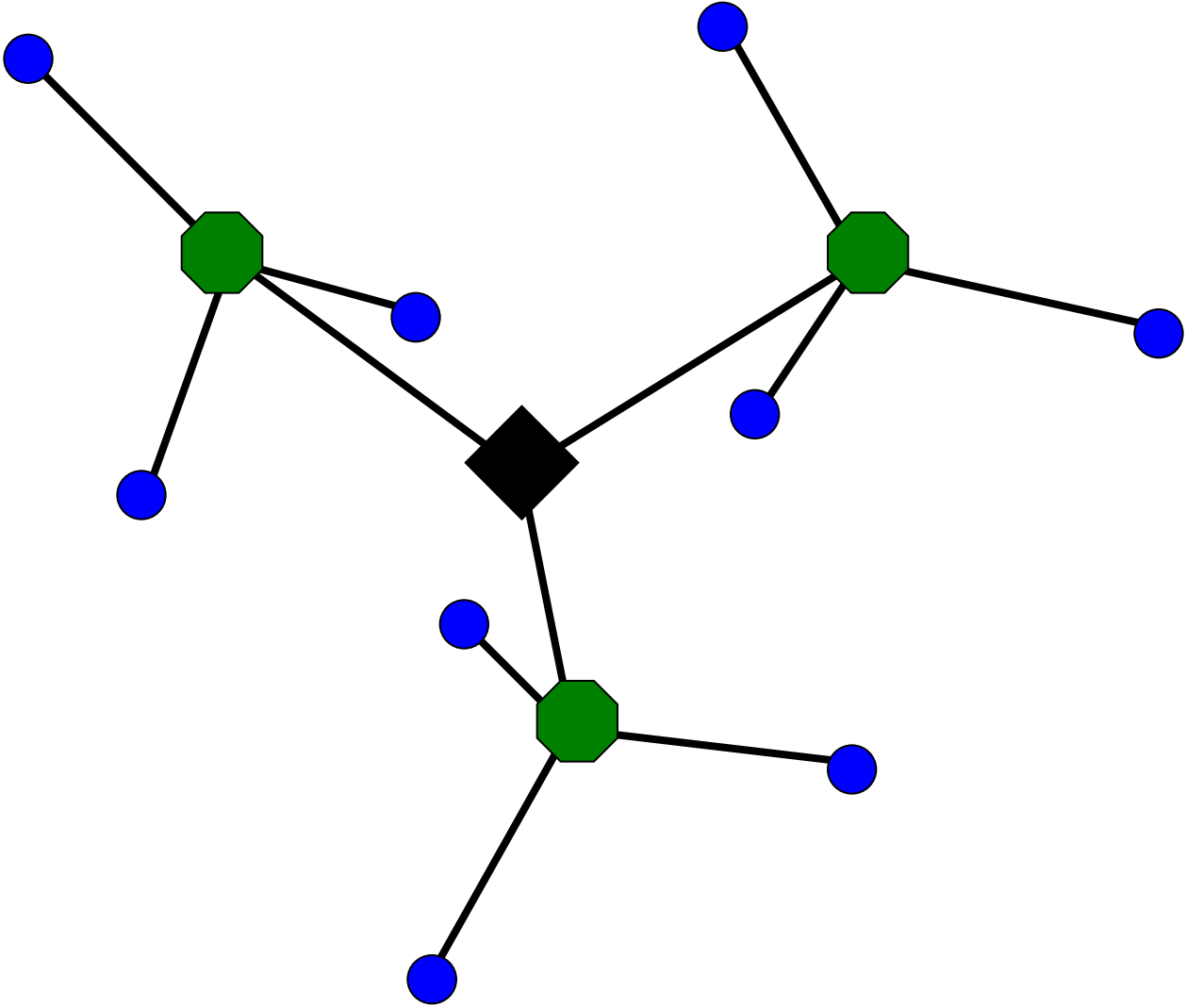


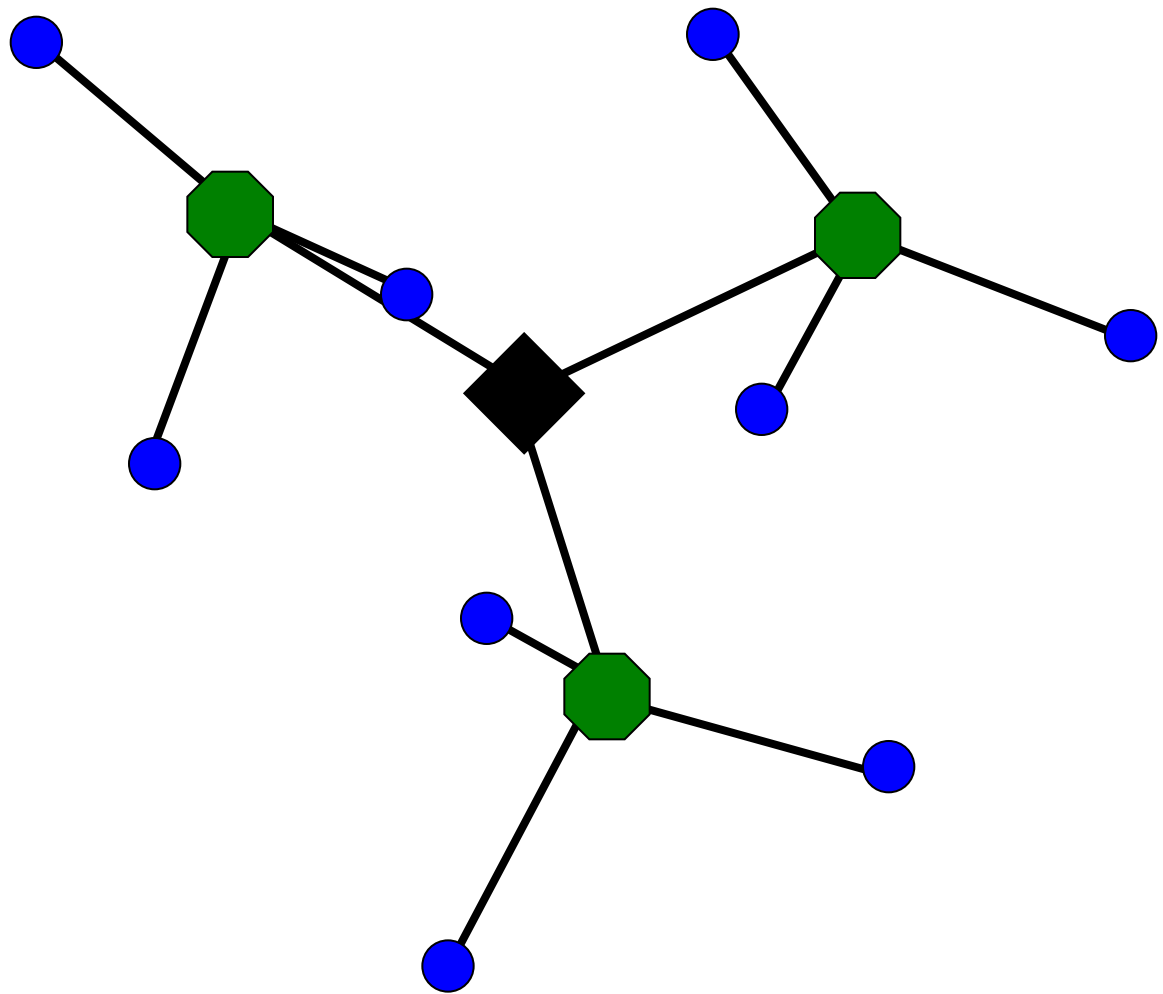
Populating the vocabulary

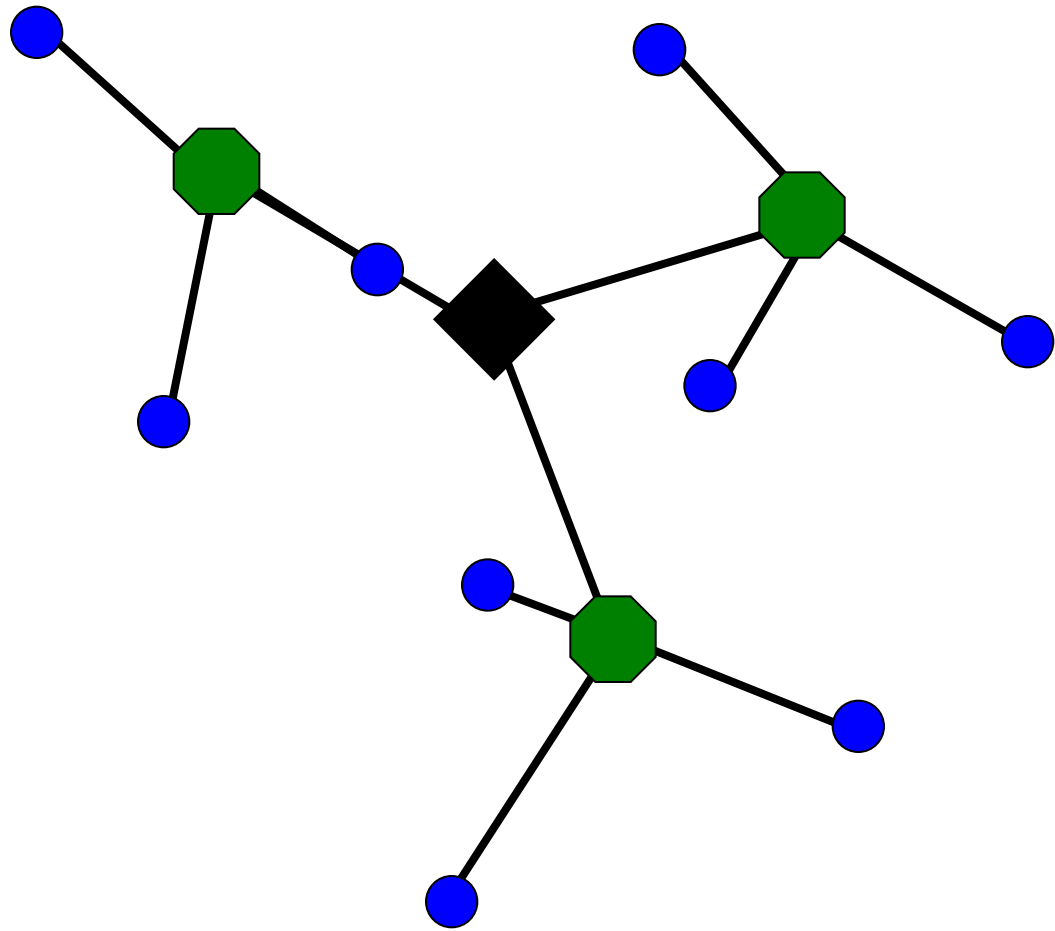


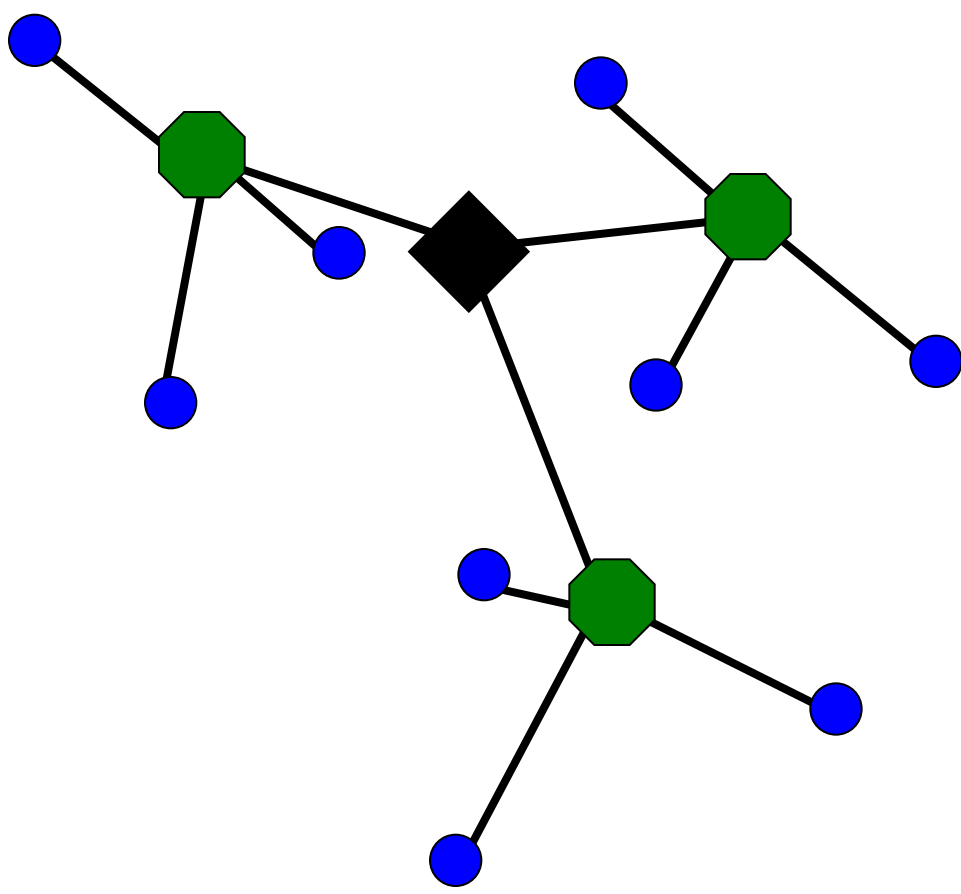


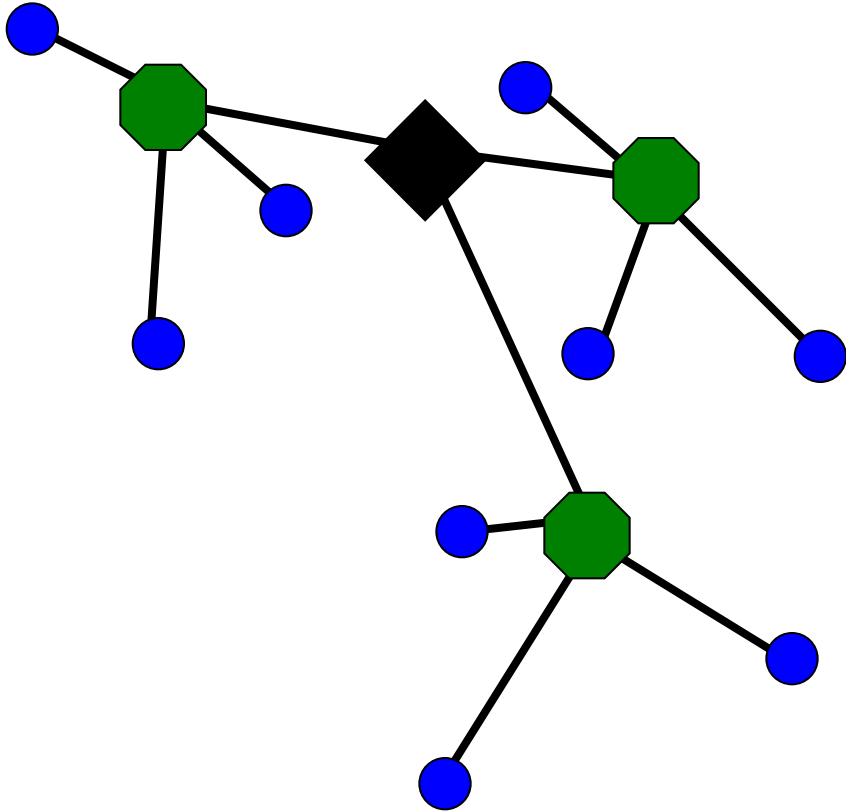


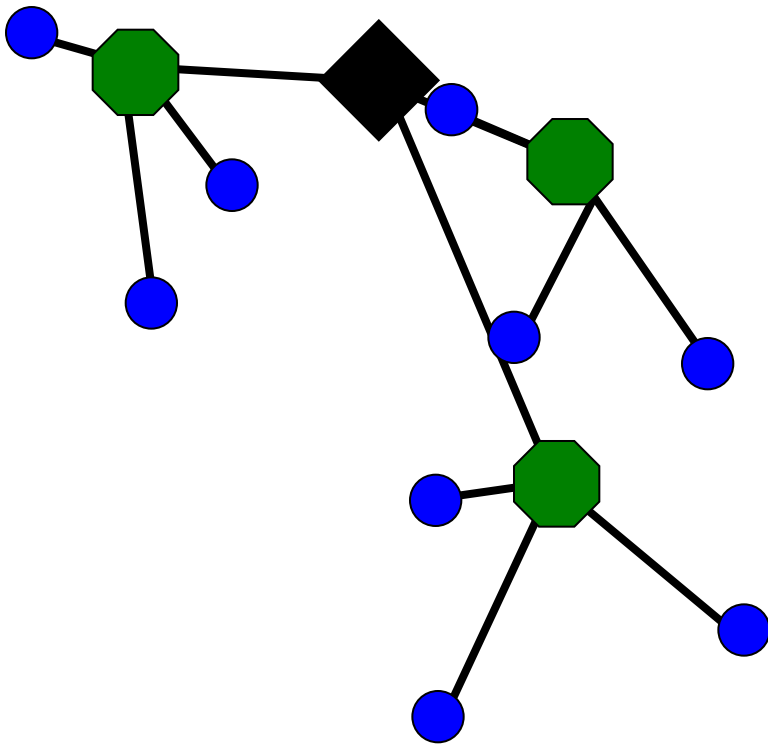


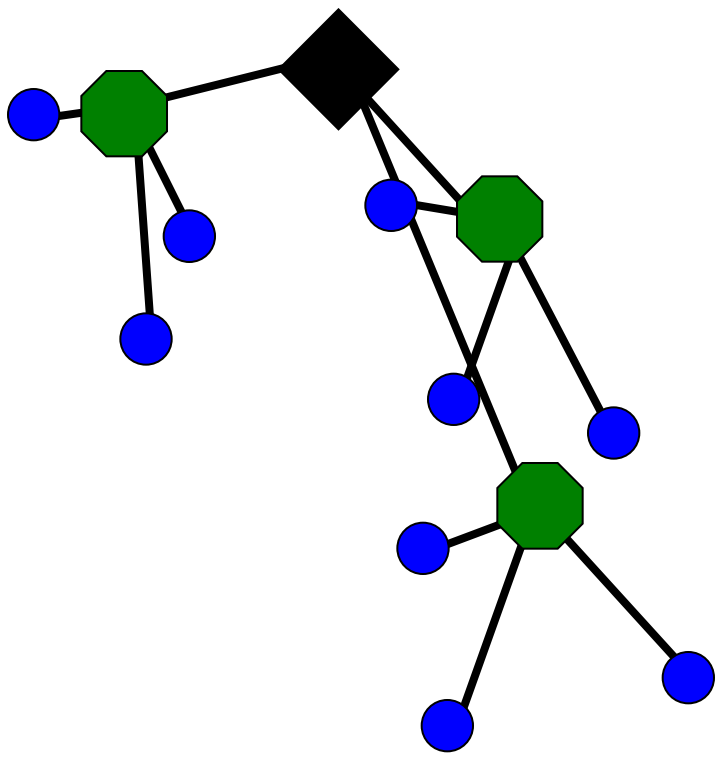


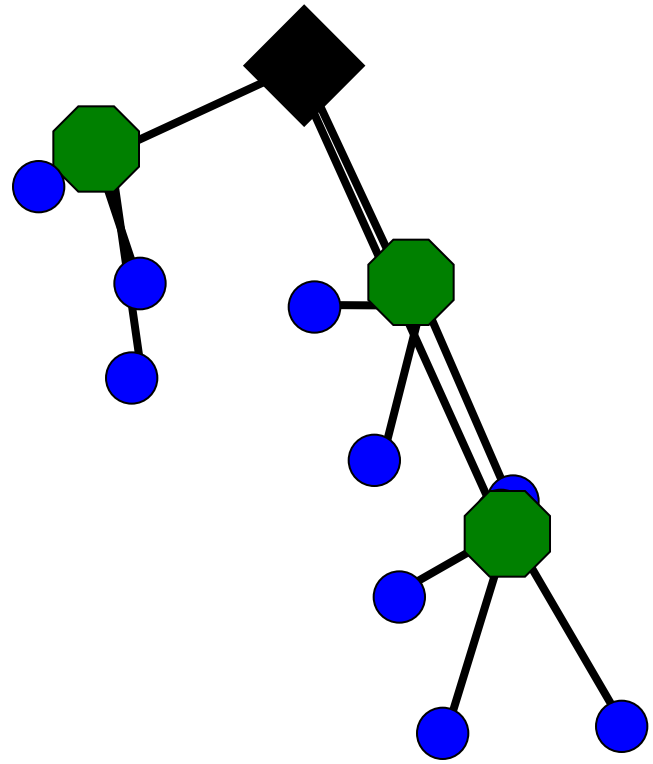


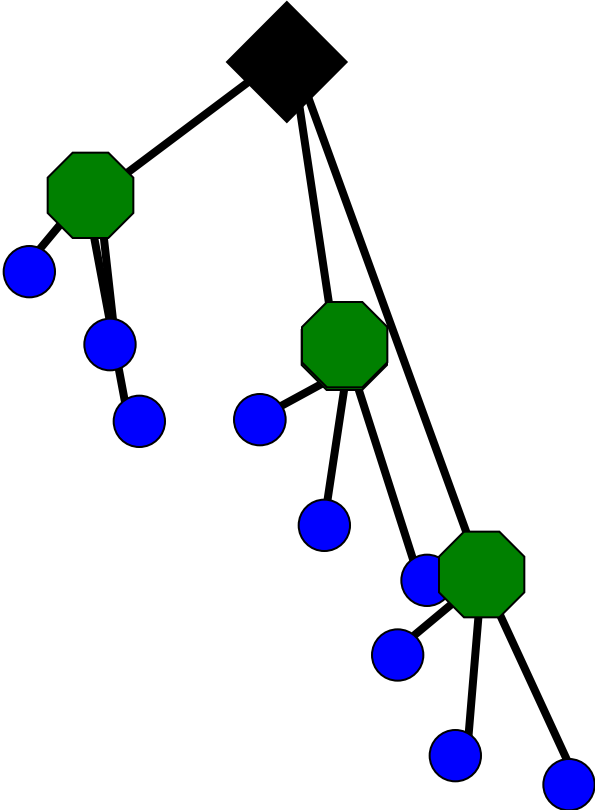


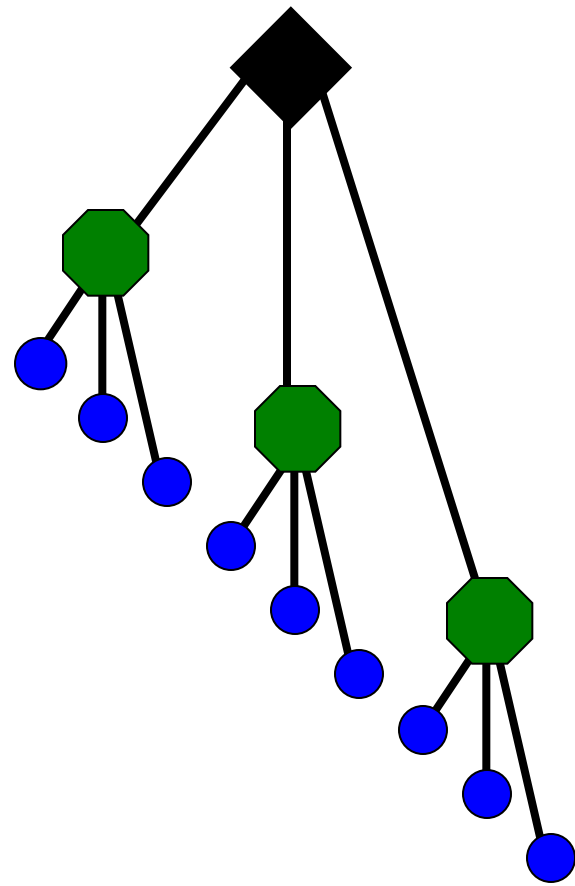




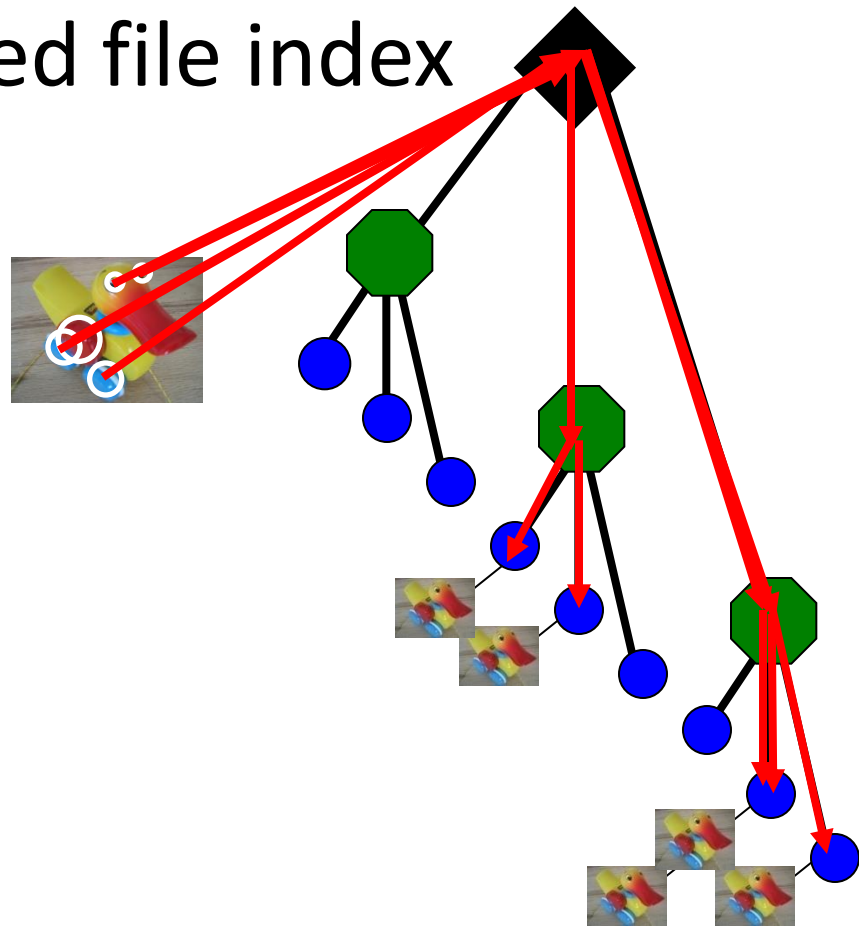




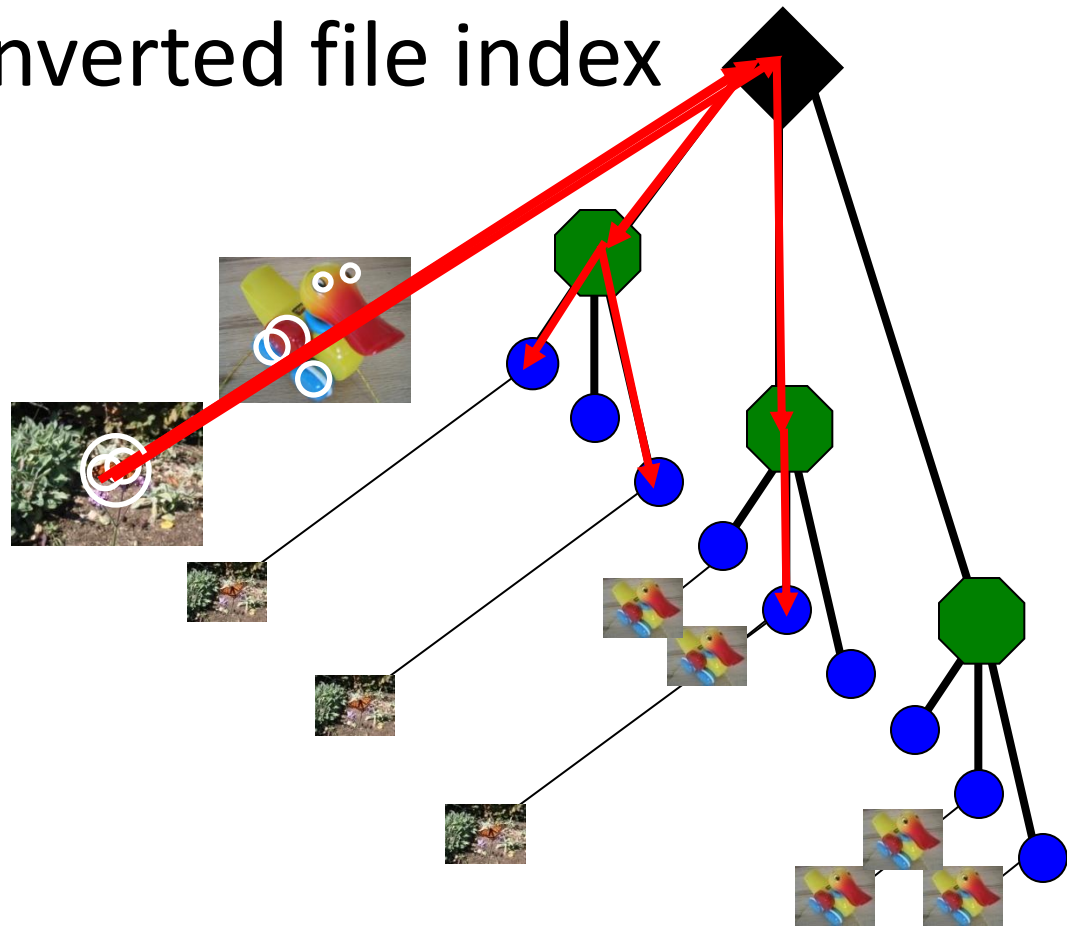




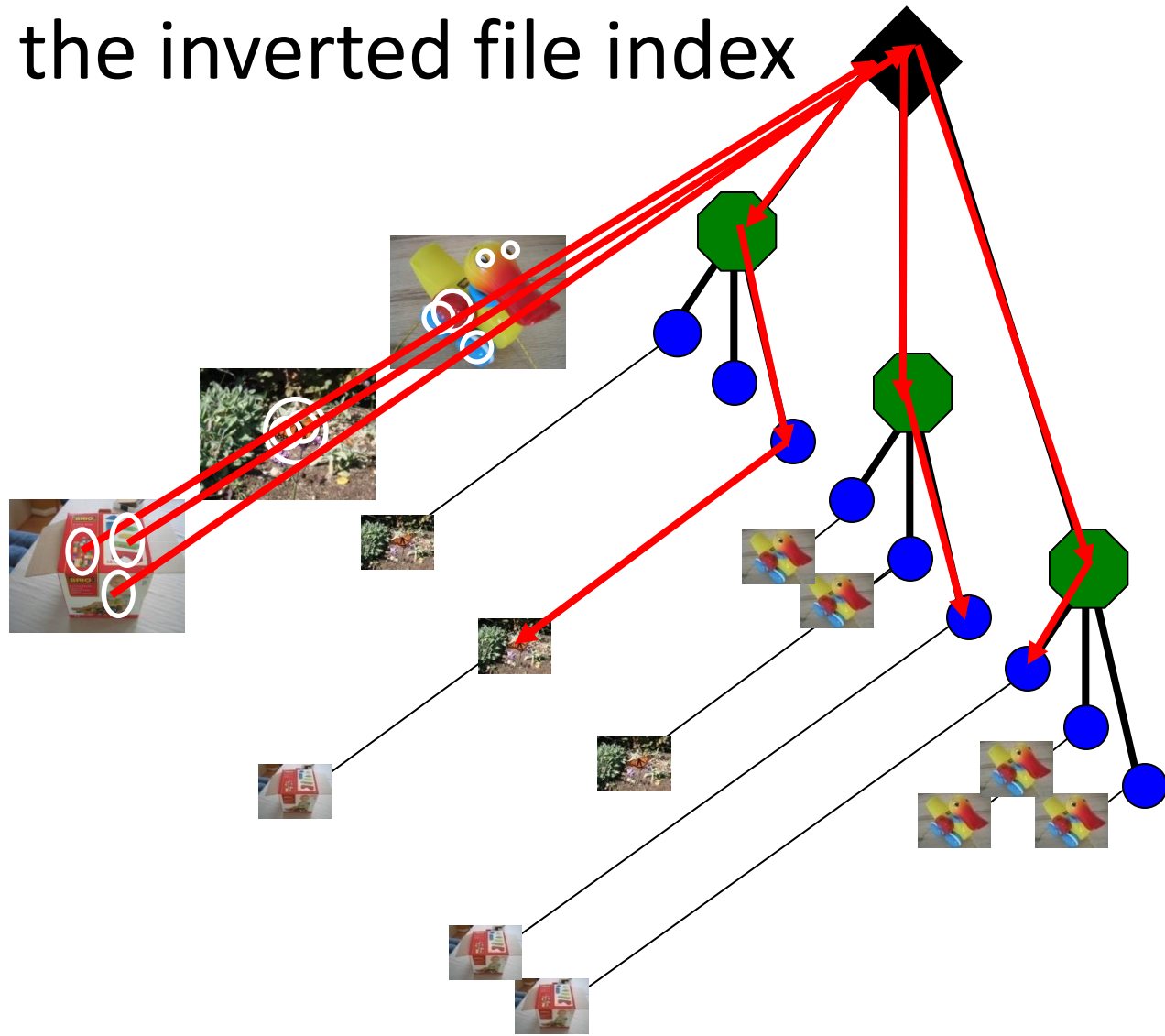
Building the inverted file index



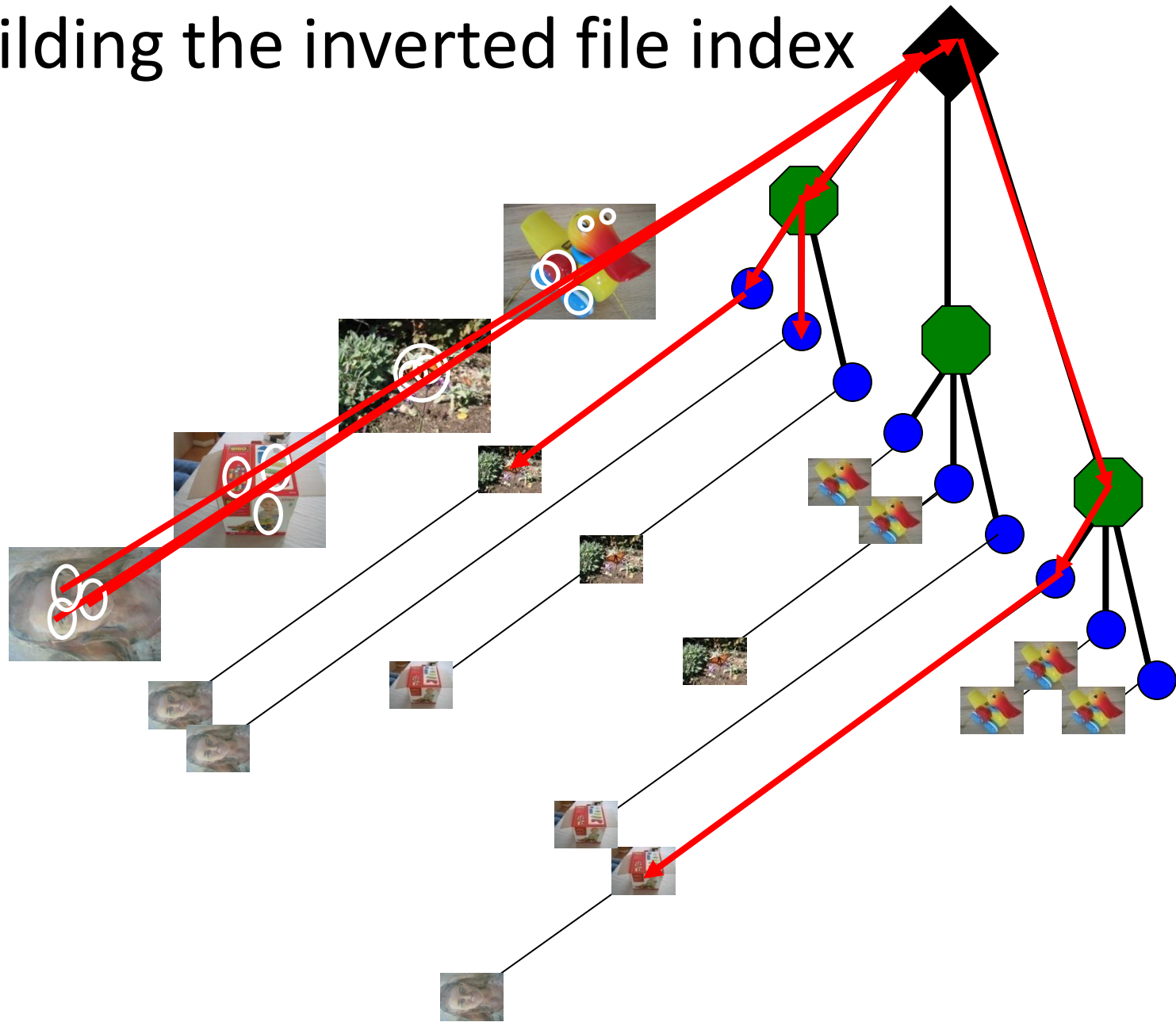
Building the inverted file index



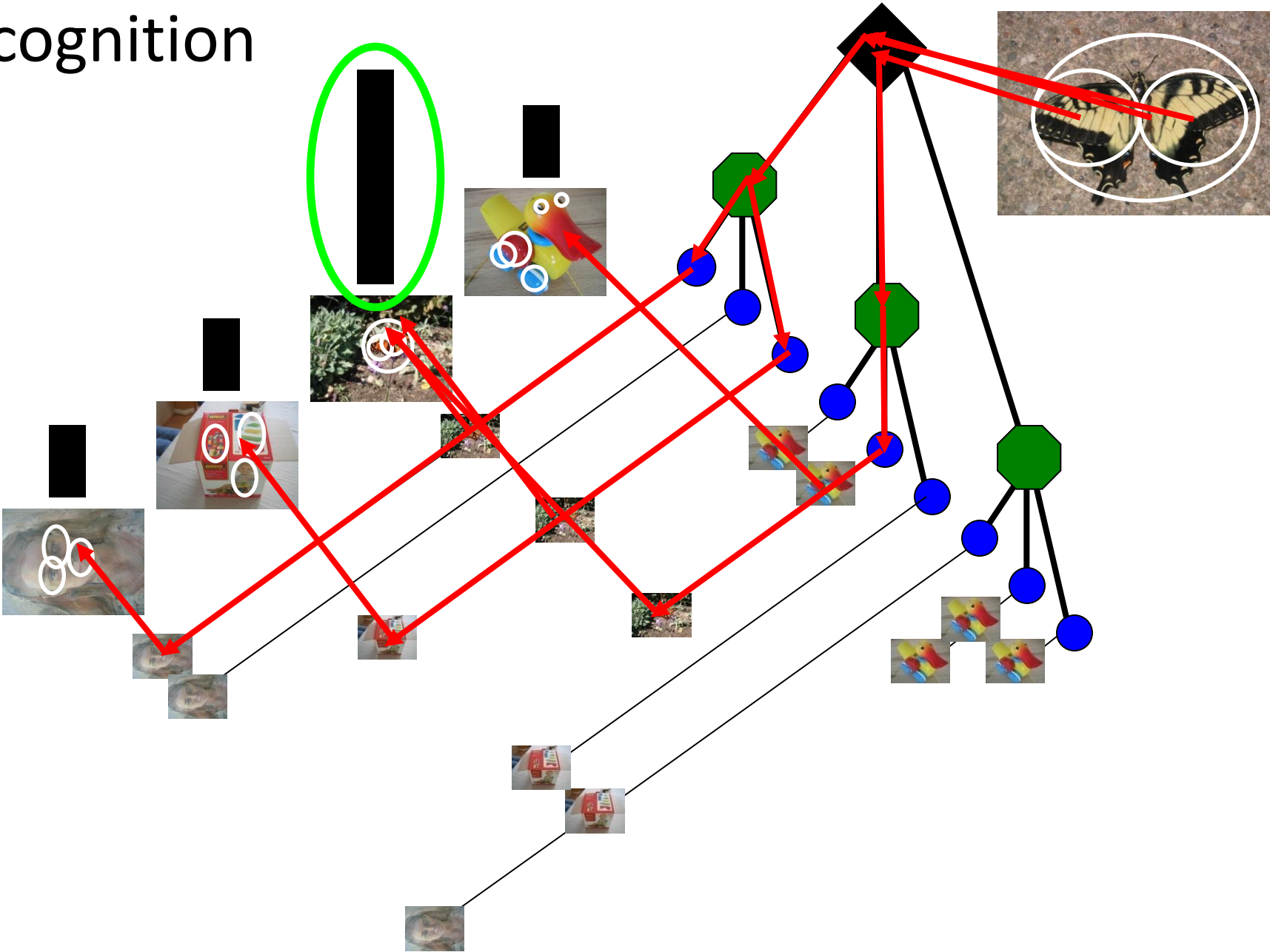
Building the inverted file index



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

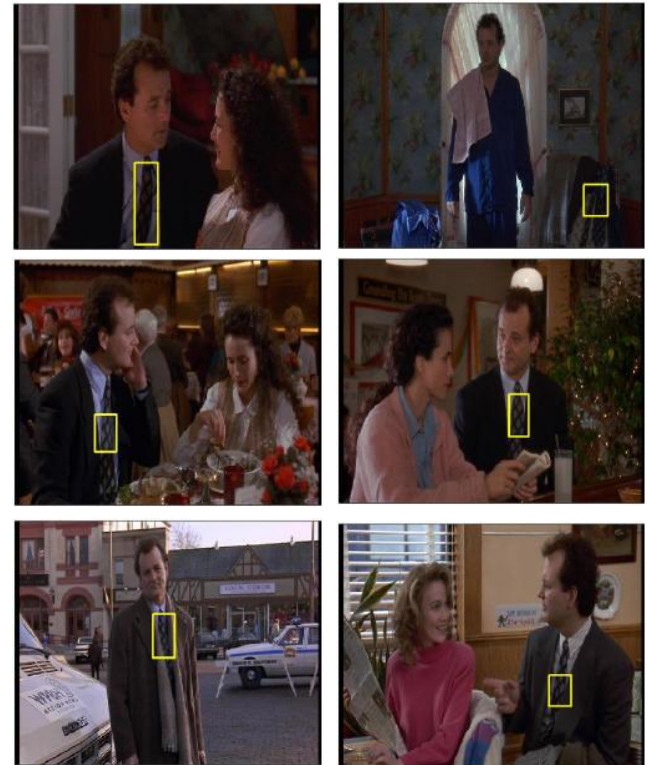
Sivic & Zisserman, ICCV 2003

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

Query region



Retrieved frames



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:
 - $P(\text{being at a known place})$
 - $P(\text{being at a new place})$
- Very high performance
- Binaries available [online](#)
- [Open FABMAP](#)

