AEROSP 740 - Visual Navigation for Autonomous Aerial Vehicles (VNA2V)

Unmute Start Video

Lectures start at 1:00pm EST

Vasileios Tzoumas

Lecture 23

COLLEGE OF ENGINEERING
AEROSPACE ENGINEERING
UNIVERSITY OF MICHIGAN

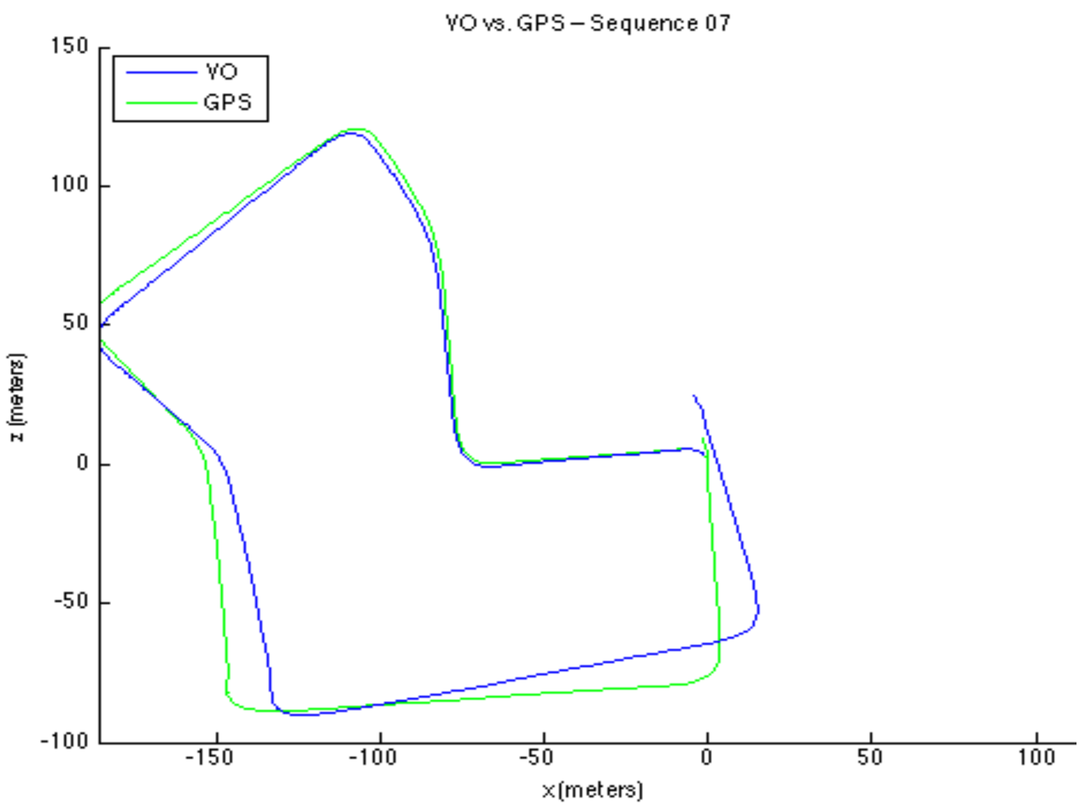
To ask questions:

Raise Hand or Chat

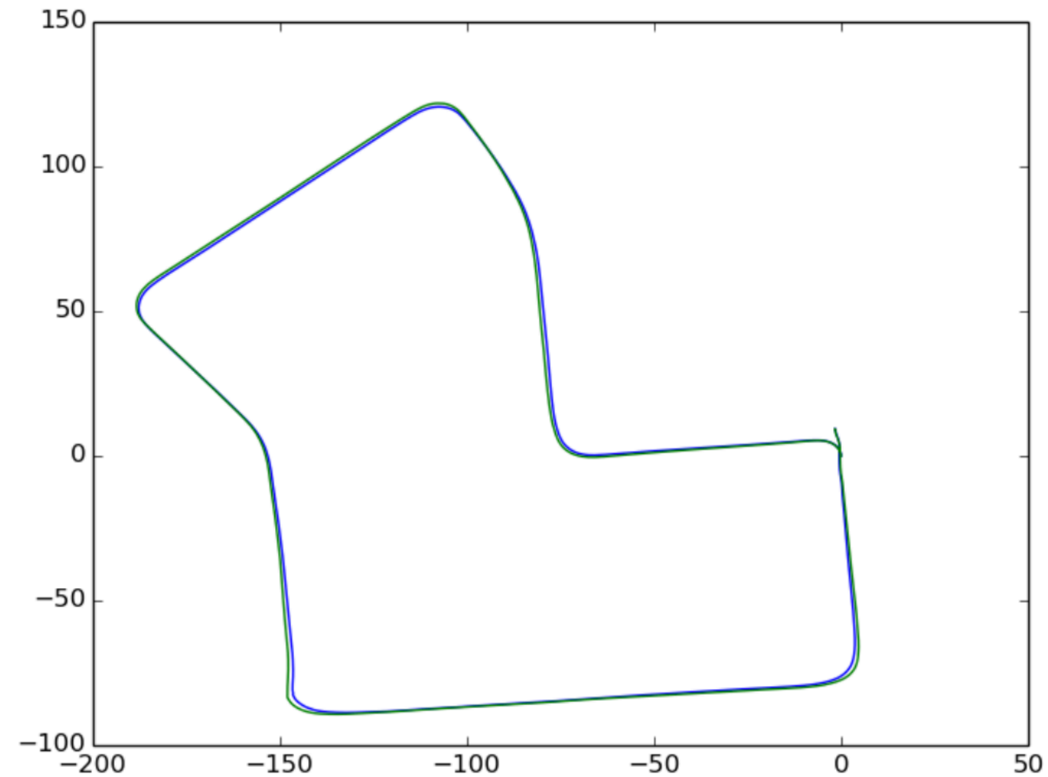
Based on slides made by Luca Carlone @ MIT AEROASTRO

Next Week: SLAM

Visual odometry



SLAM



SLAM requires:

- place recognition => loop closure detection

and / or

- Object detection => landmark detection

Today

- **Place recognition - Bag of Words**
- **Object detection / recognition**

Visual Place Recognition: A Survey

Stephanie Lowry, Niko Sünderhauf, Paul Newman, *Fellow, IEEE*, John J. Leonard, *Fellow, IEEE*, David Cox, Peter Corke, *Fellow, IEEE*, and Michael J. Milford, *Member, IEEE*

***Abstract*—Visual place recognition is a challenging problem due to the vast range of ways in which the appearance of real-world places can vary. In recent years, improvements in visual sensing capabilities, an ever-increasing focus on long-term mobile robot autonomy, and the ability to draw on state-of-the-art research in other disciplines—particularly recognition in computer vision and animal navigation in neuroscience—have all contributed to significant advances in visual place recognition systems. This paper presents a survey of the visual place recognition research landscape. We start by introducing the concepts behind place recognition—the role of place recognition in the animal kingdom, how a "place" is defined in a robotics context, and the major components of a place recognition system. Long-term robot operations have revealed that changing appearance can be a significant factor in visual place recognition failure; therefore, we discuss how place recognition solutions can implicitly or explicitly account for appearance change within the environment. Finally, we close with a discussion on the future of**

(a)

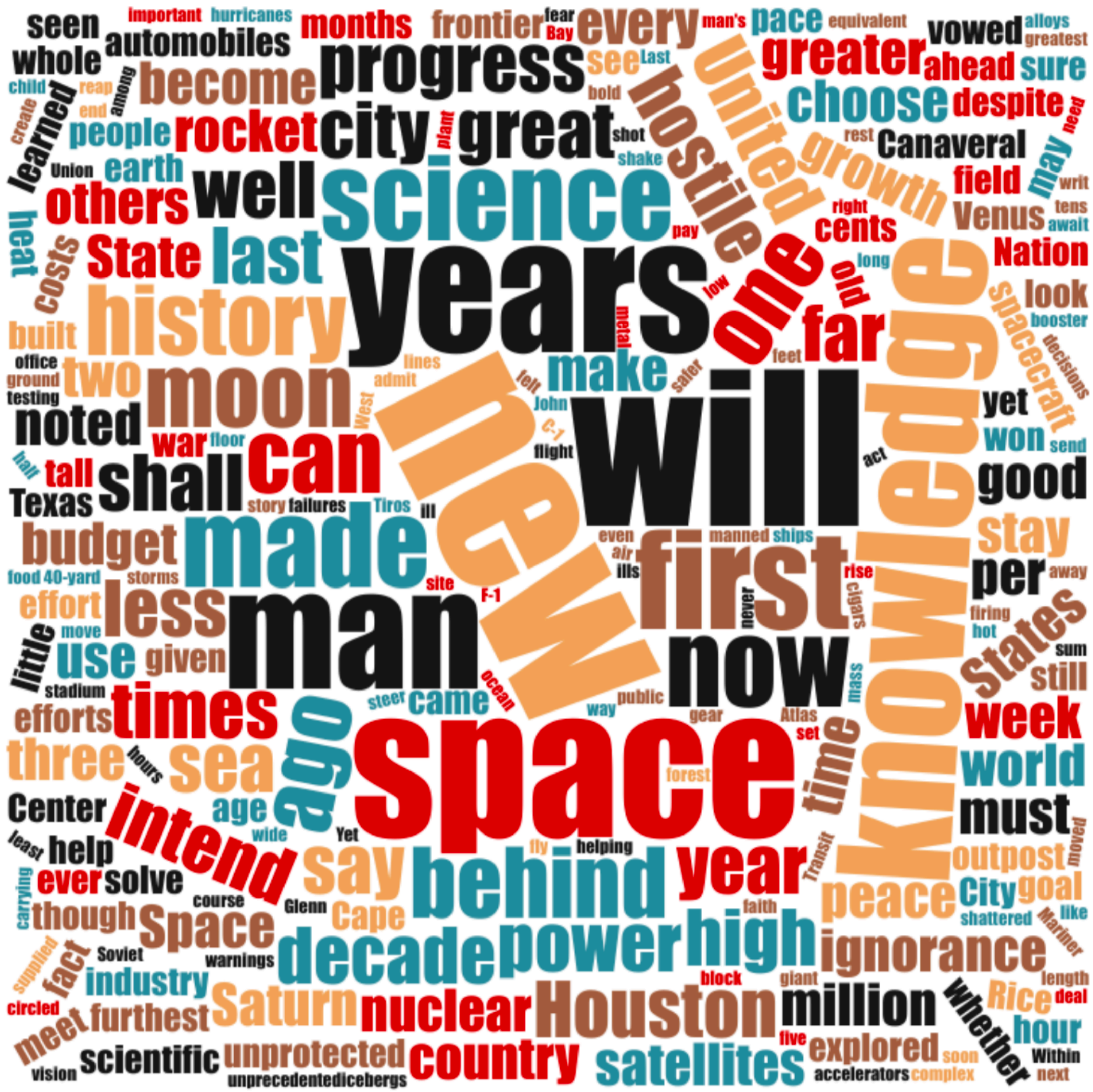
(b)

Fig. 1. Visual place recognition systems must be able to (a) successfully match very perceptually different images while (b) also rejecting incorrect matches between aliased image pairs of different places.

+ a few more recent papers

Guess the Speaker - Speech #1

Vocabulary	Freq
America	3
new	21
knowledge	8
first	8
years	9
made	6
now	6
history	5
man	8
science	6
will	20
space	18
Iran	0
behind	5
moon	5
sanctions	0



Guess the Speaker - Speech #1

Vocabulary	Freq
America	3
new	21
knowledge	8
first	8
years	9
made	6
now	6
history	5
man	8
science	6
will	20
space	18
Iran	0
behind	5
moon	5
sanctions	0

Credit: *NASA*

President John F. Kennedy: 'We choose to go to the moon'

October 9, 2017 | 2:07 PM EDT

President John F. Kennedy gave a speech at Rice University in 1962 about the quest to put a man on the moon. “We choose to go to the moon in this decade and do the other things, not because they are easy, but because they are hard," he said to a cheering crowd.

Guess the Speaker - Speech #2

Guess the Speaker - Speech #2

Vocabulary	Freq (#2)	Freq (#1)
America	12	3
new	10	21
knowledge	0	8
first	2	8
years	0	9
made	5	6
now	3	6
history	4	5
man	0	8
science	0	6
will	40	20
space	0	18
Iran	11	0
behind	1	5
moon	0	5
sanctions	4	0

Are they similar? Not quite …

For this particular vocabulary, the angle between the two vectors (histograms) is about 50 [deg]

Idea

Use the **distribution of a special set of words** to **efficiently** retrieve a query document (or find similar ones) from a **large** database

Bag of Words (Natural Language Processing)

Representation:

- Build a **vocabulary**
- Represent documents as distributions (histograms) over the vocabulary

$$\text{BoW} : \text{document} \mapsto \text{histogram}(\text{document}|\text{vocabulary})$$

Document Retrieval:

- Store the BoW histogram for every document in a DB
- Represent the **query** document as a histogram
- Compare the query histogram with histograms of documents in DB
- Return the best (or best n) matches
- Verify potential matches

Bag of Visual Words

Representation:

- Build a visual **vocabulary**
- Represent images as distributions (histograms) over the vocabulary

$$\text{BoVW} : \text{image} \mapsto \text{histogram}(\text{image}|\text{vocabulary})$$

Image Retrieval:

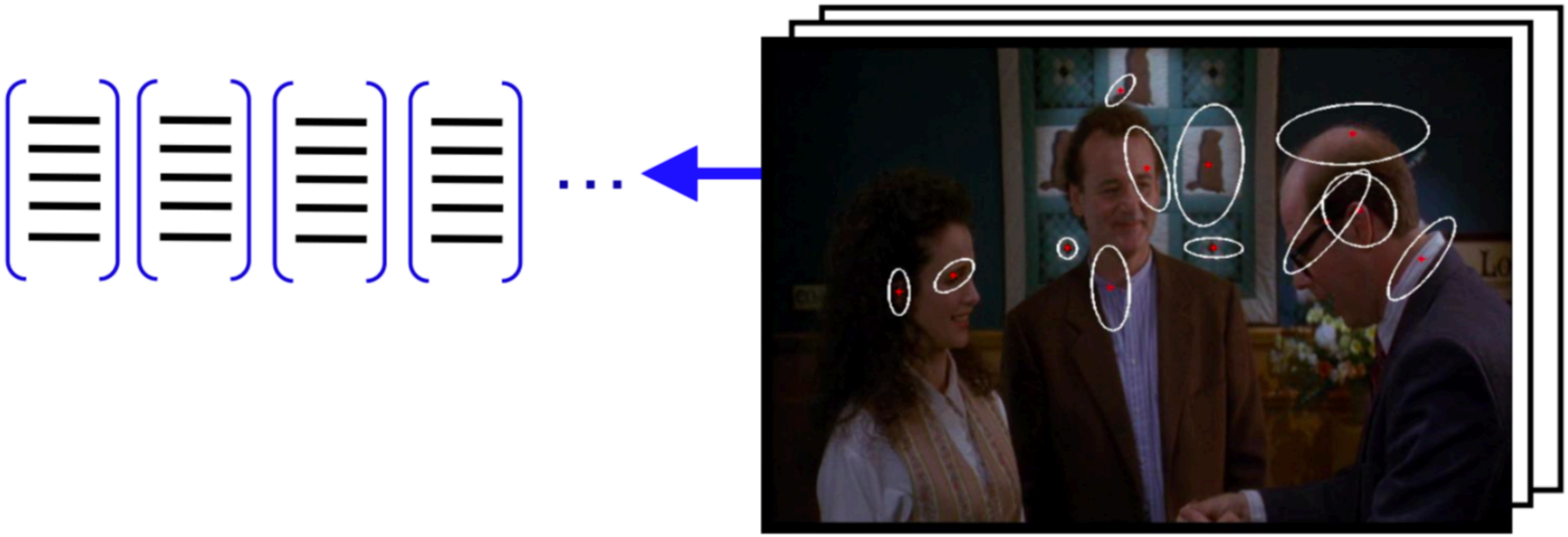
- Store the BoW histogram for every image in a DB
- Represent the **query** image as a histogram
- Compare the query histogram with histograms of images in DB
- Return the best (or best n) matches
- Verify potential matches using geometric/spatial verification (**RANSAC**)

Video Google: A text retrieval approach to object matching in videos 6758 2003
J Sivic, A Zisserman
Computer Vision, 2003. ICCV 2003. IEEE International Conference on, 1470

Build the Vocabulary

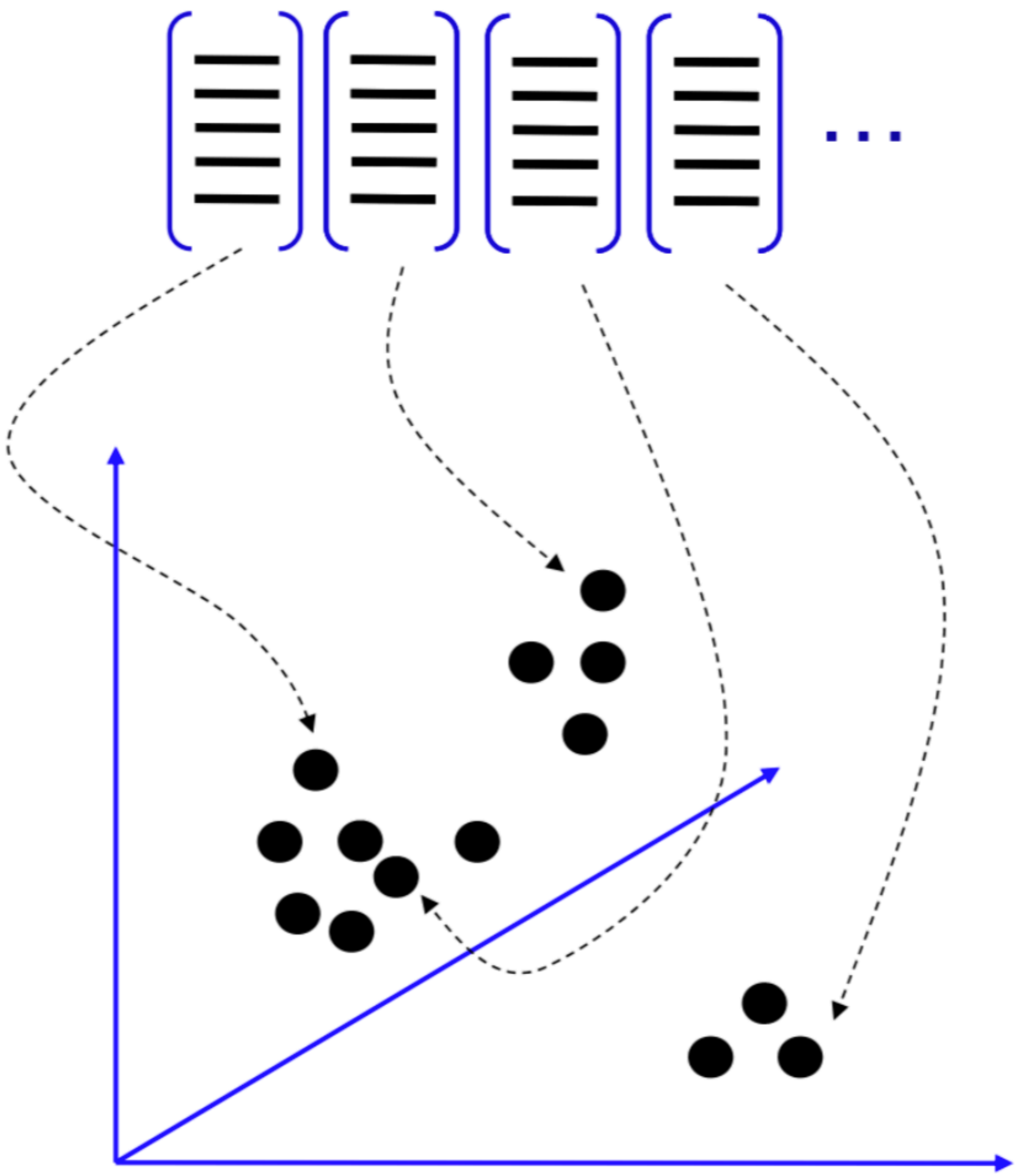
1. Pick a set of images
2. Extract keypoints and their descriptors from every image
   - Need to be fast, invariant to viewpoint variations, etc.
3. Cluster the descriptors into k clusters (using, e.g., k-means)
4. Pick the k cluster centers as your vocabulary

Extract Keypoints and Descriptors



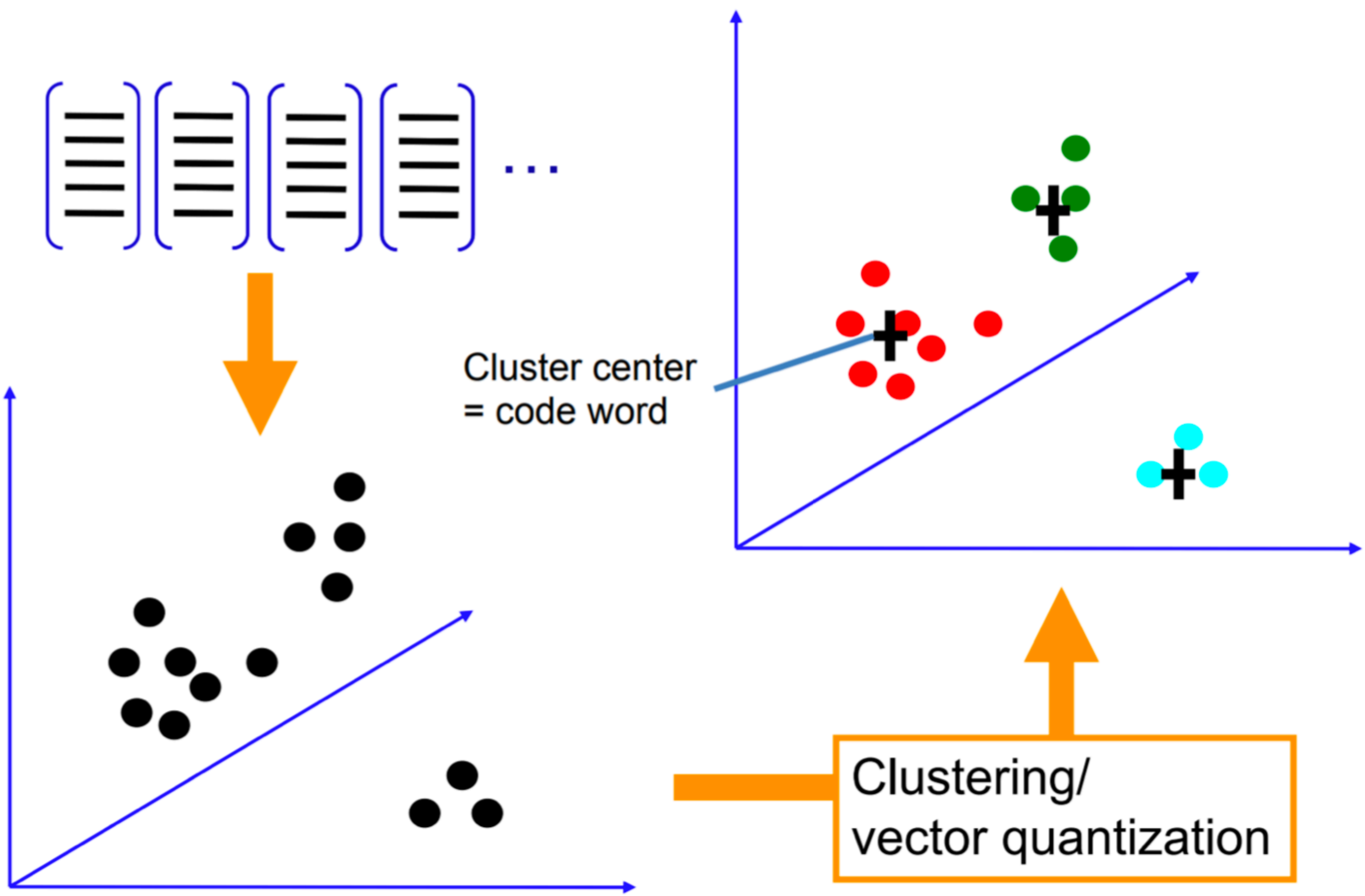
Credit: *Fei-Fei Li*

Descriptor Space



Credit: *Fei-Fei Li*

Cluster the the Descriptors to Build the Vocabulary



Credit: *Fei-Fei Li*

k-means Clustering

Find a k-partitioning (clustering) $\{\mathcal{C}_i\}_{i=1}^k$ for $\mathcal{X}$ by minimizing

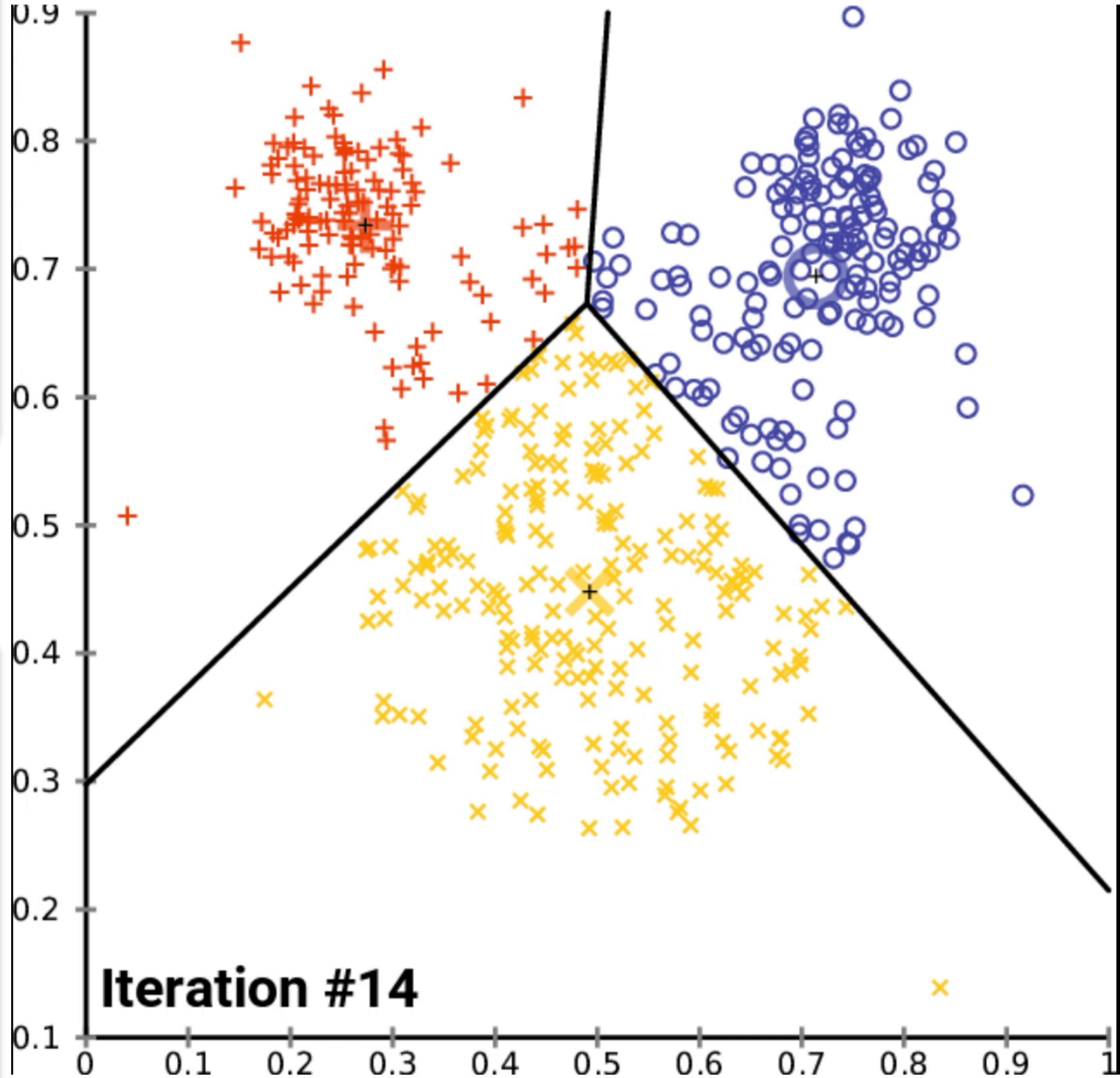
$$\sum_{i=1}^{k} \sum_{x \in \mathcal{C}_i} \|x - \mu_i\|^2$$

where μ_i is the mean of cluster $\mathcal{C}_i$

This is NP-hard - A simple idea:

Initialize cluster centers and, until convergence, alternate between

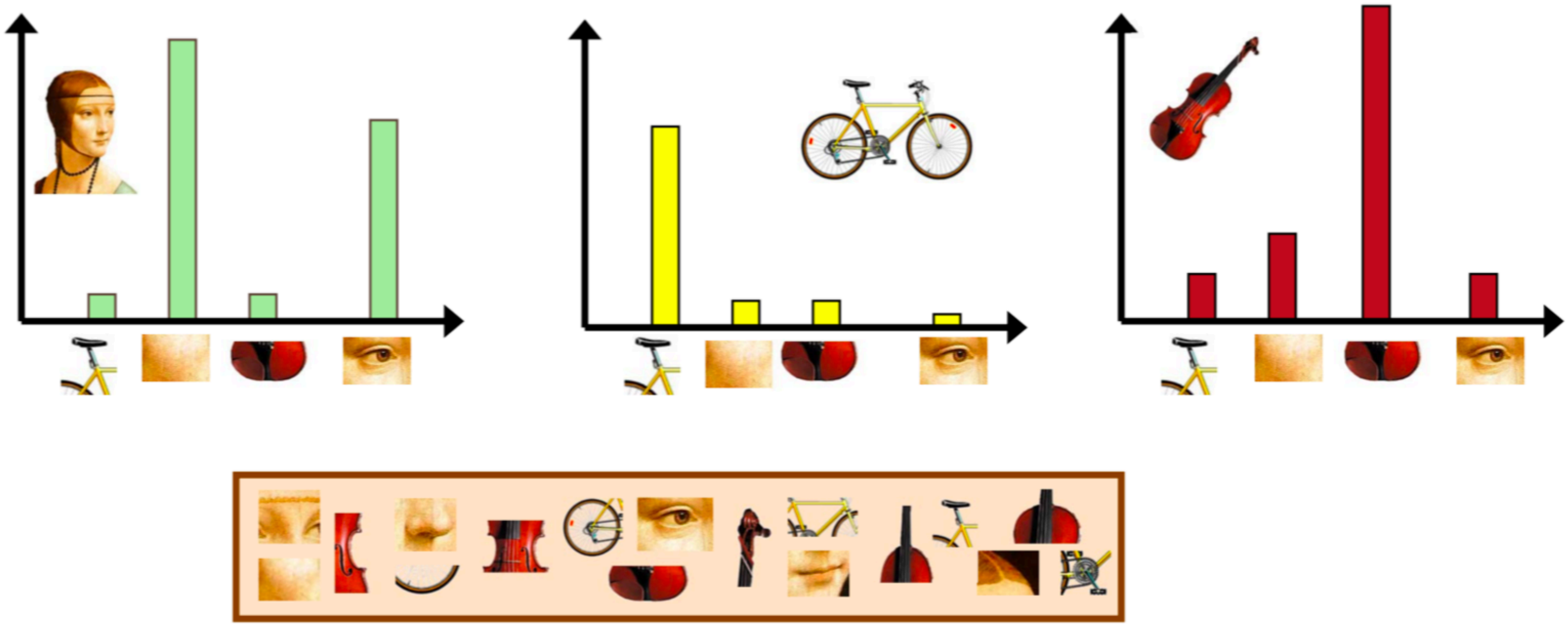
1. Associating points to nearest cluster centers
   $\Leftrightarrow$ solve for $\mathcal{C}_i$'s given μ_i's
2. Computing cluster centers given the associations
   $\Leftrightarrow$ solve for μ_i's given $\mathcal{C}_i$'s



Representation

Credit: *Fei-Fei Li*

Representation



Credit: *Fei-Fei Li*

Search in DB for a Query Image via Inverted File Index

- Given a query image, we need to search the database for similar images
- The database is large - in SLAM, it's always growing!

- Idea: for each visual word, maintain a list of images that contain that word
- Given a query image:
  1. Extract visual words (i.e., BoW representation)
  2. Look up the inverted file index (DB) to find documents containing same words
  3. Sort candidates based on weighted distance/similarity between BoW vectors

Index

TF-IDF Weights

- **Issue:** Relying on very common words could be misleading (e.g., "the", "is")
  - Both speeches contained many instances of "will" — any speech in the world would contain tons of these!
- On the other hand, unique/rare words are very informative
  - Not every presidential speech contains the word "moon"!
- **Solution:** For each word in the vocabulary, multiply its "term frequency" (TF) (i.e., histogram bar) by its "inverse document frequency" (IDF) in the (training) database

$$\text{IDF weight for word } i \triangleq \log\left(\frac{\#\text{ "documents"}}{\#\text{ "documents" that contain } i\text{th word}}\right)$$

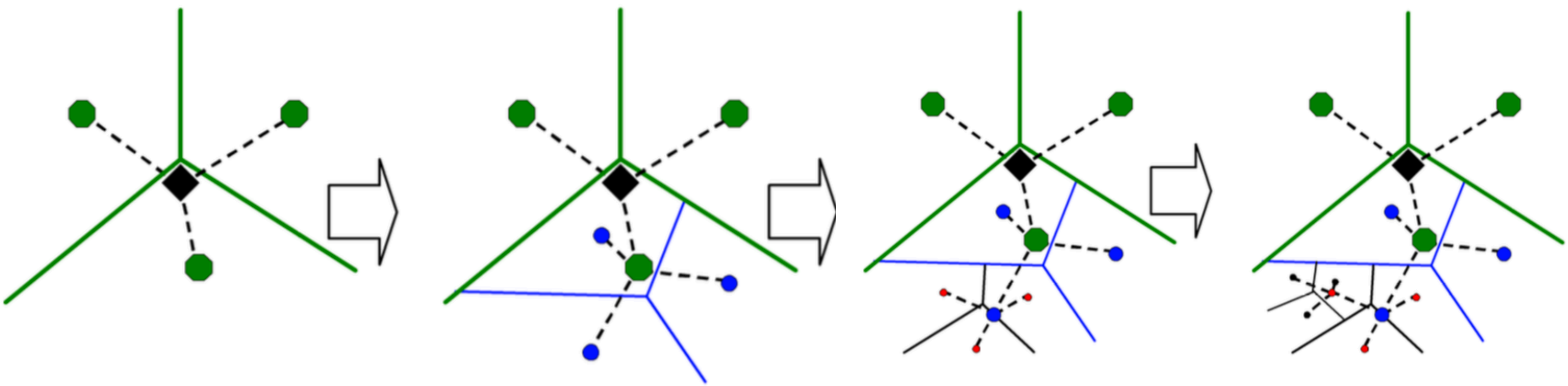
- Total weight for word $i = \text{TF}_i \times \text{IDF}_i$ (i.e., ith component of the BoW vector)
- Comparing two (very sparse) BoW vectors:

$$\text{dist}(v_{\text{query}}, v_{\text{DB}}) = \left\| \frac{v_{\text{query}}}{\|v_{\text{query}}\|} - \frac{v_{\text{DB}}}{\|v_{\text{DB}}\|} \right\|$$

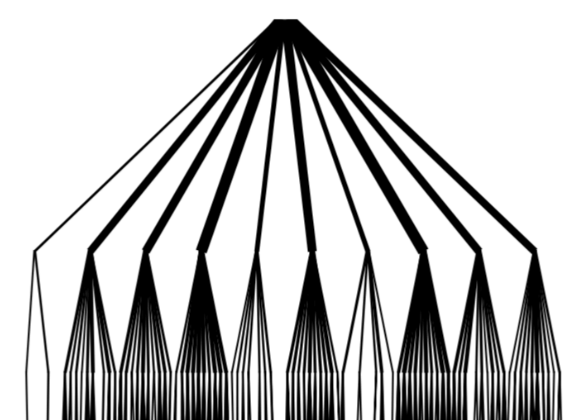
- Many dist/similarity functions, norms (ℓ_1 and ℓ_2) and normalization schemes

Need Large Vocabularies: Vocabulary Tree

- Faster quantization (logarithmic time complexity in vocabulary size)
- Therefore can afford larger vocabularies
- Hierarchical clustering:



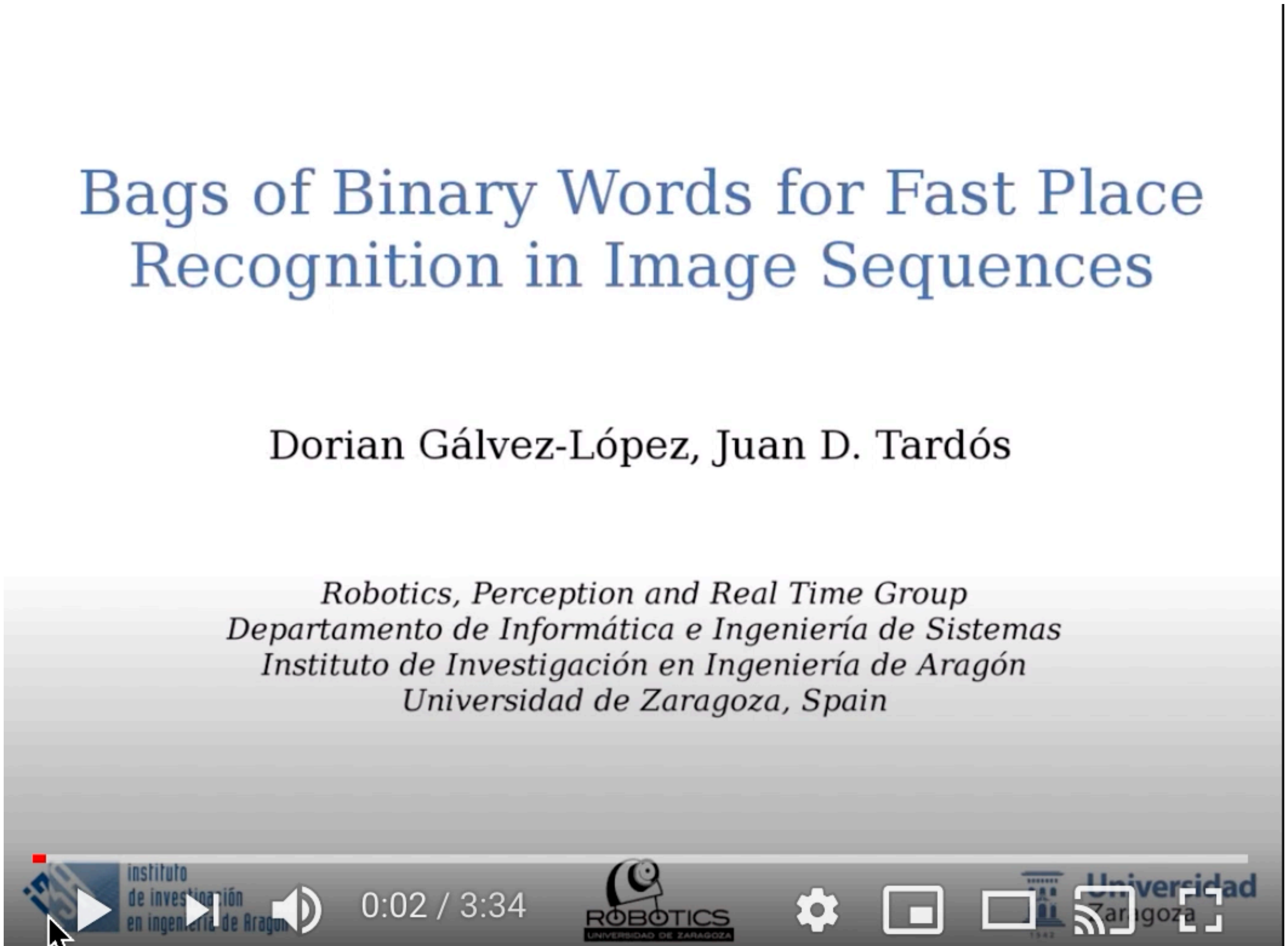
Credit: *Nister and Stewenius*



Scalable recognition with a vocabulary tree 4135 2006
D Nister, H Stewenius
2006 IEEE Computer Society Conference on Computer Vision and Pattern …

BoW-based Loop-Closure Detection in Action

Bags of Binary Words for Fast Place Recognition in Image Sequences 667 2012
D Gálvez-López, JD Tardos
IEEE Transactions on Robotics 28 (5), 1188-1197



Bags of Binary Words for Fast Place Recognition in Image Sequences

Dorian Gálvez-López, Juan D. Tardós

Robotics, Perception and Real Time Group
Departamento de Informática e Ingeniería de Sistemas
Instituto de Investigación en Ingeniería de Aragón
Universidad de Zaragoza, Spain

Today

- **Place recognition**
- **Object detection / recognition**

You Only Look Once:
Unified, Real-Time Object Detection

Joseph Redmon*, Santosh Divvala*†, Ross Girshick¶, Ali Farhadi*†
University of Washington*, Allen Institute for AI†, Facebook AI Research¶
http://pjreddie.com/yolo/

Abstract

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance.

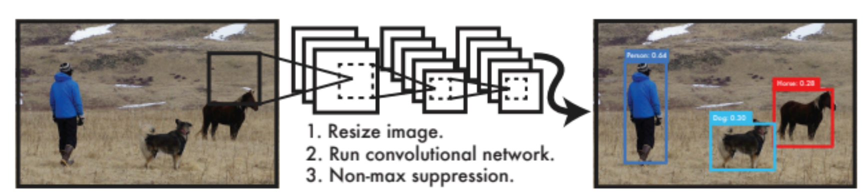
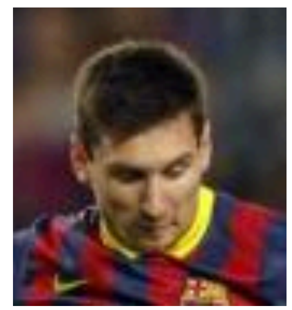


Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448×448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

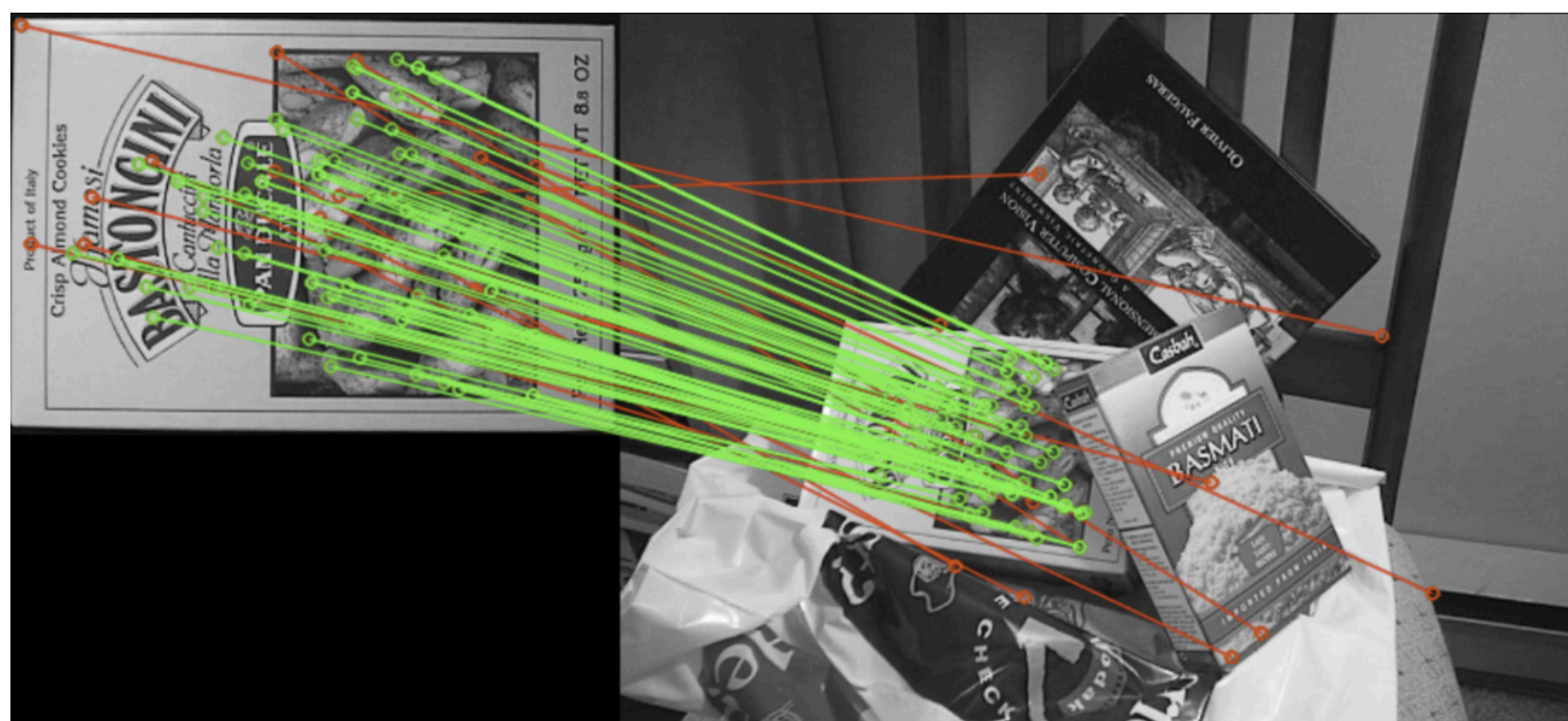
Traditional Object Detectors

- template matching (sliding window)

template



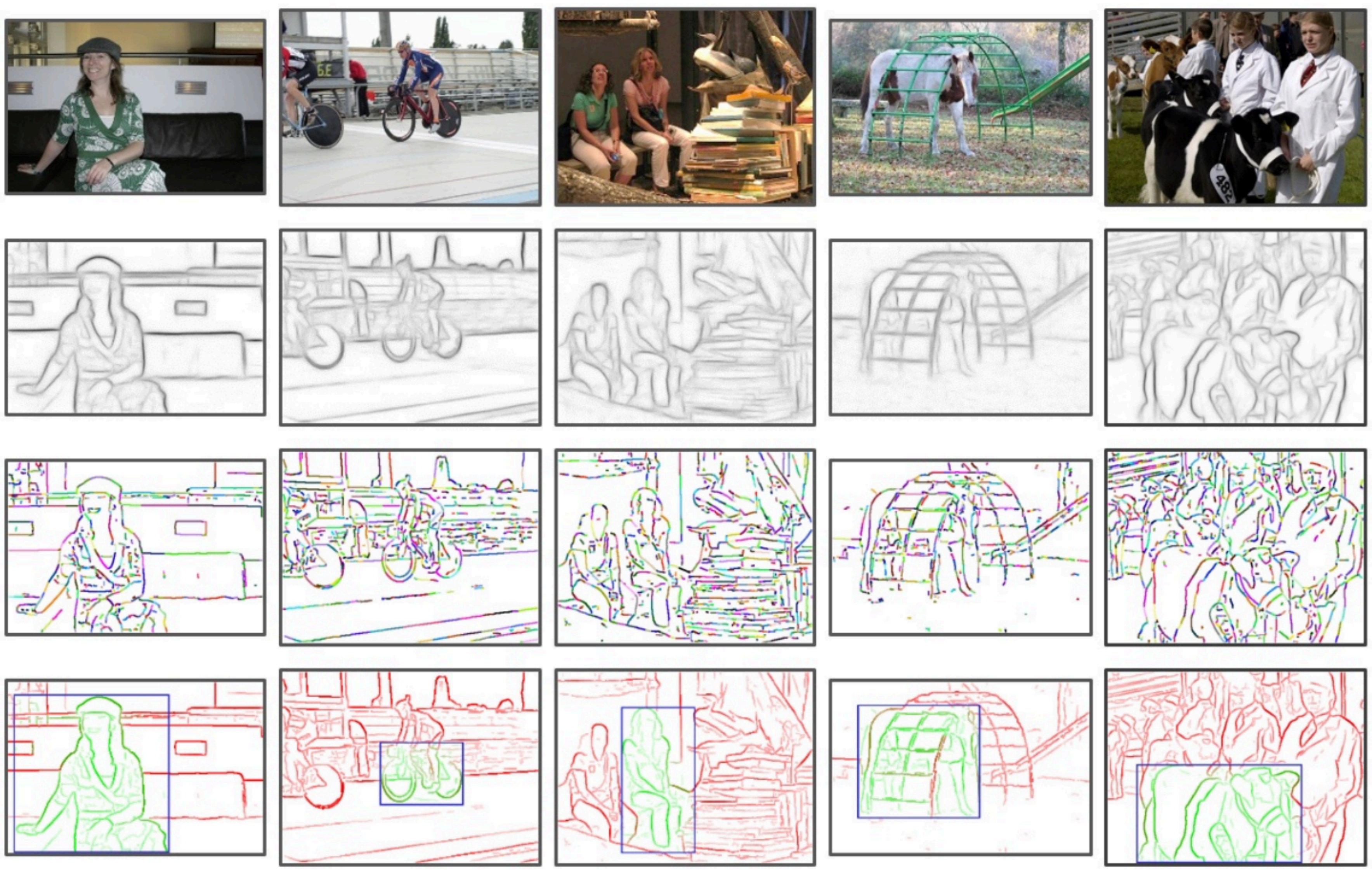
- feature-based

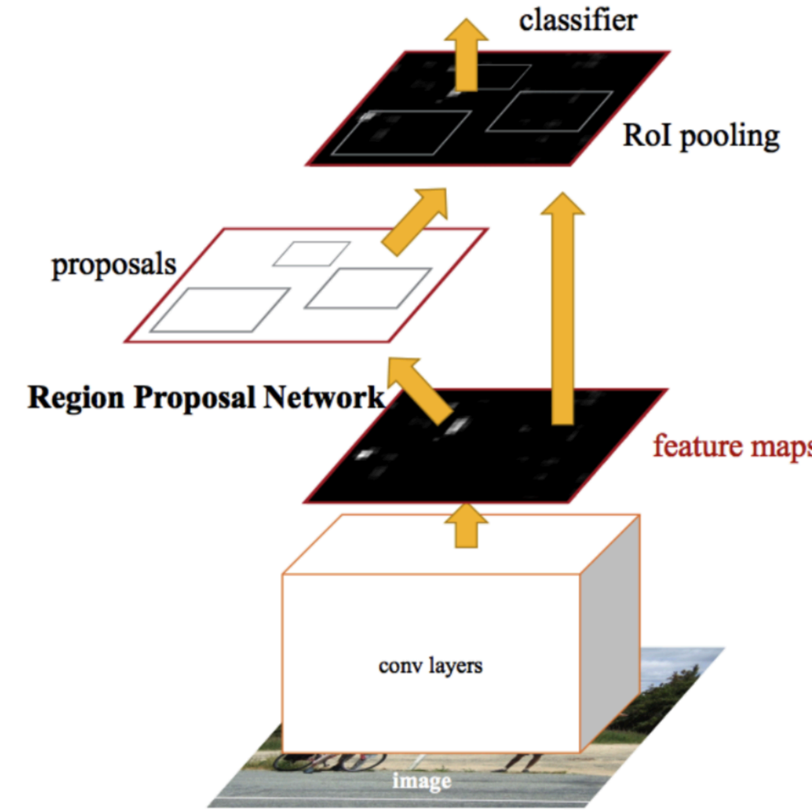


(scalability?)

Traditional Object Detectors

- Object proposal + object classification





(robustness? speed?)

L. Zitnick and P. Dollar, Edge Boxes: Locating Object Proposals from Edges, ECCV'14

Learning-based Object Detection: YOLO

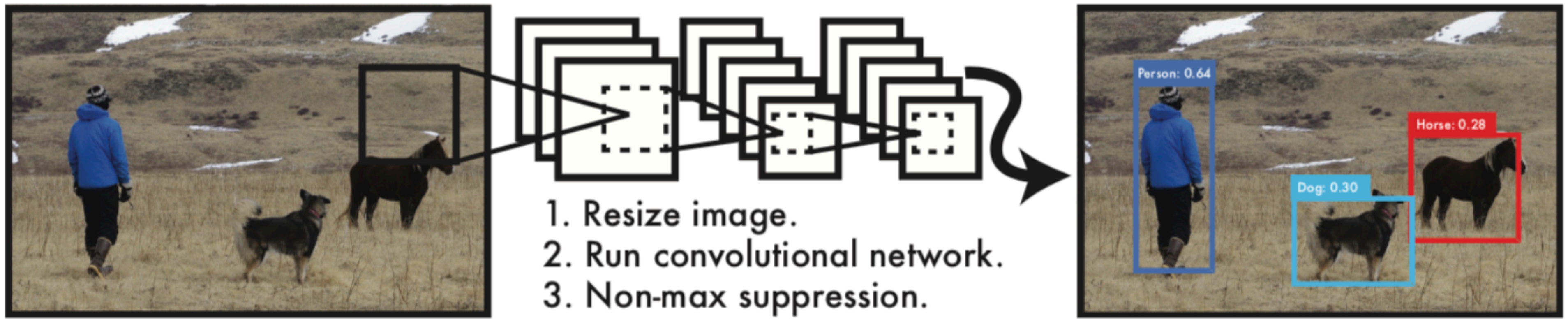


Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448×448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

- YOLO processes images 45 frames per second.
- A smaller version of the network, Fast YOLO, processes an 155fps

Redmond et al, “You Only Look Once: Unified, Real-Time Object Detection”, CVPR’16.

Learning-based Object Detection: YOLO

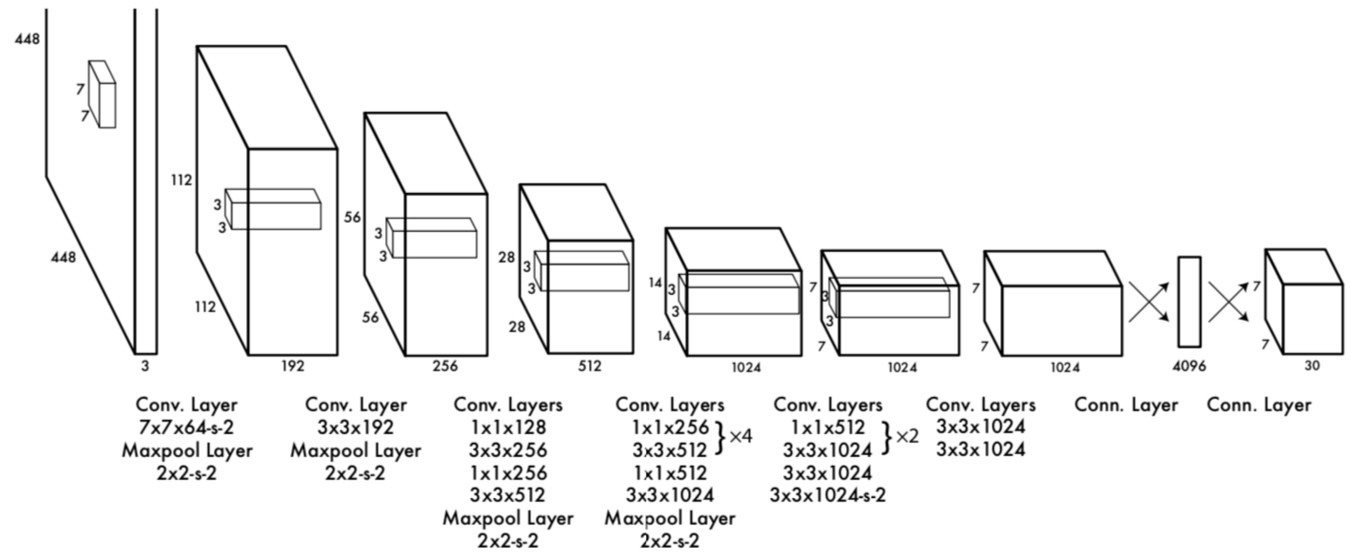


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

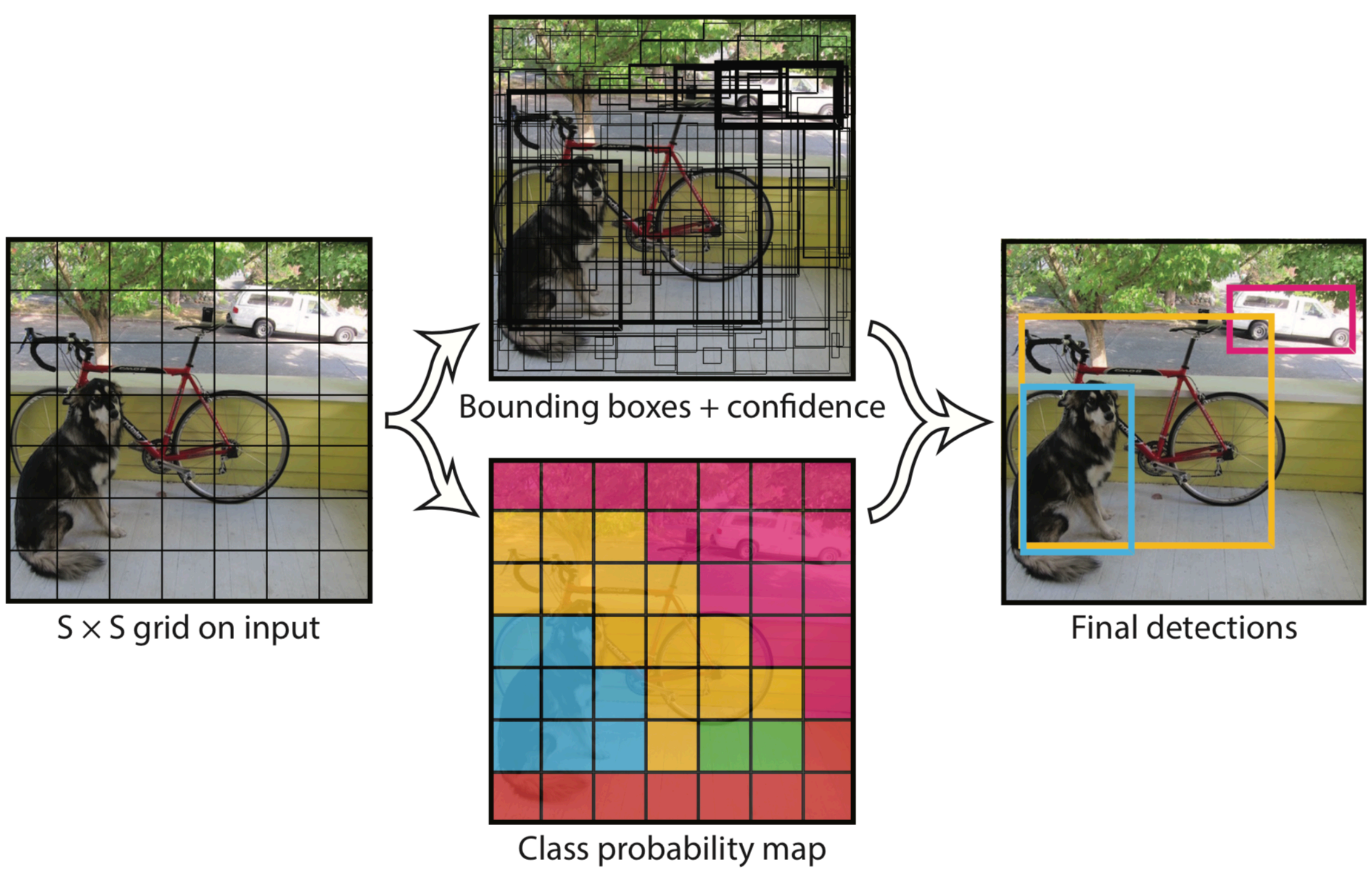


Image is split in S x S grid.

Yolo is trained to predict:

- B bounding boxes in each grid cell (x, y, h, w, confidence)
- A class label for each cell

Learning-based Object Detection: YOLO

mAP: mean Average Precision

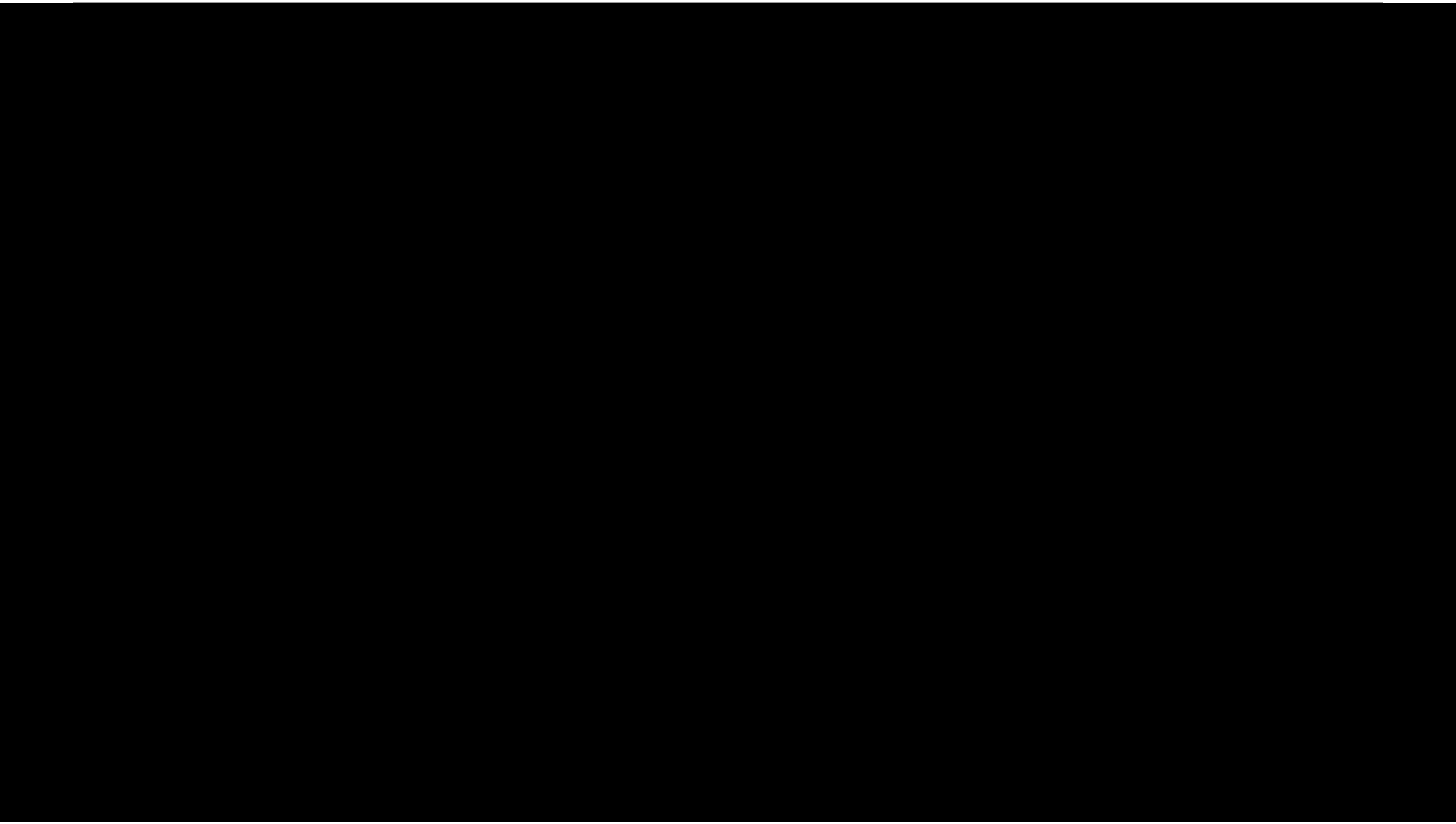
Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [38]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

Table 1: Real-Time Systems on PASCAL VOC 2007. Comparing the performance and speed of fast detectors. Fast YOLO is the fastest detector on record for PASCAL VOC detection and is still twice as accurate as any other real-time detector. YOLO is 10 mAP more accurate than the fast version while still well above real-time in speed.

Limitations of YOLO:

- **small objects**: "each grid cell only predicts B boxes and can only have one class. This spatial constraint limits the number of nearby objects that our model can predict. Our model struggles with small objects that appear in groups, such as flocks of birds."
- **generalization**: fails to detect objects in new or unusual aspect ratios or configurations.

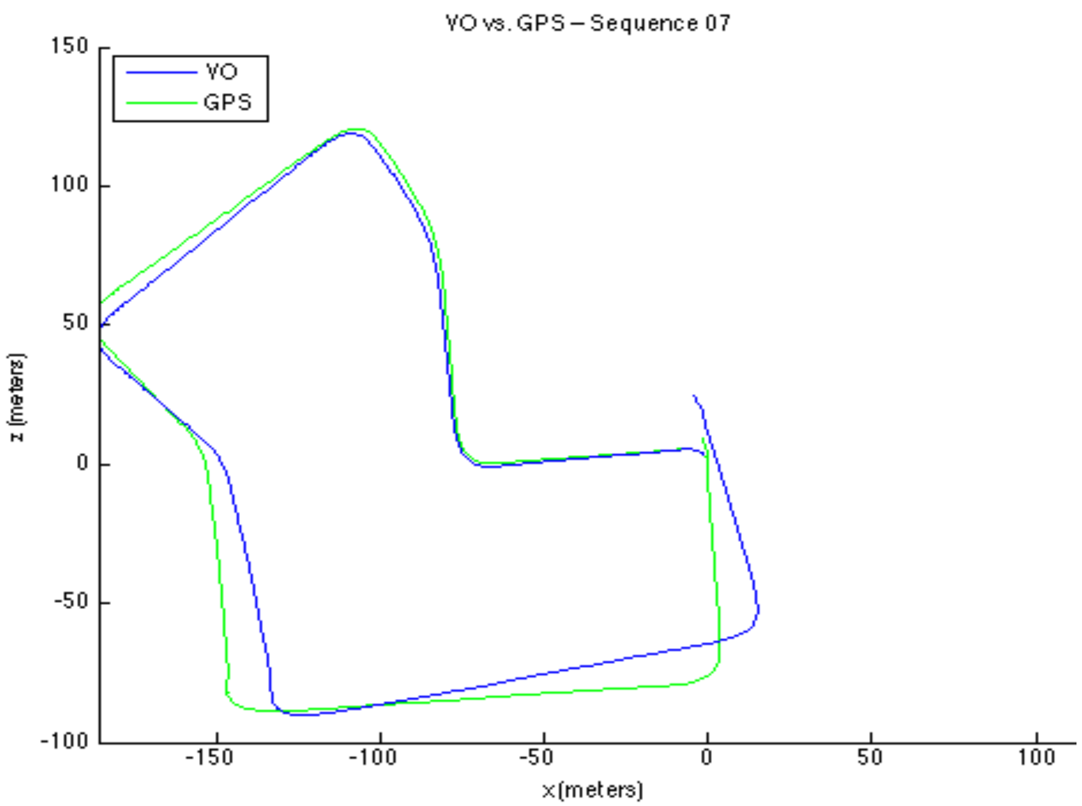
YOLO



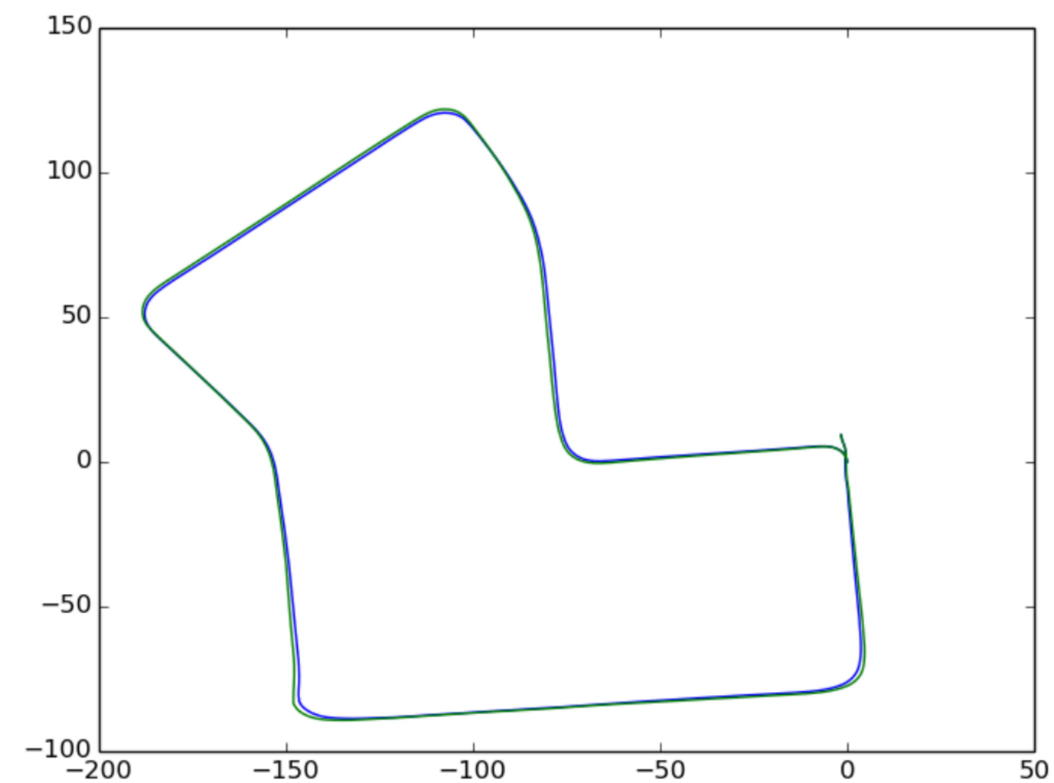
Redmond et al, “You Only Look Once: Unified, Real-Time Object Detection”, CVPR’16.
https://www.youtube.com/watch?v=uG2UOasIx2I

Next Wednesday

Visual odometry



SLAM



SLAM requires:

- place recognition => loop closure detection

and / or

- Object detection => landmark detection