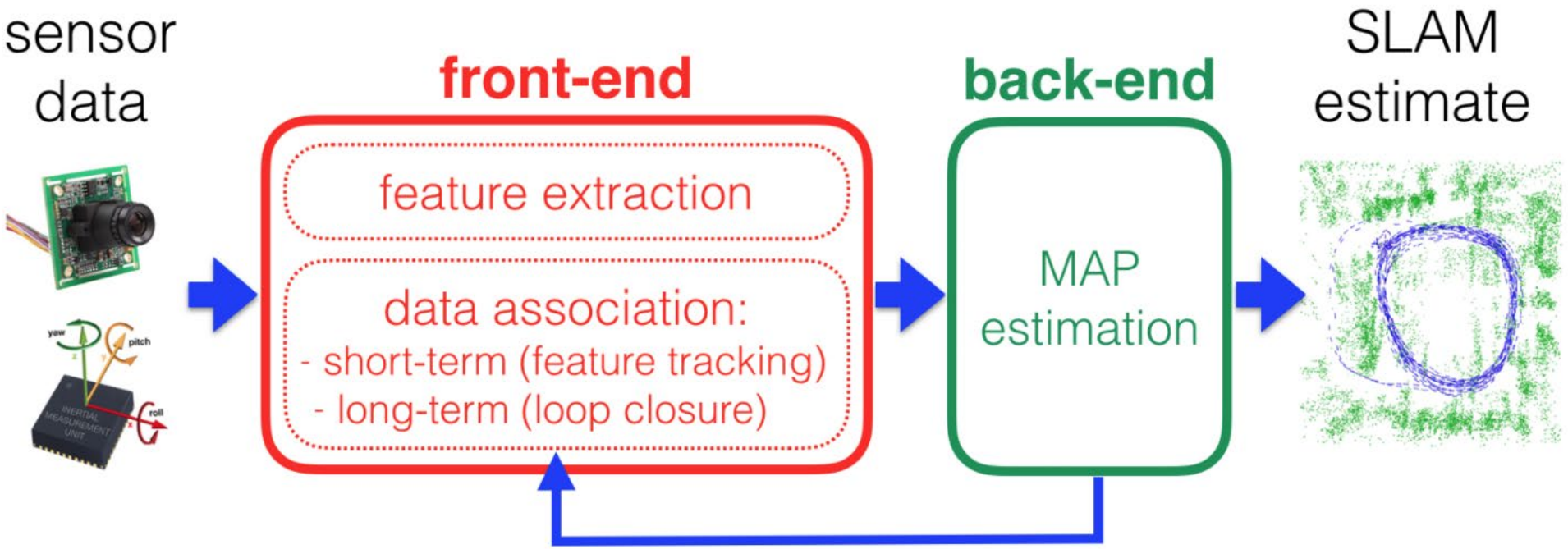
Robotic Mapping & Localization

Kaveh Fathian
Assistant Professor
Computer Science Department
Colorado School of Mines

Lec12: Loop closure

*Courtesy of Luca Carlone, Cyrill Stachniss, Olga Vysotska, Fei-Fei Li

SLAM Components



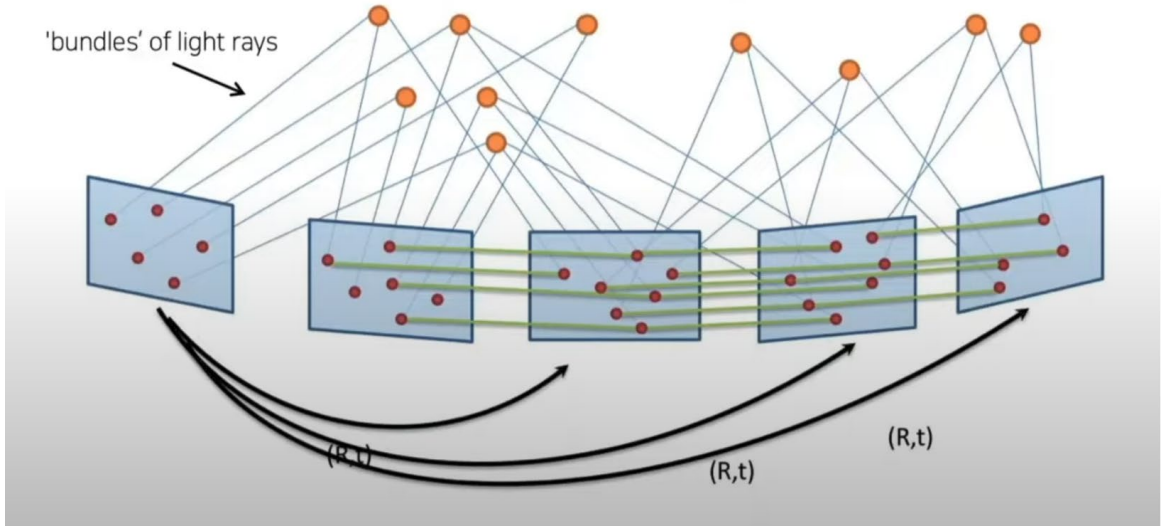
[Carlone, Visual Nav, MIT]

Standard SLAM pipeline consists of

- **Sensors**
- **Front-end** algorithms (real-time & online processing of sensor data, trajectory, map)
- **Back-end** algorithms (optimization & refinement of map, trajectory.)

Loop Closure

Front-end or short-term (feature tracking)

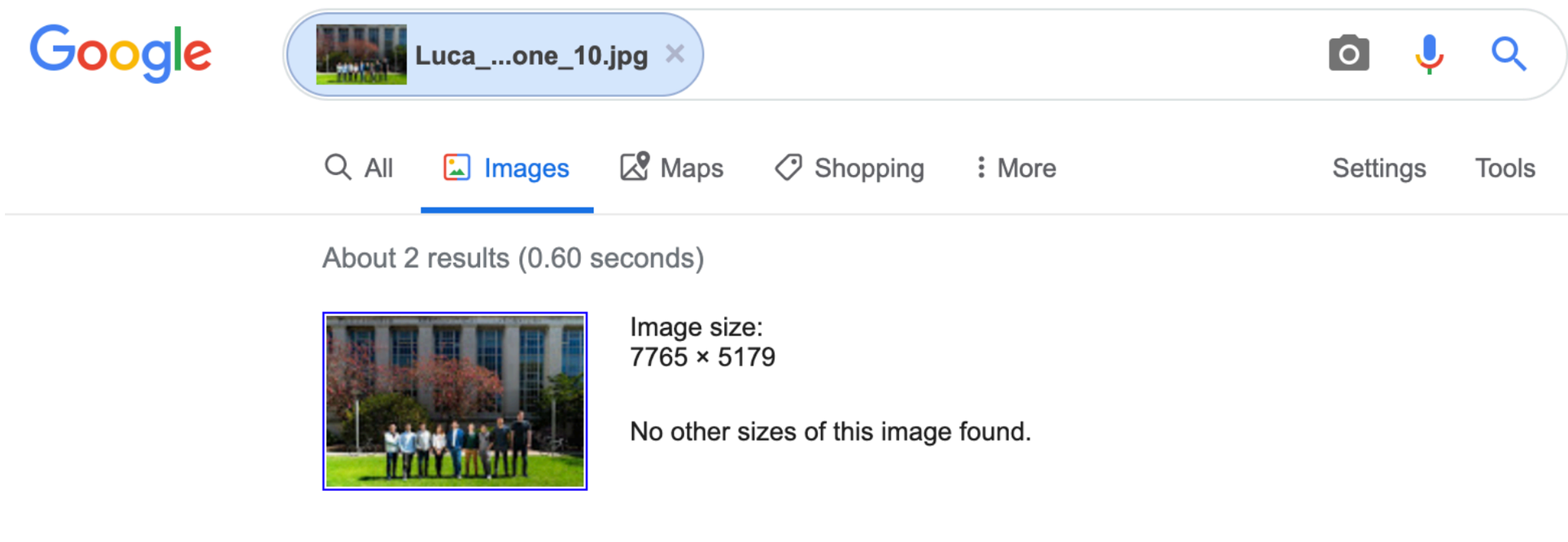


https://youtu.be/z4ldKGh12ok

Back-end or long-term (loop closure detection)

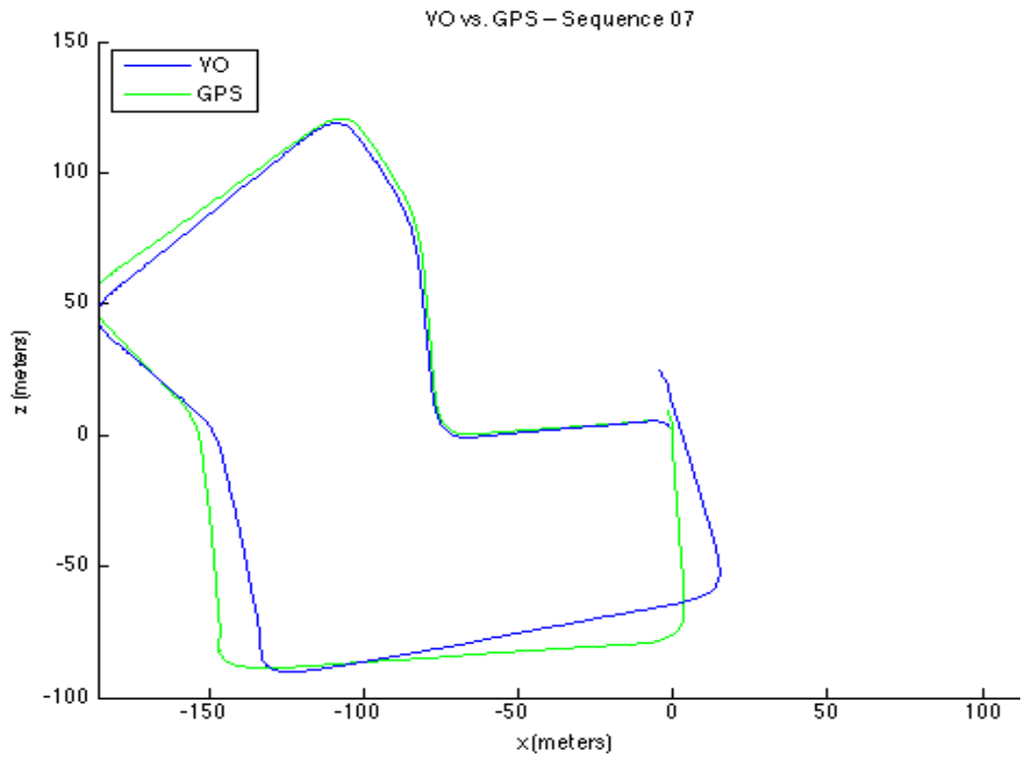
Place Recognition

Image Retrieval

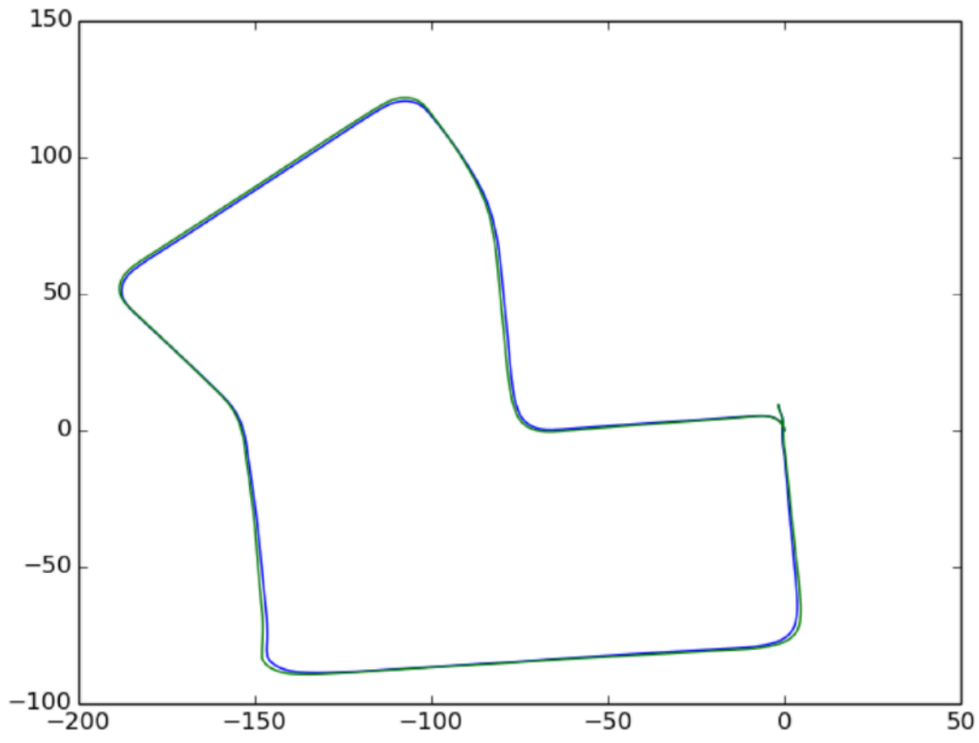


Recap

Visual odometry



SLAM



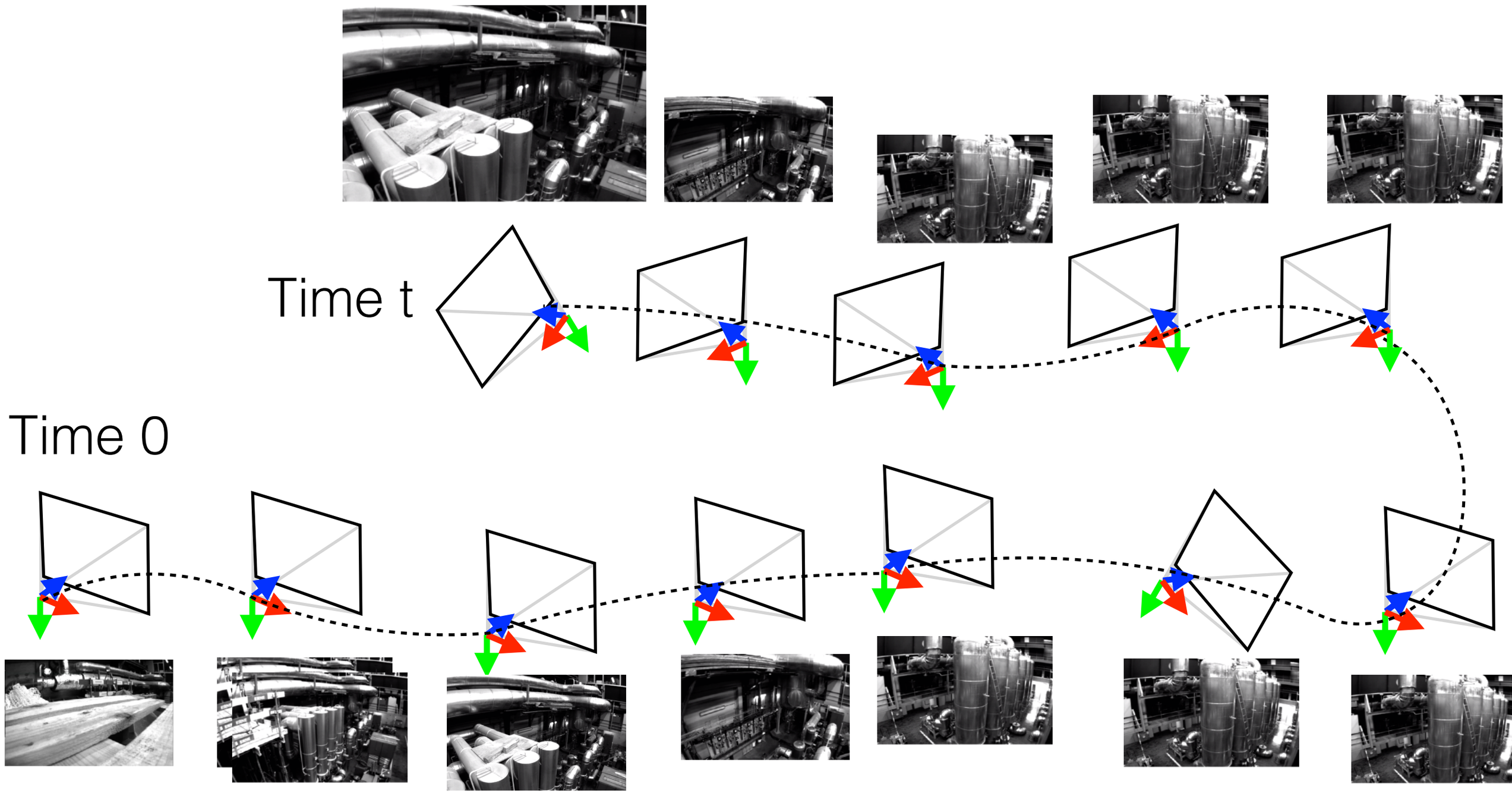
SLAM (Simultaneous Localization and Mapping) requires:

- place recognition => loop closure detection

and / or

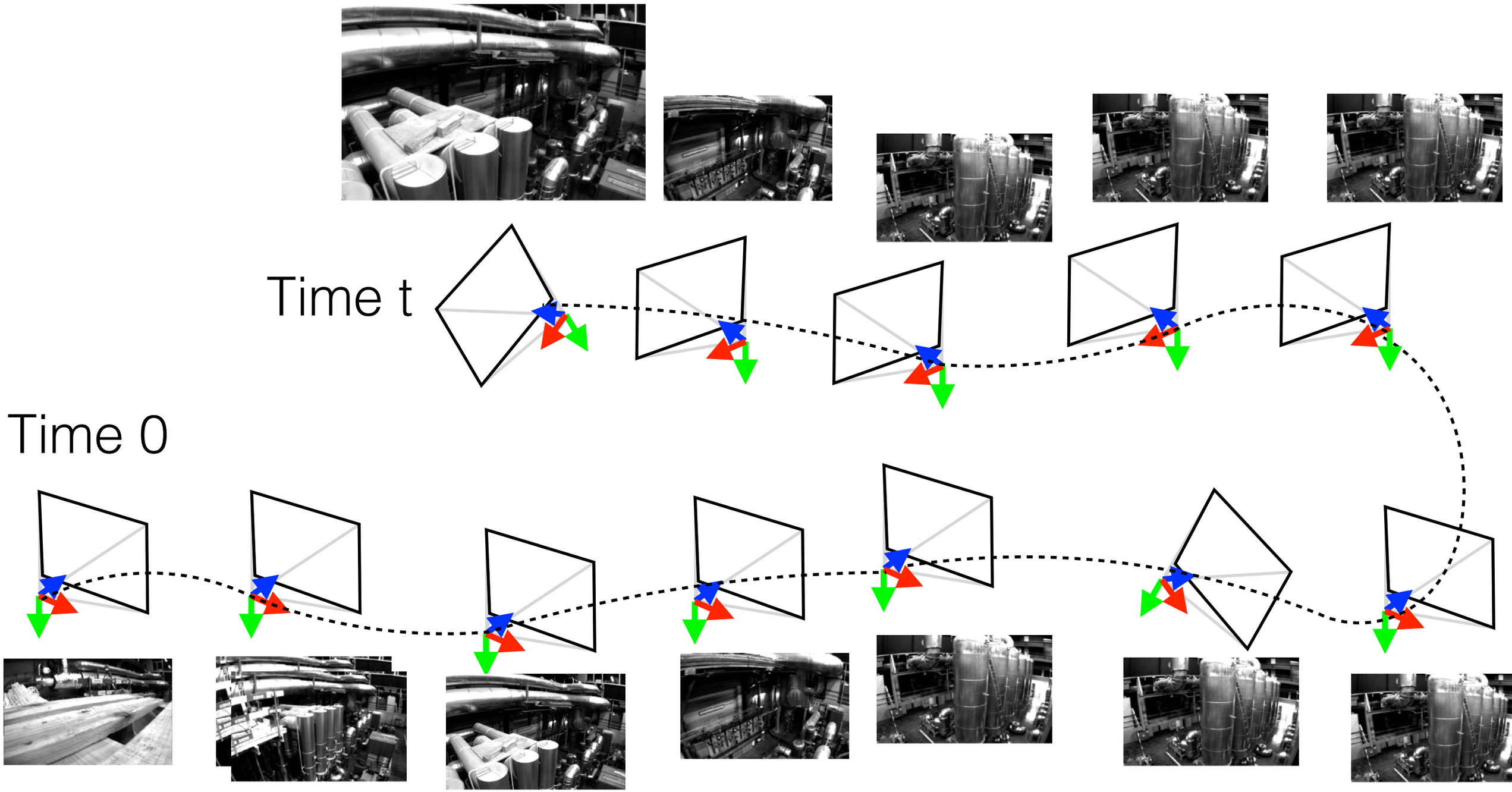
- Object detection => landmark detection

Place Recognition & Image Retrieval



Does the image at time “t” picture a place seen in previous images?

A brute force approach



Scalability is crucial..

Image retrieval/place recognition vs. pose estimation

Place Recognition: Challenges

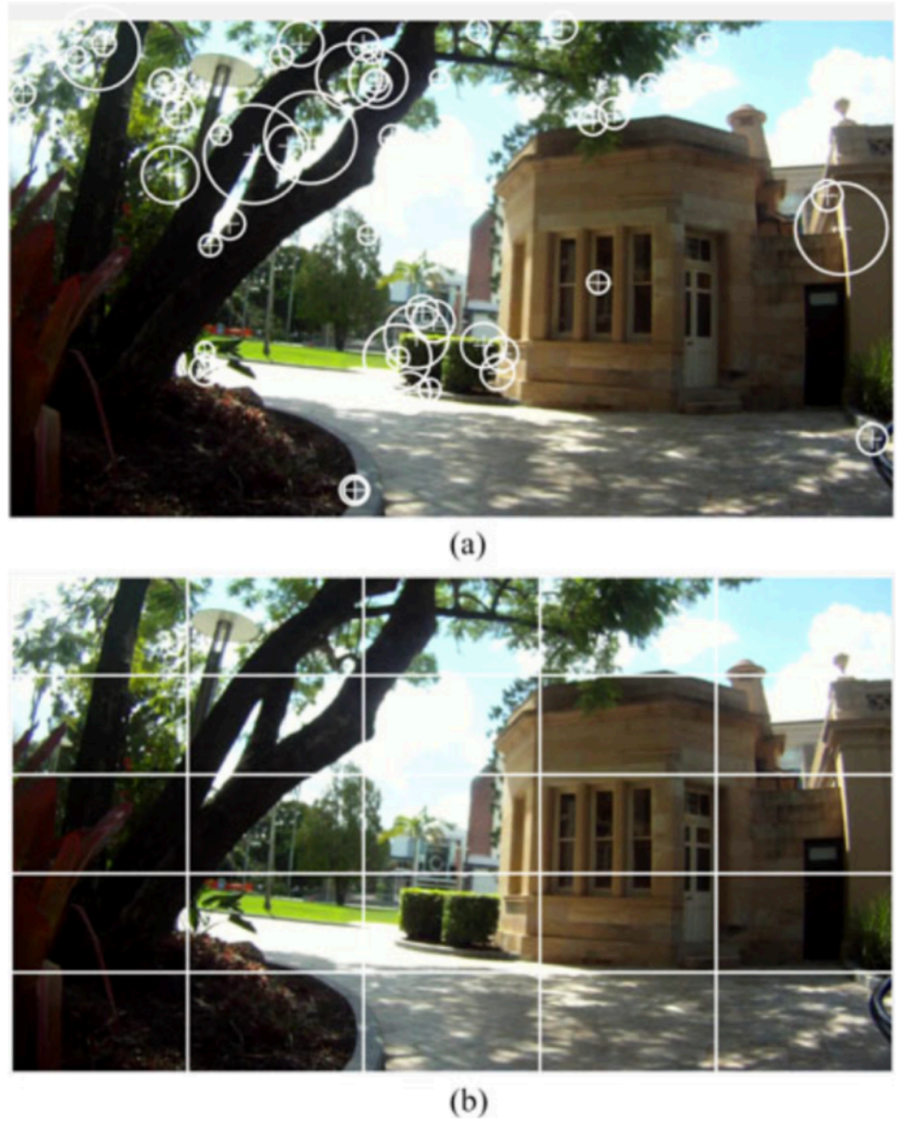
- **Appearance changes**:
  - Illumination
  - Weather conditions
  - Dynamic objects (people, cars,…)
  - Viewpoint changes
- **Perceptual aliasing**: two different places may look similar (building, roads, …)

Image Retrieval: Approaches

- Local descriptors

- Global descriptors

- Learning-based methods



(a)

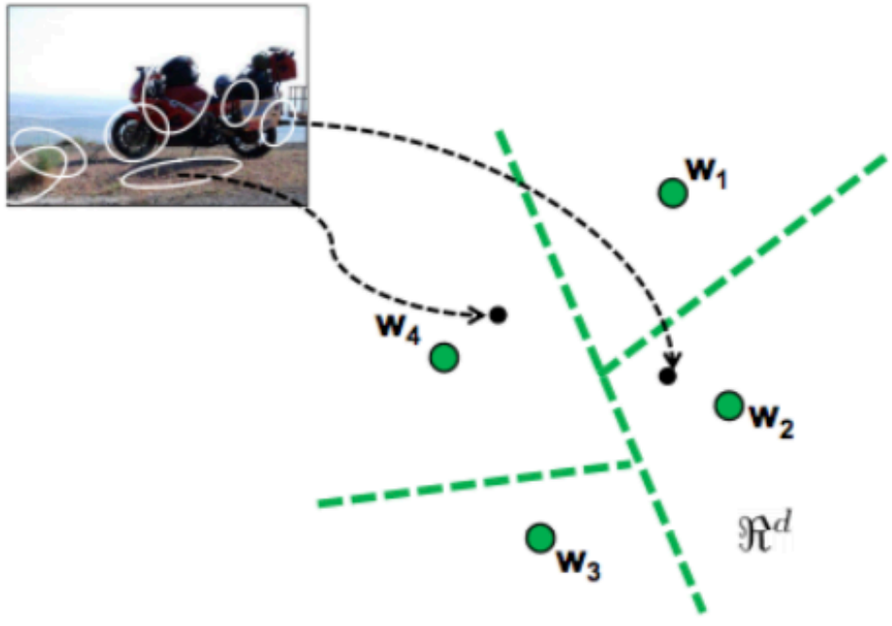
(b)

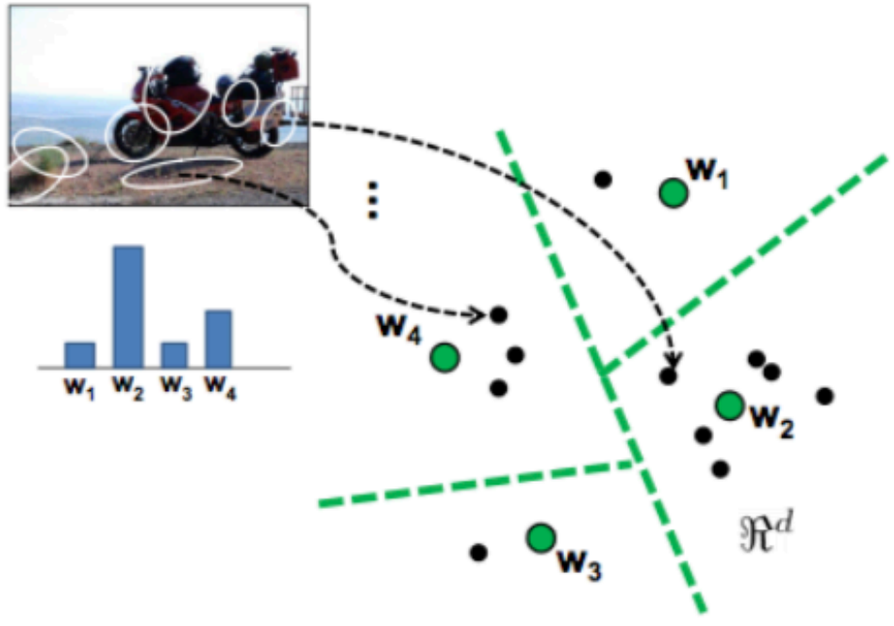
[courtesy of Lowry'16]

Local descriptors: Bag of Words

Based on text retrieval and summarization methods

1) Extract features and descriptors in image
2) Discretize feature space (clustering)
3) Store the frequency of the features for each image

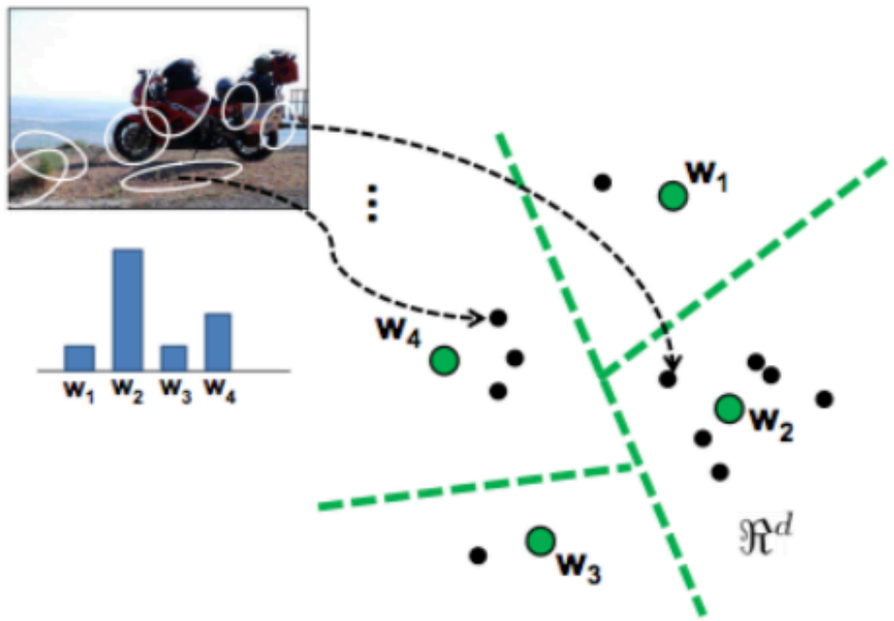


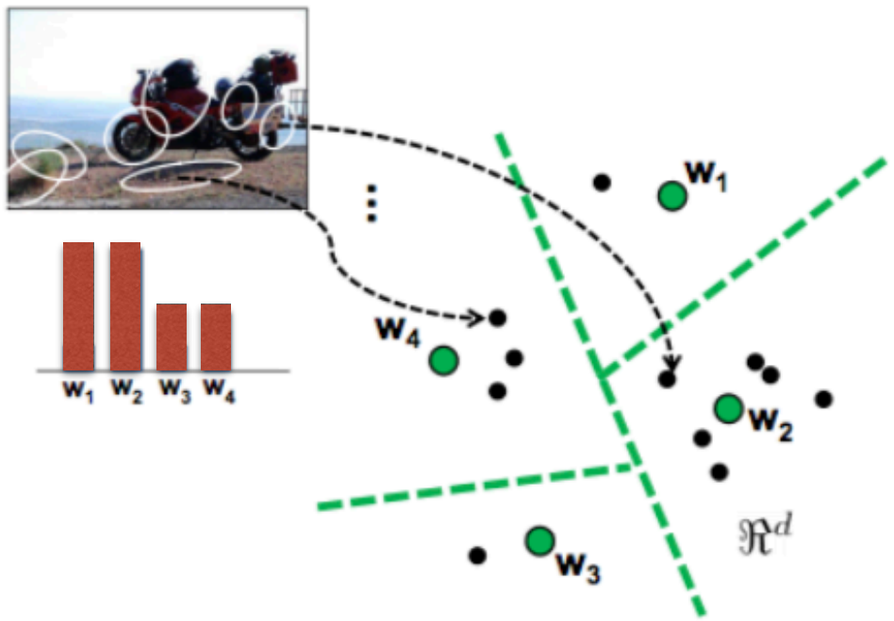


Each cluster is a “visual word”
Typically 5k-10k (up to 100k) visual words

J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

Local descriptors: Bag of Words

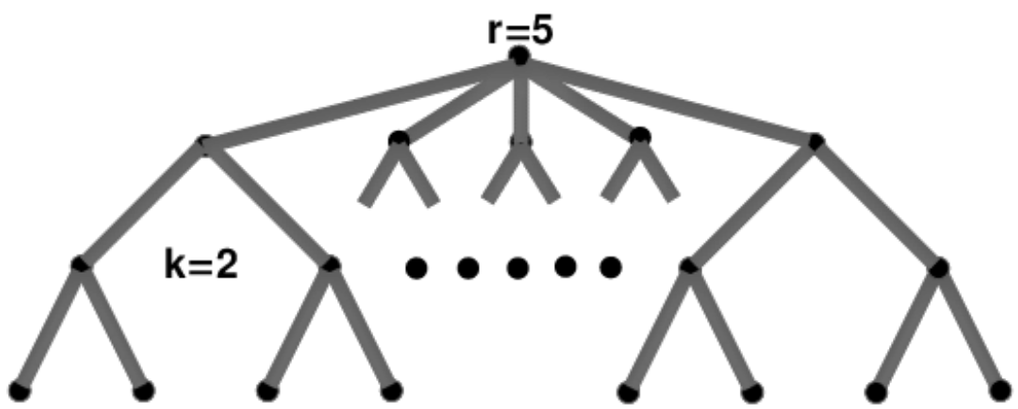




Two images are compared based on the corresponding histogram (Hamming distances, other metrics, …)

Faster version: **vocabulary tree**

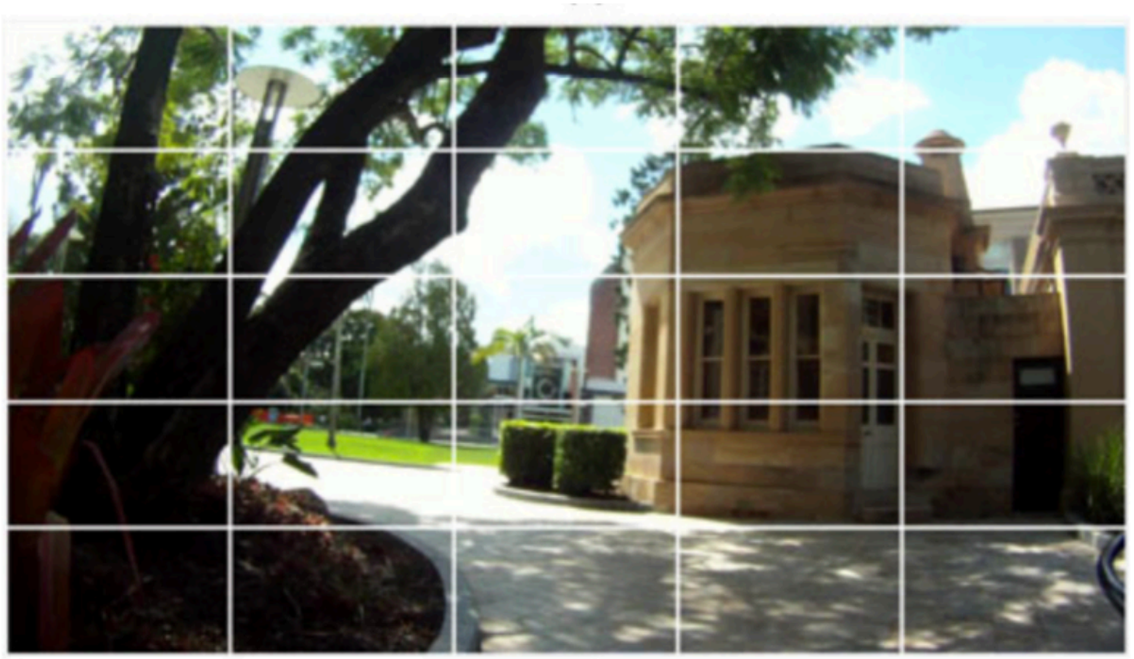
Alternatives**: VLAD** (Vector of Locally Aggregated Descriptors), **Fisher vectors**



Global descriptors

Early approaches:

- color histograms
- principal component analysis
- other statistics on edges, corners, and color patches



Early 2000:

- **GIST descriptor**:
  - image is filtered at different orientations and different frequencies to extract information from the image
  - results are averaged to generate a compact vector that represents the “gist” of a scene

A. Oliva and A. Torralba, Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope, IJCV, 2001.

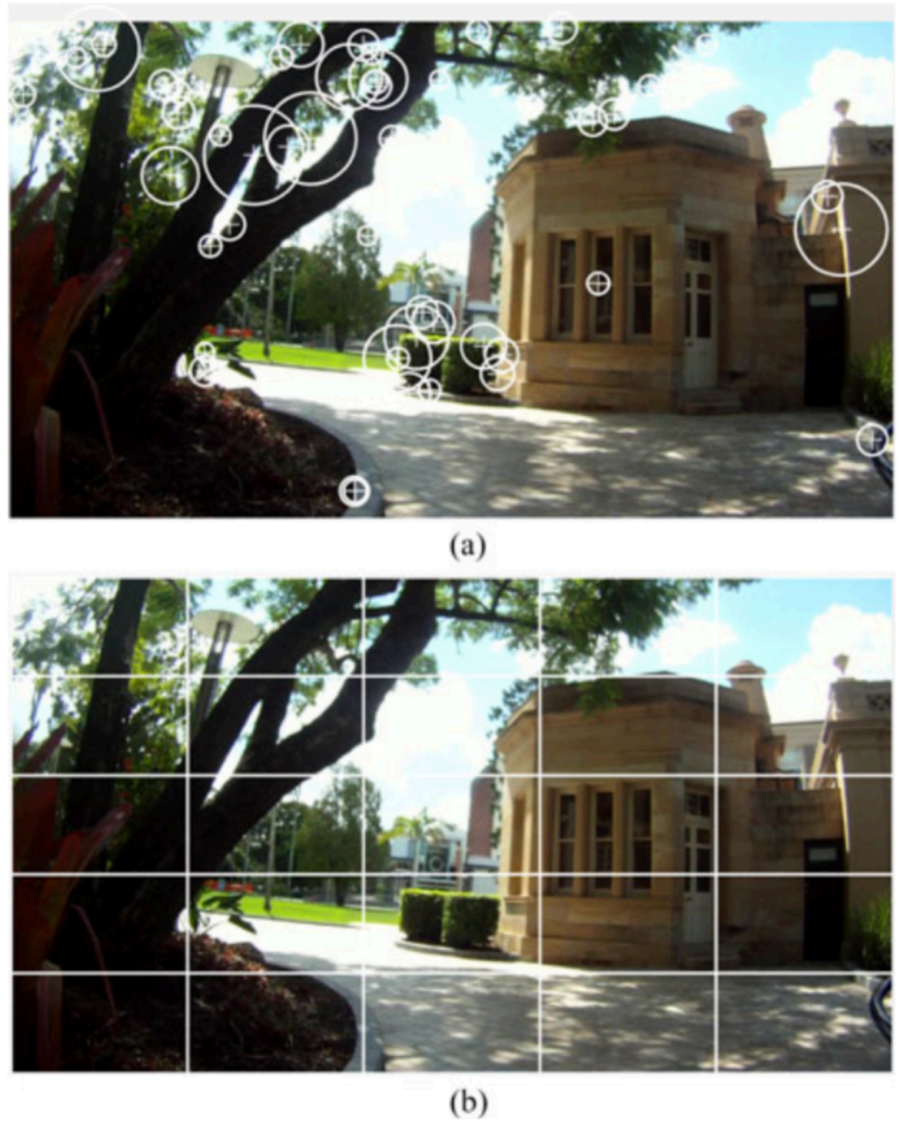
Local vs. Global Descriptors

Local descriptors:

- allow estimating feature (and camera) geometry
- sensitive to lighting conditions and seasonal variations

Global descriptors:

- better at handling lighting conditions and seasonal variation
- more sensitive to viewpoint changes



[courtesy of Lowry'16]

A. Quattoni and A. Torralba, Recognizing Indoor Scenes, CVPR'09.

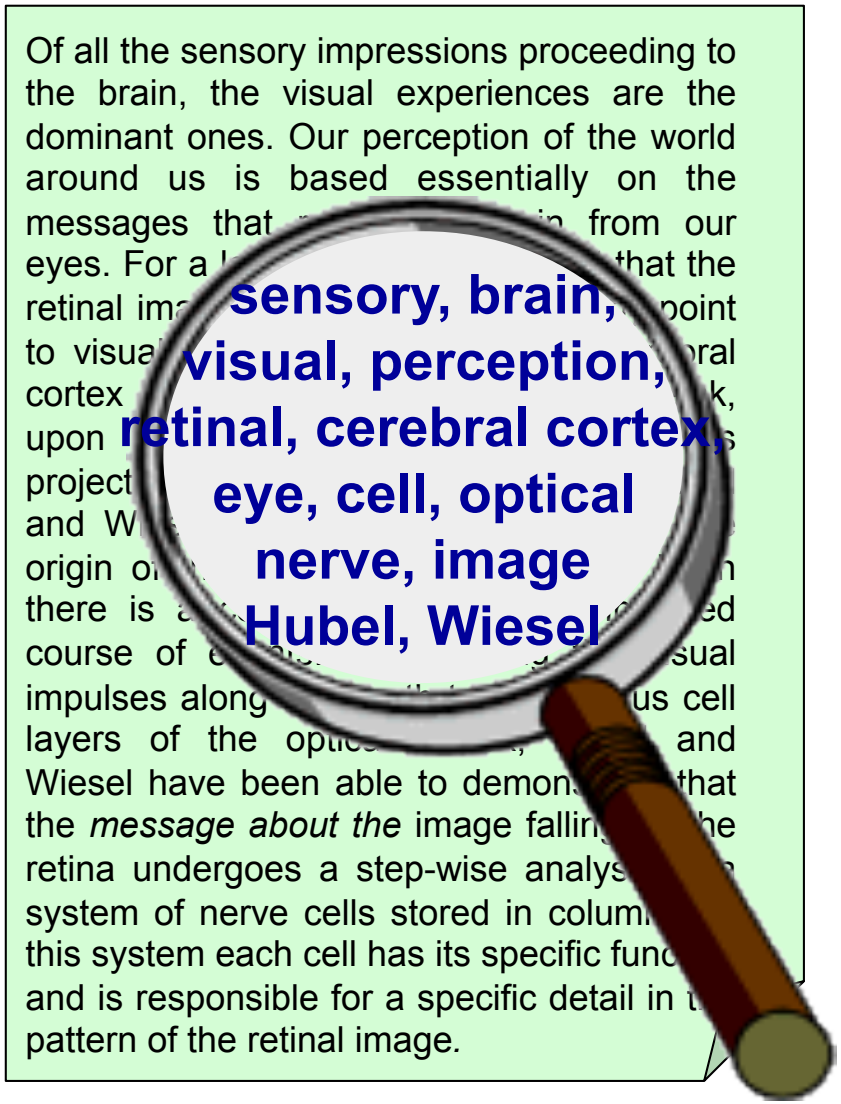
Bag of visual words

What is Bag of Visual Word for?

- Finding images in a database, which are similar to a given query image
- Computing image similarities
- Compact representation of images

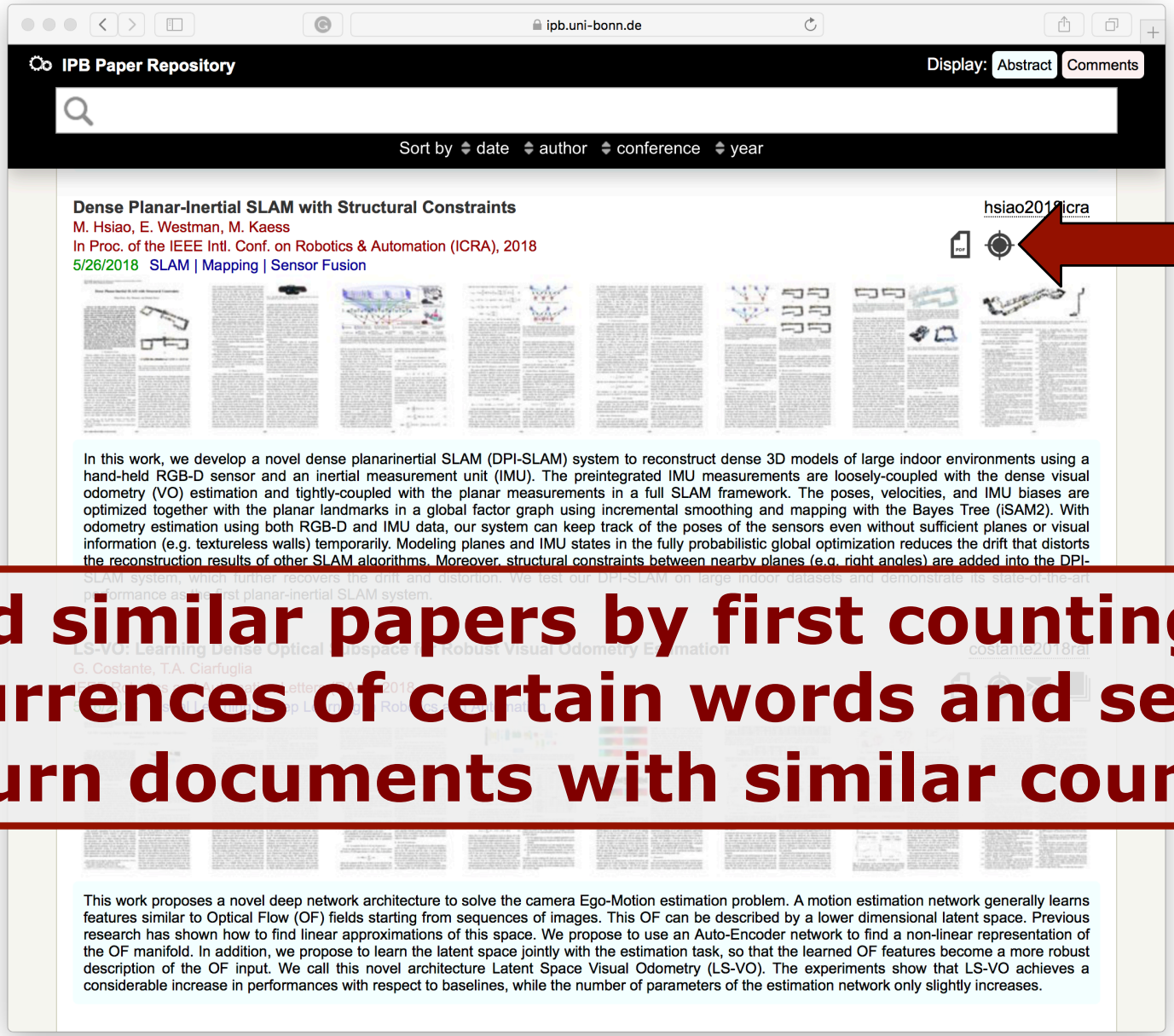
?

Analogy to Text Documents



[Image courtesy: Fei-Fei Li]

Looking for Similar Papers



"find similar papers by first counting the occurrences of certain words and second return documents with similar counts."

Bag of (Visual) Words

Analogy to documents: The content of a can be inferred from the frequency of relevant words that occur in a document

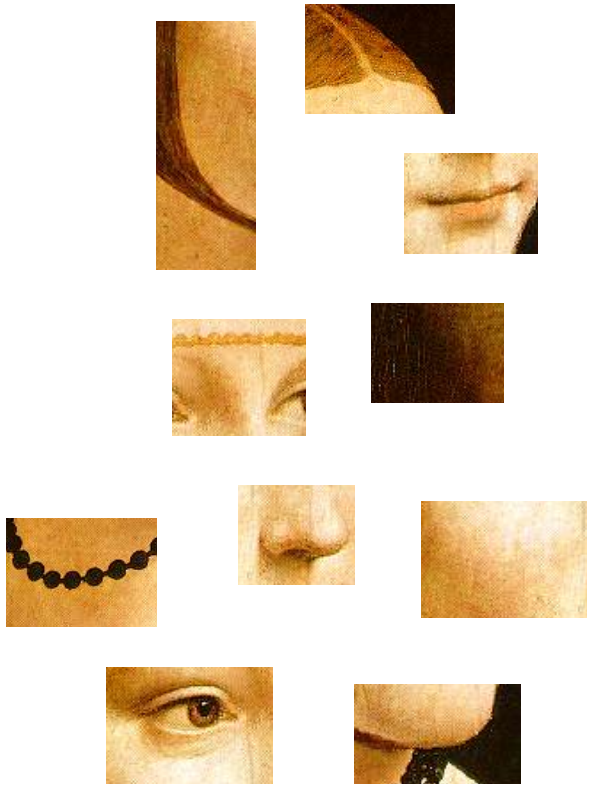
object bag of “visual words”

[Image courtesy: Fei-Fei Li]

Bag of Visual Words

- Visual words = independent features

face



features

[Image courtesy: Fei-Fei Li]

Bag of Visual Words

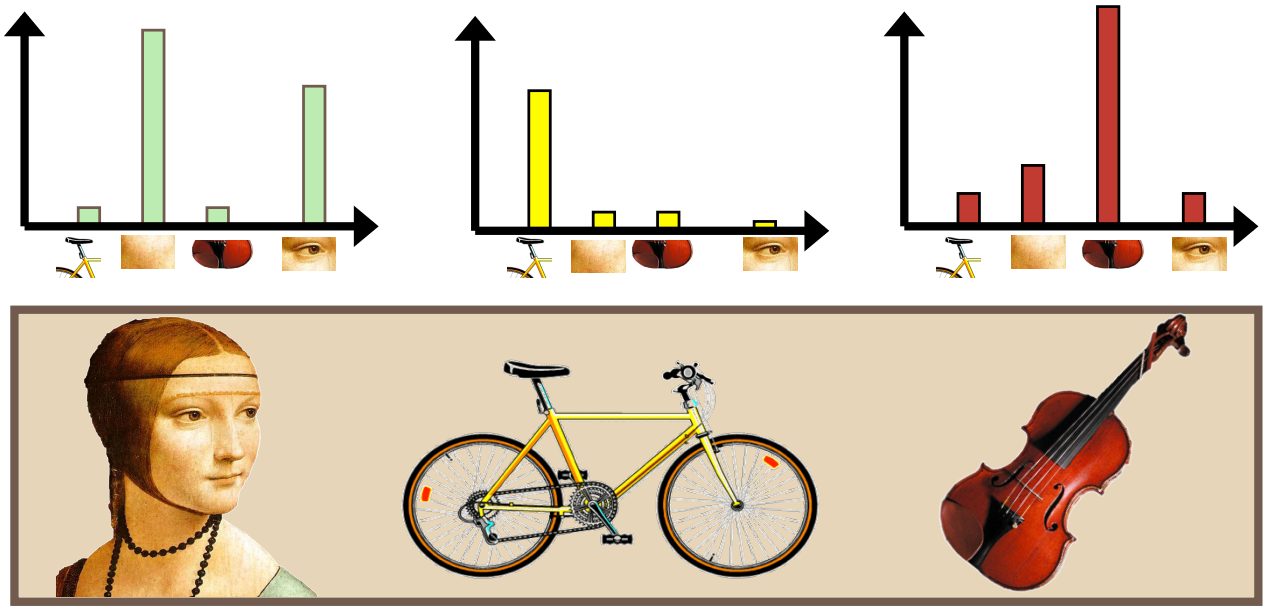
- Visual words = independent features
- Construct a dictionary of representative words
- Use only words from the dictionary

dictionary ("codebook")

[Image courtesy: Fei-Fei Li]

Bag of Visual Words

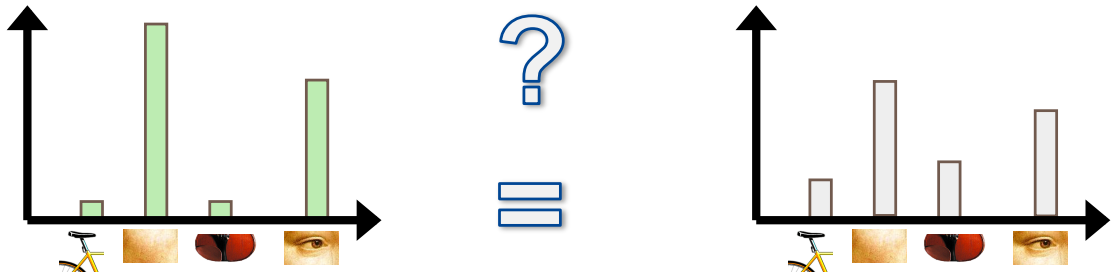
- Visual words = independent features
- Words from the dictionary
- Represent the images based on a histogram of word occurrences



[Image courtesy: Fei-Fei Li]

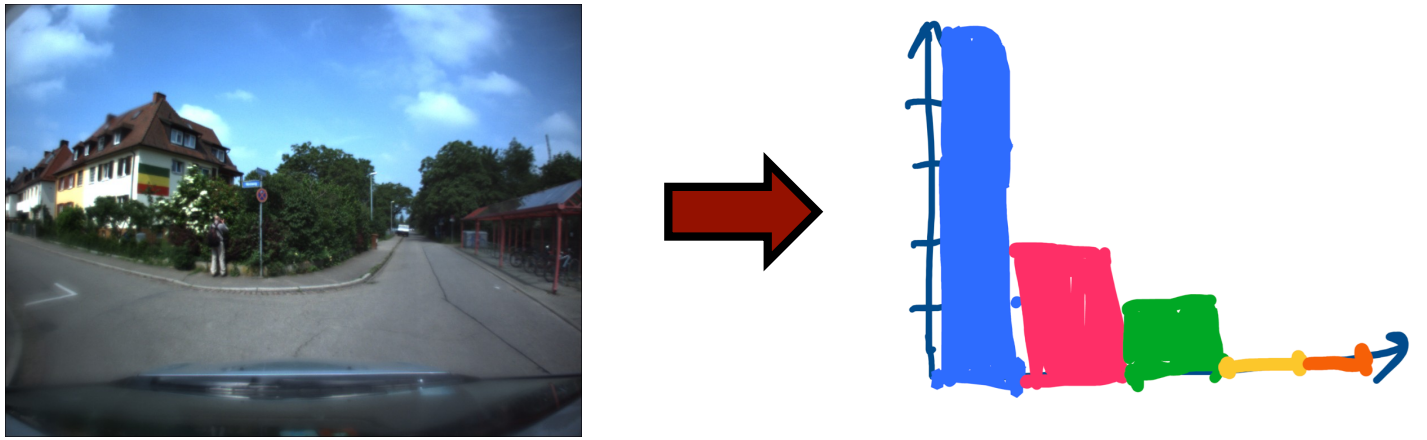
Bag of Visual Words

- Visual words = independent features
- Words from the dictionary
- Represent the images based on a histogram of word occurrences
- Image comparisons are performed based on such word histograms



[Image courtesy: Fei-Fei Li]

From Images to Histograms



[Image courtesy: Olga Vysotska]

Overview: Input Image

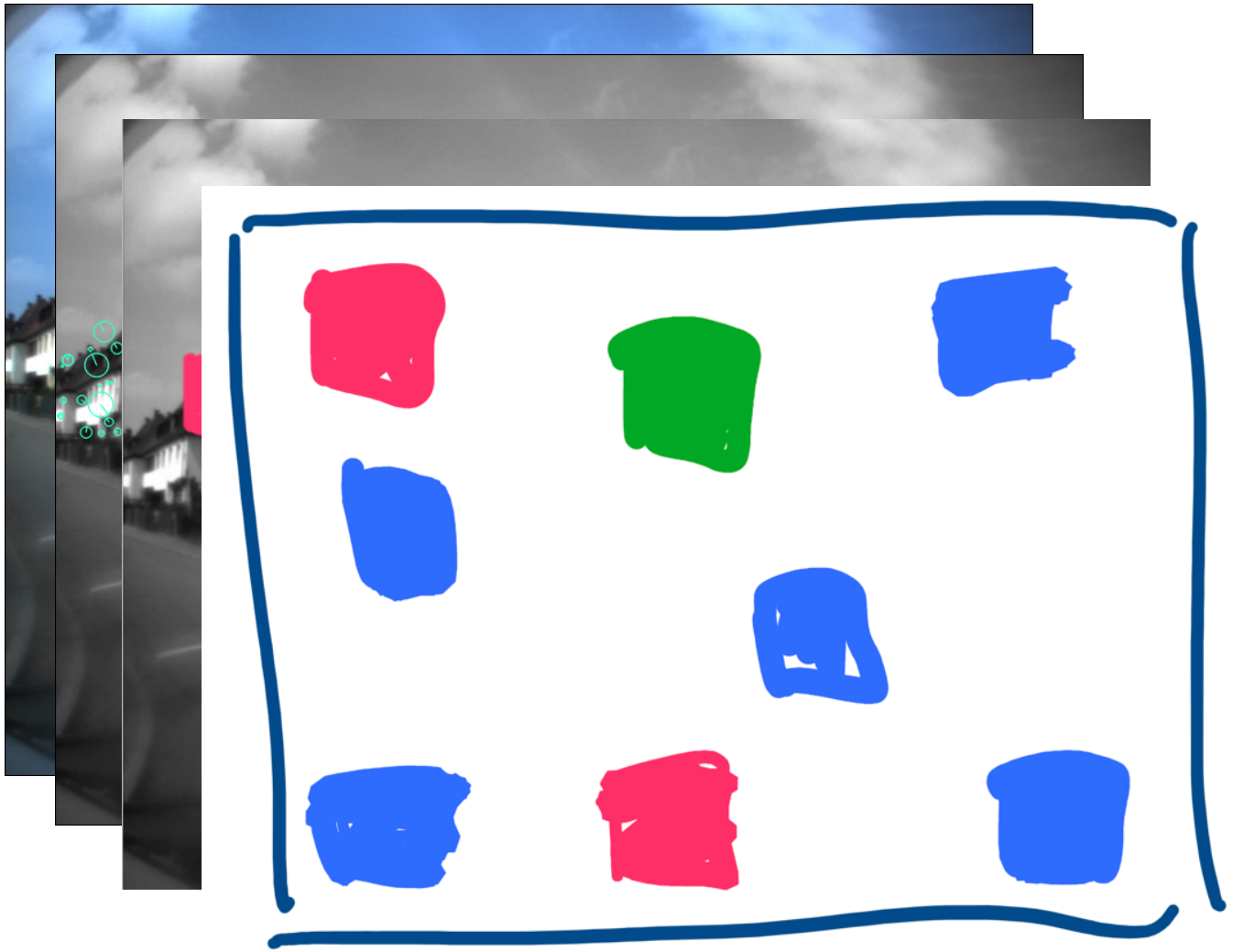
Overview: Extract Features

[Image courtesy: Olga Vysotska]

Overview: Visual Words

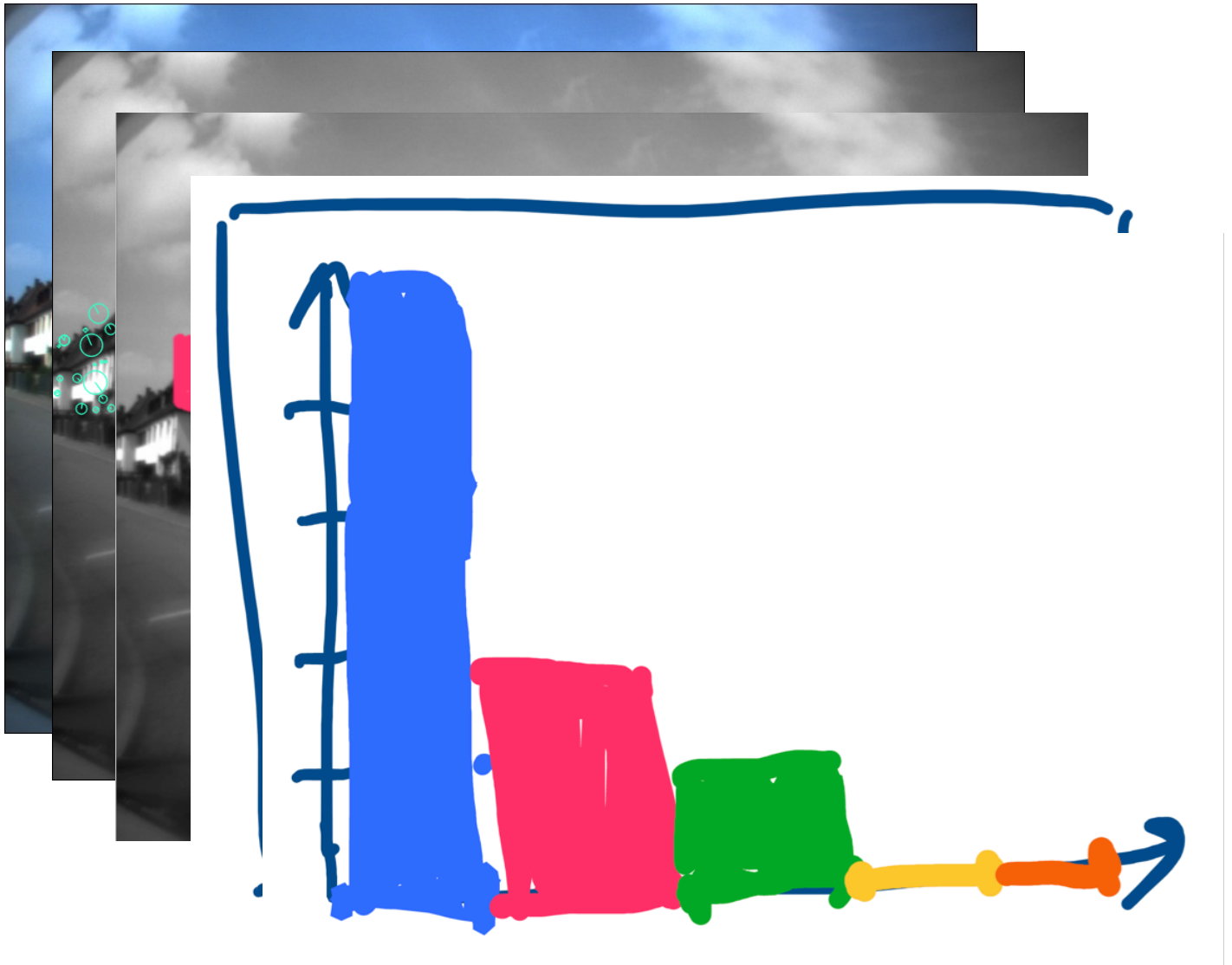
[Image courtesy: Olga Vysotska]

Overview: No Pixel Values



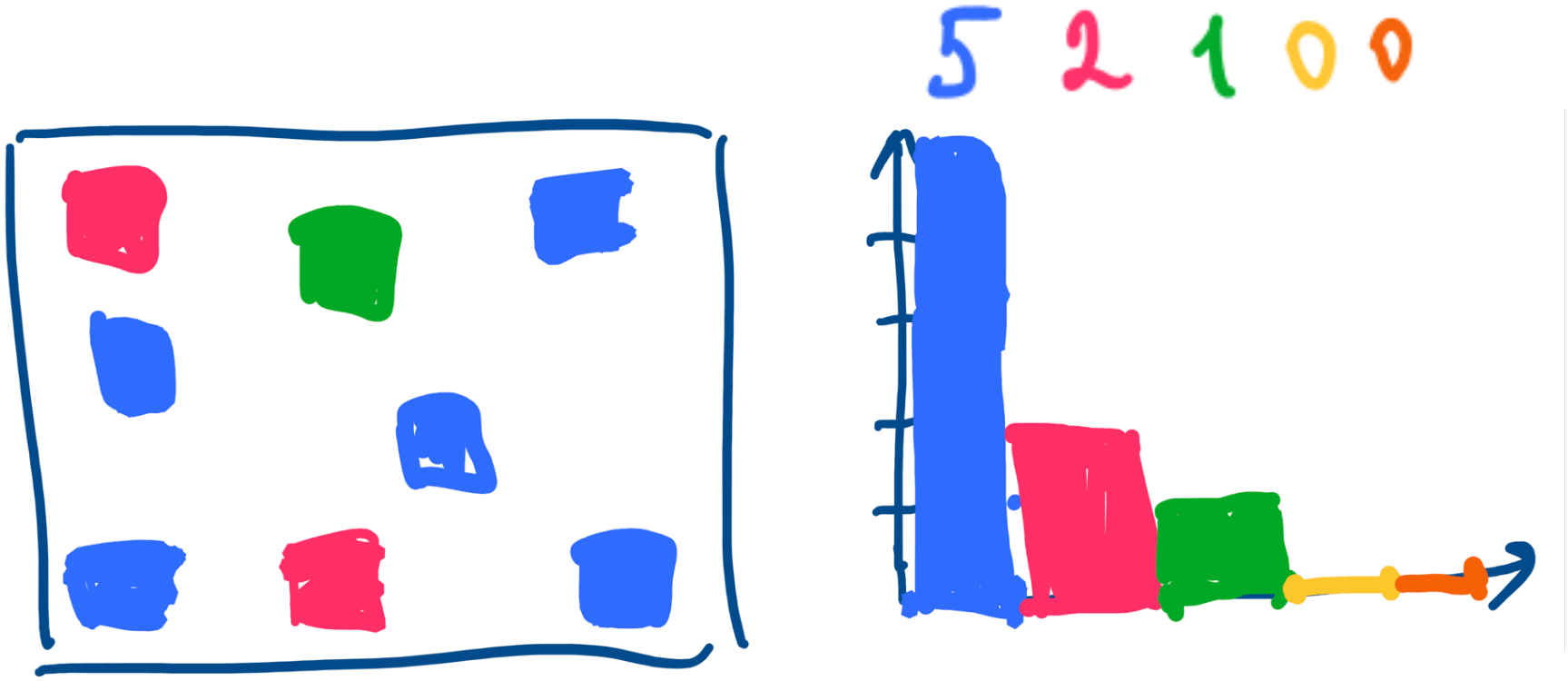
[Image courtesy: Olga Vysotska]

Overview: Word Occurrences



[Image courtesy: Olga Vysotska]

Images to Histograms

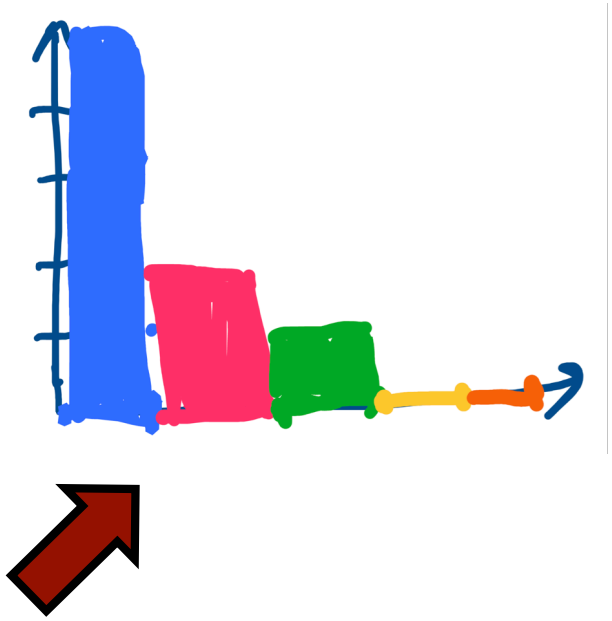


[Image courtesy: Olga Vysotska]

Where Do the Visual Words Come Form?

Dictionary

- A dictionary defines the list of words that are considered
- The dictionary defines the x-axes of all the word occurrence histograms



[Image courtesy: Olga Vysotska]

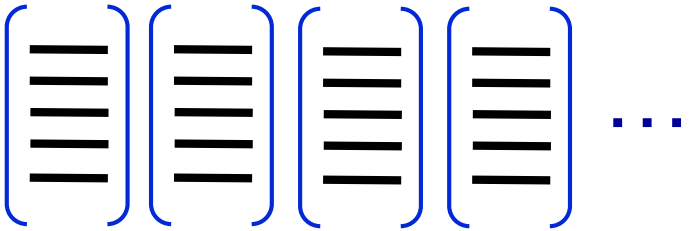
Dictionary

- A dictionary defines the list of words that are considered
- The dictionary defines the x-axes of all the word occurrence histograms
- The dictionary must remain fixed

The dictionary is typically learned from data. How can we do that?

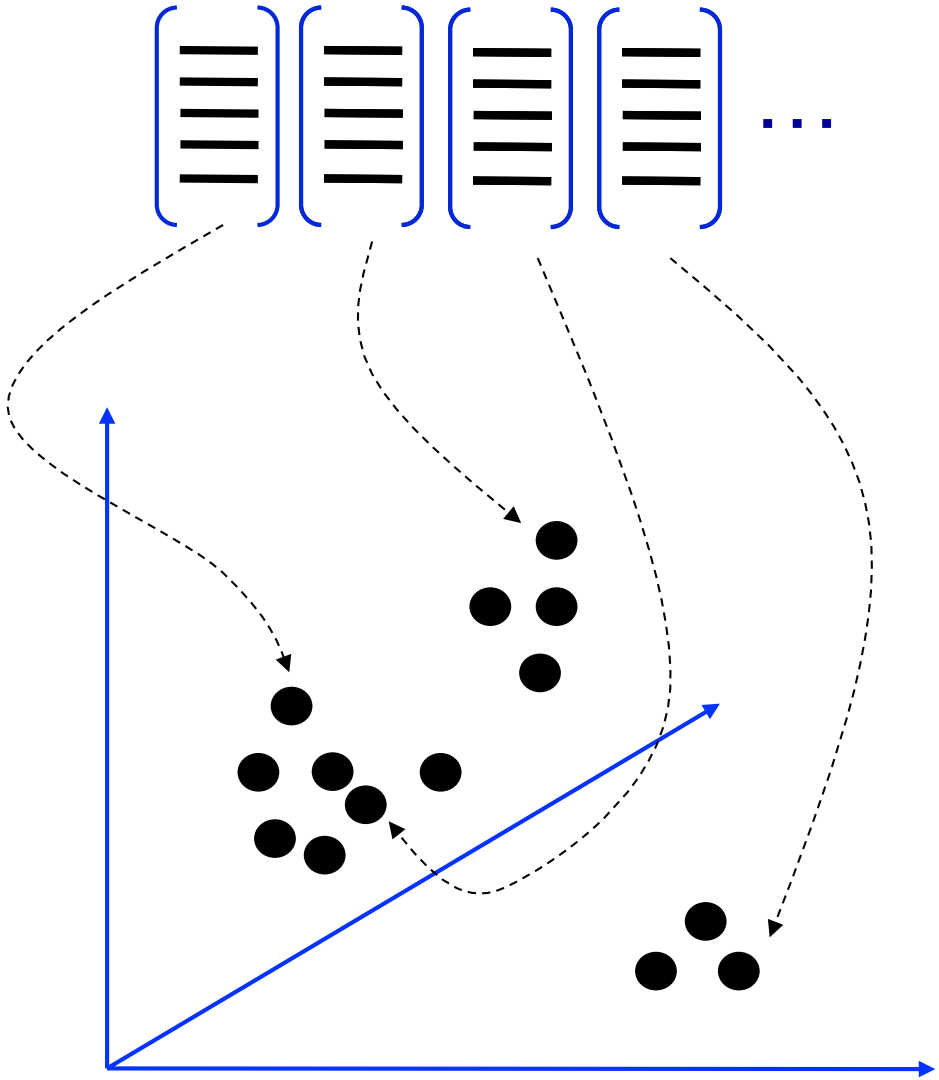
Extract Feature Descriptors from a Training Dataset

Visual feature descriptor vectors (e.g., SIFT)



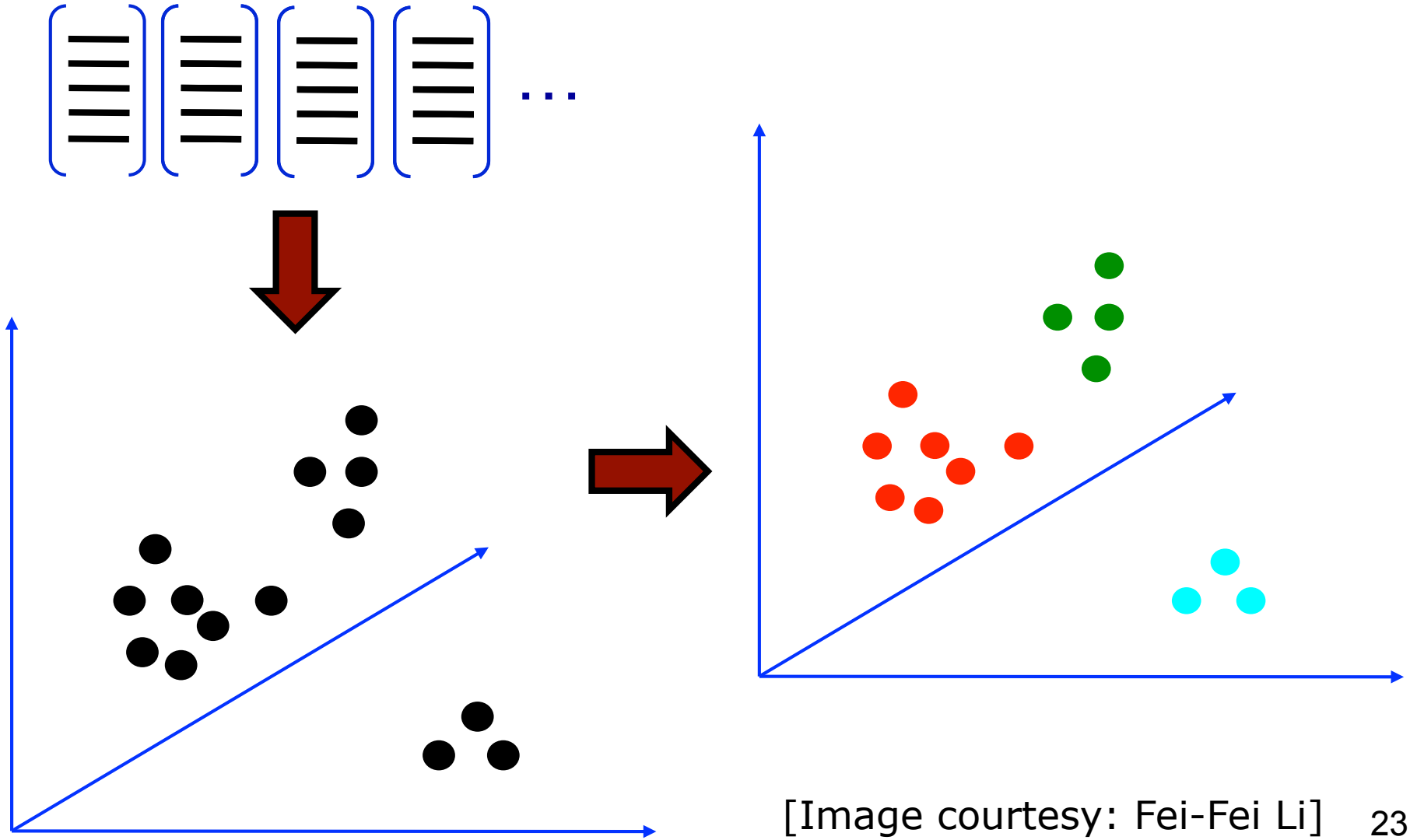
[Partial image courtesy: Fei-Fei Li]

Feature Descriptors are Points in a High-Dimensional Space



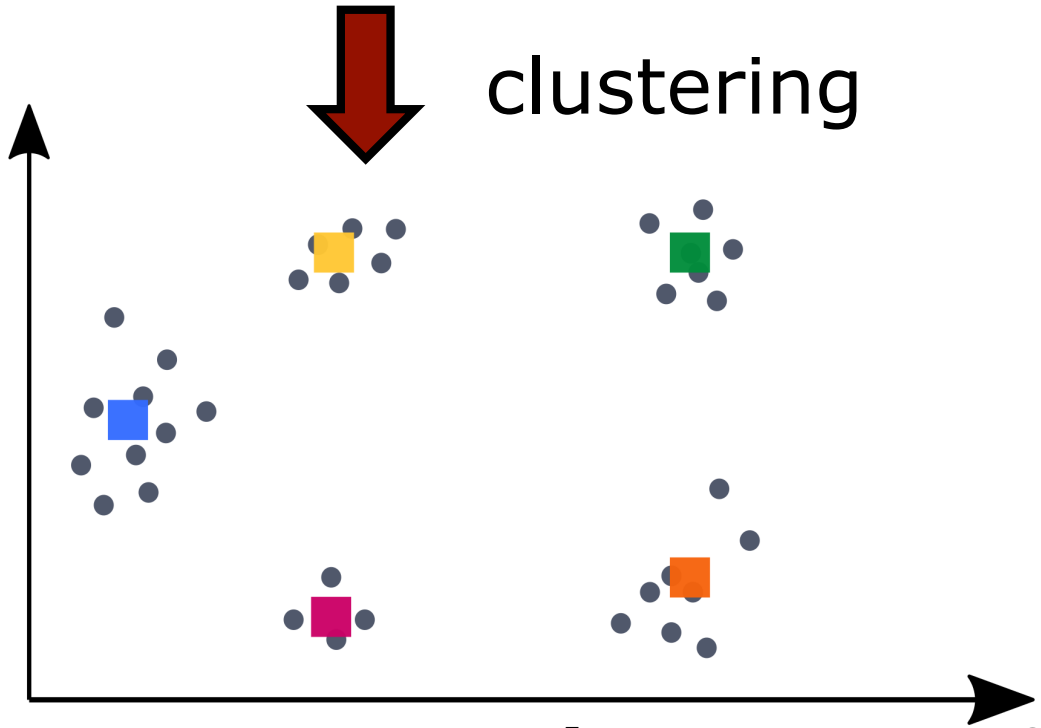
[Image courtesy: Fei-Fei Li]

Group Similar Descriptors



[Image courtesy: Fei-Fei Li]

Clusters of Descriptors from Data Forms the Dictionary



[Image courtesy: Olga Vysotska]

K-Means Clustering

K-Means Clustering

- Partitions the data into k clusters
- Clusters are represented by centroids
- A centroid is the mean of data points

Objective:

- Find the k cluster centers and assign the data points to the nearest one, such that the squared distances to the cluster centroids are minimized

K-Means Clustering for Learning the BoVW Dictionary

- Partitions the features into k groups
- The centroids form the dictionary
- Features will be assigned to the closest centroid (visual word)

Approach:

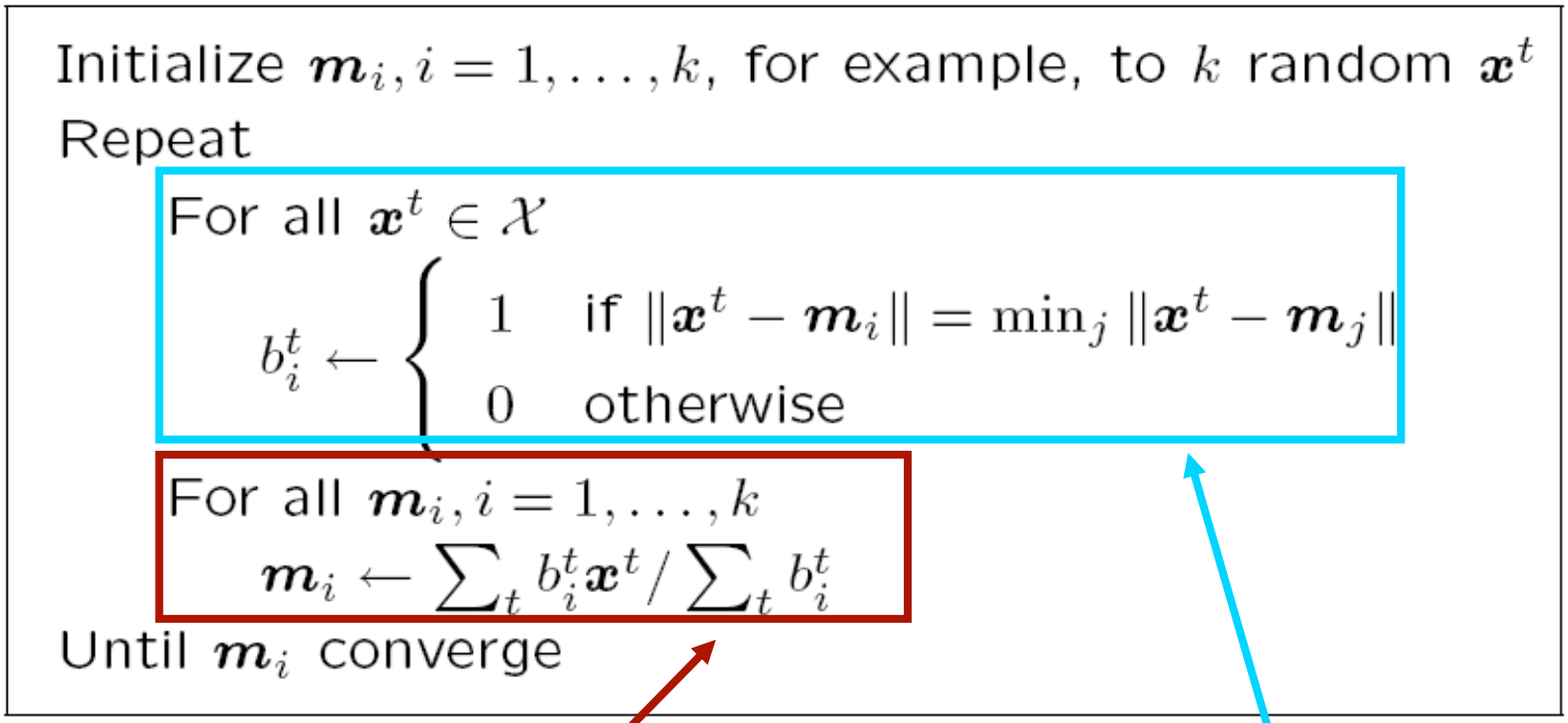
- Find k word and assign the features to the nearest word, such that the squared distances are minimized

K-Means Clustering (Informally)

- Initialization: Choose k arbitrary centroids as cluster representatives

- Repeat until convergence
  - Assign each data point to the closest centroid
  - Re-compute the centroids of the clusters based on the assigned data points

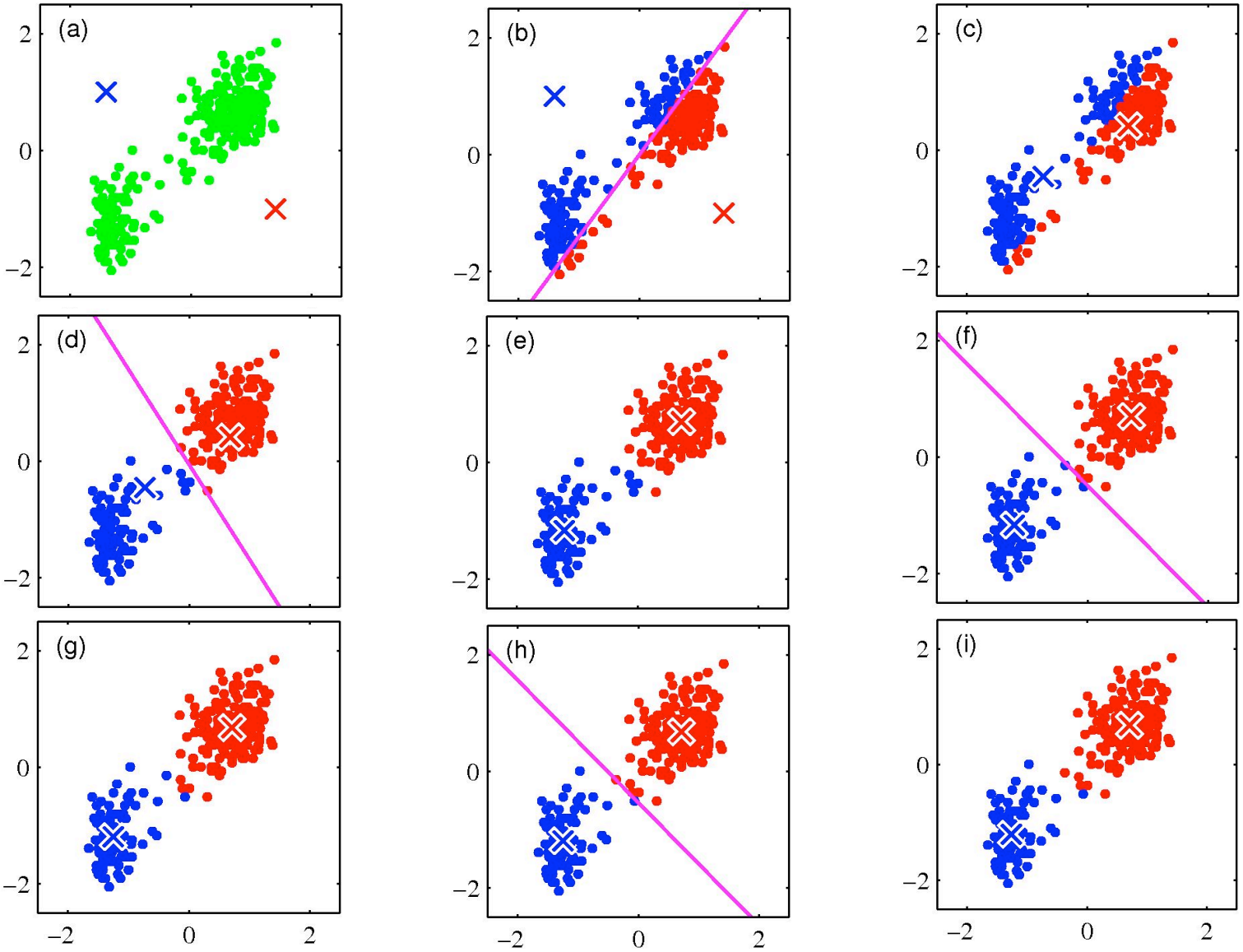
K-Means Algorithm



Re-compute the cluster means using the current cluster memberships

Assign each data point to the closest cluster

K-Means Example



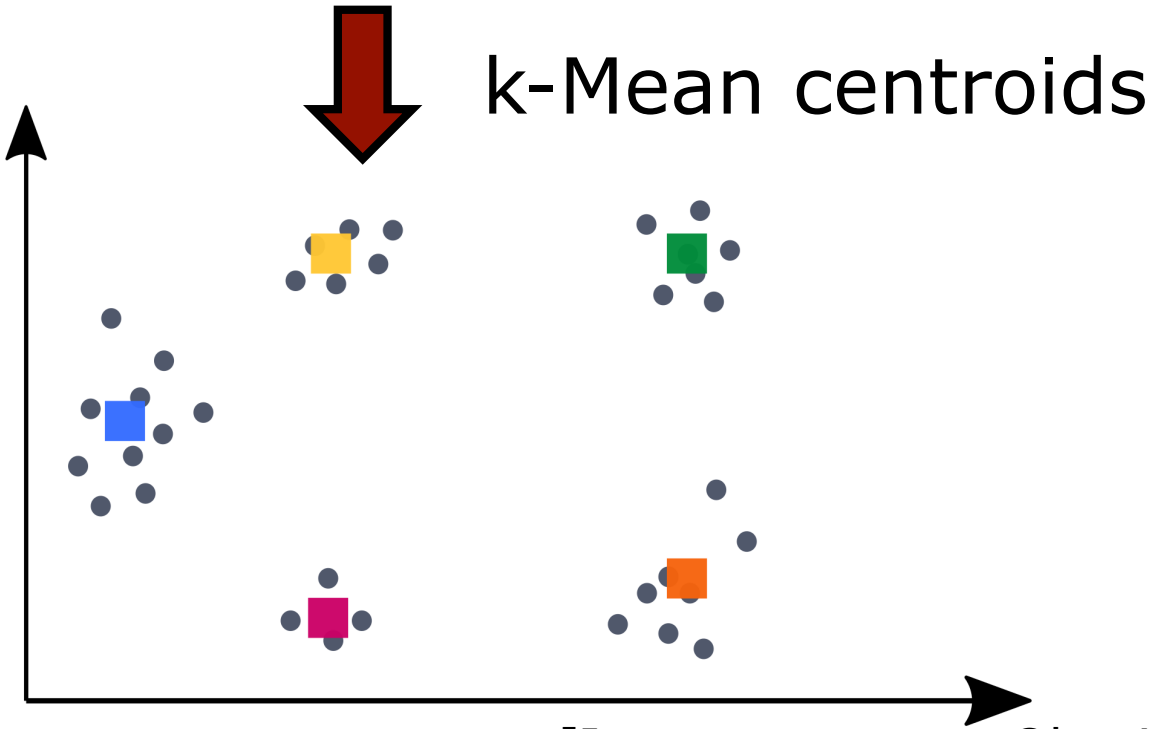
[Image courtesy: Bishop]

Summary K-Means

- Standard approach to clustering
- Simple to implement
- Number of clusters k must be chosen
- Depends on the initialization
- Sensitive to outliers
- Prone to local minima

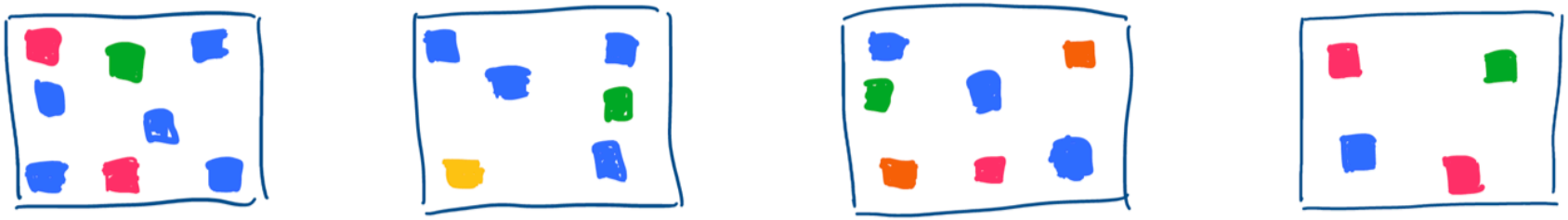
We use k-means to compute the dictionary of visual words

K-Means for Building the Dictionary from Training Data



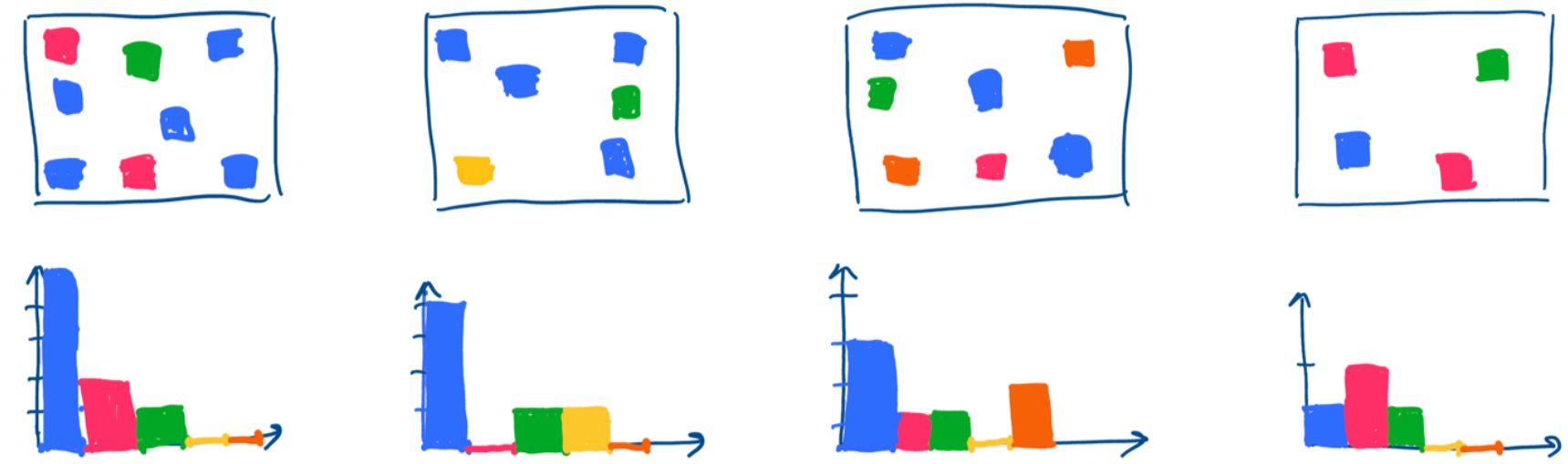
[Image courtesy: Olga Vysotska]

All Images are Reduced to Visual Words



[Image courtesy: Olga Vysotska]

All Images are Represented by Visual Word Occurrences



Every image turns into a histogram

[Image courtesy: Olga Vysotska]

Bag of Visual Words Model

- Compact summary of the image content
- Largely invariant to viewpoint changes and deformations
- Ignores the spatial arrangement
- Unclear how to choose optimal size of the vocabulary
  - Too small: Words not representative of all image regions
  - Too large: Over-fitting

How to Find Similar Images?

Task Description

- **Task:** Find similar looking images
- **Input:**
  - Database of images
  - Dictionary
  - Query image(s)
- **Output:**
  - The N most similar database images to the query image

?

Image Similarity by Comparing Word Occurrence Histograms

[Image courtesy: Olga Vysotska]

How to Compare Histograms?

- Euclidean distance of two points?
- Angle between two vectors?
- Kullback Leibler divergence (KLD)?
- Something else?

[Image courtesy: Olga Vysotska]

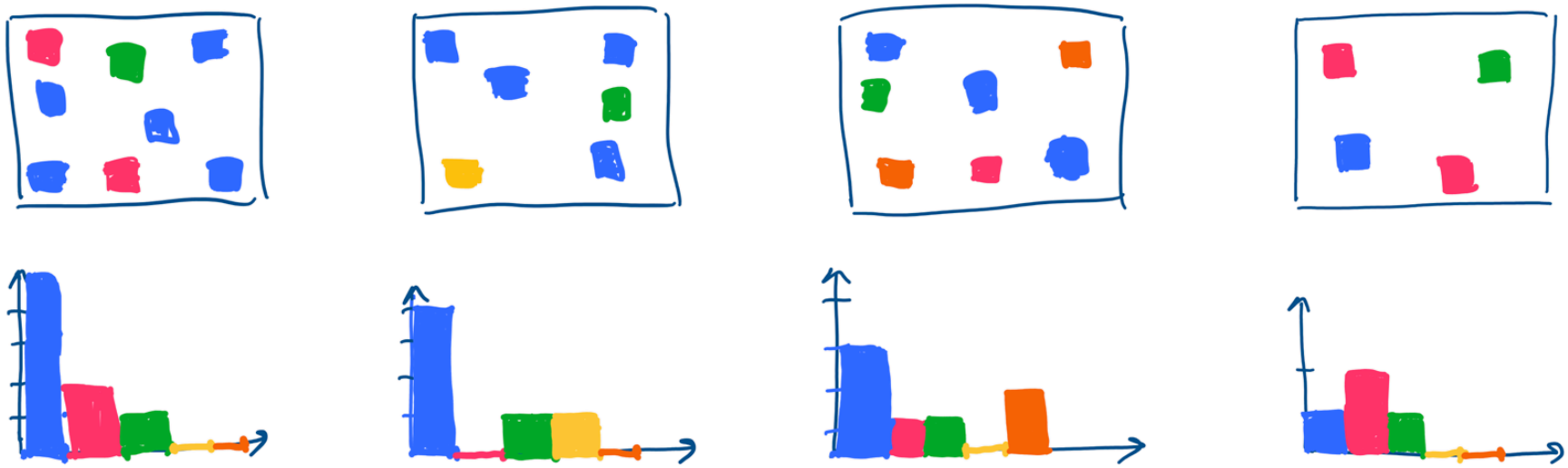
Are All Words Expressive for Comparing Histograms?

- Should all visual words be treated in the same way?
- Text analogy: What about articles?

[Image courtesy: Olga Vysotska]

Some Word are Less Expressive Than Others!

- Words that occur in every image do not help a lot for comparisons



- Example: the “green word” is useless

[Image courtesy: Olga Vysotska]

TF-IDF Reweighting

- Weight words considering the probability that they appear
- TF-IDF = term frequency – inverse document frequency
- Every bin is reweighted

$$t_{id} = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

bin **normalize** **weight**

TF-IDF

term frequency

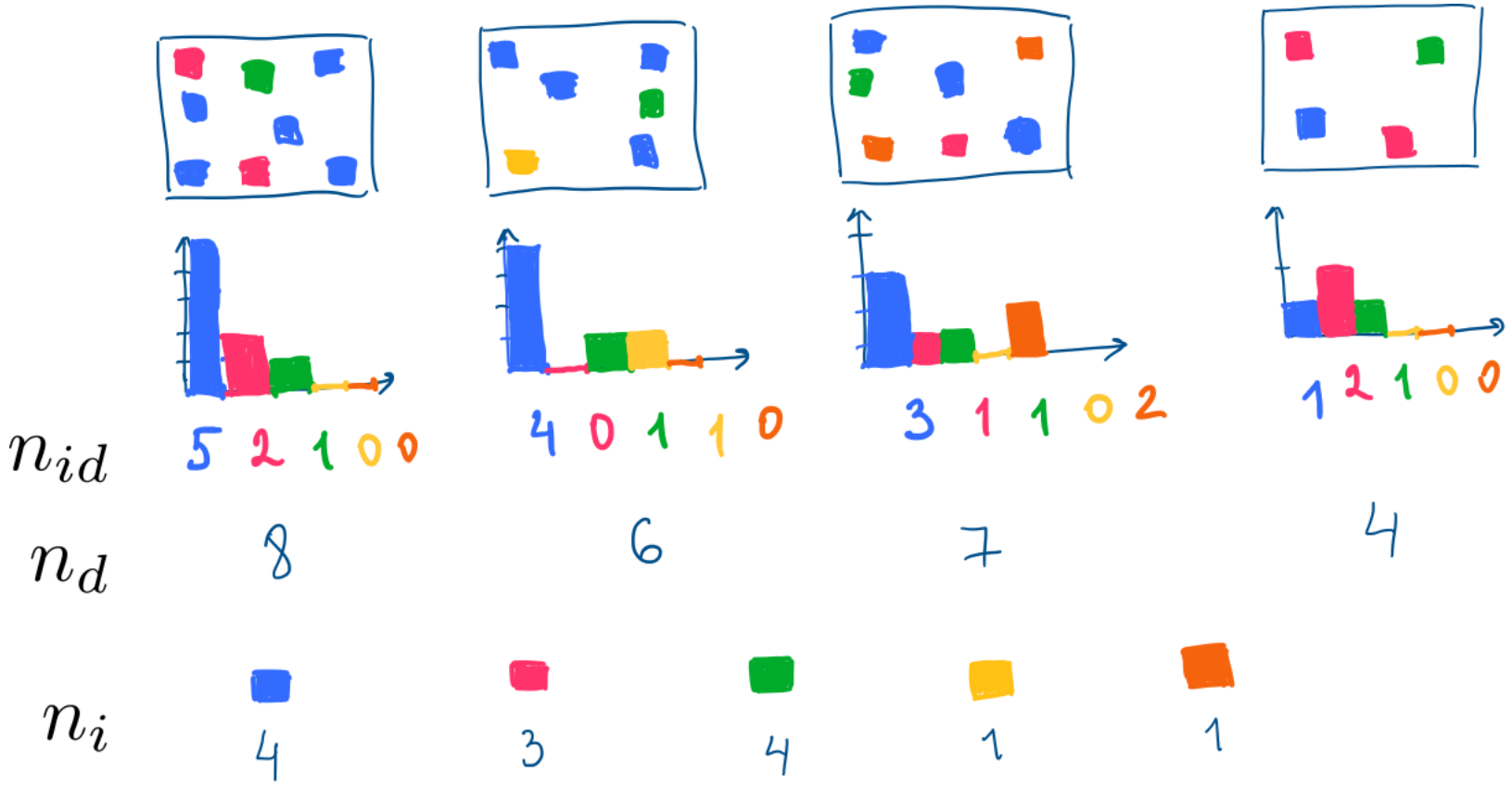
$$t_{id} = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

inverse document frequency

bin of word i in image d

- t_{id}: histogram bin of word i for image d
- n_{id}: occurances of word i in image d
- n_d: number of word occurances in image d
- n_i: number of images that contain word i
- N: number of images

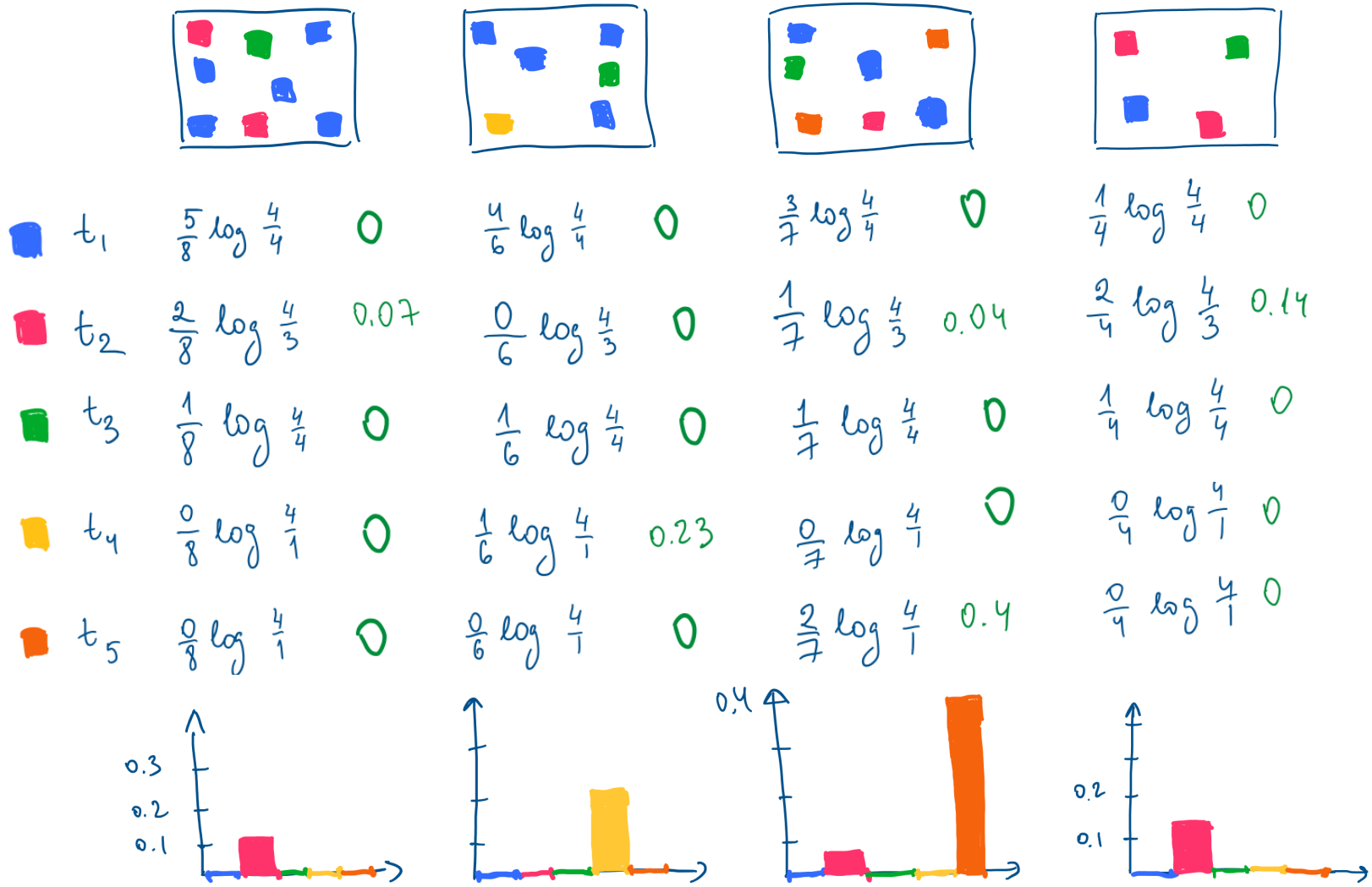
Computing the TF-IDF (1)



$$t_{id} = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

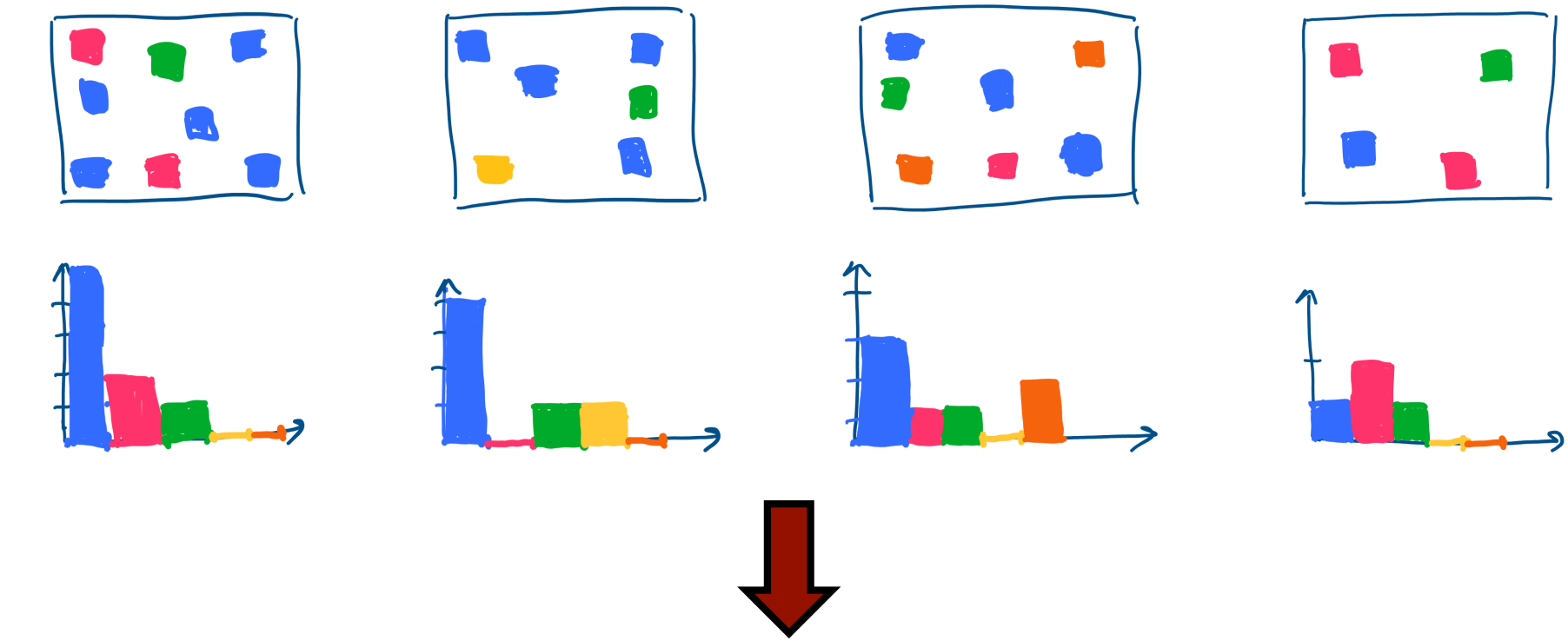
[Image courtesy: Olga Vysotska]

Computing the TF-IDF (2)



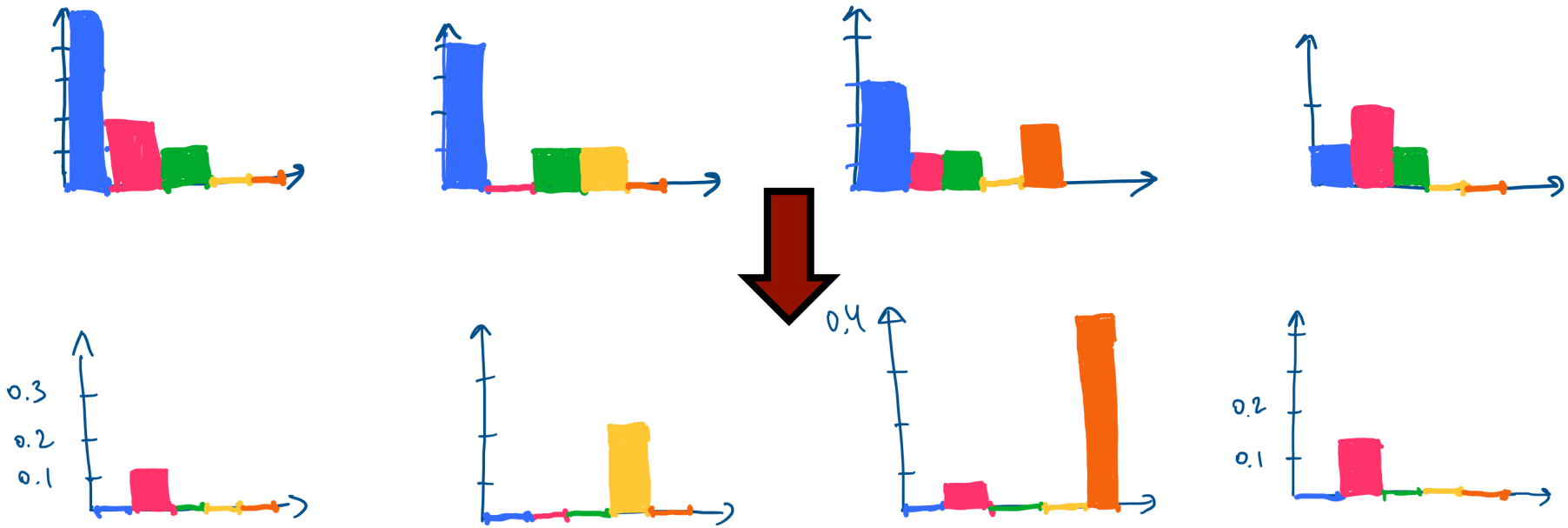
[Image courtesy: Olga Vysotska]

Reweighted Histograms



[Image courtesy: Olga Vysotska]

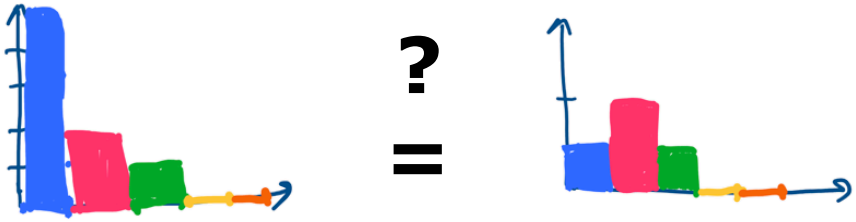
Reweighted Histograms



- Relevant words get higher weights
- Others are weighted down to zero (those occurring in every image)

[Image courtesy: Olga Vysotska]

Comparing Two Histograms

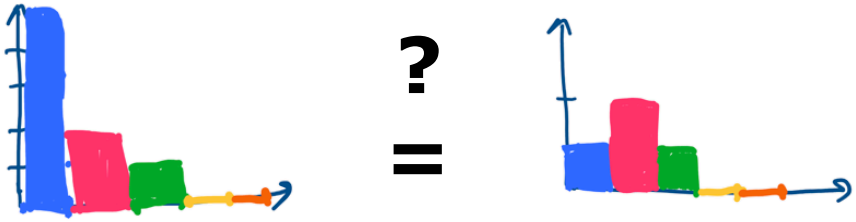


Options

- Euclidean distance of two points
- Angle between two vectors

[Image courtesy: Olga Vysotska]

Comparing Two Histograms



Options

- Euclidean distance of two vectors
- **Angle between two vectors**

BoVW approaches often use the cosine distance for comparisons

[Image courtesy: Olga Vysotska]

Cosine Similarity and Distance

- Cosine similarity considers the cosine of the angle between vectors:

$$\text{cossim}(\mathbf{x}, \mathbf{y}) = \cos(\theta) = \frac{\mathbf{x}^\top \mathbf{y}}{||\mathbf{x}|| \, ||\mathbf{y}||}$$

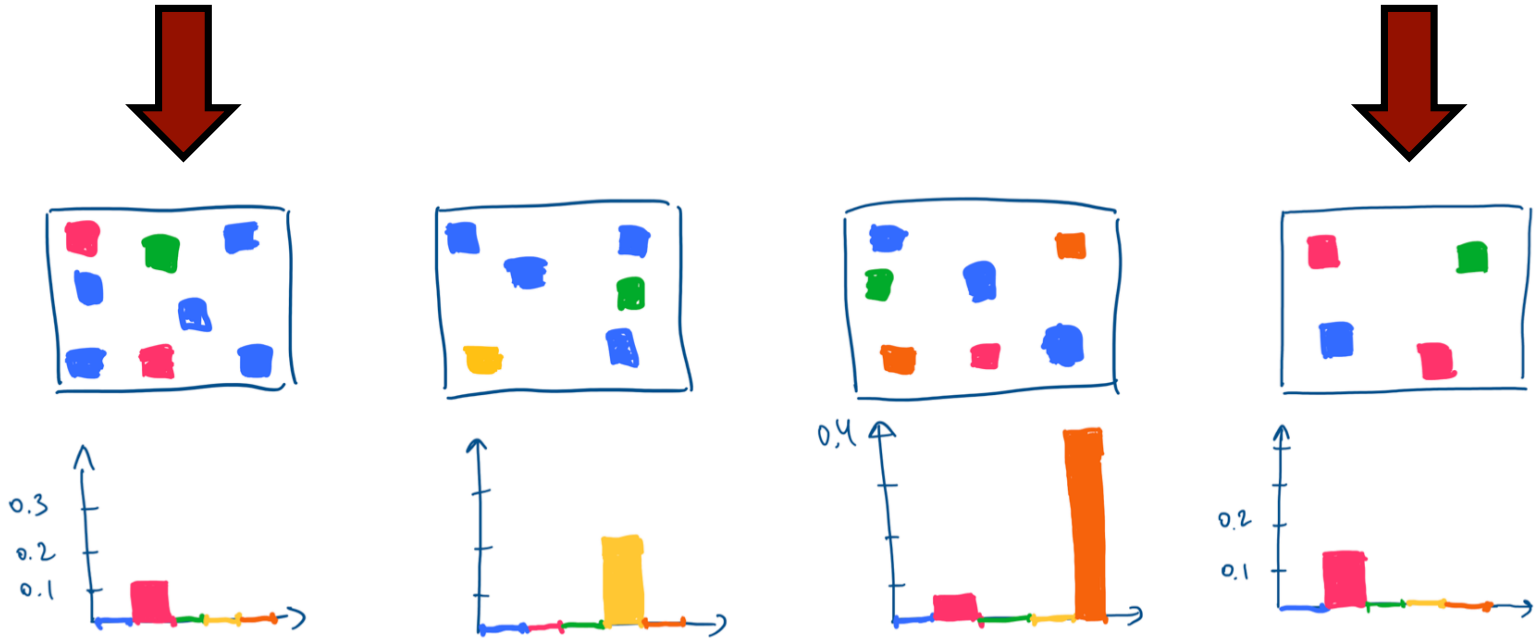
- We use the cosine distance

$$\text{d}_{\cos}(\mathbf{x}, \mathbf{y}) = 1 - \text{cossim}(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x}^\top \mathbf{y}}{||\mathbf{x}|| \, ||\mathbf{y}||}$$

- Takes values between 0 and 1 (for vectors in the 1st quadrant)

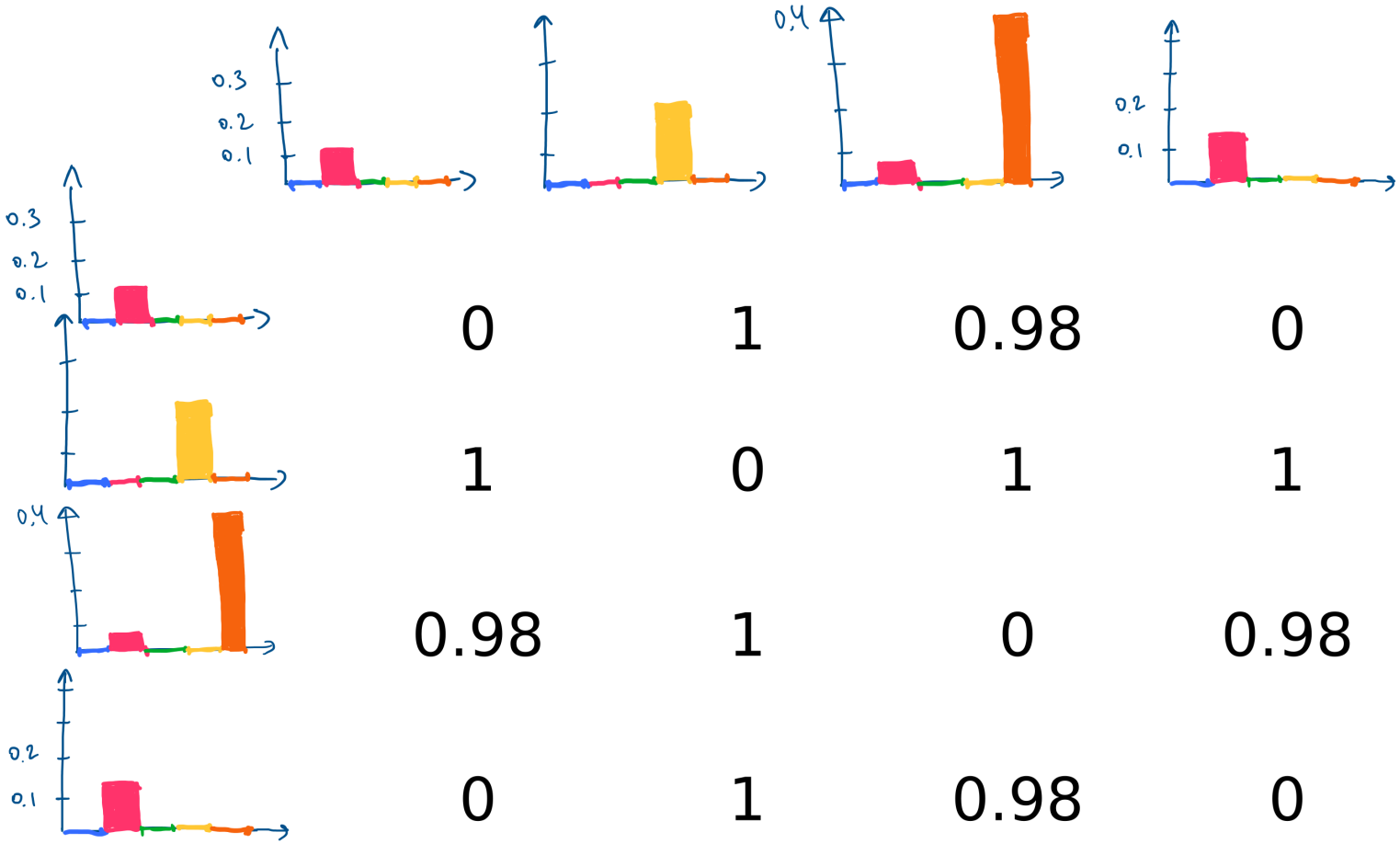
Example Comparing Histograms

- 4 images
- Image 0 and image 3 are similar



[Image courtesy: Olga Vysotska]

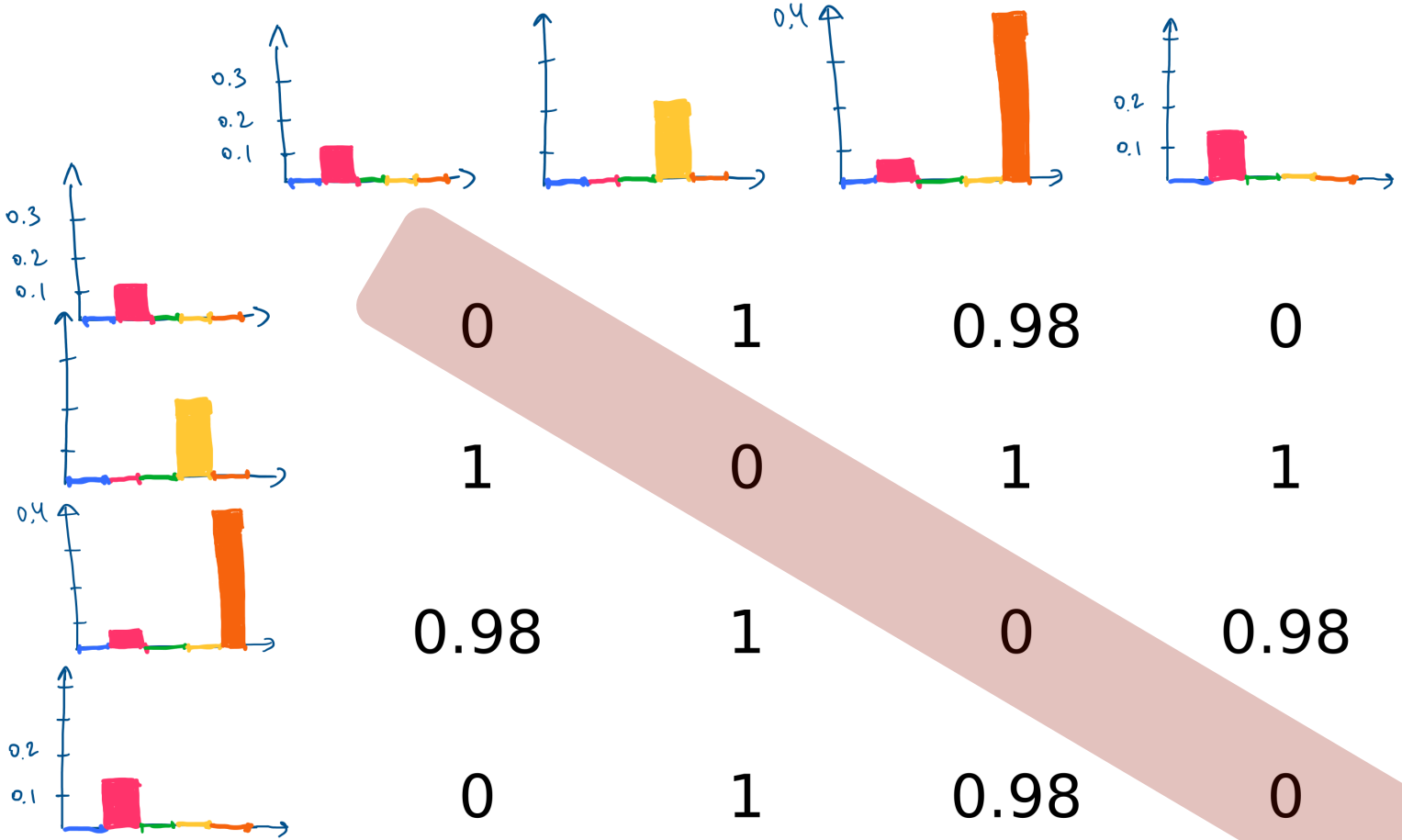
Example Comparing Histograms



0	1	0.98	0
1	0	1	1
0.98	1	0	0.98
0	1	0.98	0

[Image courtesy: Olga Vysotska]

Example Comparing Histograms

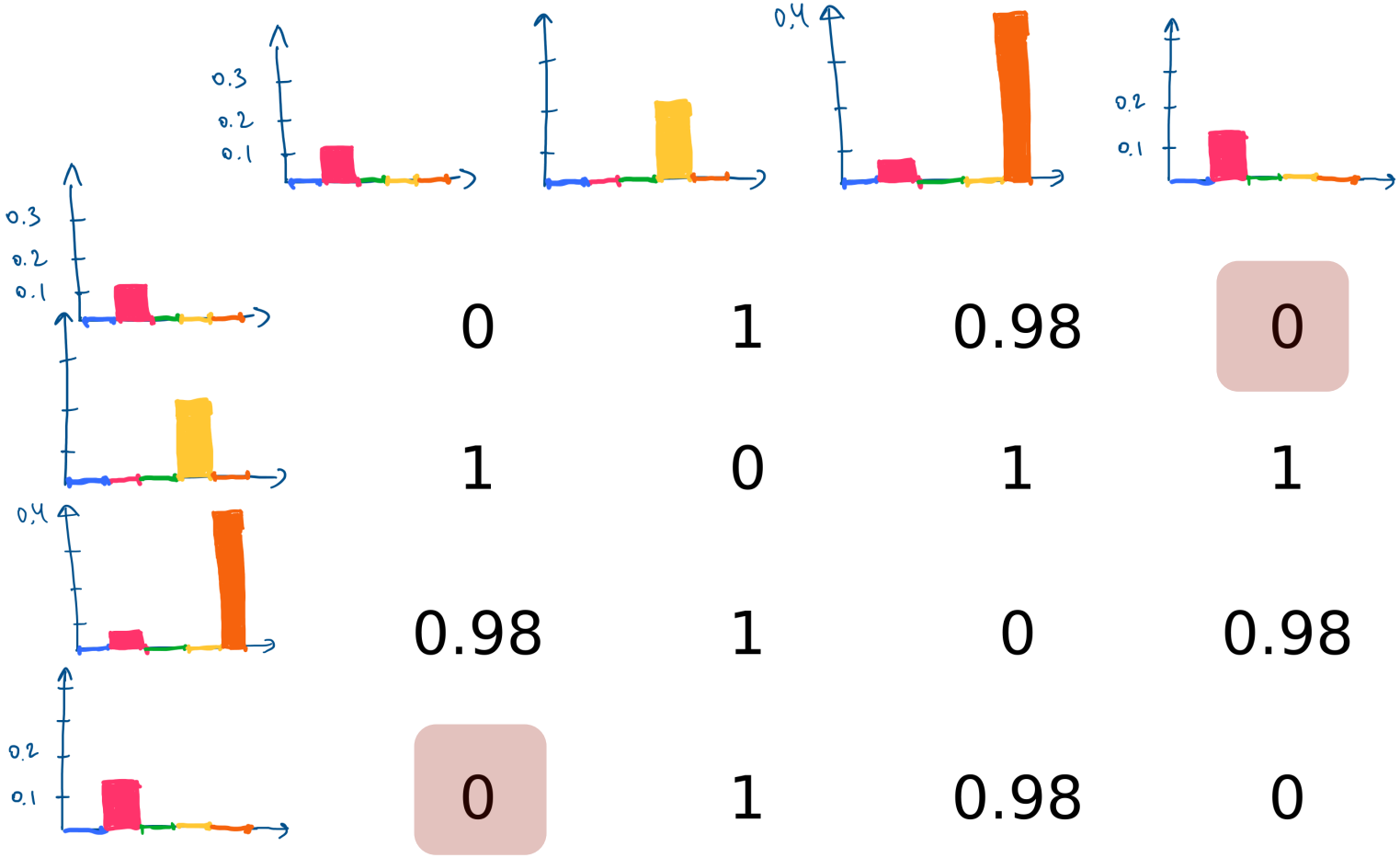


0	1	0.98	0
1	0	1	1
0.98	1	0	0.98
0	1	0.98	0

Images have a zero distance to themselves

[Image courtesy: Olga Vysotska]

Example Comparing Histograms

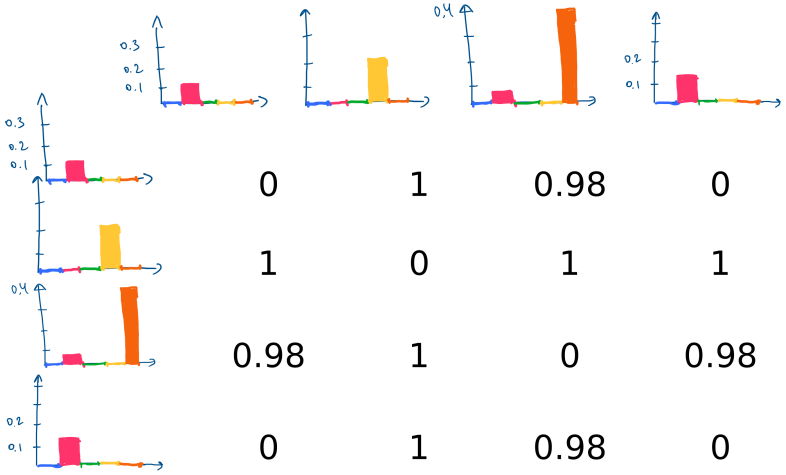


0	1	0.98	0
1	0	1	1
0.98	1	0	0.98
0	1	0.98	0

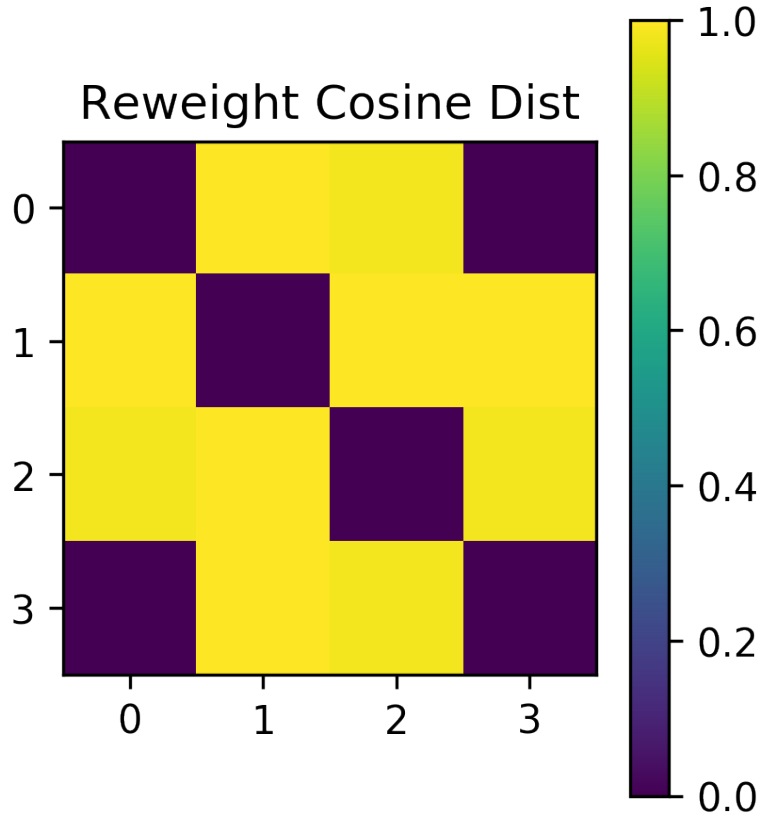
Images 0 and 3 are highly similar

[Image courtesy: Olga Vysotska]

Cost Matrix

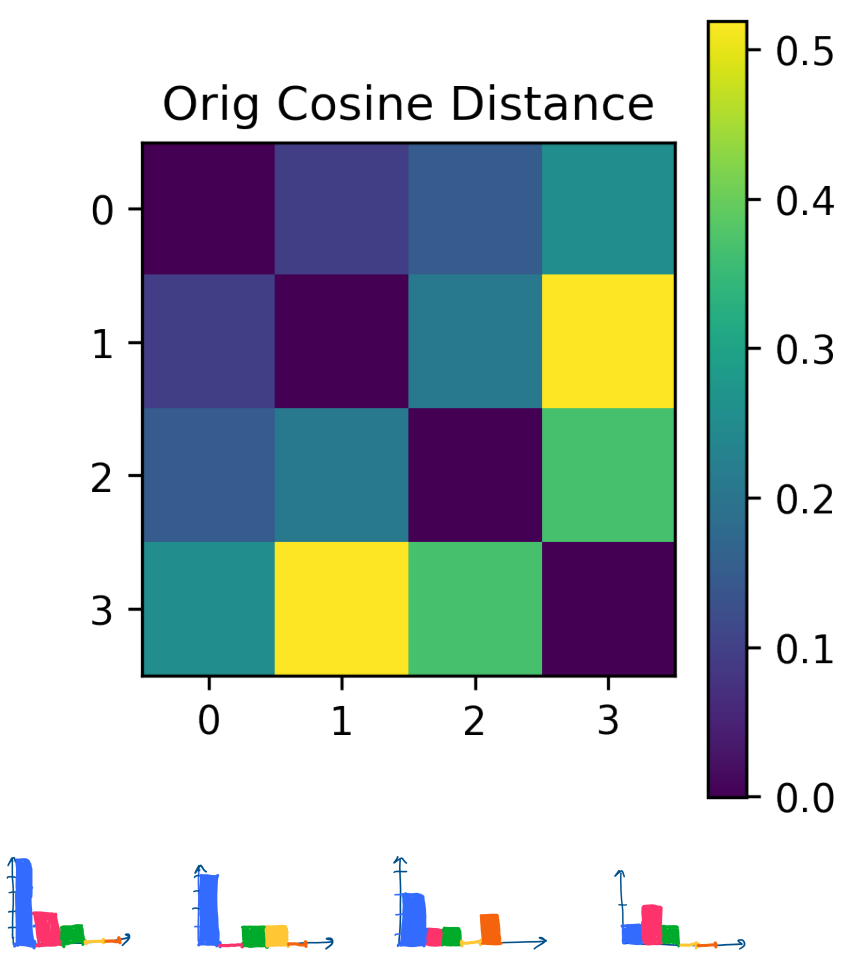


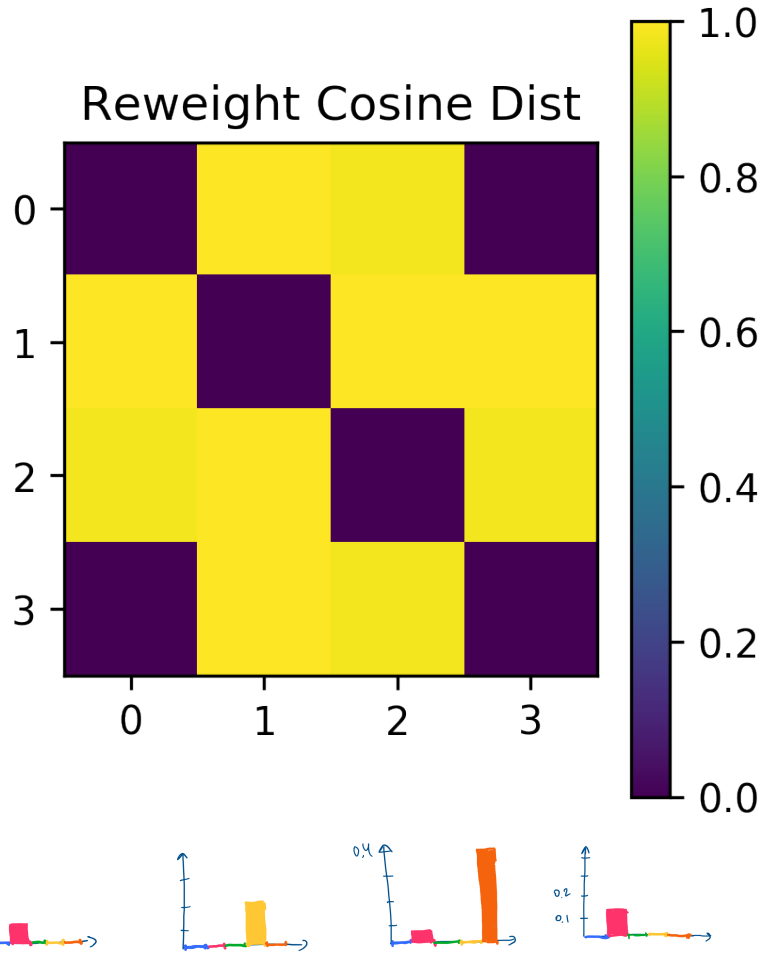
0	1	0.98	0
1	0	1	1
0.98	1	0	0.98
0	1	0.98	0



[Image courtesy: Olga Vysotska]

IF-IDF Actually Helps





original histograms

TF-IDF histograms

[Image courtesy: Olga Vysotska]

Euclidean vs. Cosine Distance

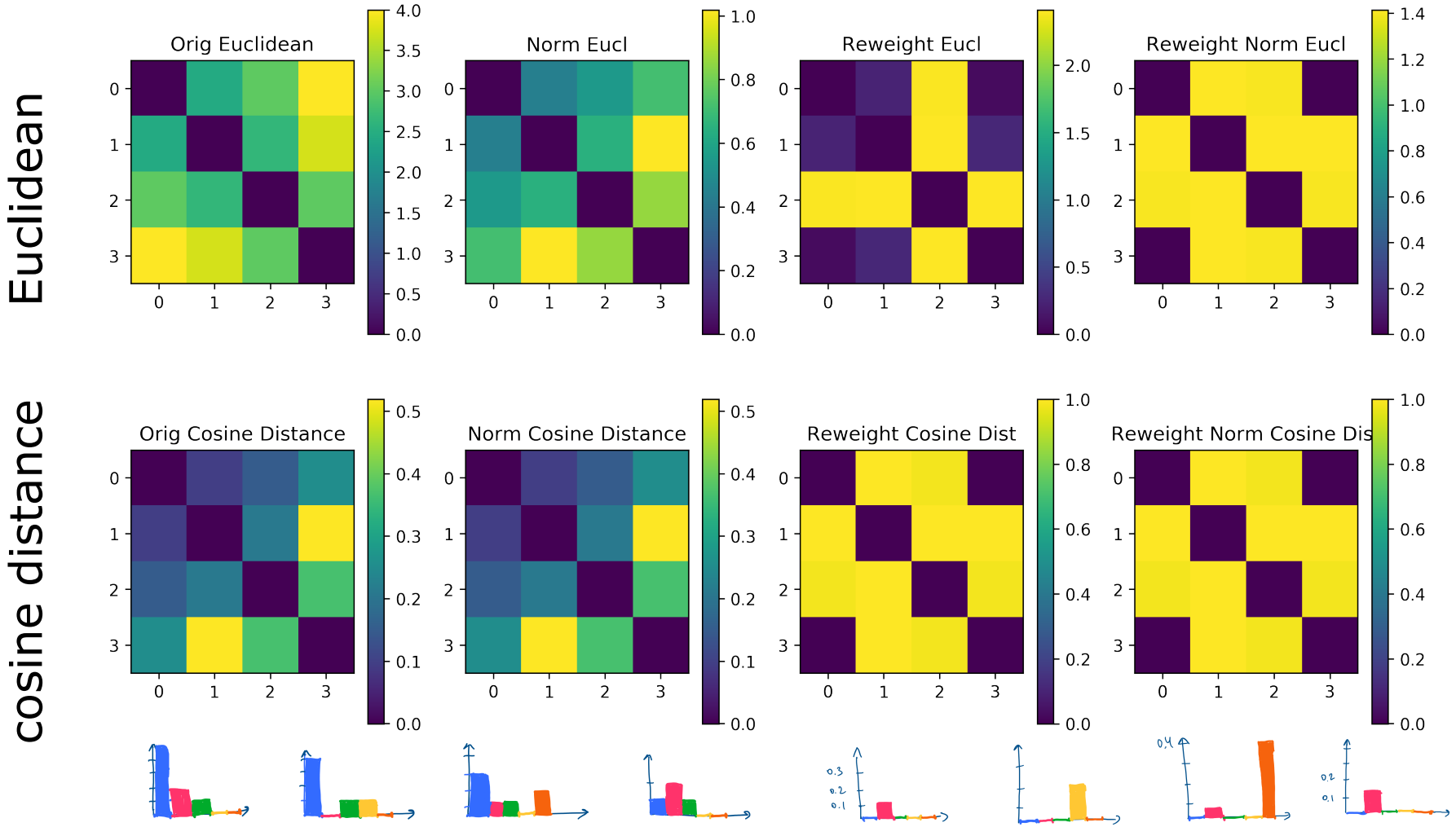
- Cosine distance ignores the length of the vectors
- **For vectors of length 1**, the squared Euclidean and the cosine distance only differ by a factor of 2:

$$\begin{aligned} ||\mathbf{x}-\mathbf{y}||^2 &= (\mathbf{x}-\mathbf{y})^\top(\mathbf{x}-\mathbf{y}) \\ &= \mathbf{x}^\top\mathbf{x} - 2\mathbf{x}^\top\mathbf{y} + \mathbf{y}^\top\mathbf{y} \end{aligned}$$

$$\text{as } ||\mathbf{x}|| = ||\mathbf{y}|| = 1$$

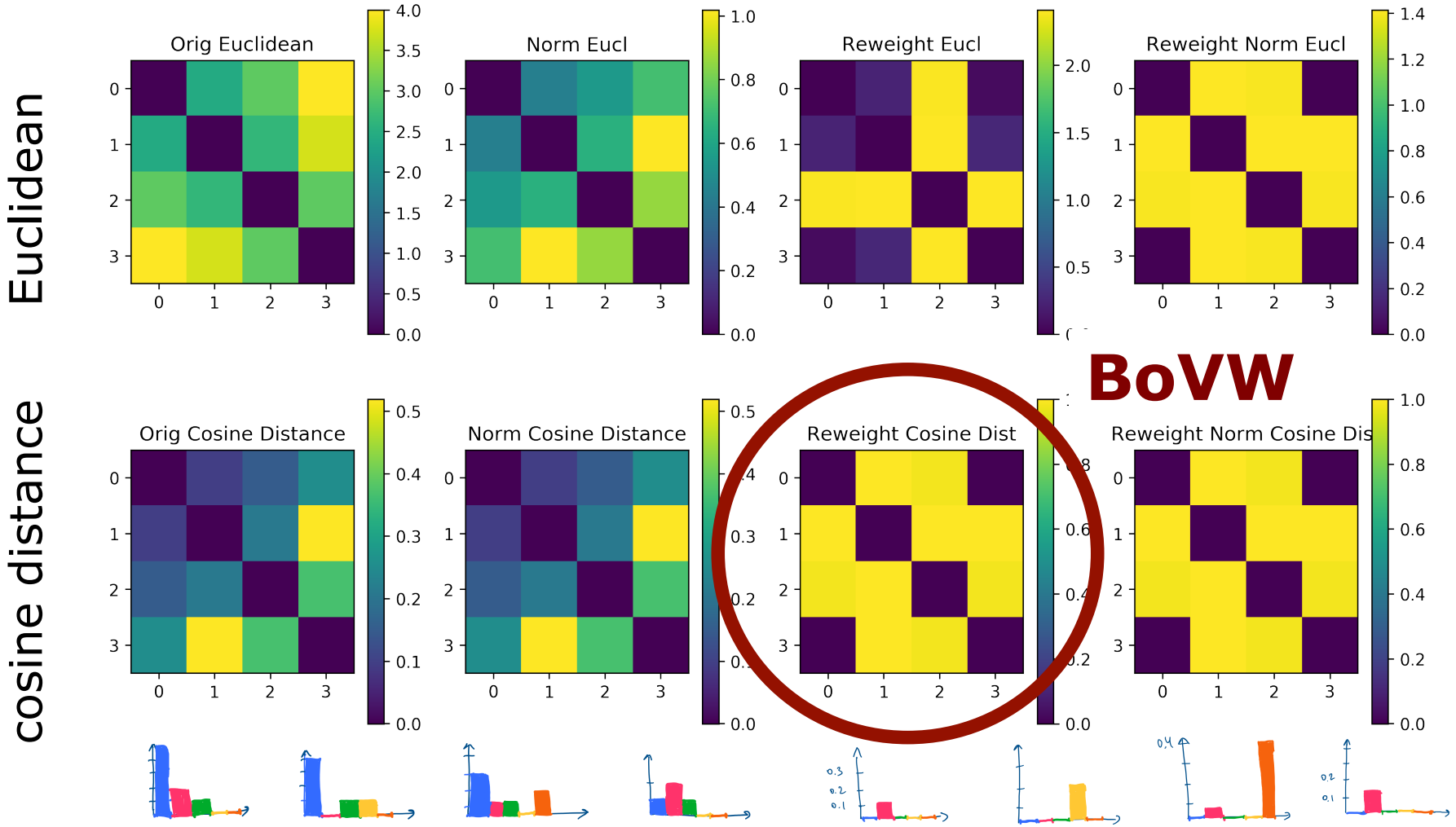
$$\begin{aligned} ||\mathbf{x}-\mathbf{y}||^2 &= 2 - 2\mathbf{x}^\top\mathbf{y} = 2 - 2\cos\theta \\ &= 2\,\mathrm{d}_{\cos}(\mathbf{x},\mathbf{y}) \end{aligned}$$

Comparison of Distance Metrics



[Image courtesy: Olga Vysotska]

Comparison of Distance Metrics



[Image courtesy: Olga Vysotska]

Similarity Queries

- Database stores TF-IDF weighted histograms for all database images

Find similar images by

- Extract features from query image
- Assign features to visual words
- Build TF-IDF histogram for query image
- Return N most similar histograms from database under cosine distance

Summary

- BoVW is an approach to compactly describe images and compute similarities between images
- Based in a set of visual words
- Images become histograms of visual word occurrences
- TF-IDF weighting for increasing the influence of expressive words
- Similarity = histogram similarity
- Cosine distance

Learning-Based Methods

Deep Learning Revolution

-new take on algorithms
-large amount of data
-large amount of computing (GPU)

2010: ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is launched

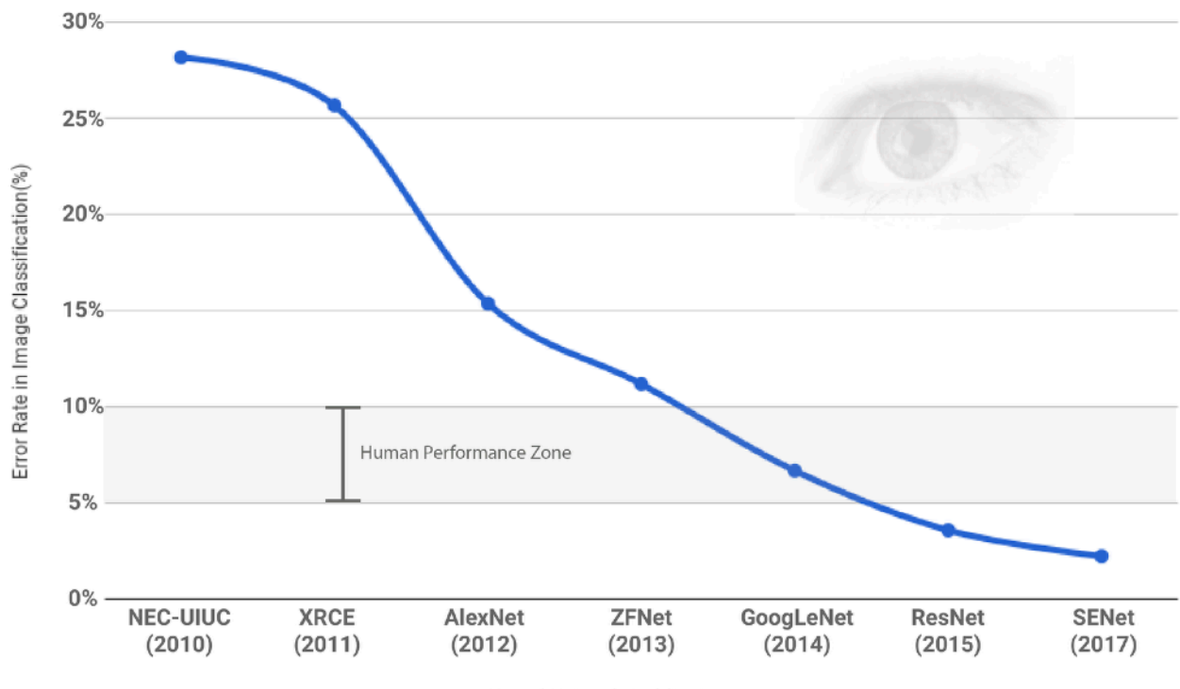
14M images

objects/ bounding boxes

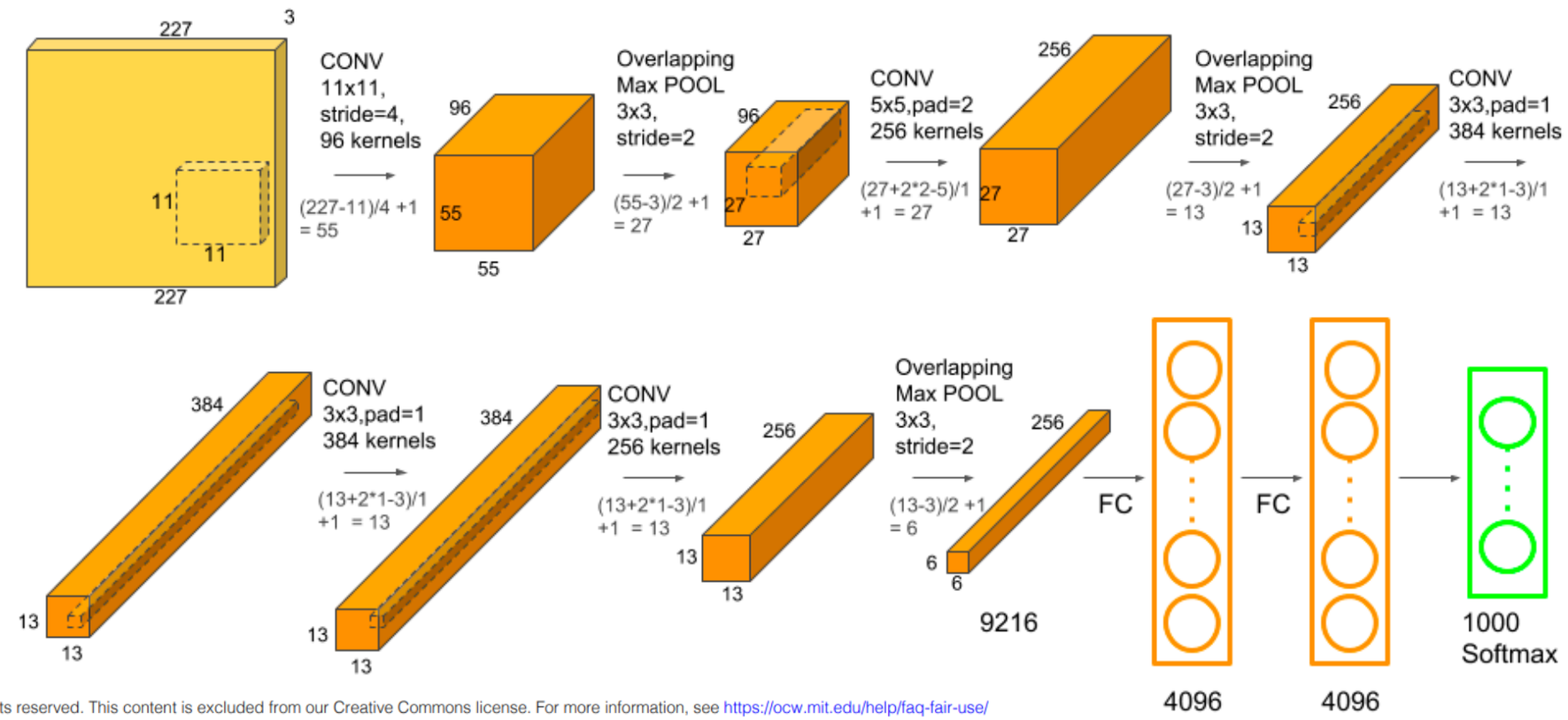
>1k classes

Deep Learning Revolution

AlexNet:
- winning entry in ILSVRC 2012
- CNN
- 10% error reduction



RGB image as input:



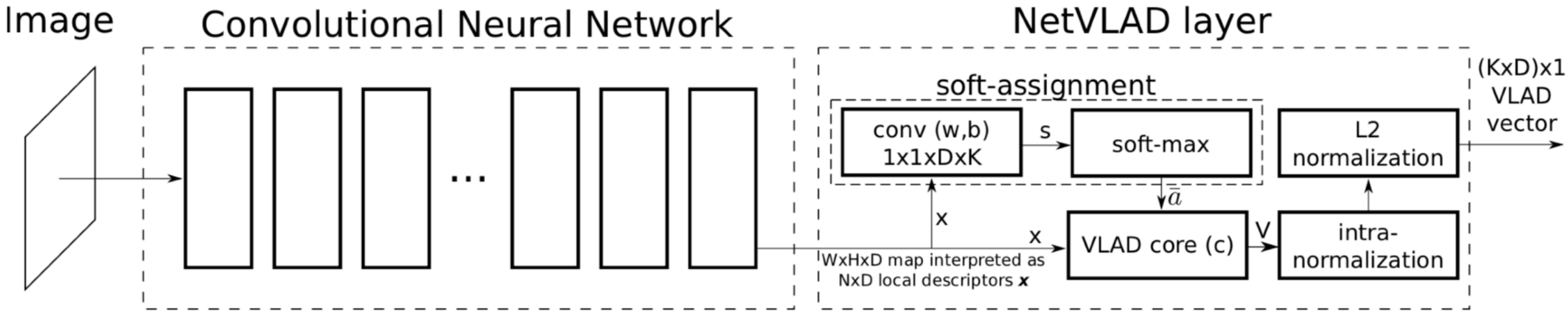
Classification results

Learning-based Descriptors: NetVLAD

Earlier approaches: using AlexNet or similar and use layers activations as descriptors

NetVLAD:

- CNN-based approach
- Trained on the task of place recognition
- Clever use of Internet data for training



R. Arandjelovix, P. Gronat, J. Sivic, NetVLAD: CNN Architecture for Weakly Supervised Place Recognition, CVPR'16.

Learning-based Descriptors: NetVLAD

How to get labeled data?

- a large dataset of panoramic images from the Google Street View Time Machine
- positions based on their (noisy) GPS
- Seasonal variations
- Illumination changes

(a) (b) (c)

Figure 4. **Google Street View Time Machine examples.** Each column shows perspective images generated from panoramas from nearby locations, taken at different times. A well designed method can use this source of imagery to learn to be invariant to changes in viewpoint and lighting (a-c), and to moderate occlusions (b). It can also learn to suppress confusing visual information such as clouds (a), vehicles and people (b-c), and to chose to either ignore vegetation or to learn a season-invariant vegetation representation (a-c). More examples are given in appendix B.

Metrics

True positives (TP): correct matches
False positives (FP): incorrect matches
False negatives (FN): missed matches

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

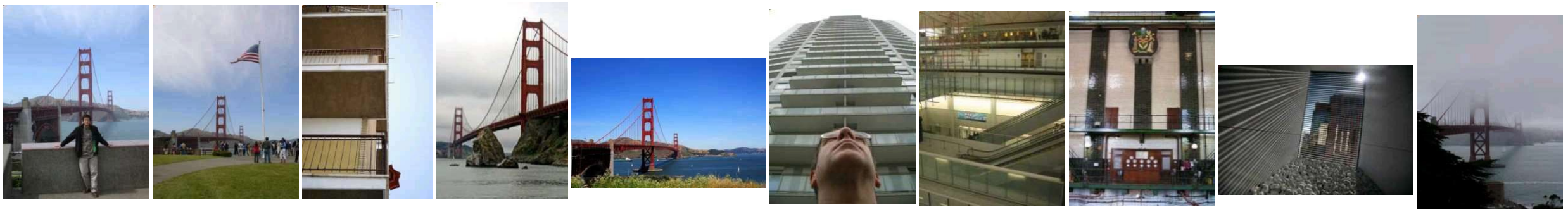
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

Perfect system:
100% precision (0 FP)
100% recall (0 FN)

Scoring Retrieval Quality Example

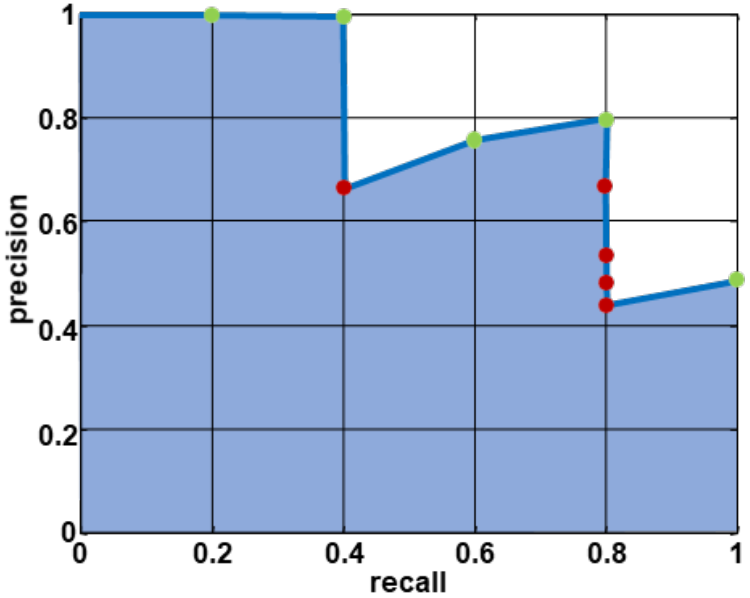
Query

Results (ordered):

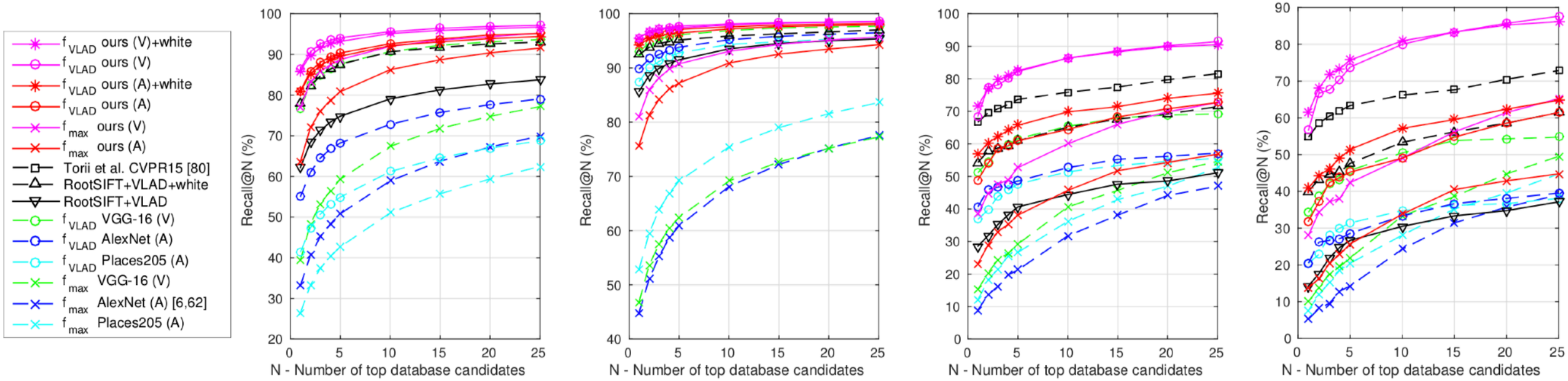


Database size: 10 images
Relevant (total): 5 images

Precision = #relevant / #returned
Recall = #relevant / #total relevant



Learning-based Descriptors: NetVLAD



(a) Pitts250k-test (b) TokyoTM-val (c) Tokyo 24/7 all queries (d) Tokyo 24/7 sunset/night

Figure 5. **Comparison of our methods versus off-the-shelf networks and state-of-the-art.** The base CNN architecture is denoted in brackets: (A)lexNet and (V)GG-16. Trained representations (red and magenta for AlexNet and VGG-16) outperform by a large margin off-the-shelf ones (blue, cyan, green for AlexNet, Places205, VGG-16), f_{VLAD} (-o-) works better than f_{max} (-x-), and our f_{VLAD}+whitening (-∗-) representation based on VGG-16 sets the state-of-the-art on all datasets. [80] only evaluated on Tokyo 24/7 as the method relies on depth data not available in other datasets. Additional results are shown in appendix C.

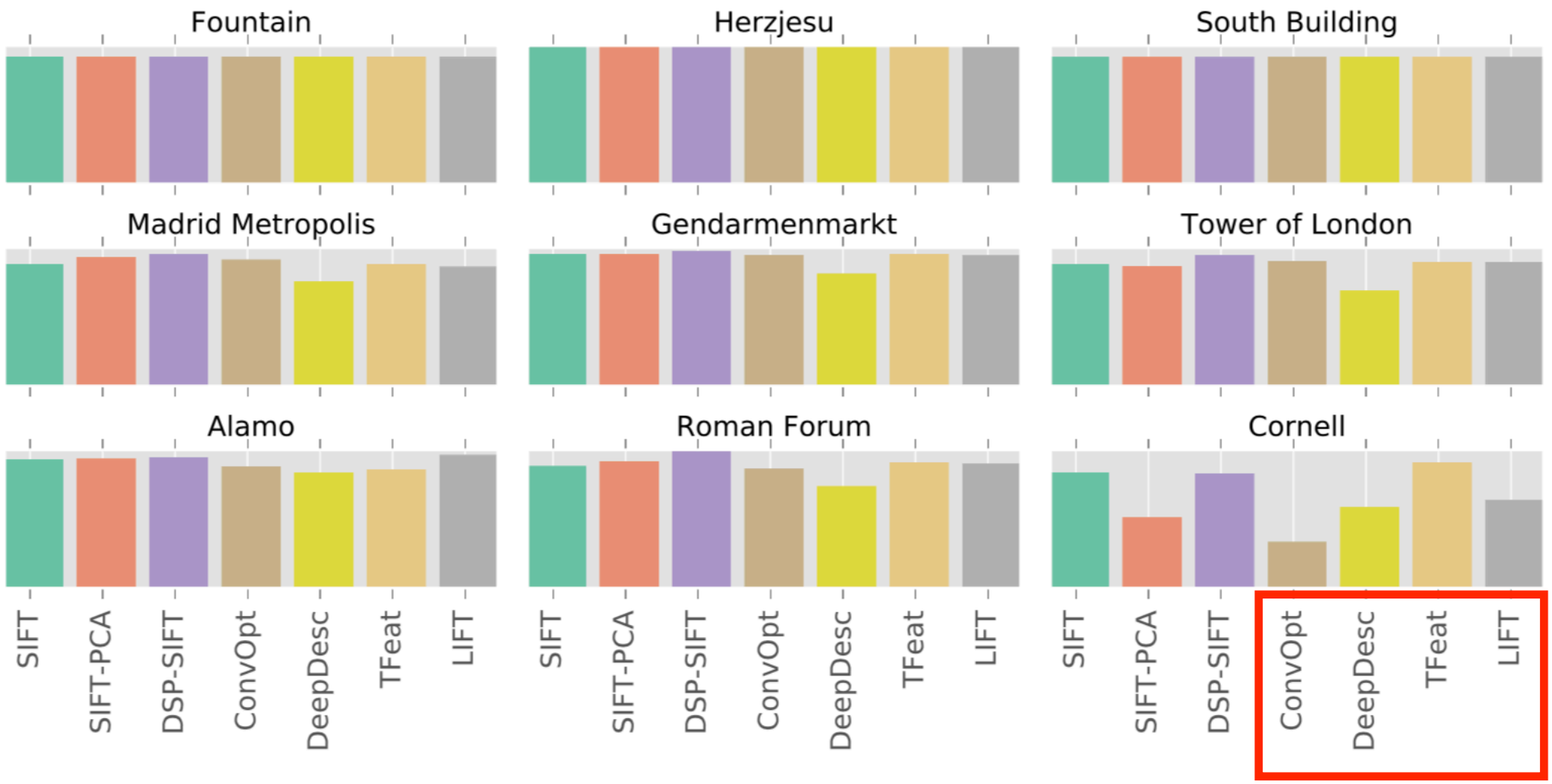
- query image is deemed correctly localized if at least one of the top N retrieved database images is within d = 25 meters from the ground truth position of the query.

- percentage of correctly recognized queries (Recall) is then plotted for different values of N

R. Arandjelovix, P. Gronat, J. Sivic, NetVLAD: CNN Architecture for Weakly Supervised Place Recognition, CVPR'16.

Handcrafted vs. Learned Local Descriptors

- learned descriptors typically outperform SIFT in terms of recall, while SIFT performs better in terms of precision
- advanced SIFT variants outperform learned features
- learned descriptors have high variance across the different datasets (i.e., over-fitting)



Number of registered images for the different methods

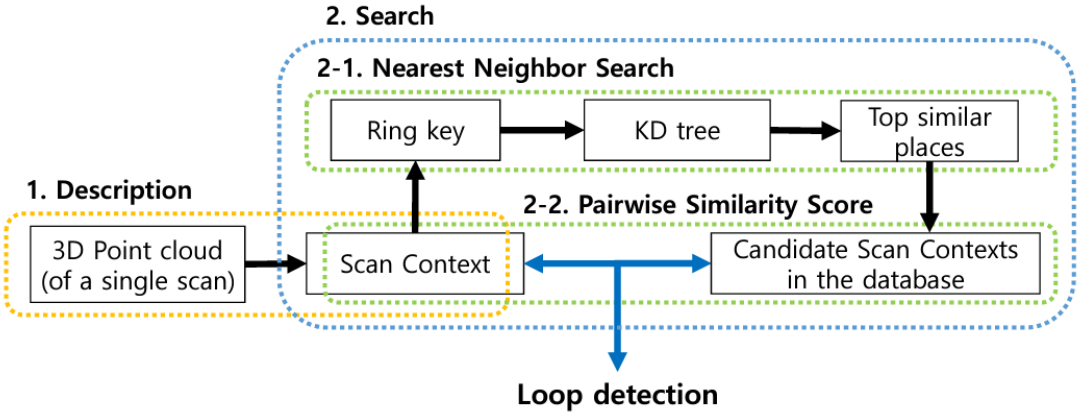
Learning-based

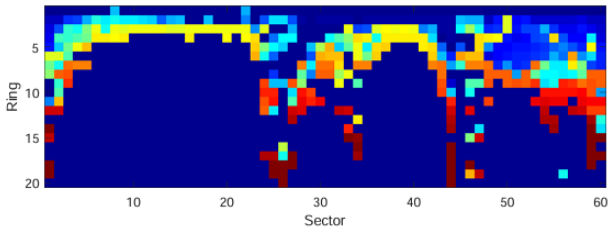
Schonberg et al., Comparative Evaluation of Hand-Crafted and Learned Local Features, CVPR'17.

LiDAR Loop Closure

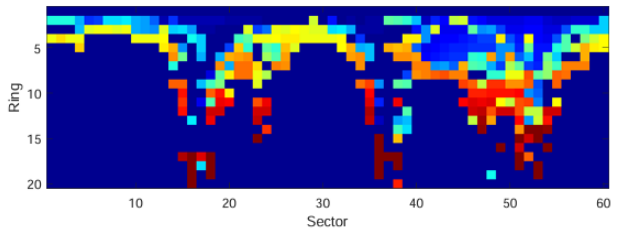
Scan Context

- Introduced by Giseop Kim & Ayoung Kim (https://gisbi-kim.github.io/publications/gkim-2018-iros.pdf)
- 2D feature descriptor for LiDAR point clouds
- Computes descriptor by binning points into concentric radial and azimuthal bins, then selecting the maximum z-height of points in each bin
- Can be used to detect loop closures

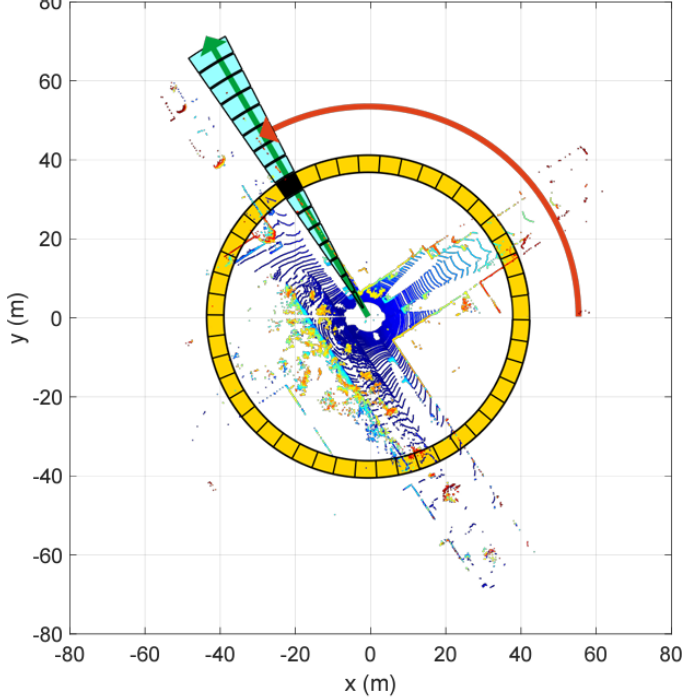




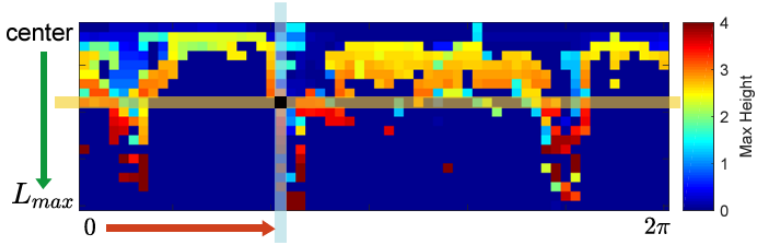
(a) The query scan context (3280th scan, KITTI00)



(b) The detected scan context (2345th scan, KITTI00)



(a) Bin division along azimuthal and radial directions



(b) Scan context

Fig. 1. Two-step scan context creation. Using the top view of a point cloud from a 3D scan (a), we partition ground areas into bins, which are split according to both azimuthal (from 0 to 2π within a LiDAR frame) and radial (from center to maximum sensing range) directions. We refer to the yellow area as a *ring*, the cyan area as a *sector*, and the black-filled area as a *bin*. Scan context is a matrix as in (b) that explicitly preserves the absolute geometrical structure of a point cloud. The ring and sector described in (a) are represented by the same-colored column and row, respectively, in (b). The representative value extracted from the points located in each bin is used as the corresponding pixel value of (b). In this paper, we use the maximum height of points in a bin.

References

- **Slides:** MIT's Visual Navigation course (https://vnav.mit.edu/)
- **Slides**: Photogrammetry II course (https://www.ipb.uni-bonn.de/photo12-2021/index.html)
- **Bag of Visual Words in 5 Minutes:** https://www.youtube.com/watch?v=a4cFONdc6nc
- **Jupyter notebook by Olga Vysotska:** https://github.com/ovysotska/in_simple_english/blob/master/bag_of_visual_words.ipynb
- **Sivic and Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos, 2003:** https://www.robots.ox.ac.uk/~vgg/publications/2003/Sivic03/sivic03.pdf
- **TF-IDF information:** https://en.wikipedia.org/wiki/Tf%E2%80%93idf