Photogrammetry & Robotics Lab

Bag of Visual Words for Finding Similar Images

Cyrill Stachniss

Slides have been created by Cyrill Stachniss.
Most images by Olga Vysotska and Fei-Fei Li.

5 Minute Preparation for Today

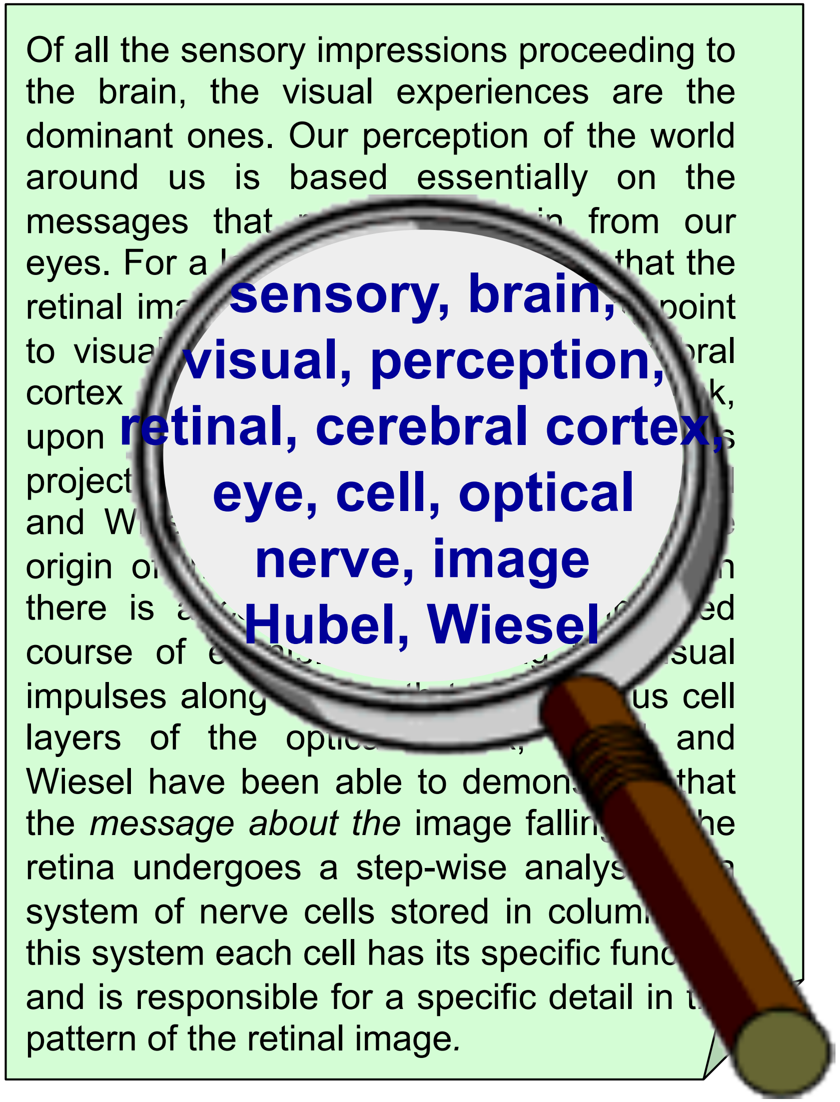
https://www.youtube.com/watch?v=a4cFONdc6nc

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

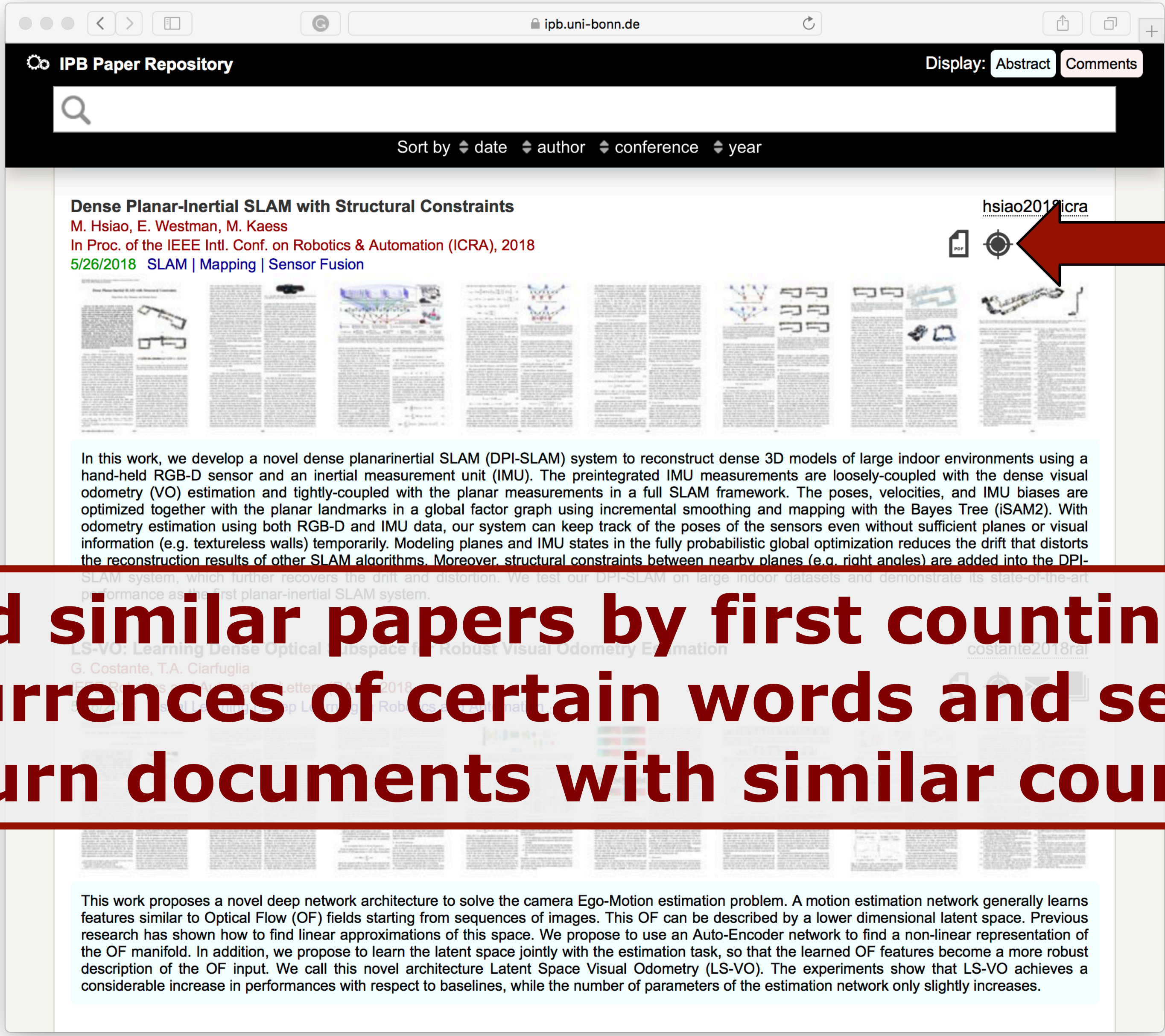
?

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

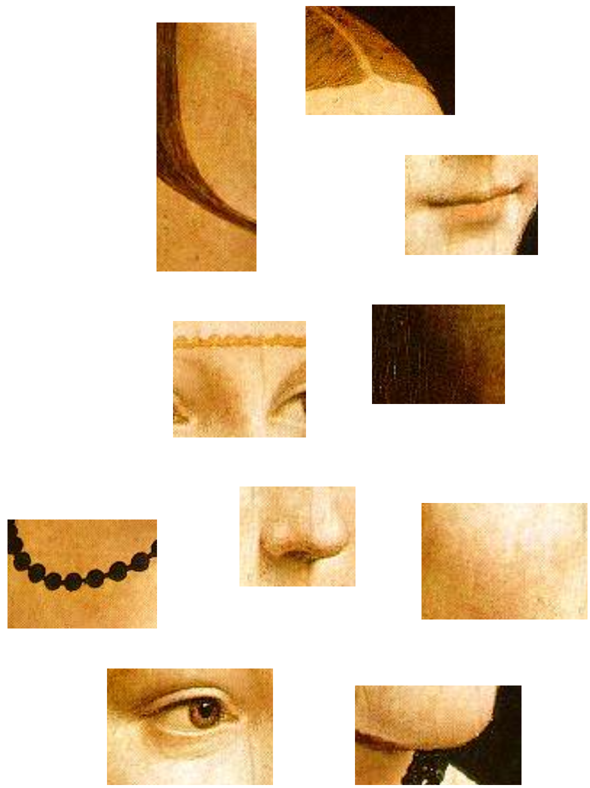
bag of “visual words”

[Image courtesy: Fei-Fei Li]

Bag of Visual Words

- Visual words = independent features

face



features

[Image courtesy: Fei-Fei Li]

Bag of Visual Words

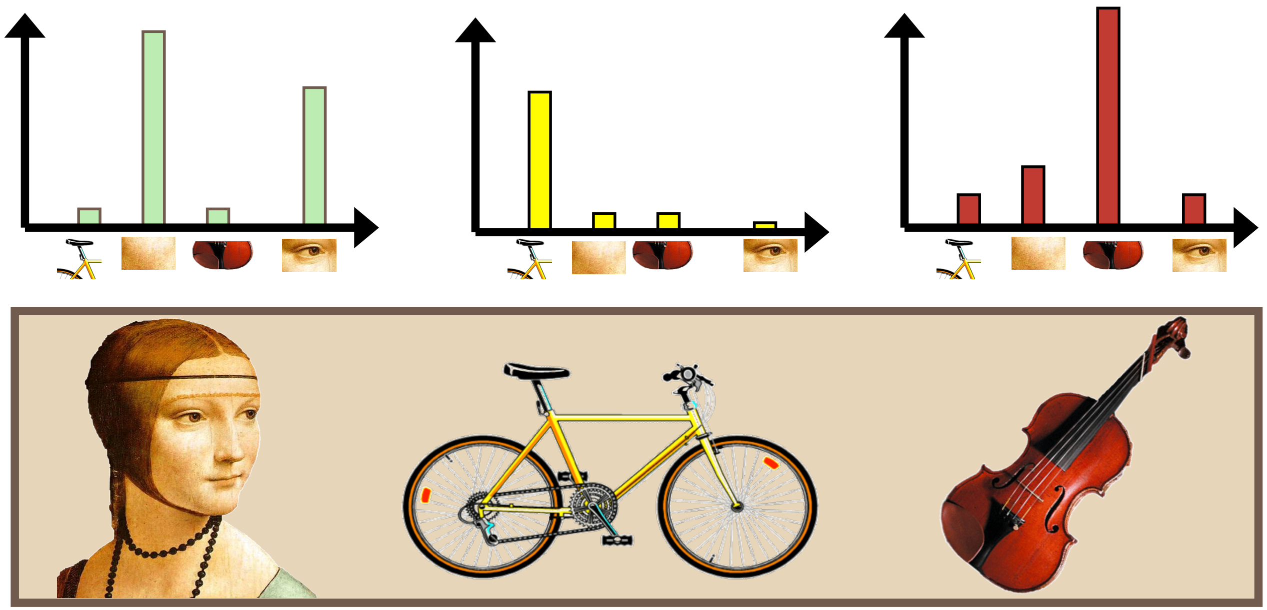
- Visual words = independent features
- Construct a dictionary of representative words
- Use only words from the dictionary

dictionary (“codebook“)

[Image courtesy: Fei-Fei Li]

Bag of Visual Words

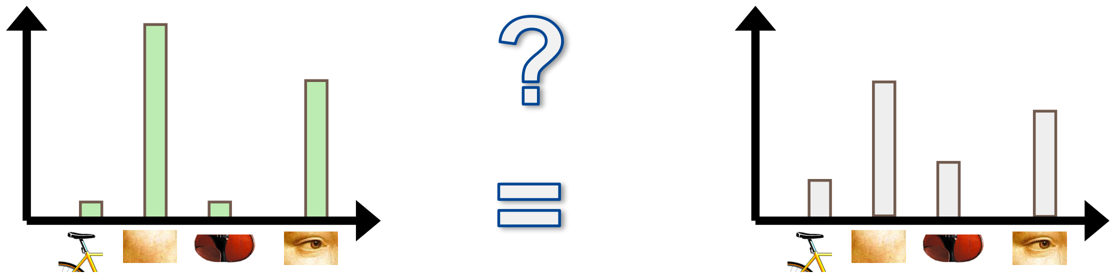
- Visual words = independent features
- Words from the dictionary
- Represent the images based on a histogram of word occurrences



[Image courtesy: Fei-Fei Li]

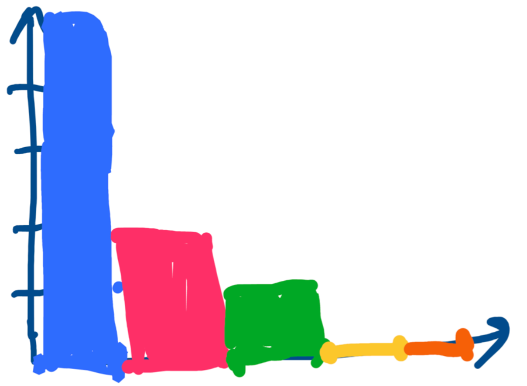
Bag of Visual Words

- 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

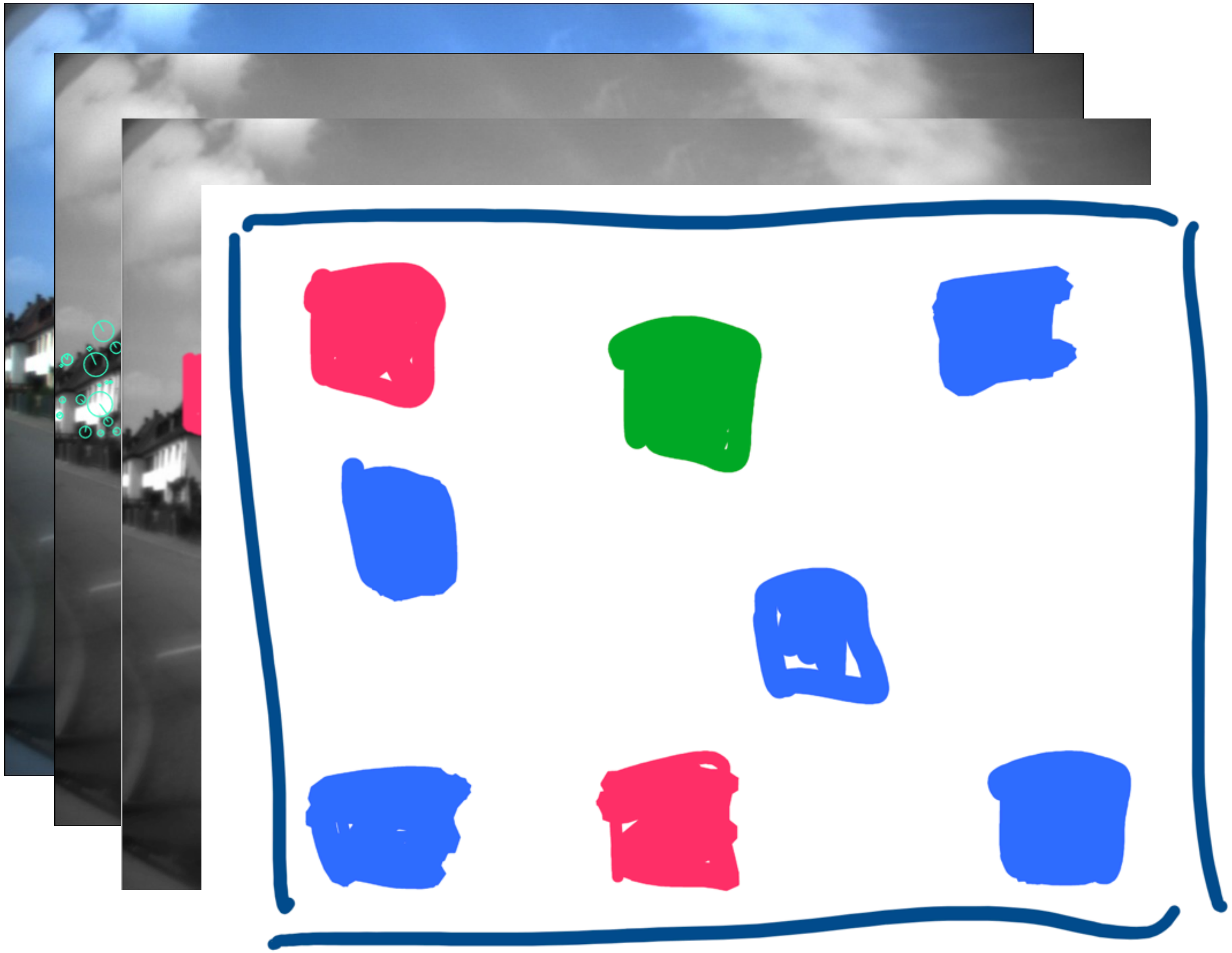
Overview: Extract Features

[Image courtesy: Olga Vysotska]

Overview: Visual Words

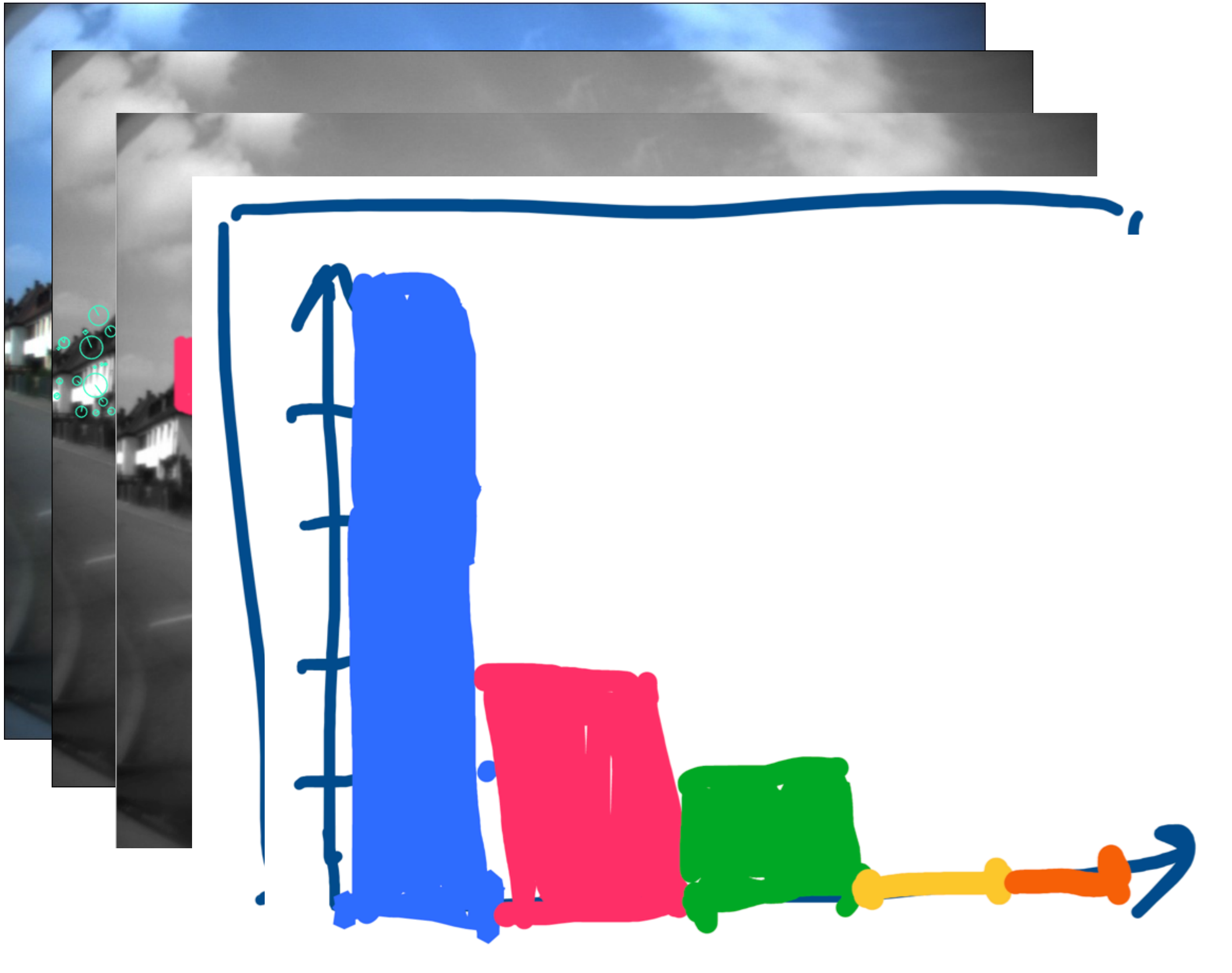
[Image courtesy: Olga Vysotska]

Overview: No Pixel Values



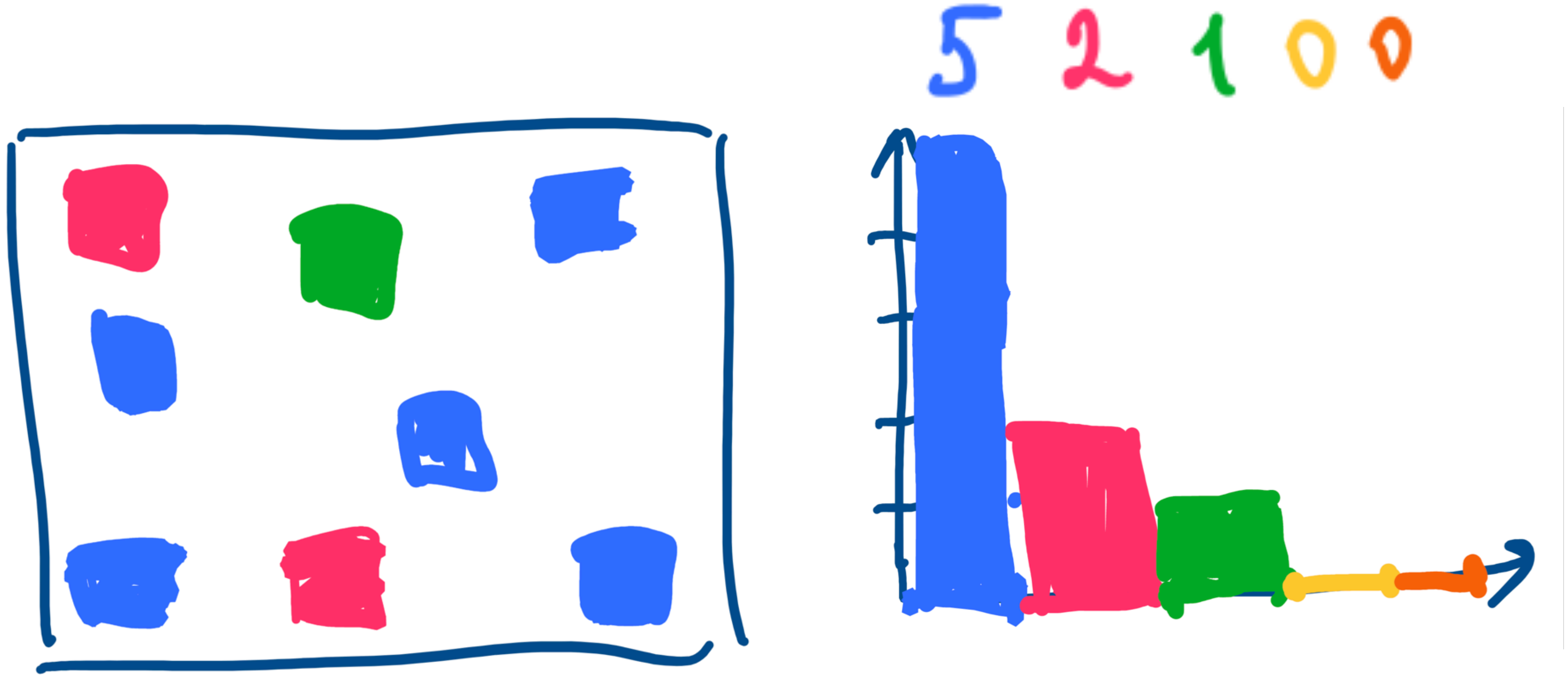
[Image courtesy: Olga Vysotska]

Overview: Word Occurrences



[Image courtesy: Olga Vysotska]

Images to Histograms

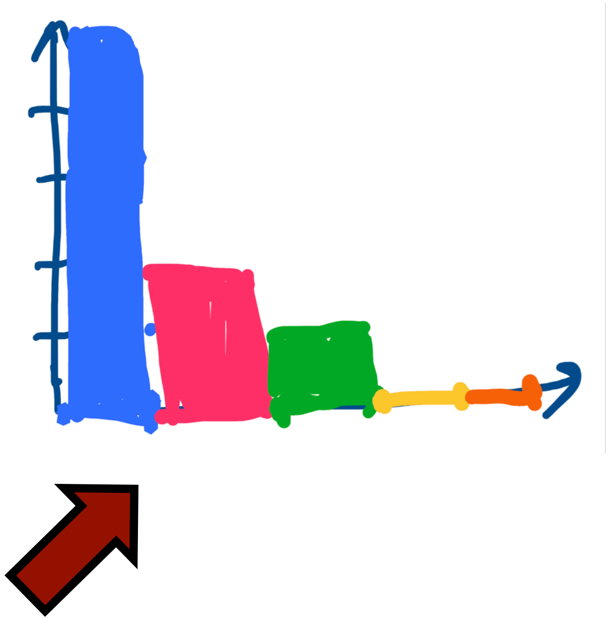


[Image courtesy: Olga Vysotska]

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



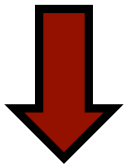
[Image courtesy: Olga Vysotska]

Dictionary

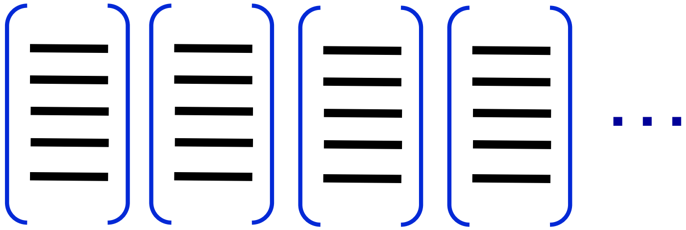
- A dictionary defines the list of words that are considered
- The dictionary defines the x-axes of all the word occurrence histograms
- The dictionary must remain fixed

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

Extract Feature Descriptors from a Training Dataset

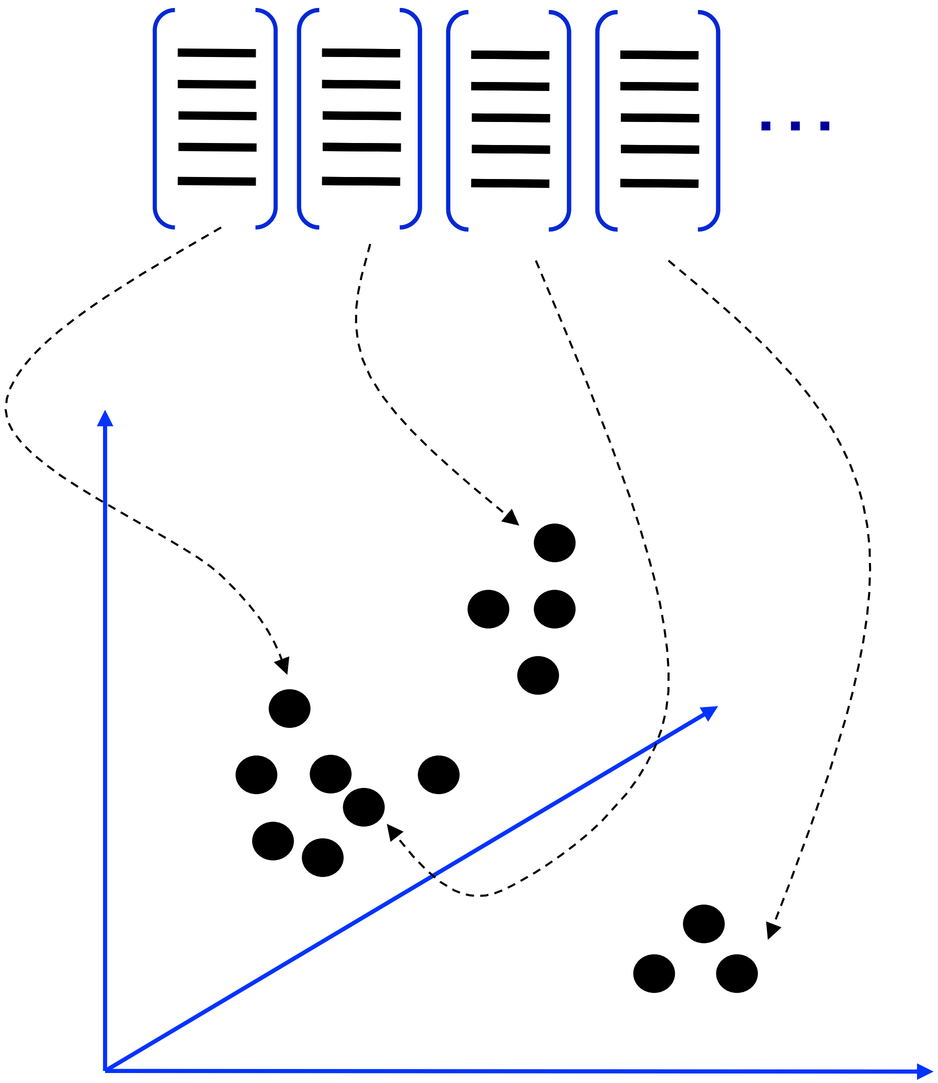


Visual feature descriptor vectors (e.g., SIFT)



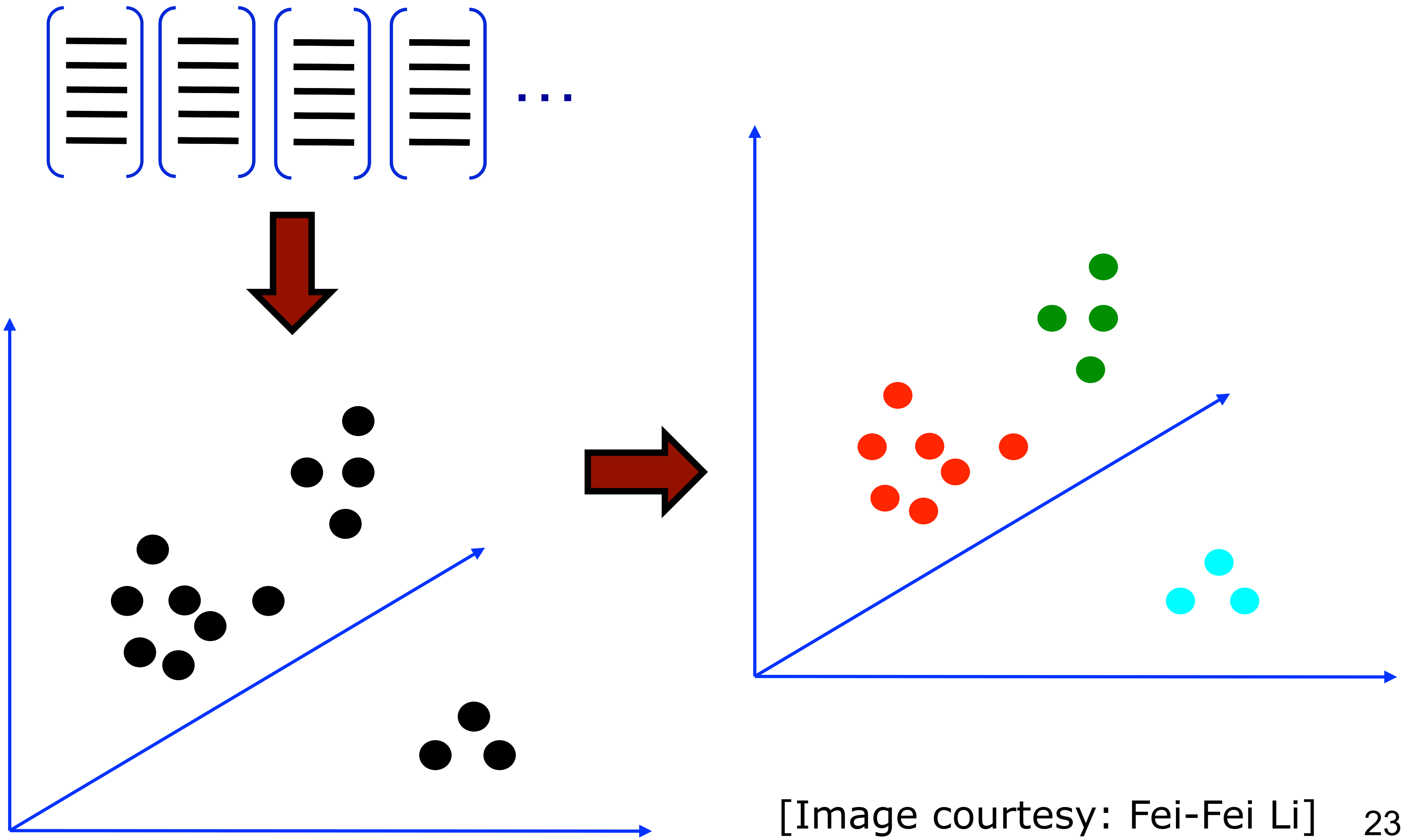
[Partial image courtesy: Fei-Fei Li]

Feature Descriptors are Points in a High-Dimensional Space



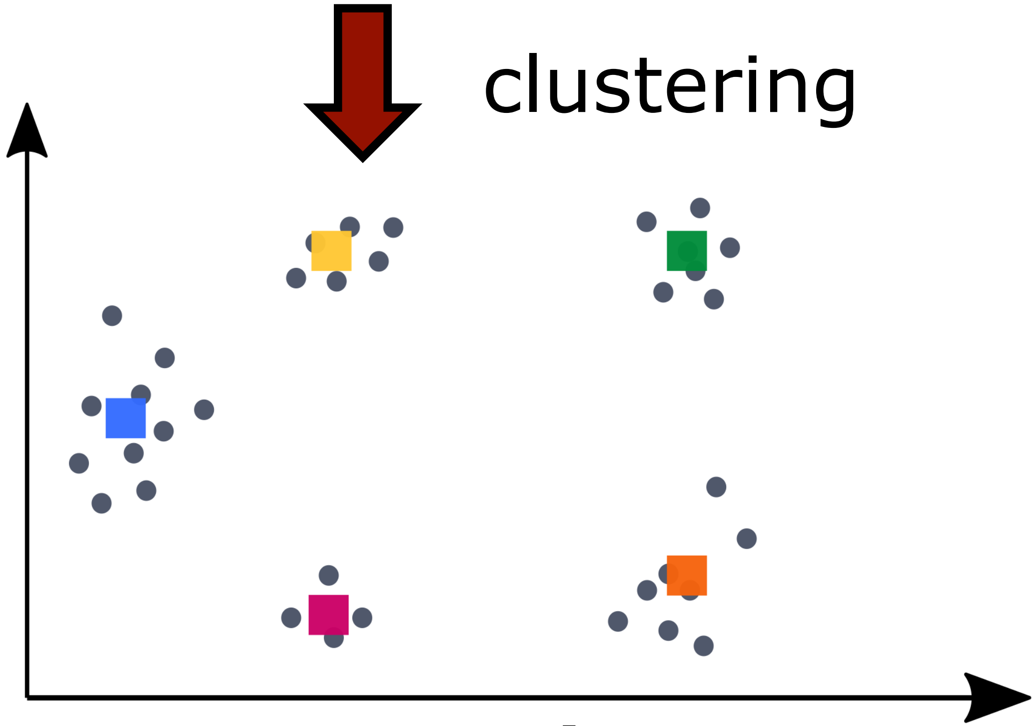
[Image courtesy: Fei-Fei Li]

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

Initialize $\boldsymbol{m}_i, i = 1, \ldots, k$, for example, to k random $\boldsymbol{x}^t$

Repeat

For all $\boldsymbol{x}^t \in \mathcal{X}$

$$b_i^t \leftarrow \begin{cases} 1 & \text{if } \|\boldsymbol{x}^t - \boldsymbol{m}_i\| = \min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\| \\ 0 & \text{otherwise} \end{cases}$$

For all $\boldsymbol{m}_i, i = 1, \ldots, k$

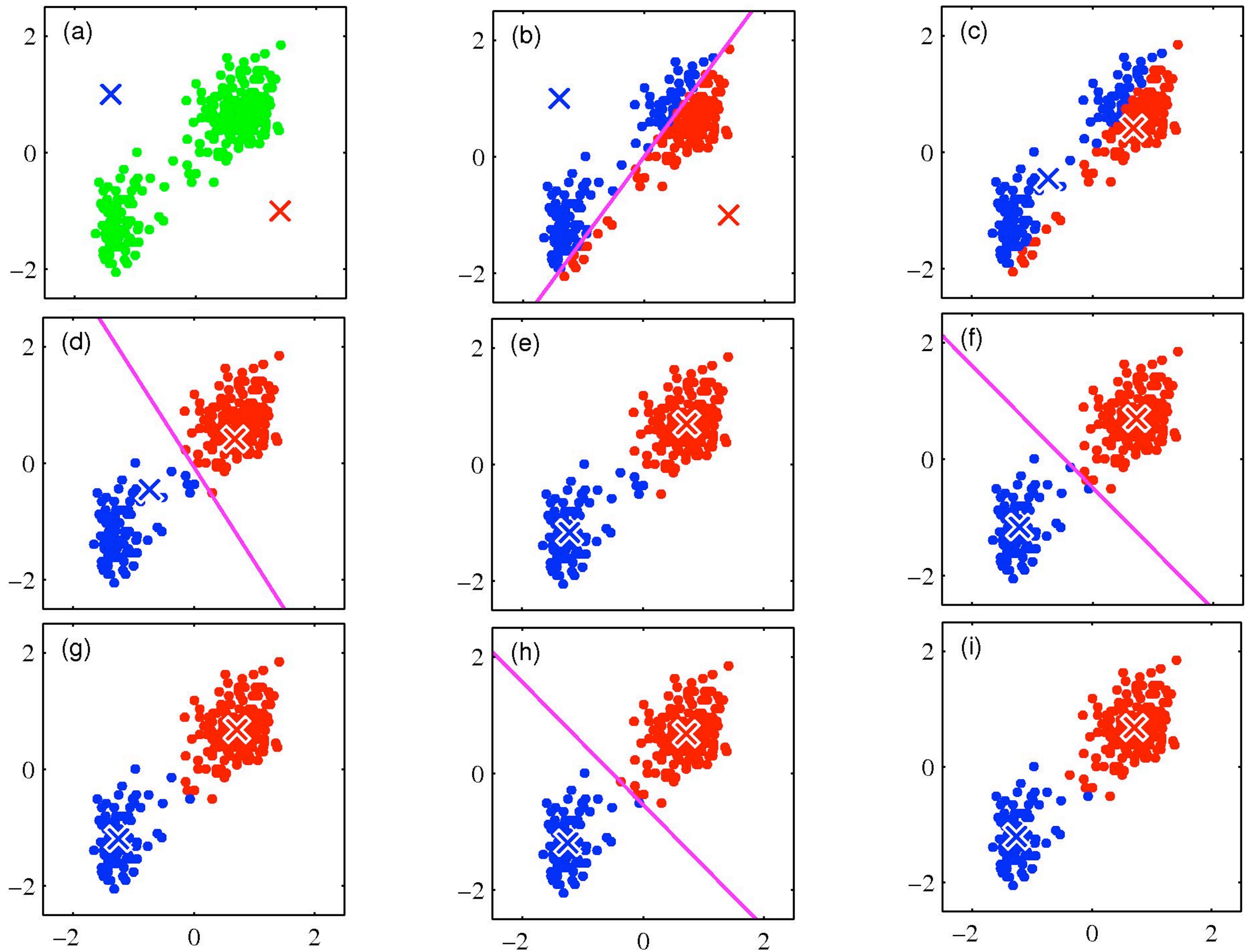
$$\boldsymbol{m}_i \leftarrow \sum_t b_i^t \boldsymbol{x}^t / \sum_t b_i^t$$

Until $\boldsymbol{m}_i$ converge

Re-compute the cluster means using the current cluster memberships

Assign each data point to the closest cluster

K-Means Example



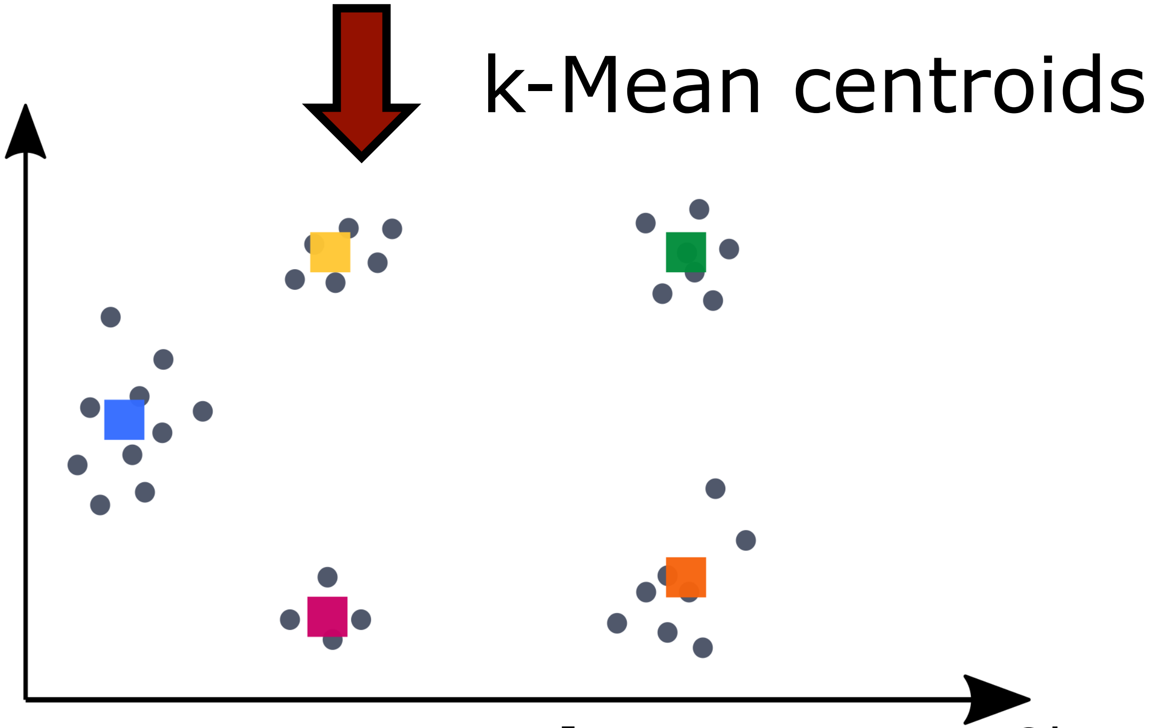
[Image courtesy: Bishop]

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

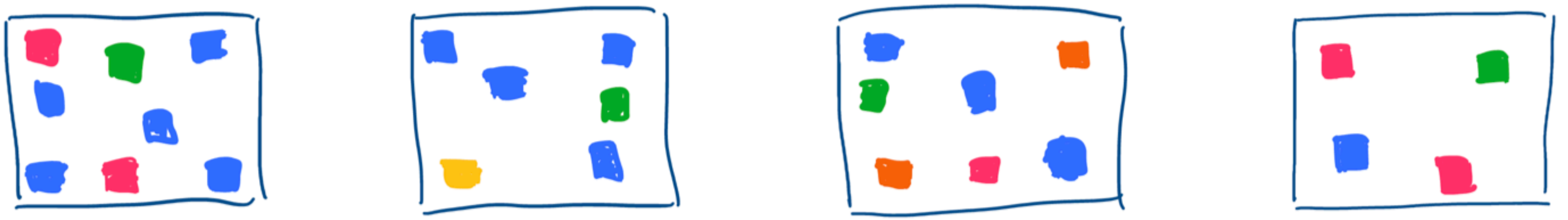
We use k-means to compute the dictionary of visual words

K-Means for Building the Dictionary from Training Data



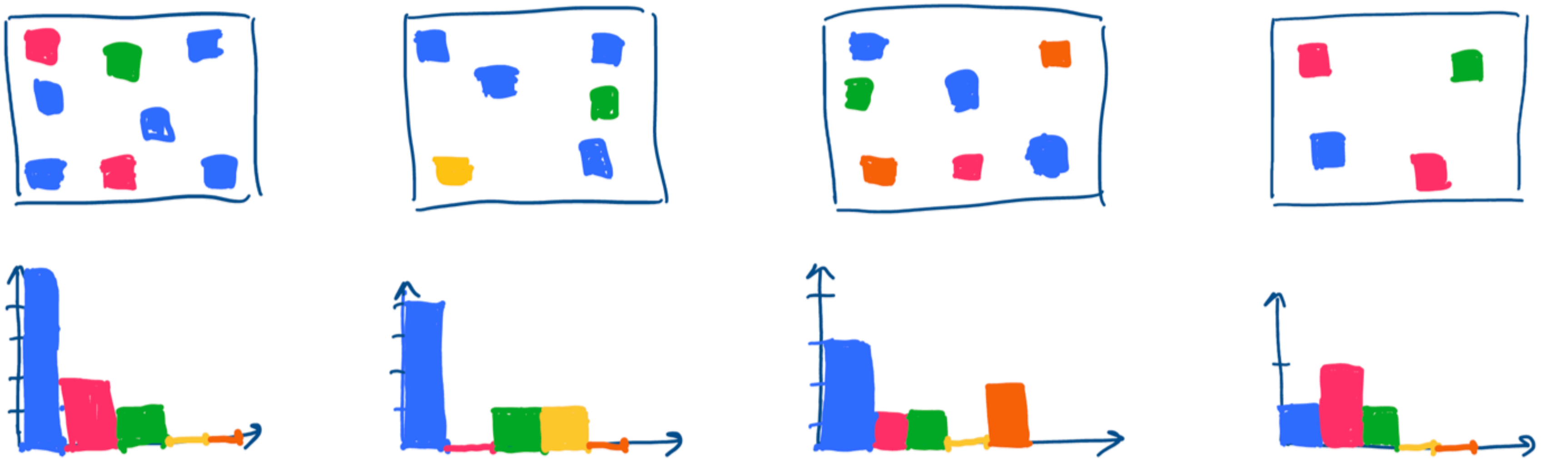
[Image courtesy: Olga Vysotska]

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]

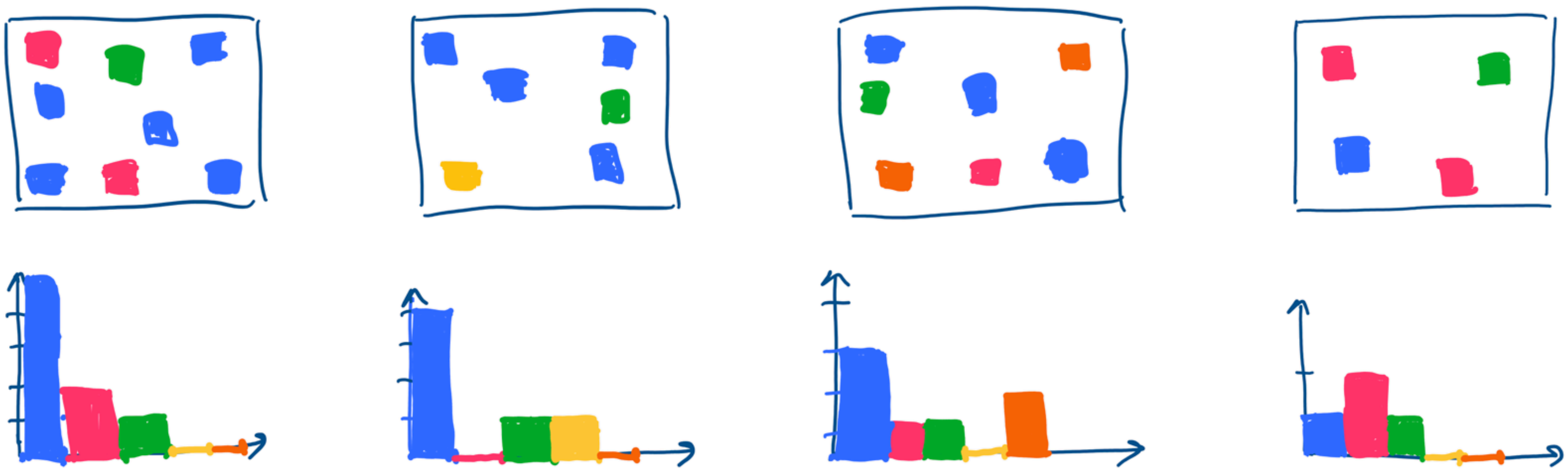
Are All Words Expressive for Comparing Histograms?

- 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

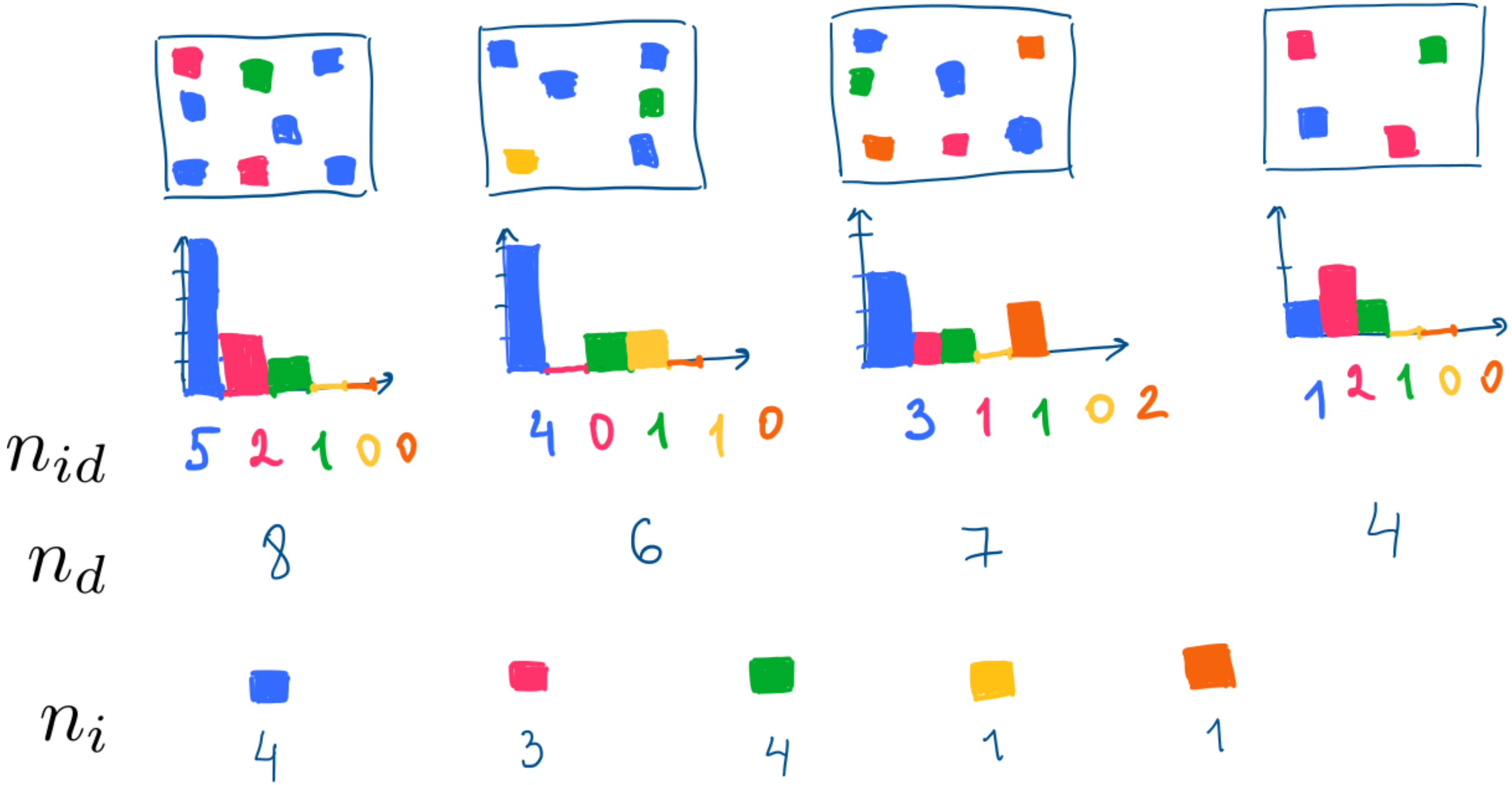
$$t_{id} = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

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

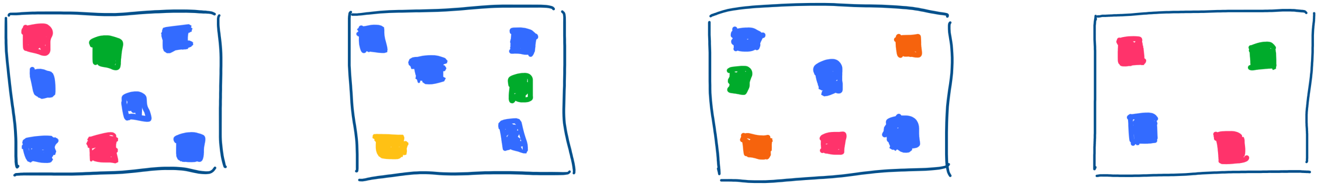
Computing the TF-IDF (1)



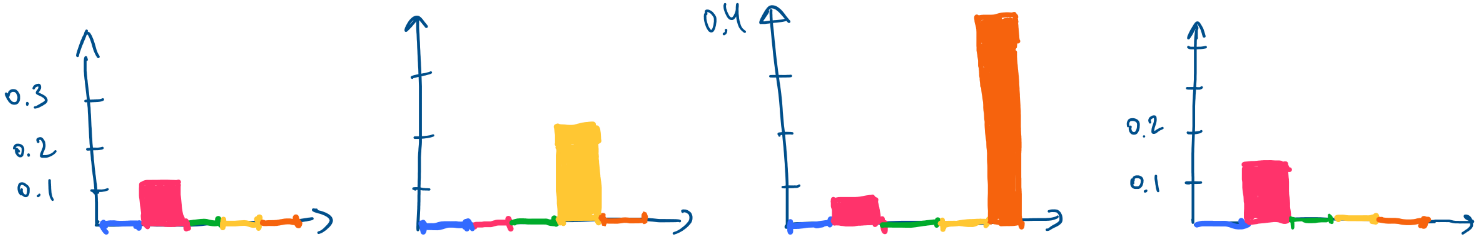
$$t_{id} = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

[Image courtesy: Olga Vysotska]

Computing the TF-IDF (2)

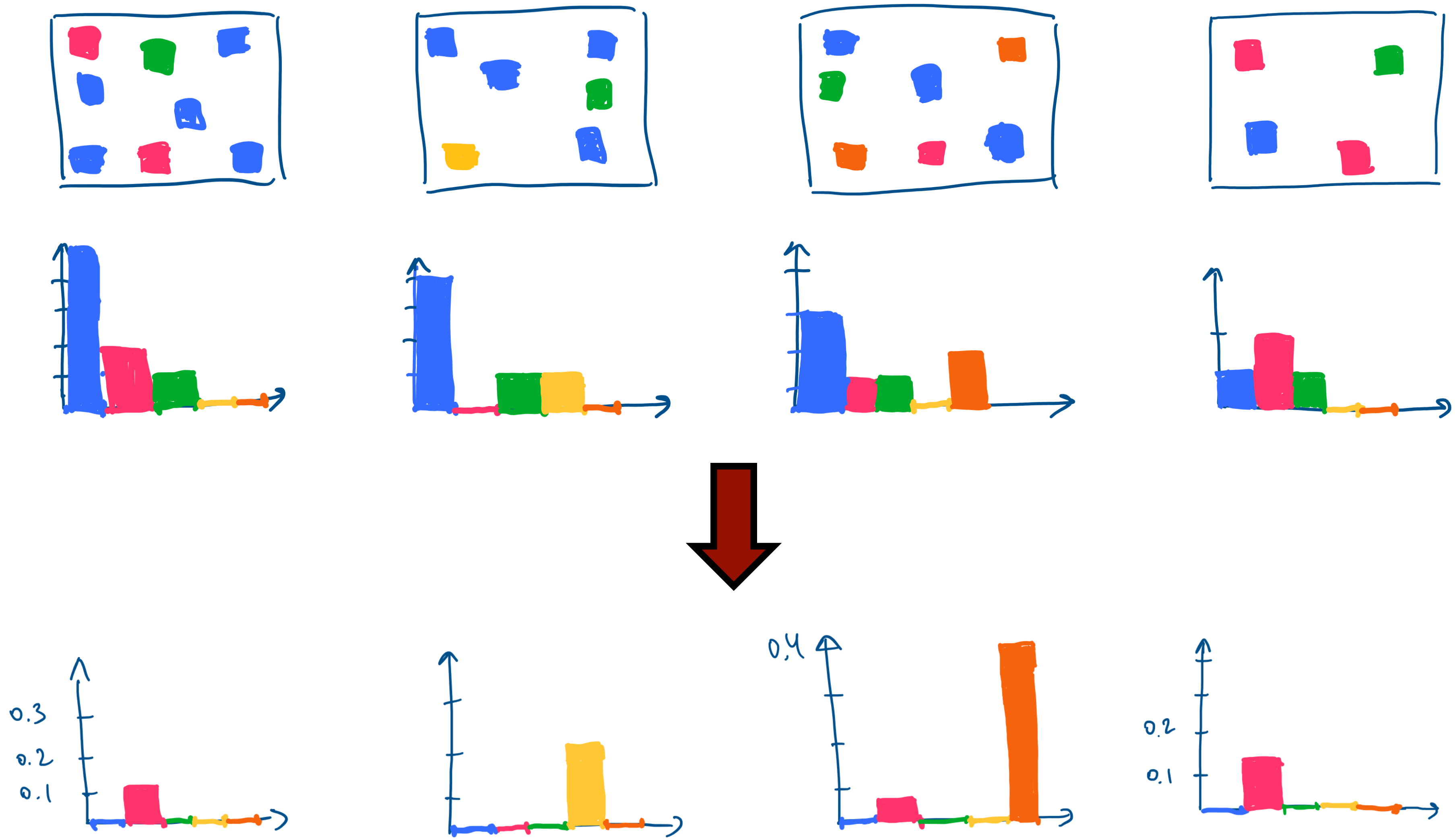


t_1	$\frac{5}{8}\log\frac{4}{4}$	0	$\frac{4}{6}\log\frac{4}{4}$	0	$\frac{3}{7}\log\frac{4}{4}$	0	$\frac{1}{4}\log\frac{4}{4}$	0
t_2	$\frac{2}{8}\log\frac{4}{3}$	0.07	$\frac{0}{6}\log\frac{4}{3}$	0	$\frac{1}{7}\log\frac{4}{3}$	0.04	$\frac{2}{4}\log\frac{4}{3}$	0.14
t_3	$\frac{1}{8}\log\frac{4}{4}$	0	$\frac{1}{6}\log\frac{4}{4}$	0	$\frac{1}{7}\log\frac{4}{4}$	0	$\frac{1}{4}\log\frac{4}{4}$	0
t_4	$\frac{0}{8}\log\frac{4}{1}$	0	$\frac{1}{6}\log\frac{4}{1}$	0.23	$\frac{0}{7}\log\frac{4}{1}$	0	$\frac{0}{4}\log\frac{4}{1}$	0
t_5	$\frac{0}{8}\log\frac{4}{1}$	0	$\frac{0}{6}\log\frac{4}{1}$	0	$\frac{2}{7}\log\frac{4}{1}$	0.4	$\frac{0}{4}\log\frac{4}{1}$	0



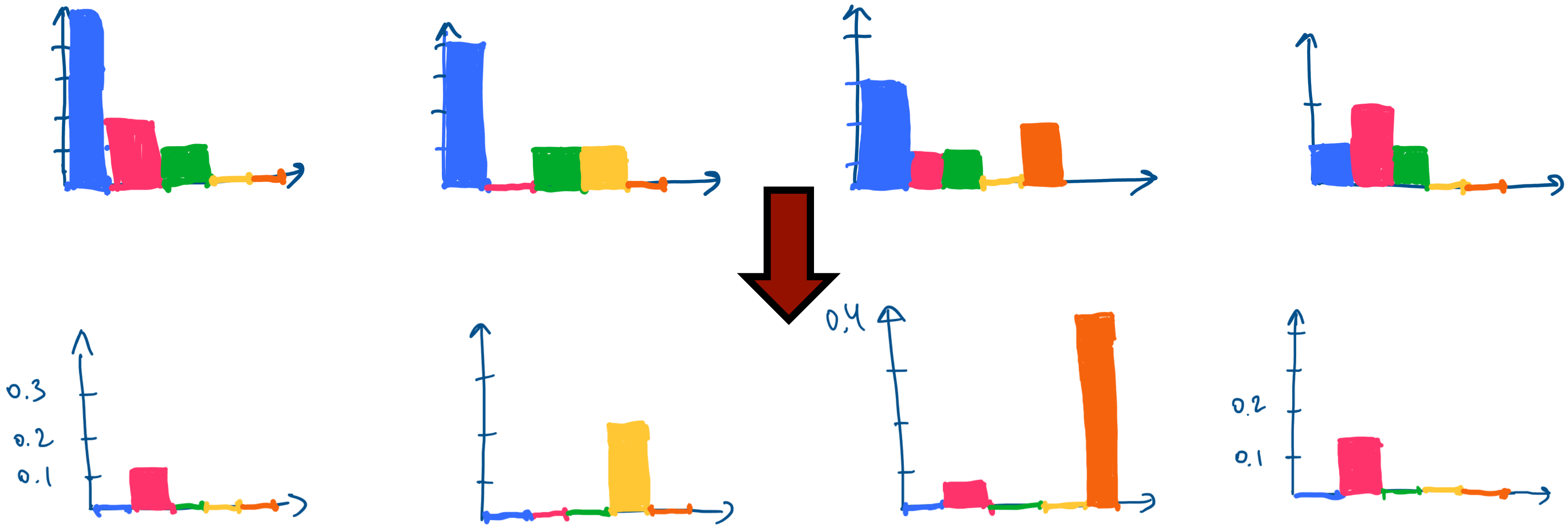
[Image courtesy: Olga Vysotska]

Reweighted Histograms



[Image courtesy: Olga Vysotska]

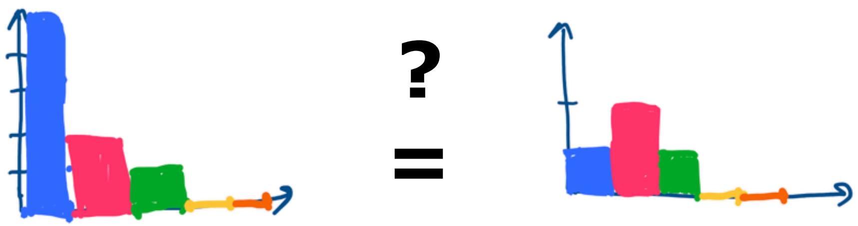
Reweighted Histograms



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

[Image courtesy: Olga Vysotska]

Comparing Two Histograms

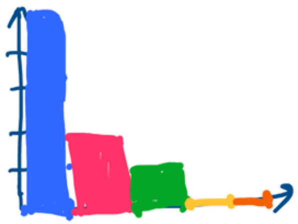


Options

- Euclidean distance of two points
- Angle between two vectors
- ~~Kullback Leibler divergence (KLD)~~

[Image courtesy: Olga Vysotska]

Comparing Two Histograms



$\stackrel{?}{=}$

Options

- Euclidean distance of two vectors
- **Angle between two vectors**
- ~~Kullback Leibler divergence (KLD)~~

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:

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

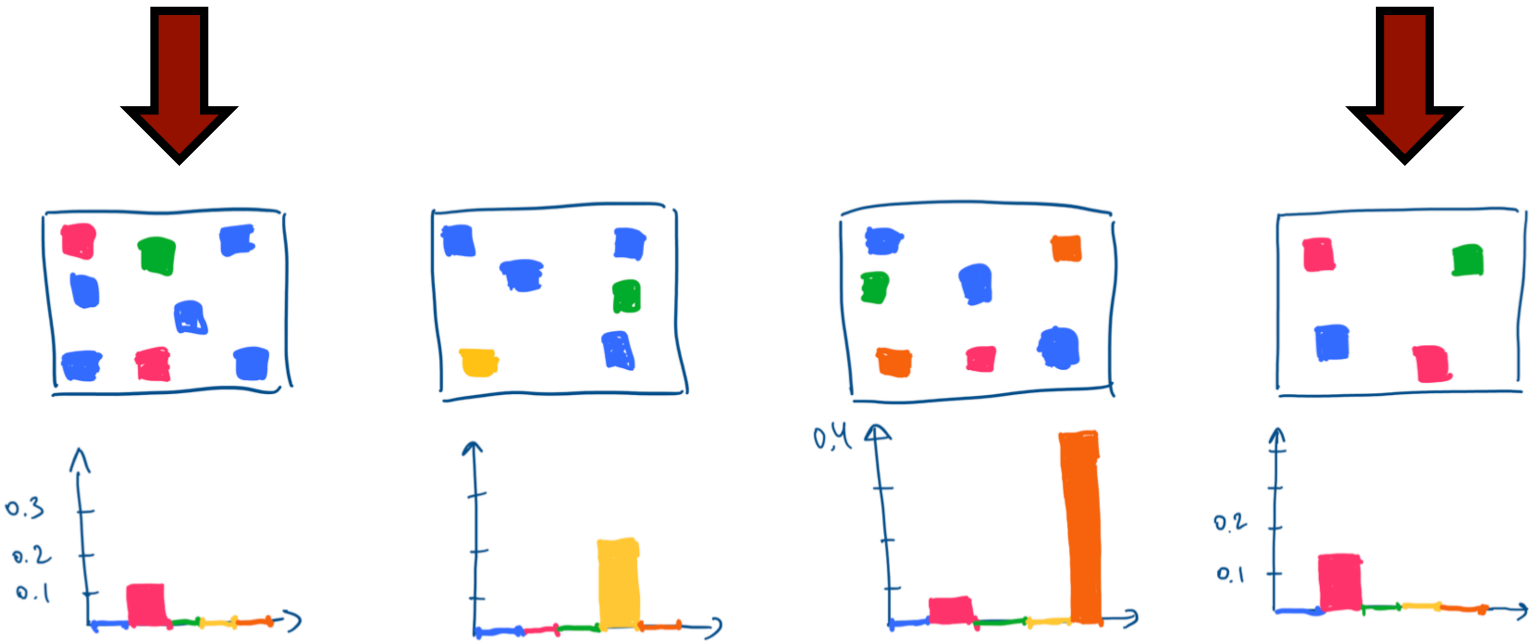
- We use the cosine distance

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

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

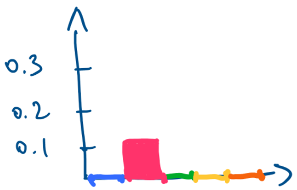
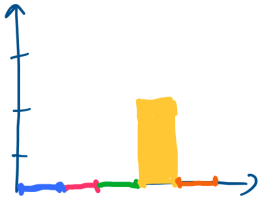
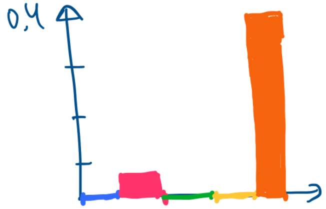
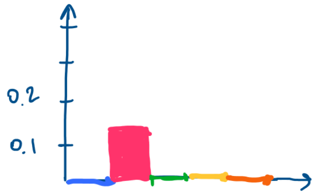
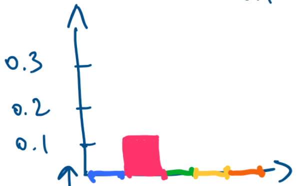
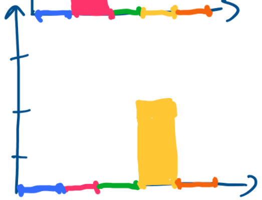
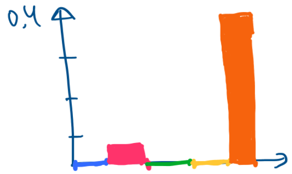
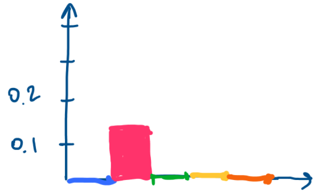
Example Comparing Histograms

- 4 images
- Image 0 and image 3 are similar



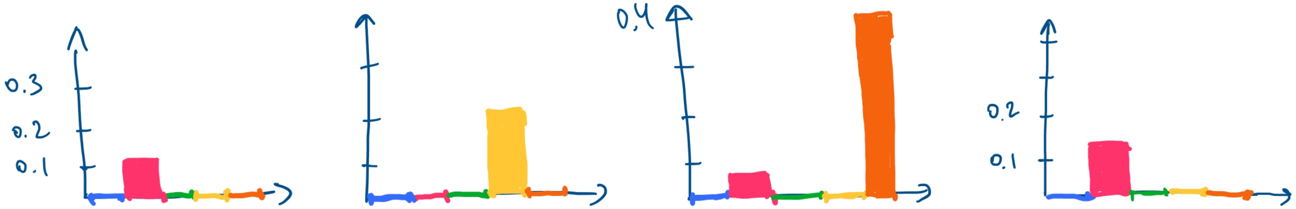
[Image courtesy: Olga Vysotska]

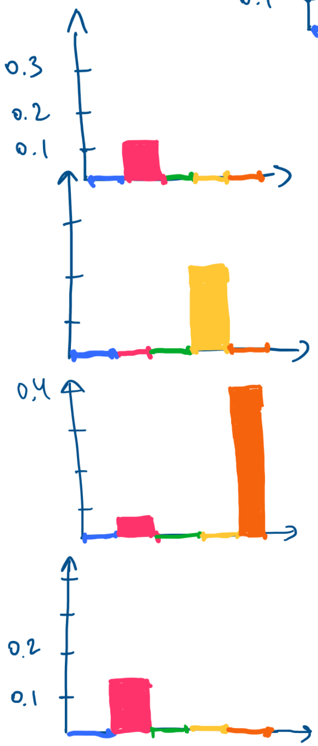
Example Comparing Histograms

				
	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



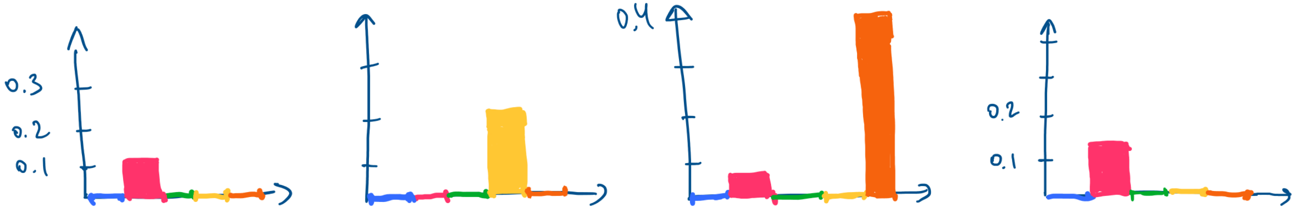


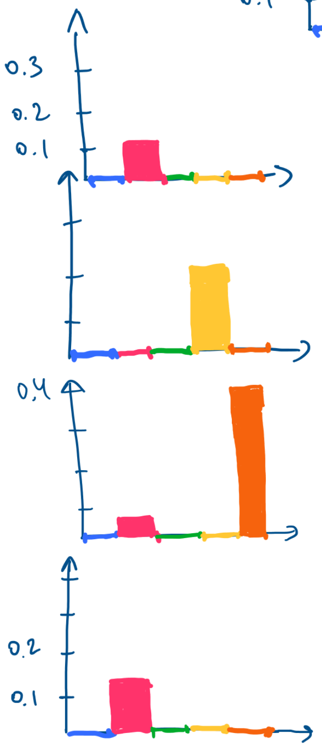
0	1	0.98	0
1	0	1	1
0.98	1	0	0.98
0	1	0.98	0

Images have a zero distance to themselves

[Image courtesy: Olga Vysotska]

Example Comparing Histograms



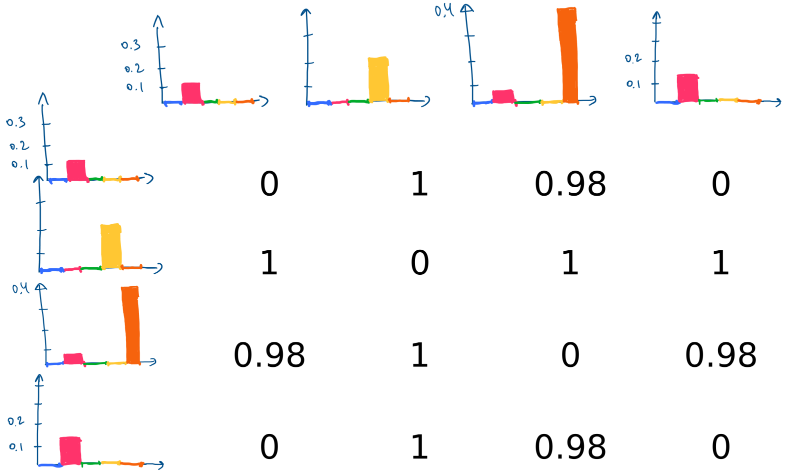


0	1	0.98	0
1	0	1	1
0.98	1	0	0.98
0	1	0.98	0

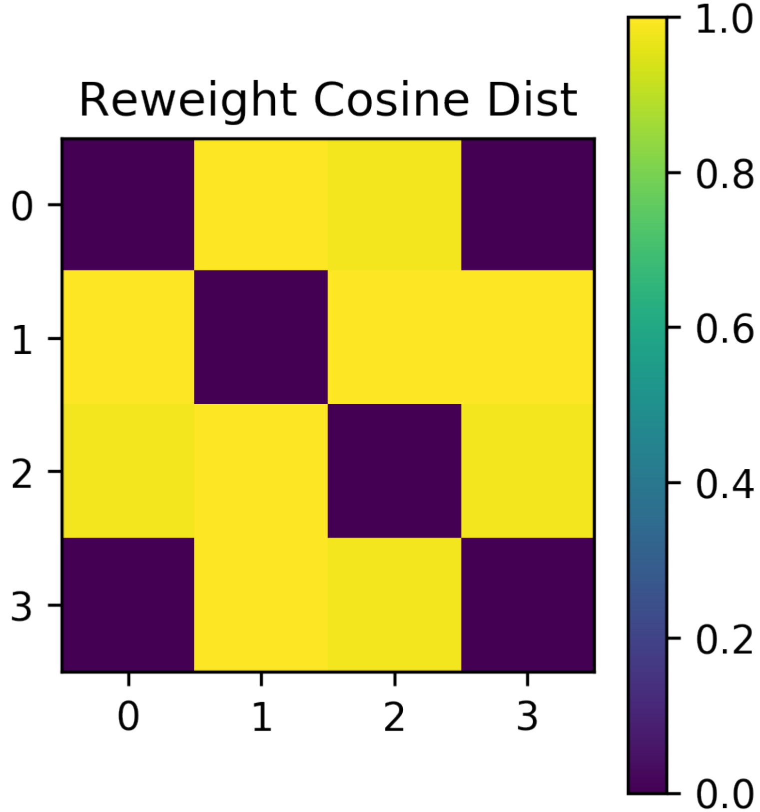
Images 0 and 3 are highly similar

[Image courtesy: Olga Vysotska]

Cost Matrix

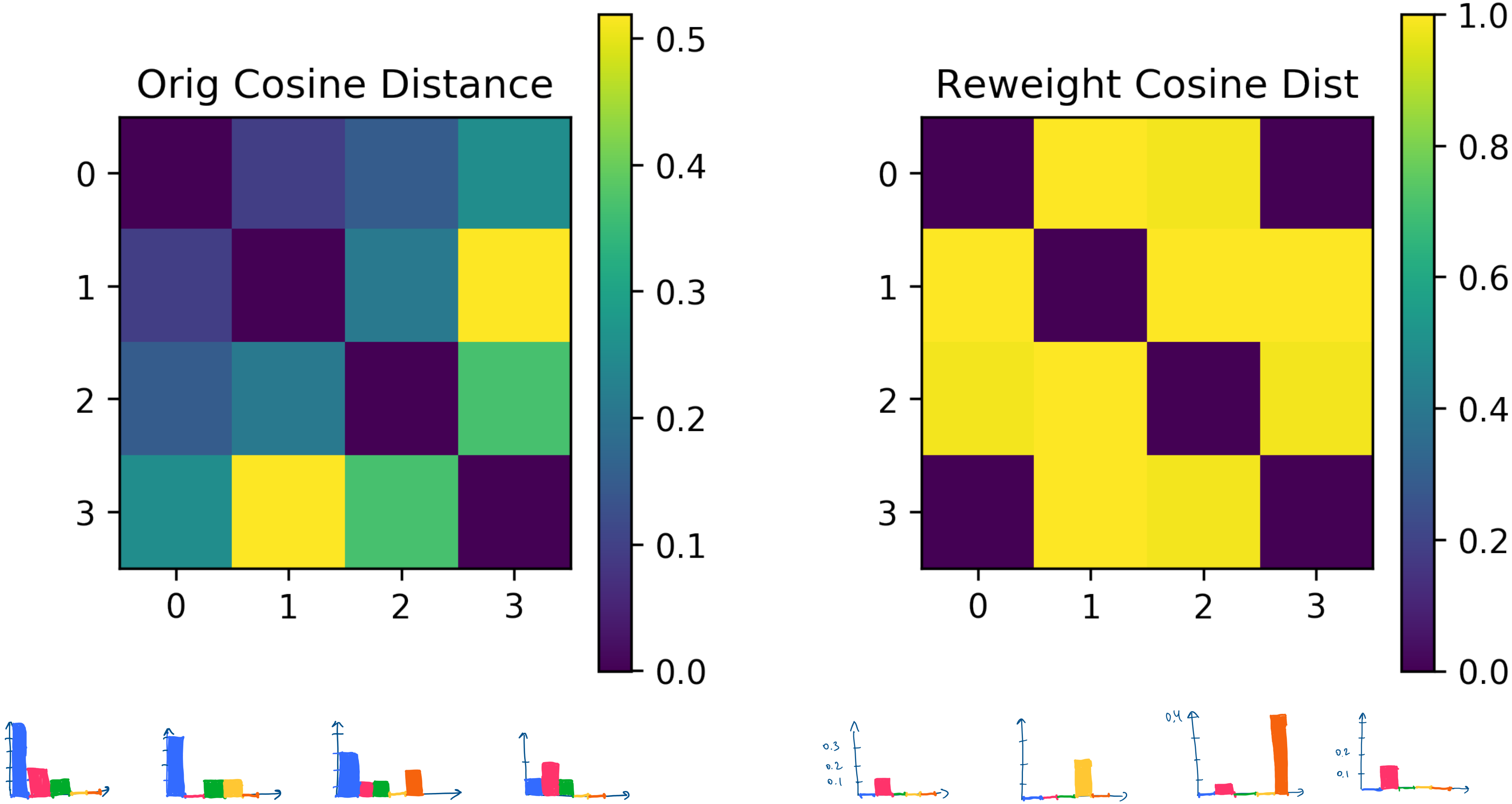


0	1	0.98	0
1	0	1	1
0.98	1	0	0.98
0	1	0.98	0



[Image courtesy: Olga Vysotska]

IF-IDF Actually Helps



original histograms

TF-IDF histograms

[Image courtesy: Olga Vysotska]

Euclidean vs. Cosine Distance

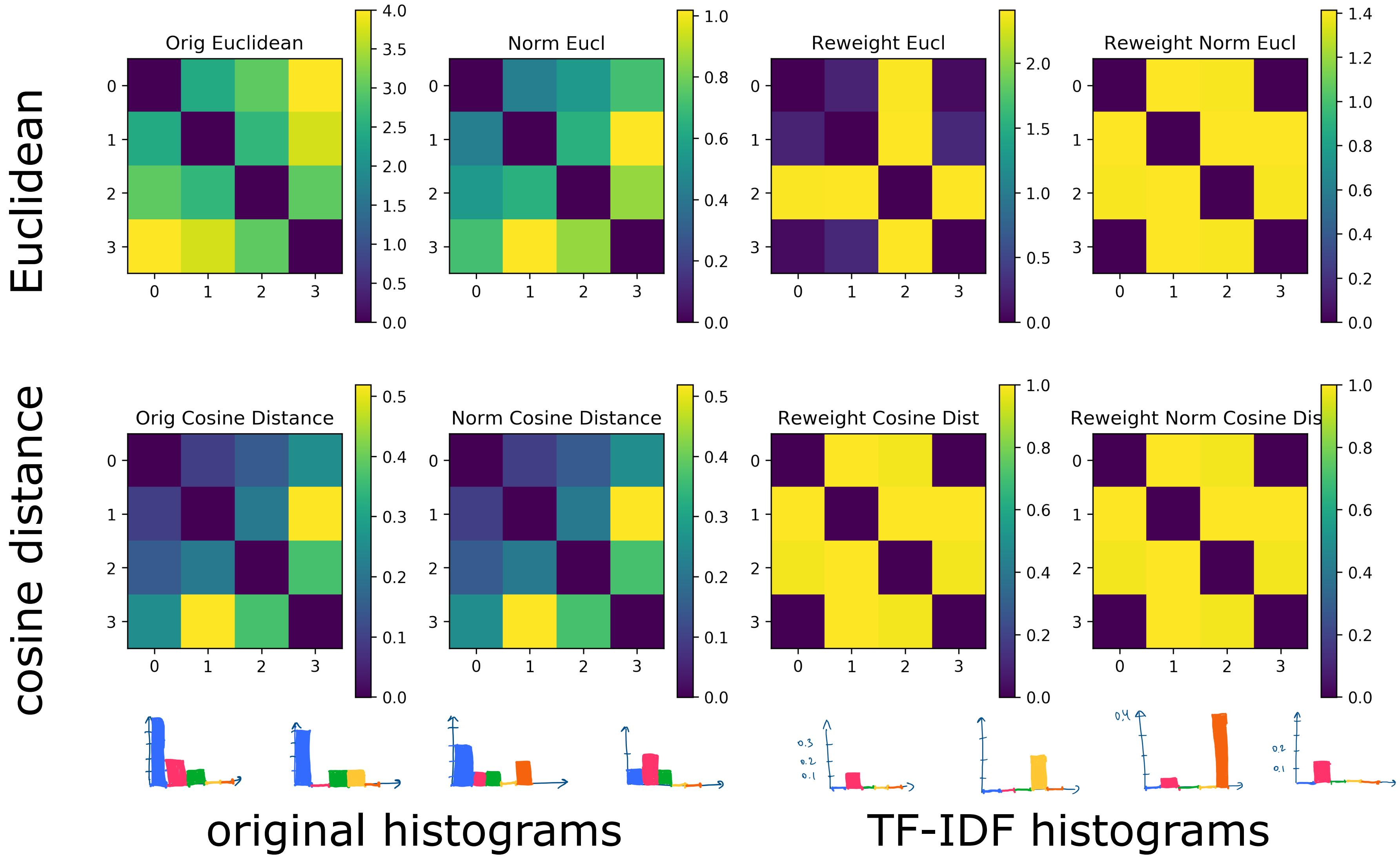
- Cosine distance ignores the length of the vectors
- **For vectors of length 1**, the squared Euclidean and the cosine distance only differ by a factor of 2:

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

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

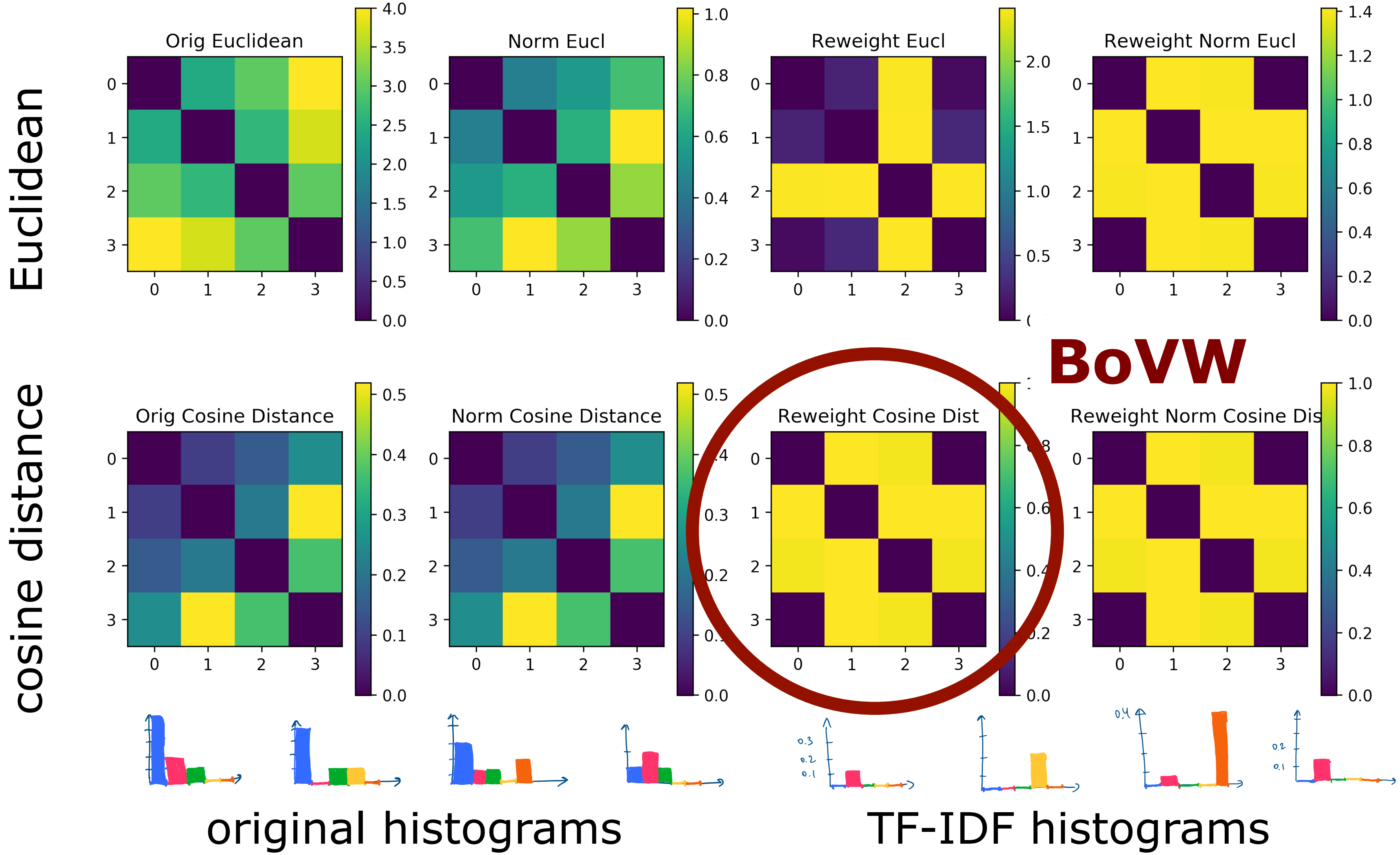
$$\begin{aligned} ||\mathbf{x}-\mathbf{y}||^2 &= 2 - 2\mathbf{x}^\top\mathbf{y} = 2 - 2\cos\theta \\ &= 2\,\mathrm{d}_{\cos}(\mathbf{x},\mathbf{y}) \end{aligned}$$

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

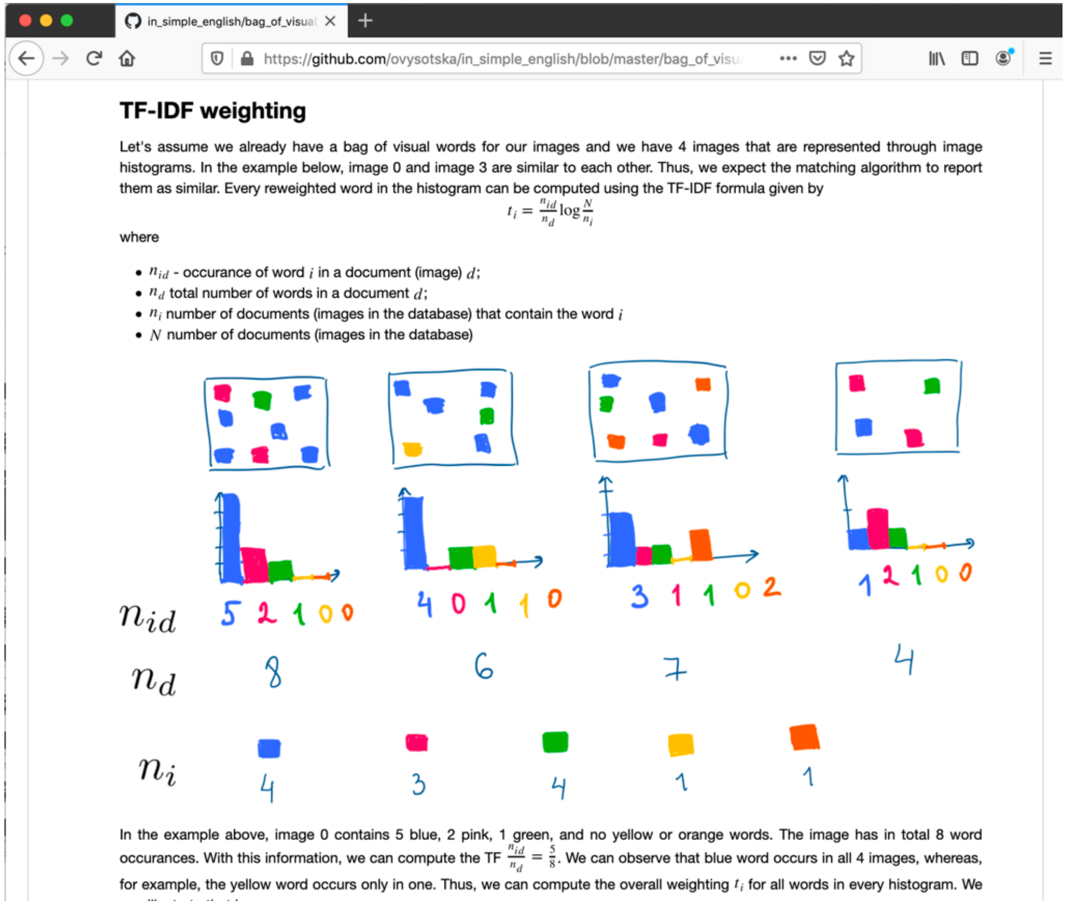
- Extract features from query image
- Assign features to visual words
- Build TF-IDF histogram for query image
- Return N most similar histograms from database under cosine distance

Further Material

- Bag of Visual Words in 5 Minutes:
  https://www.youtube.com/watch?v=a4cFONdc6nc

Further Material

- Jupyter notebook by Olga Vysotska: https://github.com/ovysotska/in_simple_english/blob/master/bag_of_visual_words.ipynb



Further Material

- 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:
  http://www.robots.ox.ac.uk/~vgg/publications/papers/sivic03.pdf
- TF-IDF information:
  https://en.wikipedia.org/wiki/Tf%E2%80%93idf

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