

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>

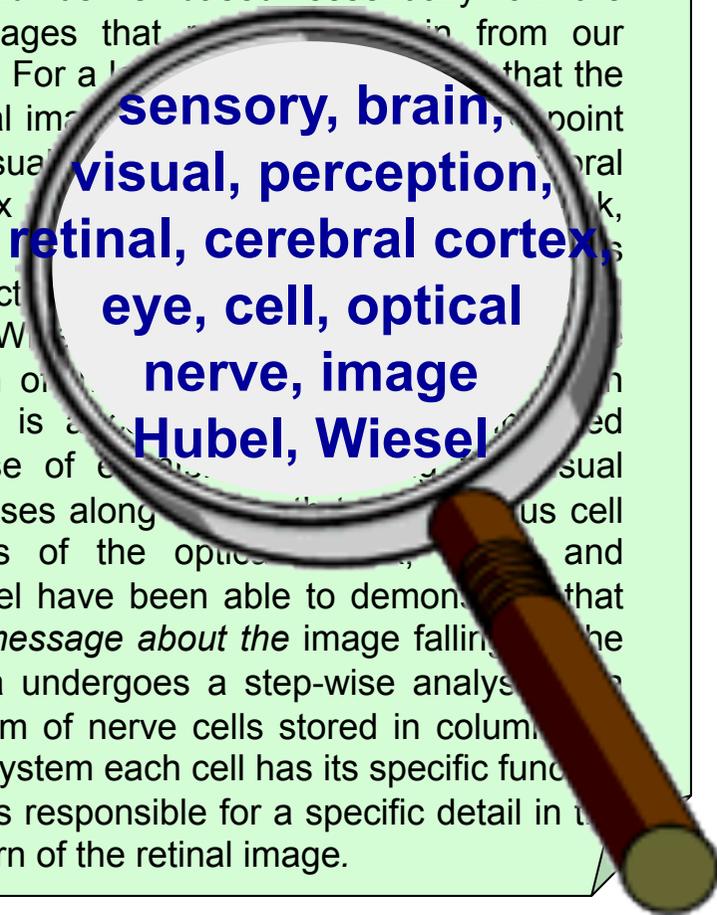
What is Bag of Visual Word for?

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



Analogy to Text Documents

Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant ones. Our perception of the world around us is based essentially on the messages that come from our eyes. For a long time it was thought that the retinal image is a point-to-point projection to visual cortex. However, Hubel and Wiesel have been able to demonstrate that the *message about the image* falling on the retina undergoes a step-wise analysis by a system of nerve cells stored in columns. In this system each cell has its specific function and is responsible for a specific detail in the pattern of the retinal image.



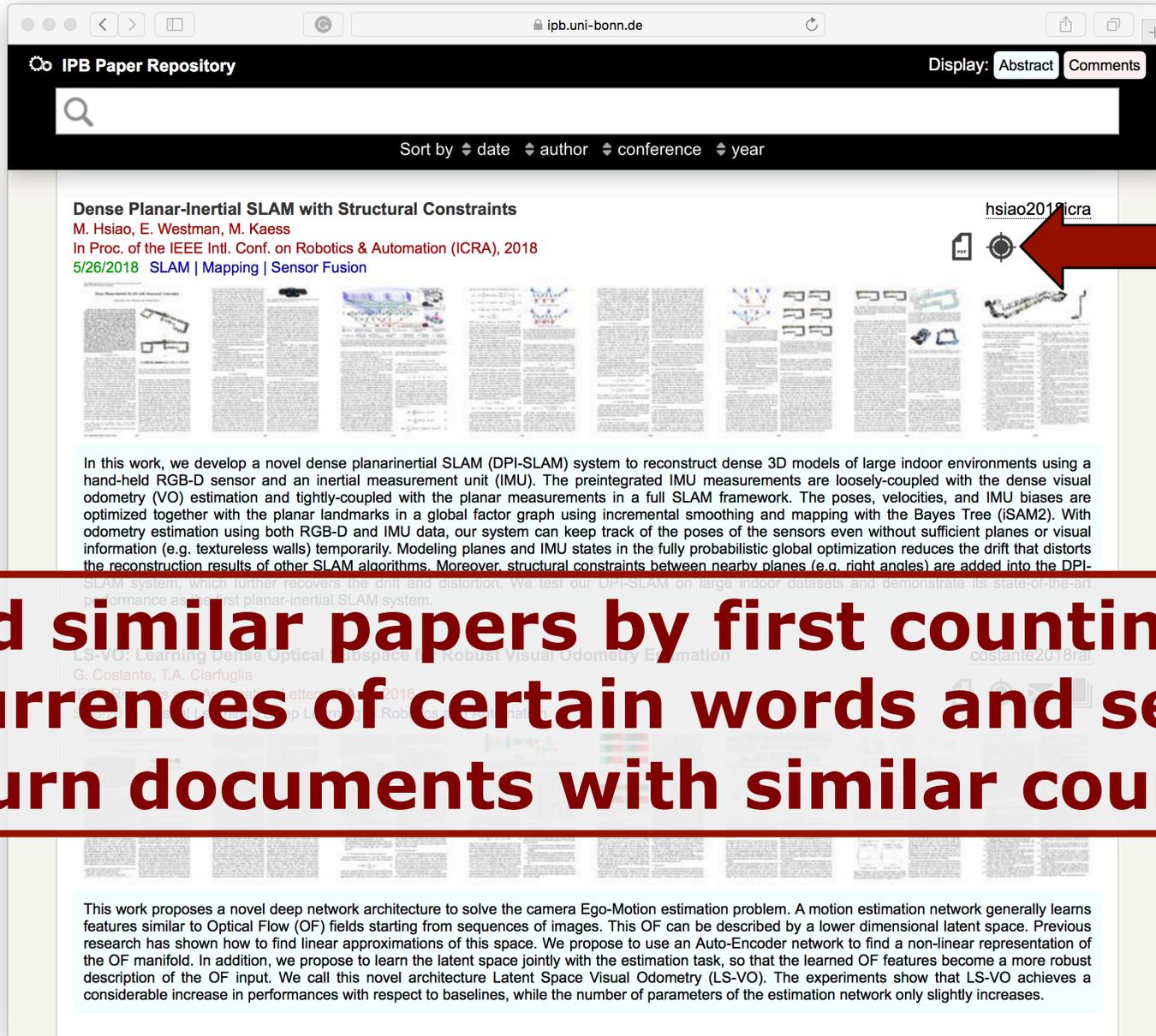
**sensory, brain,
visual, perception,
retinal, cerebral cortex,
eye, cell, optical
nerve, image
Hubel, Wiesel**

China is forecasting a trade surplus of \$90bn (£51bn) to \$100bn this year, a threefold increase on 2004's \$32bn. The Commerce Ministry says the surplus would be created by a jump in exports of 18% and a 10% rise in imports. The government says the rise in exports is likely to be helped by a long-term plan to help the yuan, bank, domestic, foreign, increase, trade, value.



**China, trade,
surplus, commerce,
exports, imports, US,
yuan, bank, domestic,
foreign, increase,
trade, value**

Looking for Similar Papers



The screenshot shows a web browser window with the URL `ipb.uni-bonn.de`. The page is titled "IPB Paper Repository" and has a search bar and sorting options (date, author, conference, year). The main content is a search result for the paper "Dense Planar-Inertial SLAM with Structural Constraints" by M. Hsiao, E. Westman, and M. Kaess, published in the IEEE Intl. Conf. on Robotics & Automation (ICRA) in 2018. The paper's abstract is visible, and a red arrow points to the "hsiao2018icra" citation link. Below the abstract, there is a large red-bordered box containing a quote.

“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



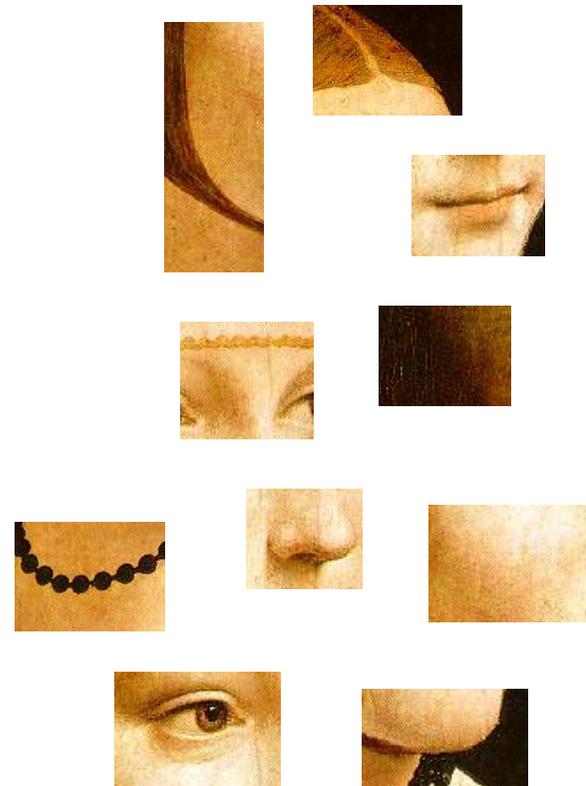
bag of "visual words"

Bag of Visual Words

- Visual words = independent features



face



features

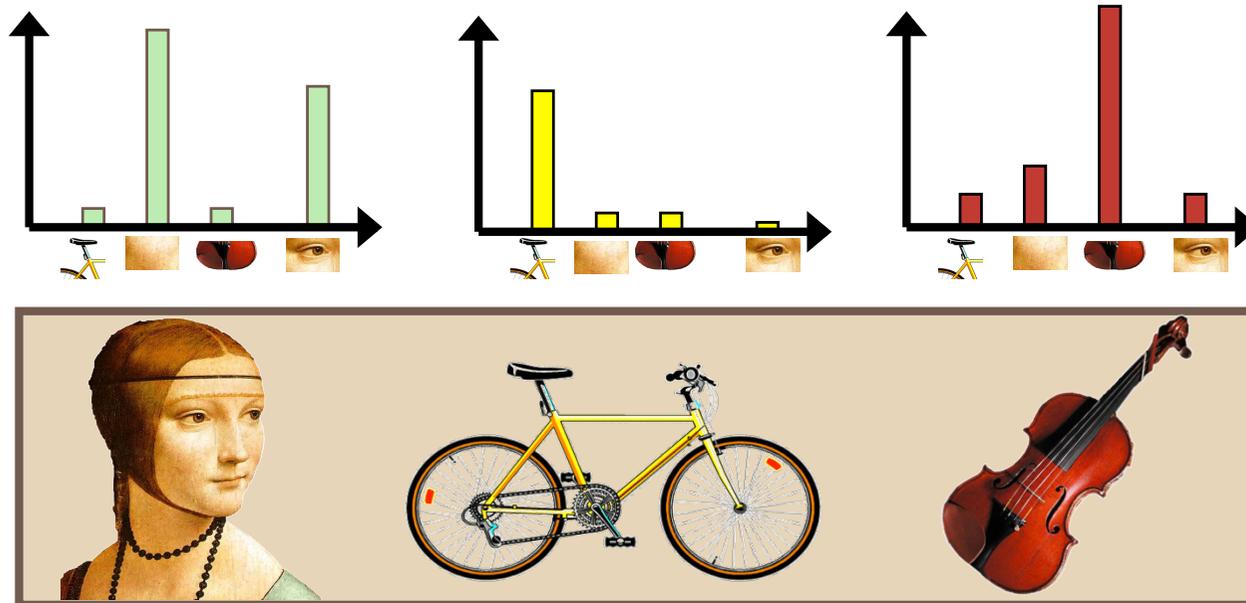
Bag of Visual Words

- Visual words = independent features
 - Construct a dictionary of representative words
 - Use only words from the dictionary
- dictionary (“codebook”)



Bag of Visual Words

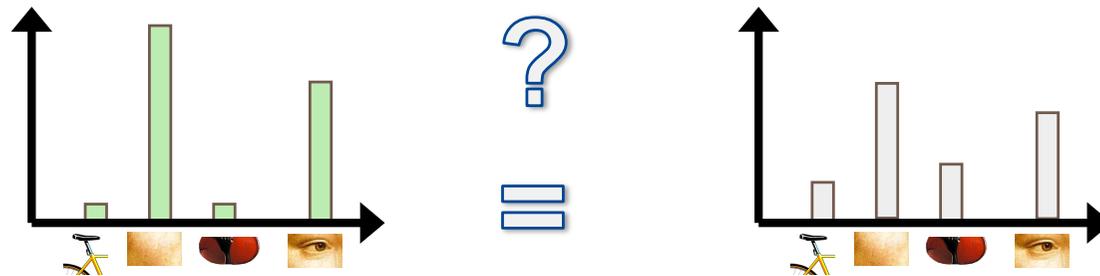
- Visual words = independent features
- Words from the dictionary
- Represent the images based on a histogram of word occurrences



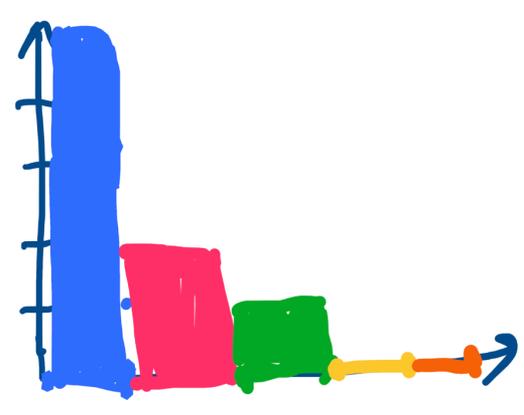
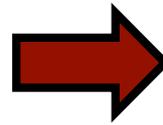
[Image courtesy: Fei-Fei Li]

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



From Images to Histograms



[Image courtesy: Olga Vysotska]

Overview: Input Image



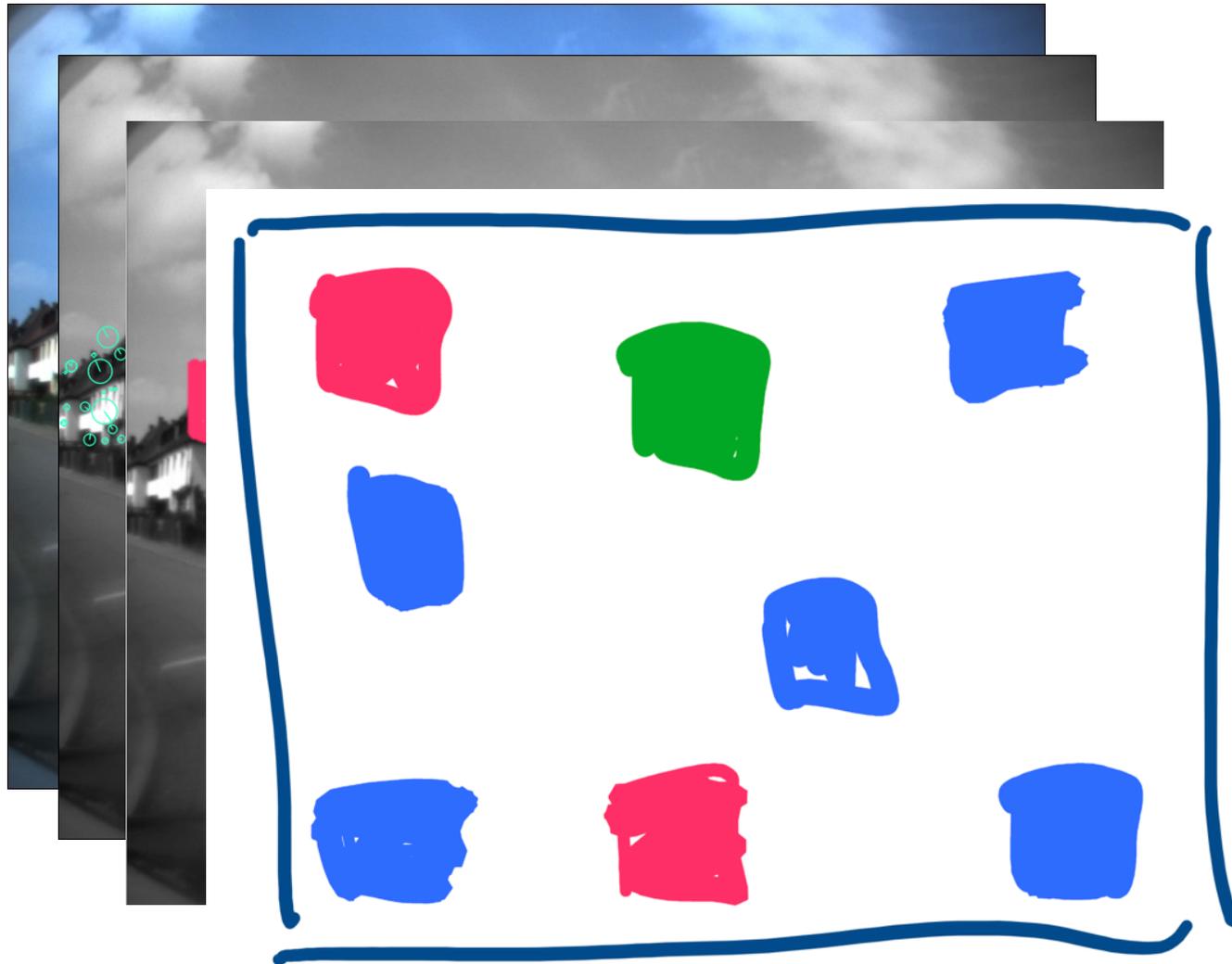
Overview: Extract Features



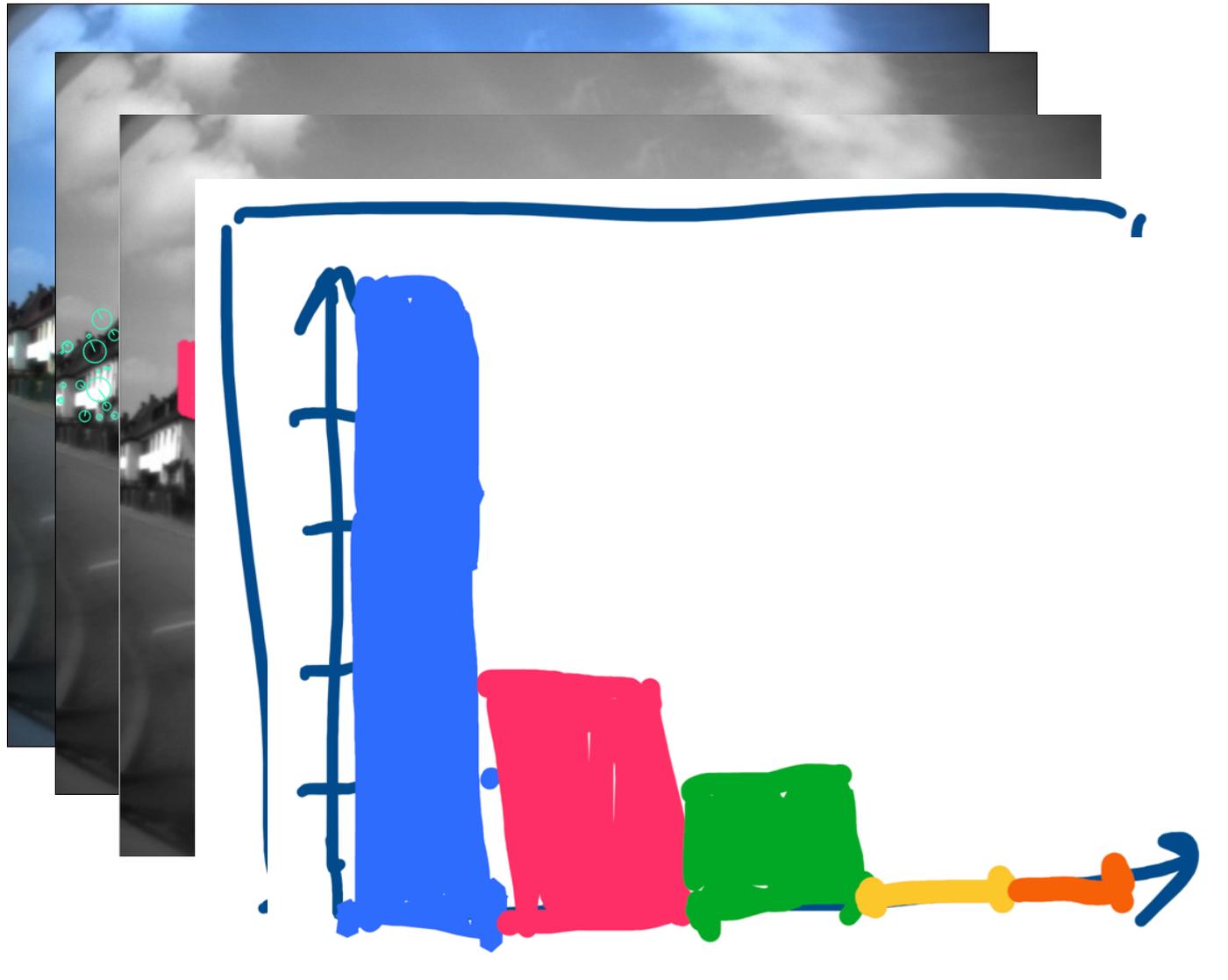
Overview: Visual Words



Overview: No Pixel Values

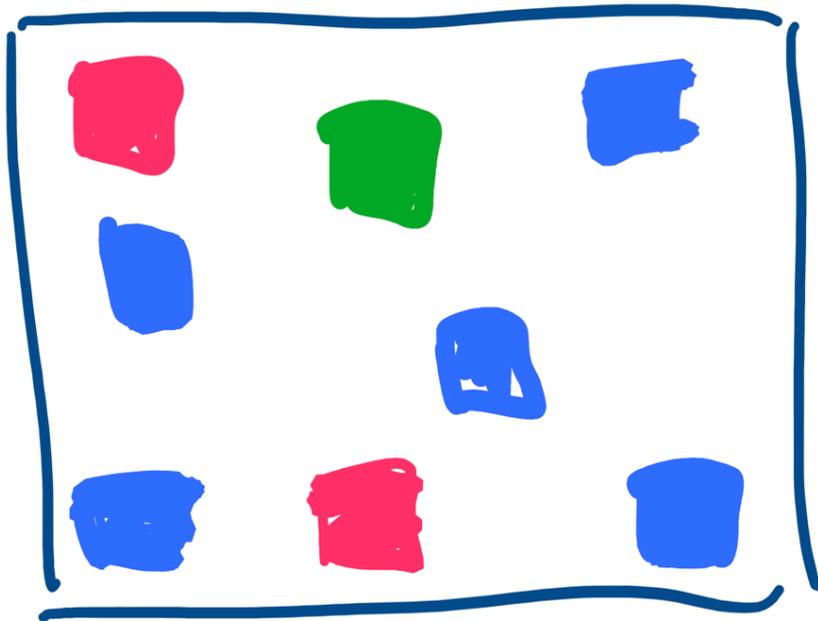


Overview: Word Occurrences



[Image courtesy: Olga Vysotska]

Images to Histograms

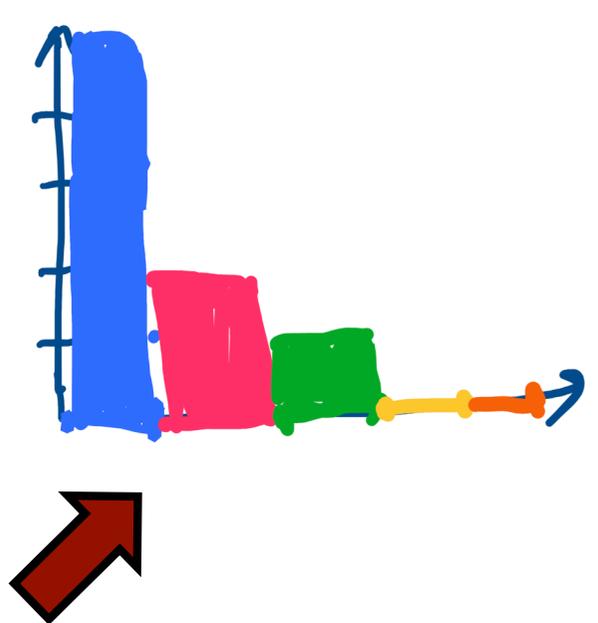


[Image courtesy: Olga Vysotska]

Where Do the Visual Words Come From?

Dictionary

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



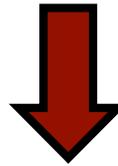
[Image courtesy: Olga Vysotska]

Dictionary

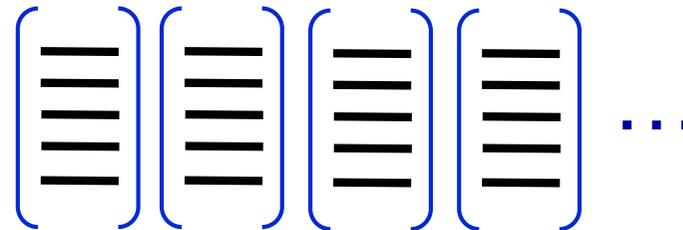
- A dictionary defines the list of words that are considered
- The dictionary defines the x-axes of all the word occurrence histograms
- The dictionary must remain fixed

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

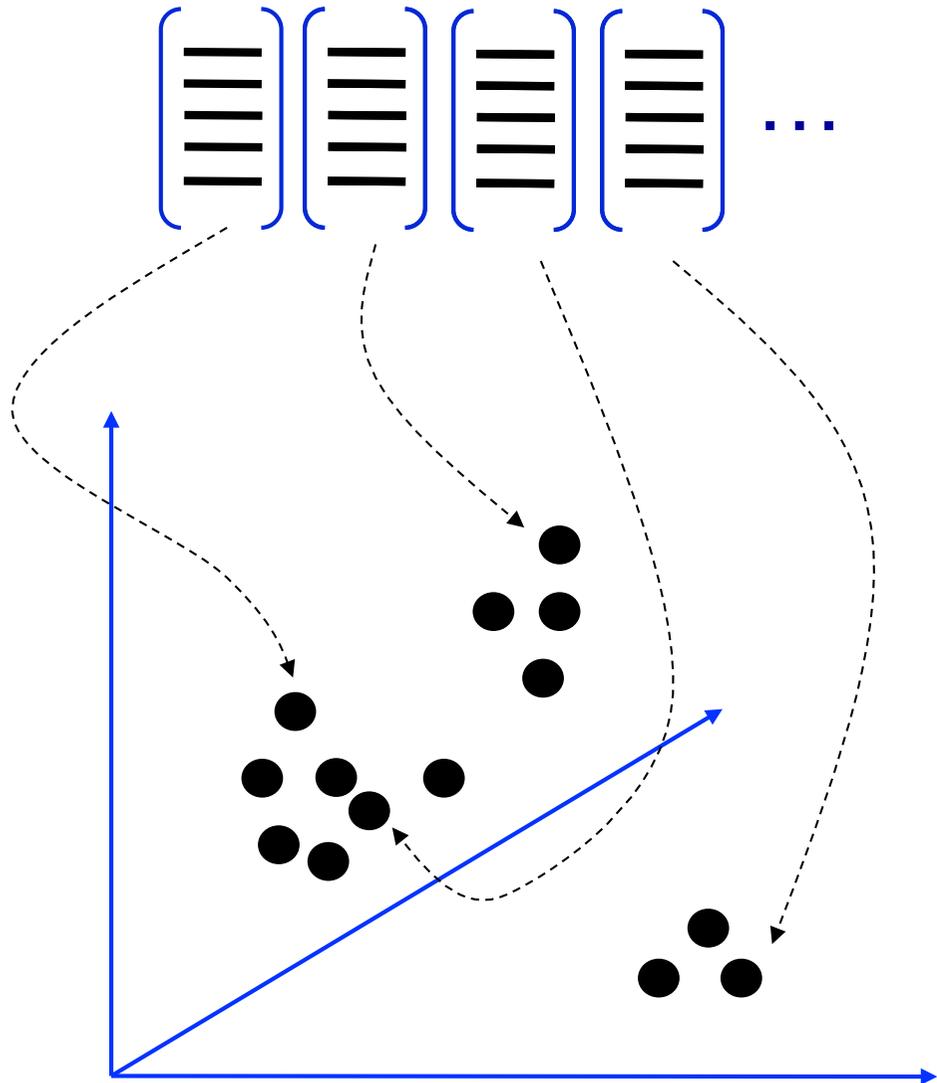
Extract Feature Descriptors from a Training Dataset



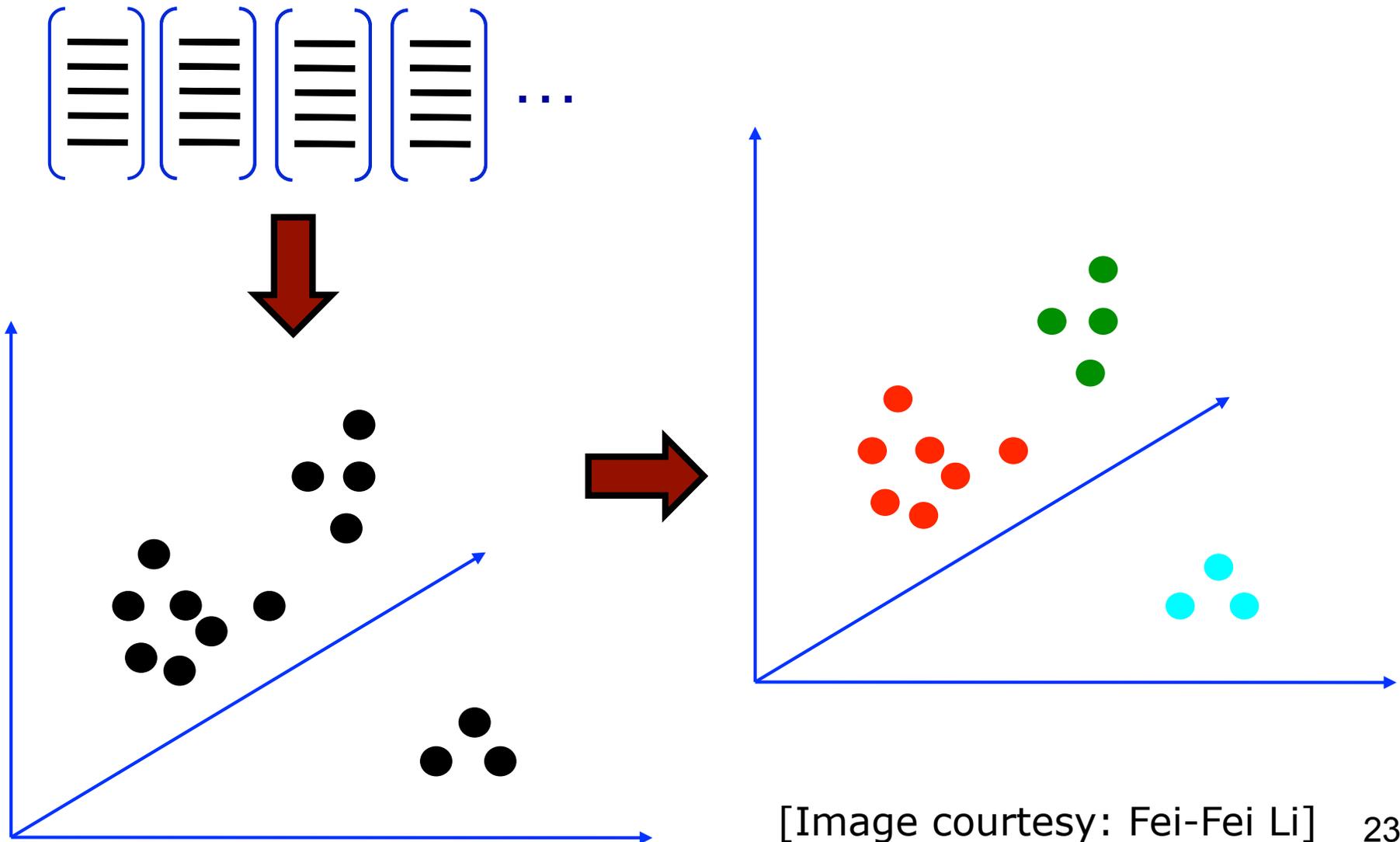
Visual feature
descriptor vectors
(e.g., SIFT)



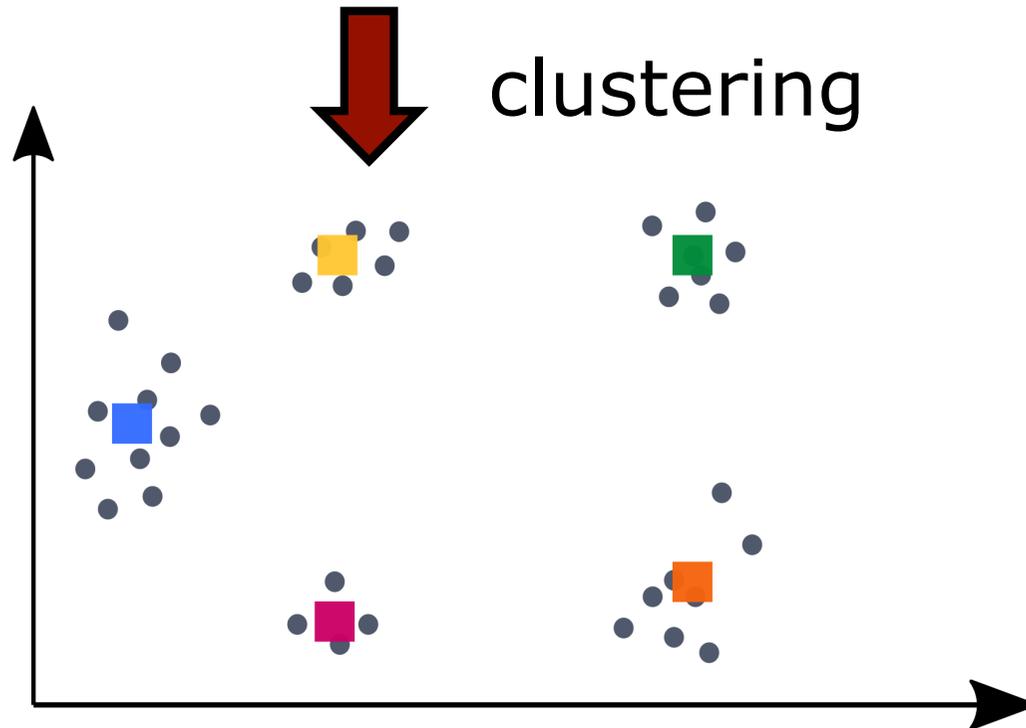
Feature Descriptors are Points in a High-Dimensional Space



Group Similar Descriptors



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 $\mathbf{m}_i, i = 1, \dots, k$, for example, to k random \mathbf{x}^t

Repeat

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

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

For all $\mathbf{m}_i, i = 1, \dots, k$

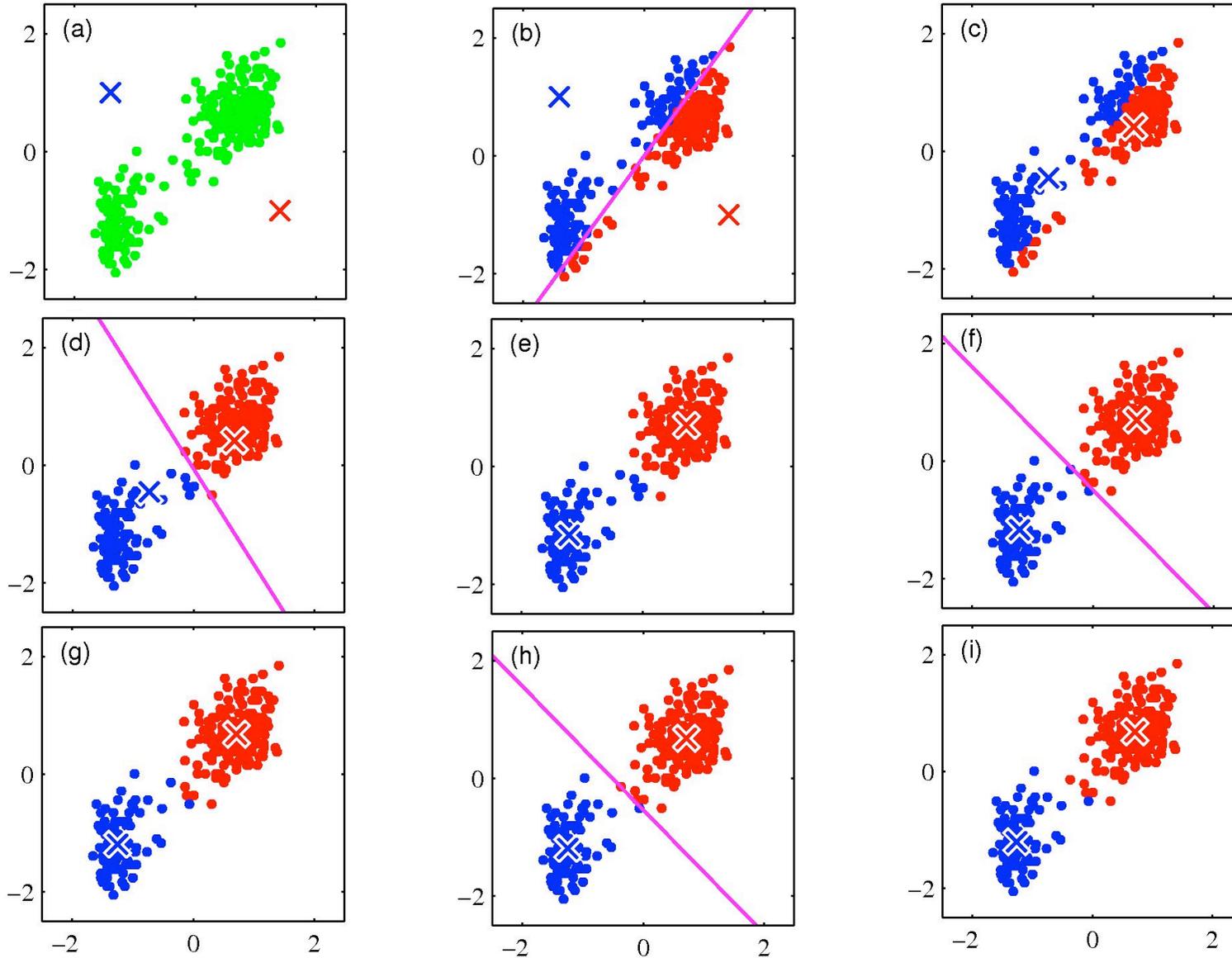
$$\mathbf{m}_i \leftarrow \sum_t b_i^t \mathbf{x}^t / \sum_t b_i^t$$

Until \mathbf{m}_i converge

Re-compute the cluster means using the current cluster memberships

Assign each data point to the closest cluster

K-Means Example

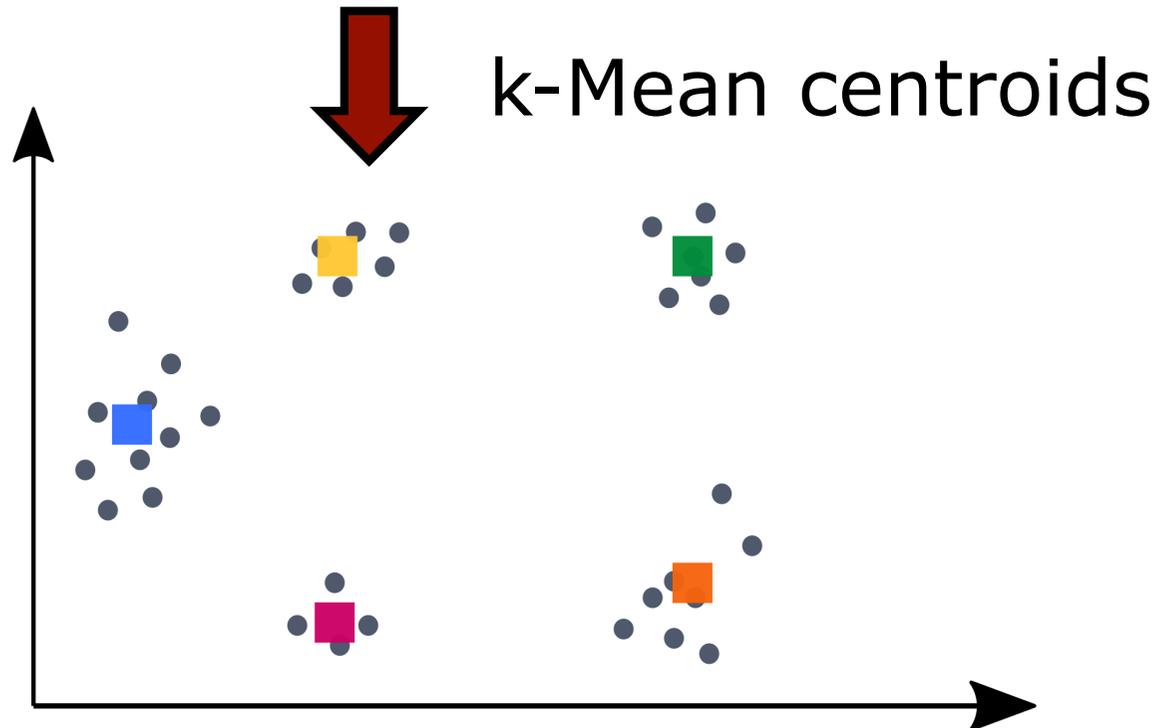


Summary K-Means

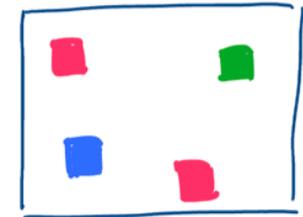
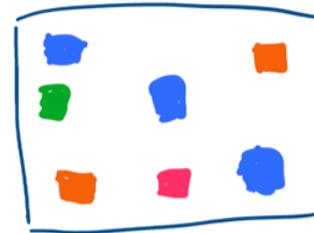
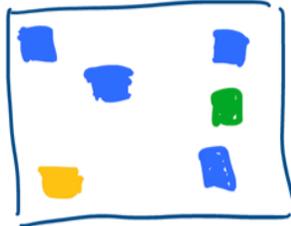
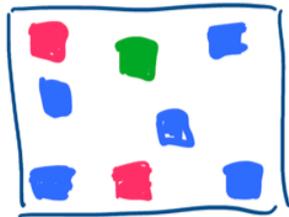
- Standard approach to clustering
- Simple to implement
- Number of clusters k must be chosen
- Depends on the initialization
- Sensitive to outliers
- Prone to local minima

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

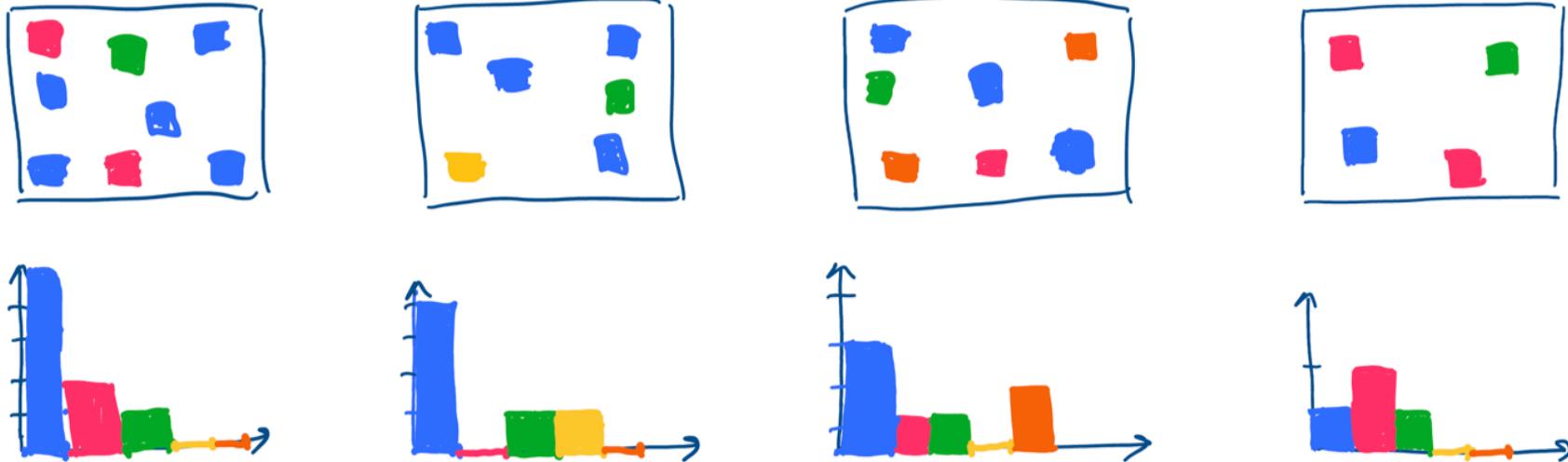
K-Means for Building the Dictionary from Training Data



All Images are Reduced to Visual Words



All Images are Represented by Visual Word Occurrences



Every image turns into a histogram

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

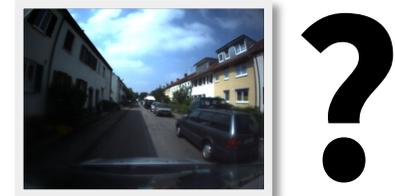
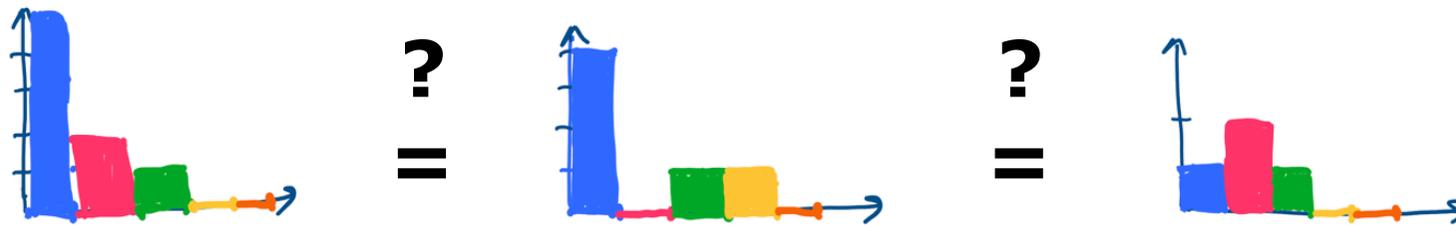
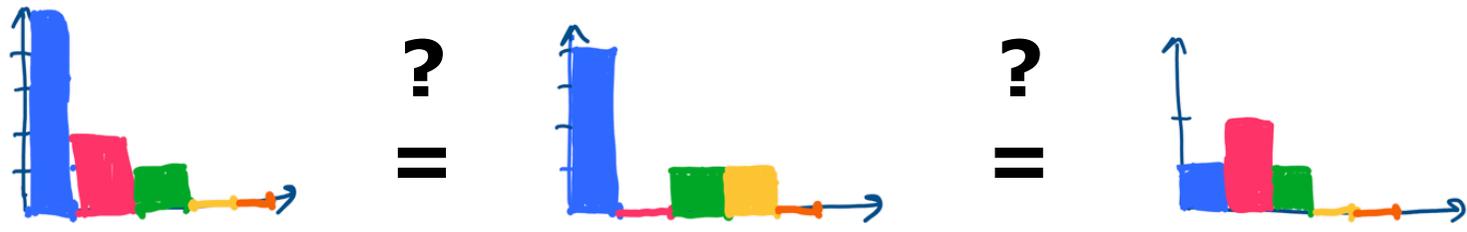


Image Similarity by Comparing Word Occurrence Histograms



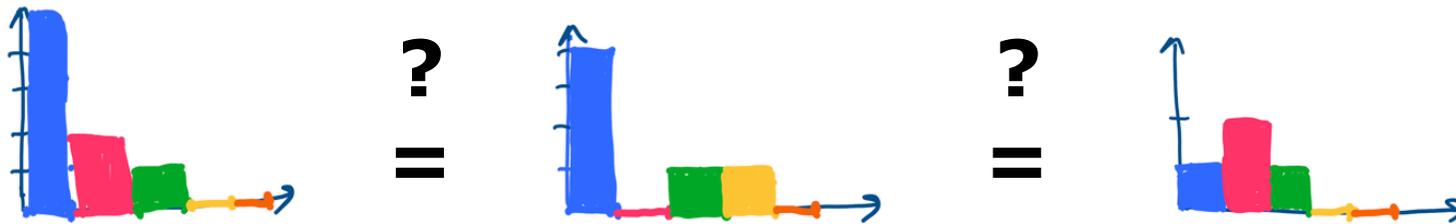
How to Compare Histograms?

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



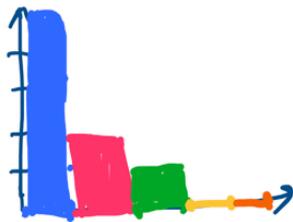
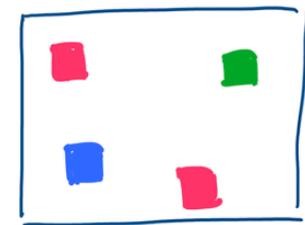
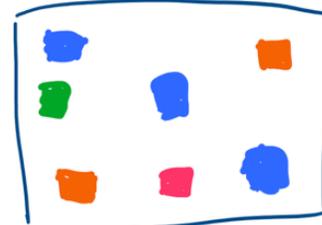
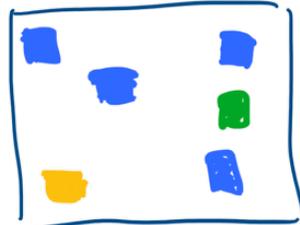
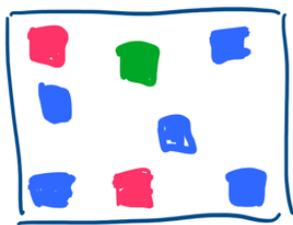
Are All Words Expressive for Comparing Histograms?

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



Some Words are Less Expressive Than Others!

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



- Example: the “green word” is useless

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

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

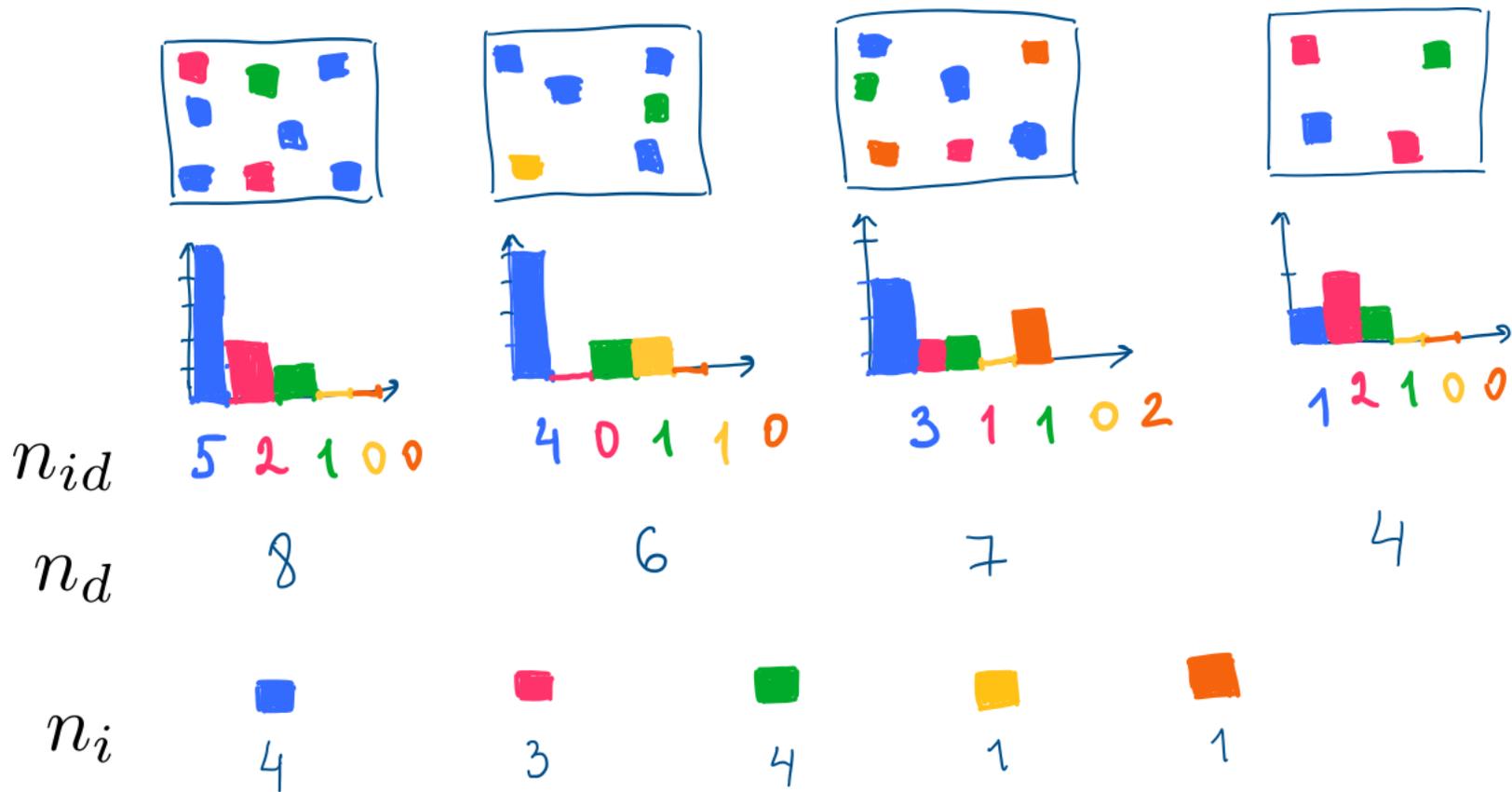
term frequency (arrow pointing to n_{id})

inverse document frequency (arrow pointing to $\frac{N}{n_i}$)

bin of word i in image d (arrow pointing to t_{id})

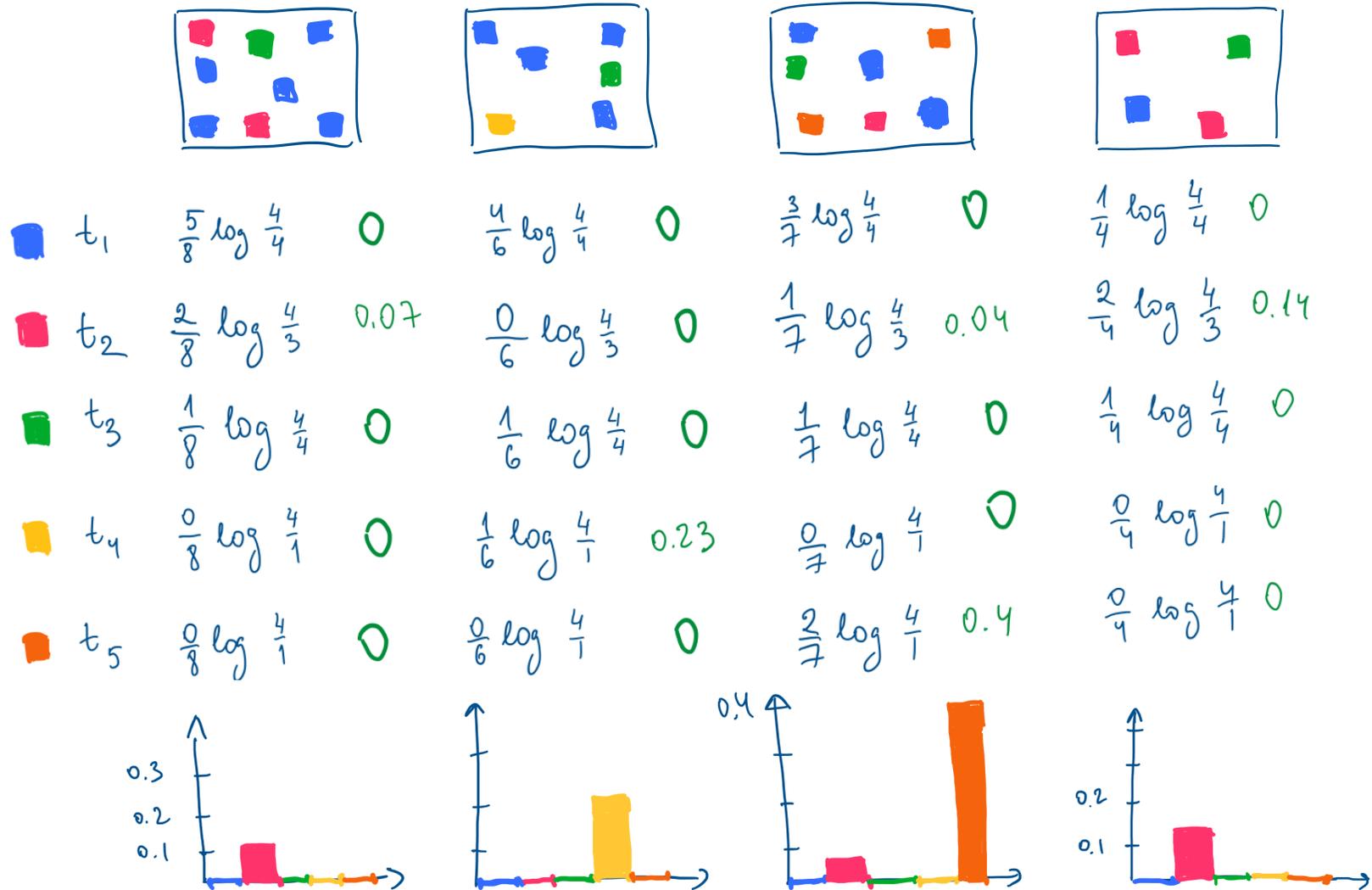
- t_{id} : histogram bin of word i for image d
- n_{id} : occurrences of word i in image d
- n_d : number of word occurrences in image d
- n_i : number of images that contain word i
- N : number of images

Computing the TF-IDF (1)



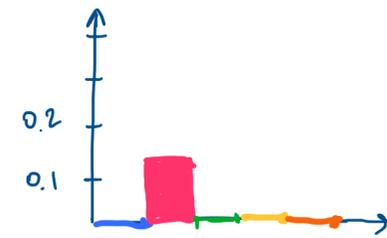
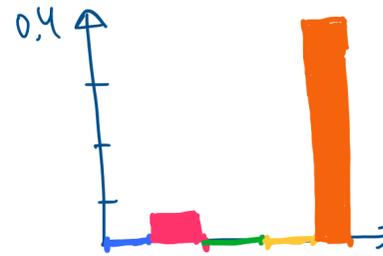
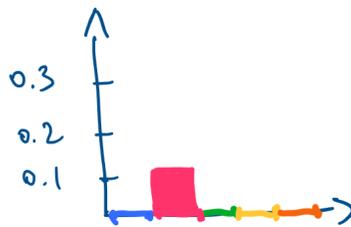
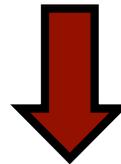
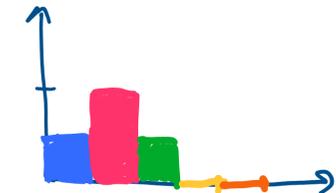
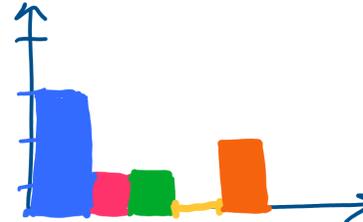
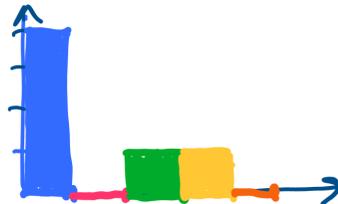
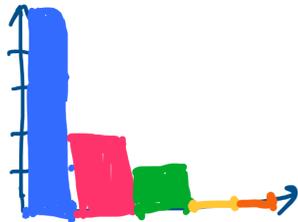
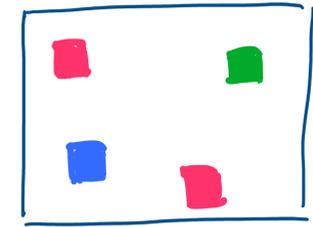
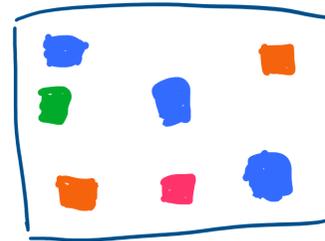
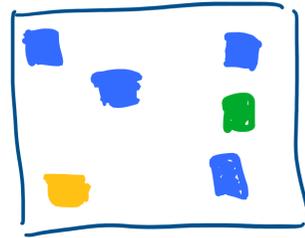
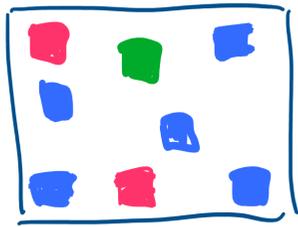
$$t_{id} = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

Computing the TF-IDF (2)



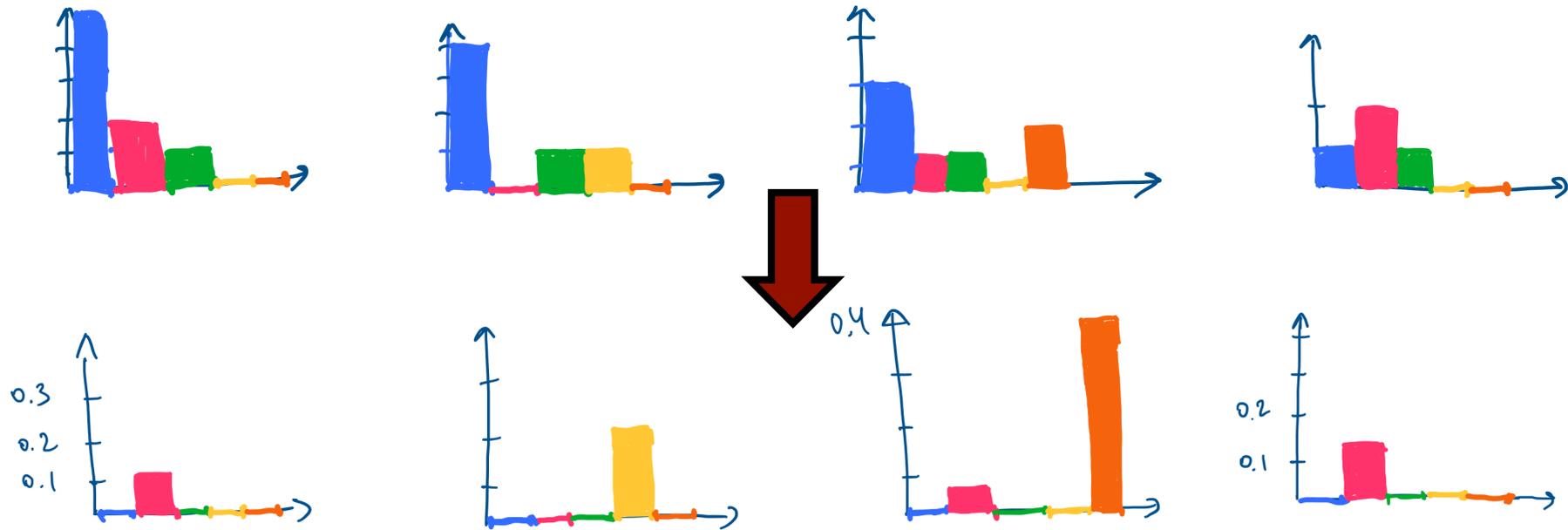
[Image courtesy: Olga Vysotska]

Reweighted Histograms



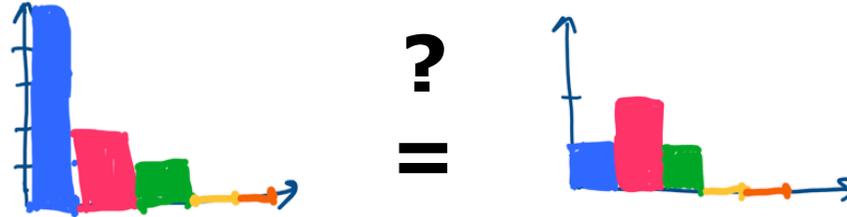
[Image courtesy: Olga Vysotska]

Reweighted Histograms



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

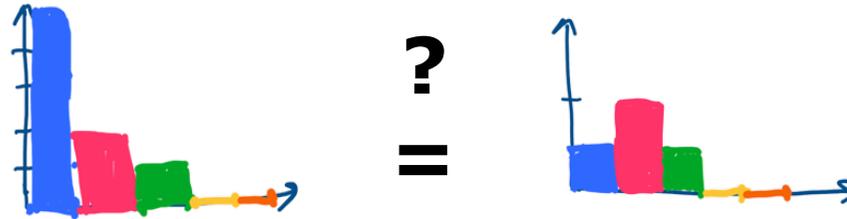
Comparing Two Histograms



Options

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

Comparing Two Histograms



Options

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

BoVW approaches often use the cosine distance for comparisons

Cosine Similarity and Distance

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

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

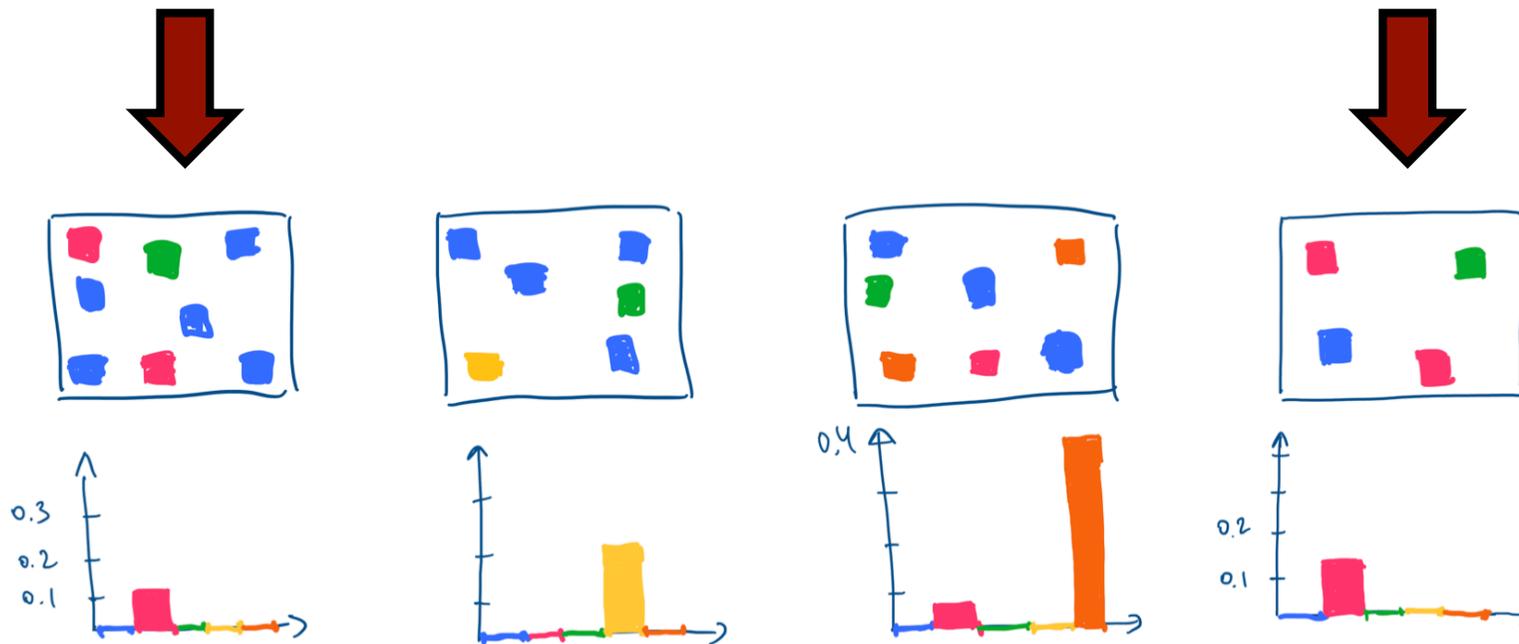
- We use the cosine distance

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

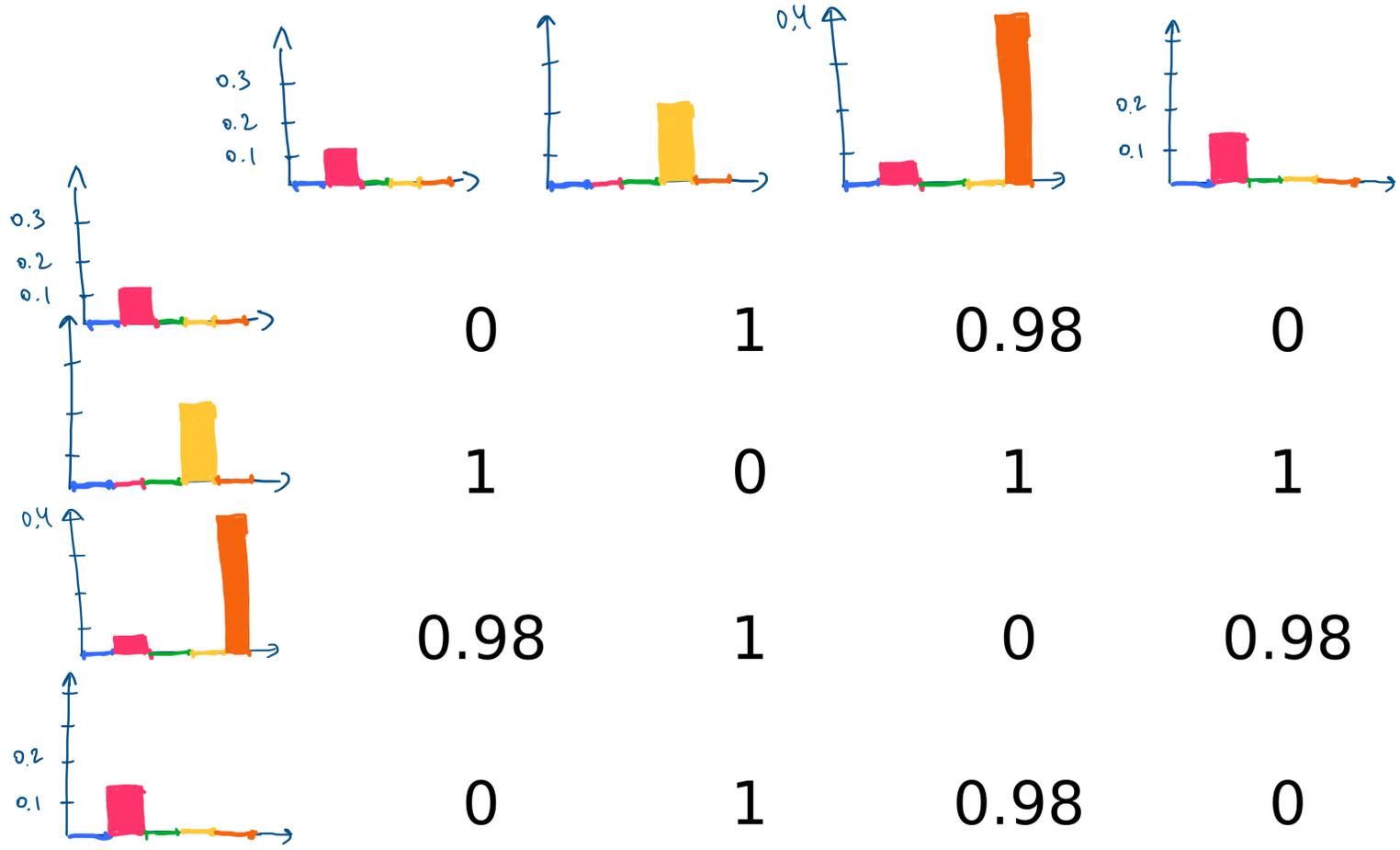
- Takes values between 0 and 1 (for vectors in the 1st quadrant)

Example Comparing Histograms

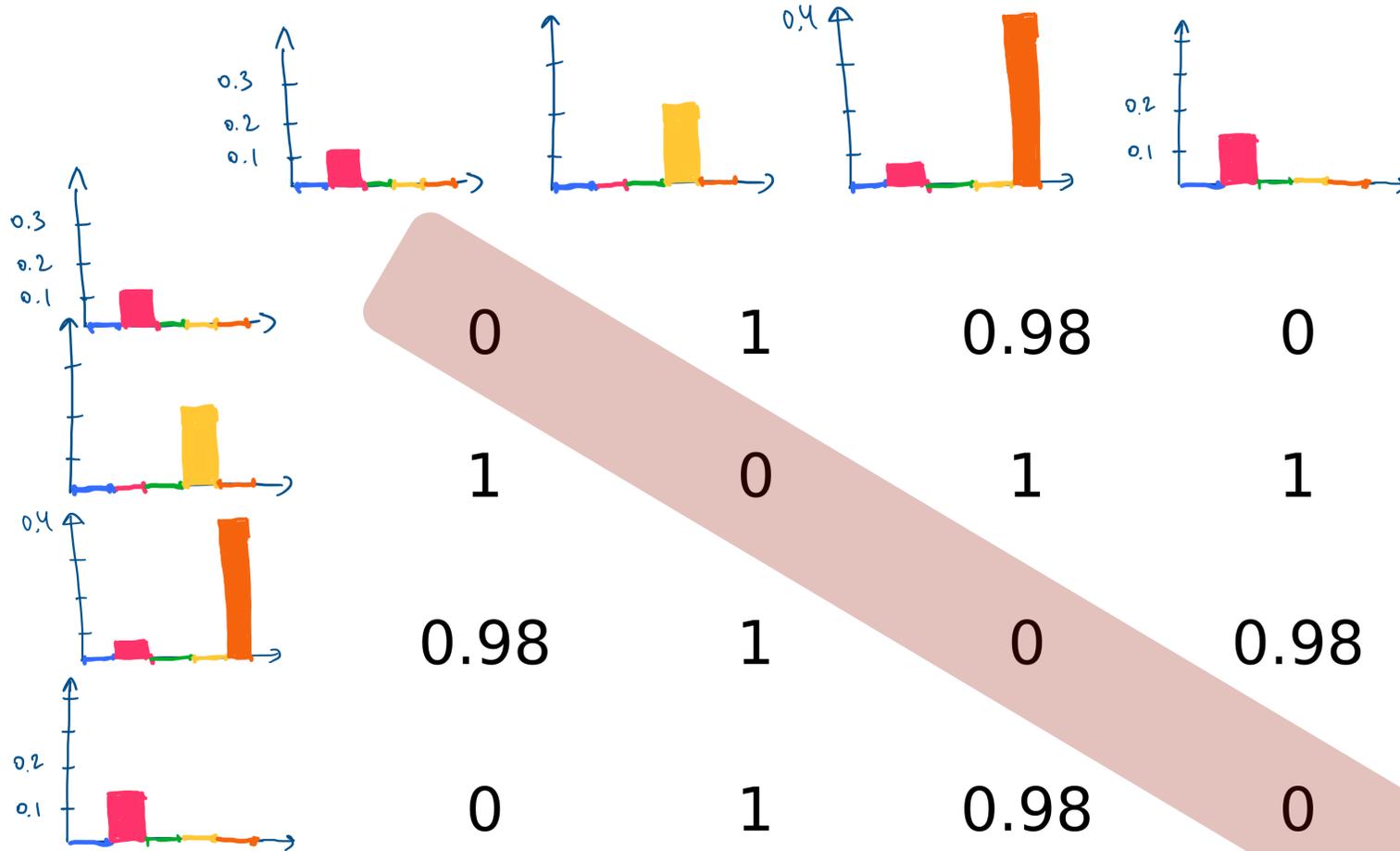
- 4 images
- Image 0 and image 3 are similar



Example Comparing Histograms

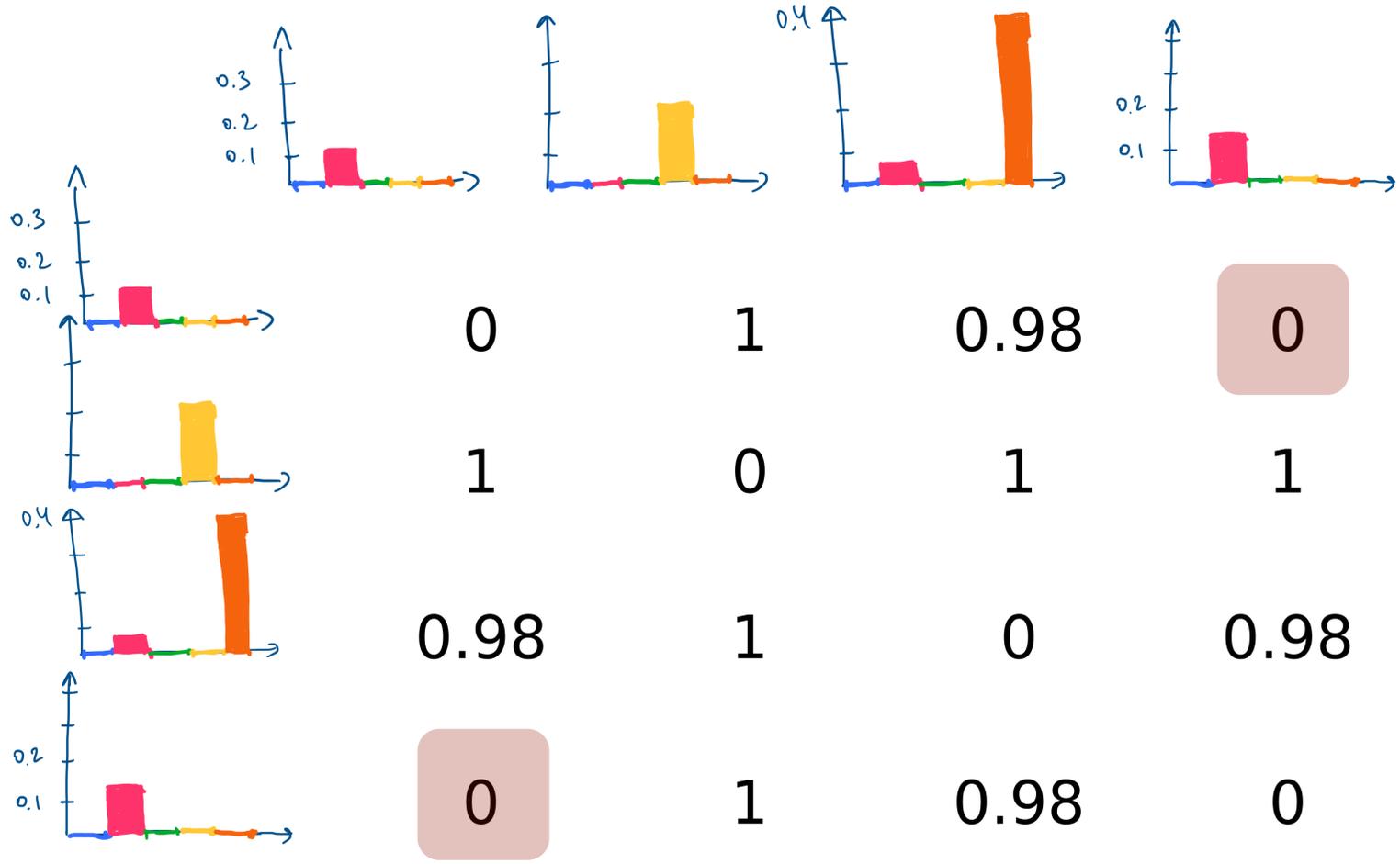


Example Comparing Histograms



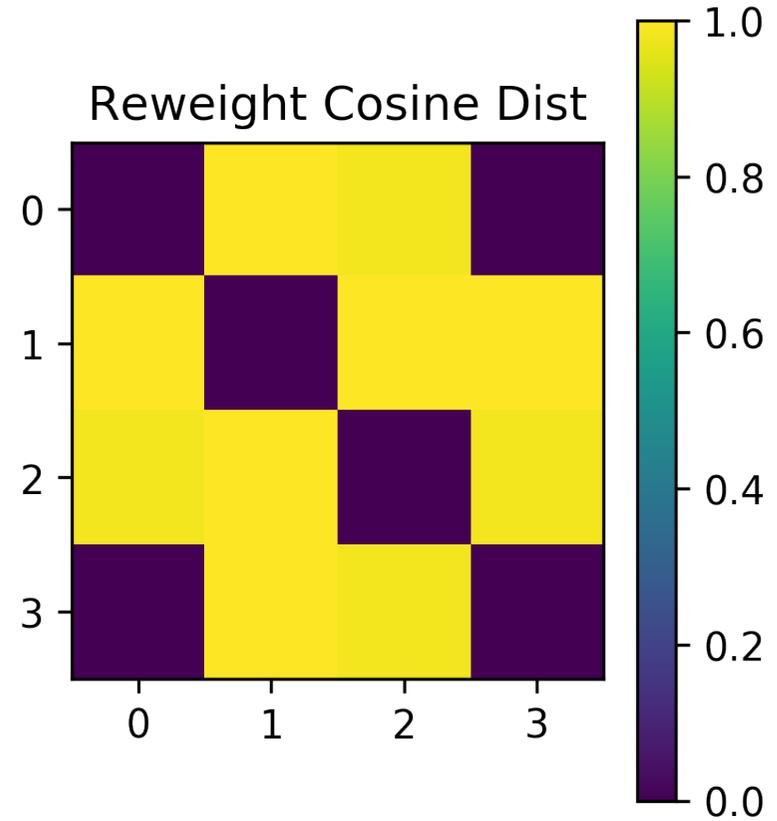
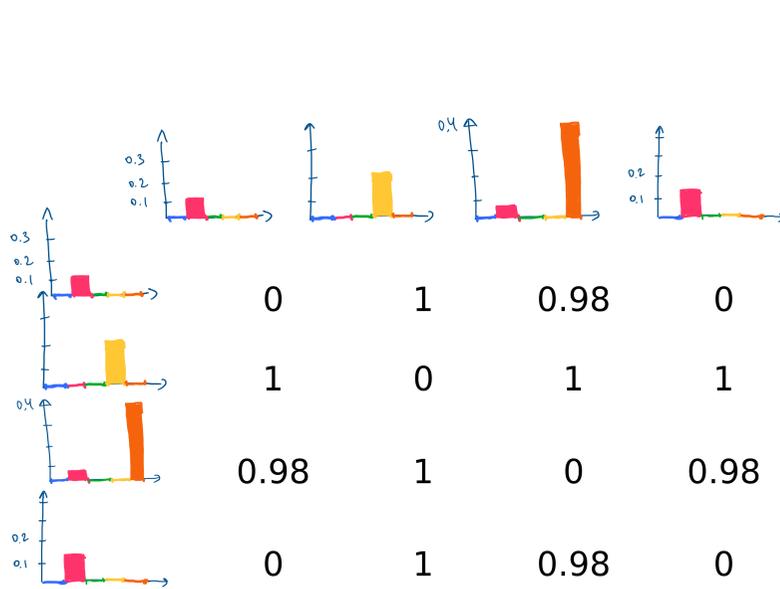
Images have a zero distance to themselves

Example Comparing Histograms

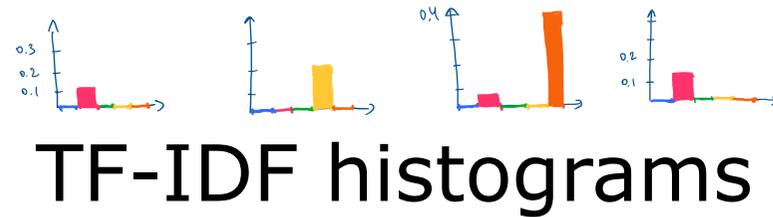
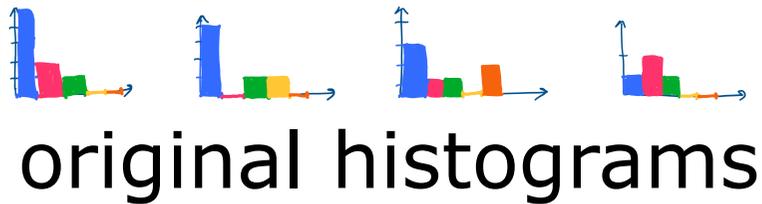
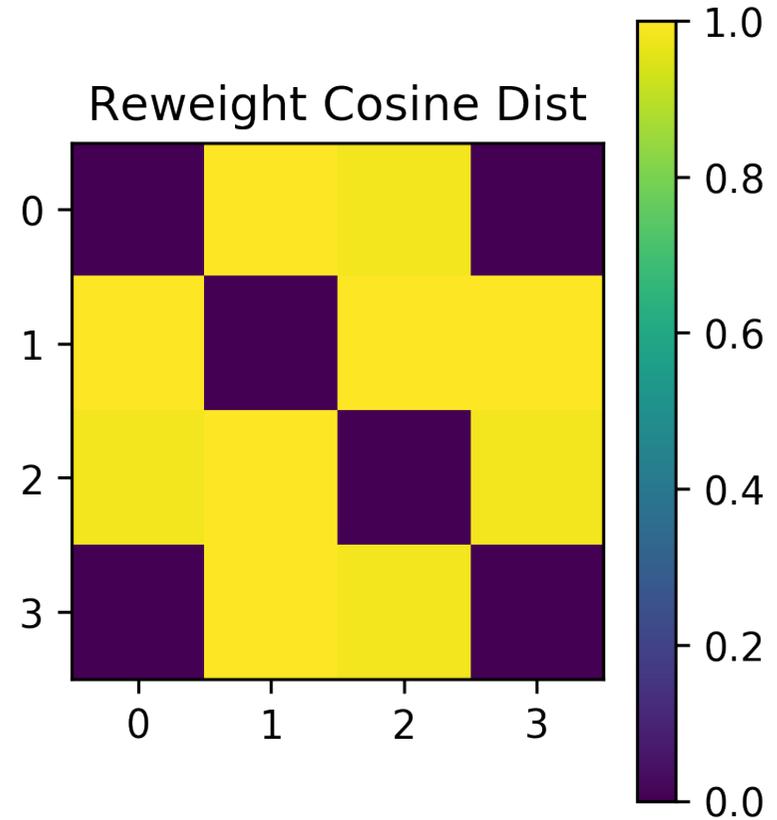
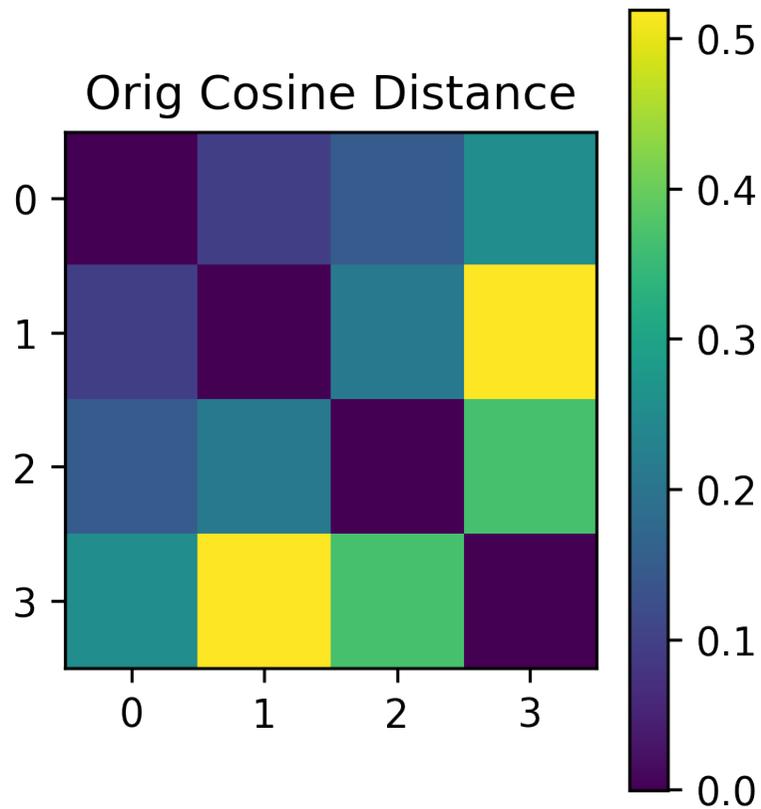


Images 0 and 3 are highly similar

Cost Matrix



IF-IDF Actually Helps



Euclidean vs. Cosine Distance

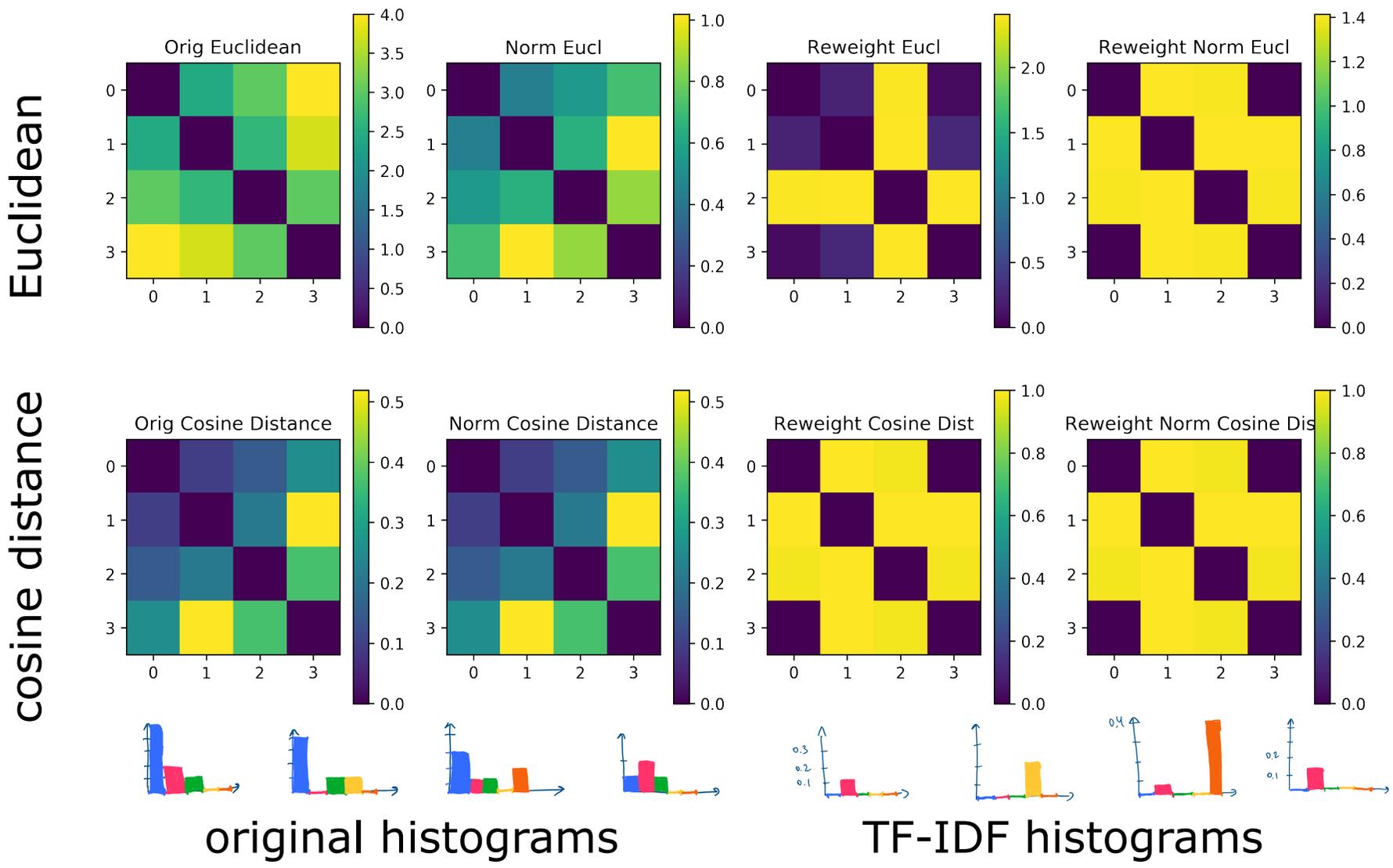
- Cosine distance ignores the length of the vectors
- **For vectors of length 1**, the squared Euclidean and the cosine distance only differ by a factor of 2:

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

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

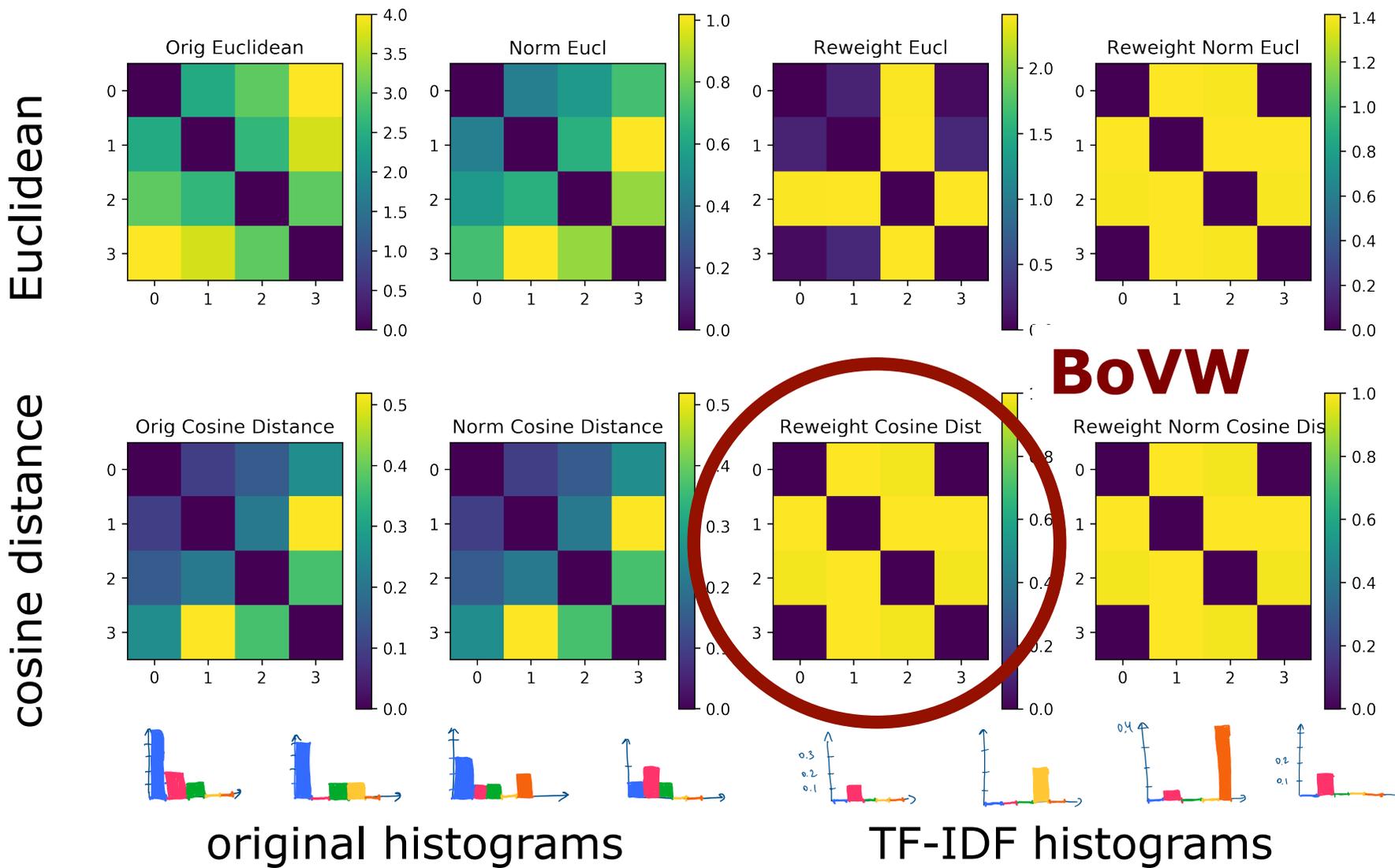
$$\begin{aligned}\|\mathbf{x} - \mathbf{y}\|^2 &= 2 - 2\mathbf{x}^\top \mathbf{y} = 2 - 2 \cos \theta \\ &= 2 d_{\cos}(\mathbf{x}, \mathbf{y})\end{aligned}$$

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

TF-IDF weighting

Let's assume we already have a bag of visual words for our images and we have 4 images that are represented through image histograms. In the example below, image 0 and image 3 are similar to each other. Thus, we expect the matching algorithm to report them as similar. Every reweighted word in the histogram can be computed using the TF-IDF formula given by

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

where

- n_{id} - occurrence of word i in a document (image) d ;
- n_d total number of words in a document d ;
- n_i number of documents (images in the database) that contain the word i
- N number of documents (images in the database)

n_{id}	5 2 1 0 0	4 0 1 1 0	3 1 1 0 2	1 2 1 0 0	
n_d	8	6	7	4	
n_i	4	3	4	1	1

In the example above, image 0 contains 5 blue, 2 pink, 1 green, and no yellow or orange words. The image has in total 8 word occurrences. With this information, we can compute the TF $\frac{n_{id}}{n_d} = \frac{5}{8}$. We can observe that blue word occurs in all 4 images, whereas, for example, the yellow word occurs only in one. Thus, we can compute the overall weighting t_i for all words in every histogram. We

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