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
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 arrive in the brain from our eyes. For a long time it was thought that the retinal image was transmitted point by point to visual centers in the cerebral cortex, as if the eye were a movie projector and the cerebral cortex the screen upon which the image was projected. Through the discoveries of Hubel and Wiesel we now know that behind the origin of the visual perception in the brain there is a considerably more complicated course of events. By following the visual impulses along the pathway from the retina to the various cell layers of the optical cortex, Hubel and Wiesel have been able to demonstrate that the message about the image falling on the retina undergoes a step-wise analysis in 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.

China is forecasting a trade surplus of $90bn (£51bn) to $100bn this year, a threefold increase on 2004's $32bn. The Commerce Ministry said the surplus would be created by a predicted 30% jump in exports to $750bn, compared with a 18% rise in imports to $660bn. The US has long argued that China's exports are unfairly helped by a deliberately undervalued yuan. Beijing agrees the surplus is too high but says the yuan is only one factor. China increased the value of the yuan against the dollar by 2.1% in July and permitted it to trade within a narrow band, but the US wants the yuan to be allowed to trade freely. However, Beijing has made it clear that it will take its time and tread carefully before allowing the yuan to rise further in value.
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.
Bag of Visual Words

- Visual words = independent 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")
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 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)

[Partial image courtesy: Fei-Fei Li]
Feature Descriptors are Points in a High-Dimensional Space

[Image courtesy: Fei-Fei Li]
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 $m_i, i = 1, \ldots, k$, for example, to $k$ random $x^t$

Repeat

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

\[ b^t_i \left\{ \begin{array}{ll} 1 & \text{if } ||x^t - m_i|| = \min_j ||x^t - m_j|| \\ 0 & \text{otherwise} \end{array} \right. \]

For all $m_i, i = 1, \ldots, k$

\[ m_i \leftarrow \frac{\sum_t b^t_i x^t}{\sum_t b^t_i} \]

Until $m_i$ 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

k-Mean centroids

[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
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

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

- \( t_{id} \): 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)

\[ t_{id} = \frac{n_{id}}{n_{d}} \log \frac{N}{n_{i}} \]

[Image courtesy: Olga Vysotska]
Computing the TF-IDF (2)

\[ t_1 \quad \frac{5}{8} \log \frac{4}{4} \quad 0 \]
\[ t_2 \quad \frac{2}{8} \log \frac{4}{5} \quad 0.07 \]
\[ t_3 \quad \frac{1}{8} \log \frac{4}{4} \quad 0 \]
\[ t_4 \quad \frac{0}{8} \log \frac{4}{4} \quad 0 \]
\[ t_5 \quad \frac{3}{8} \log \frac{4}{4} \quad 0 \]

[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

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:
  \[
  \text{cossim}(x, y) = \cos(\theta) = \frac{x^\top y}{\|x\| \|y\|}
  \]

- We use the cosine distance
  \[
  d_{\text{cos}}(x, y) = 1 - \text{cossim}(x, y) = 1 - \frac{x^\top y}{\|x\| \|y\|}
  \]

- 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

[Image courtesy: Olga Vysotska]
Example Comparing Histograms

Images have a zero distance to themselves

[Image courtesy: Olga Vysotska]
Example Comparing Histograms

Images 0 and 3 are highly similar

[Image courtesy: Olga Vysotska]
Cost Matrix

[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:

\[
\|x - y\|^2 = (x - y)^\top (x - y)
\]

\[
= x^\top x - 2x^\top y + y^\top y
\]

as \(\|x\| = \|y\| = 1\)

\[
\|x - y\|^2 = 2 - 2x^\top y = 2 - 2 \cos \theta
\]

\[
= 2 \, d_{\cos}(x, y)
\]
Comparison of Distance Metrics

- Euclidean Distance
  - Orig Euclidean
  - Norm Eucl
  - Reweight Eucl
  - Reweight Norm Eucl

- Cosine Distance
  - Orig Cosine Distance
  - Norm Cosine Distance
  - Reweight Cosine Dist
  - Reweight Norm Cosine Dist

Original histograms vs. TF-IDF histograms

[Image courtesy: Olga Vysotska]
Comparison of Distance Metrics

Original histograms

TF-IDF histograms

BoVW

[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


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

\[ t_i = \frac{n_w}{N} \log \frac{N}{n_d} \]

where

- \( n_w \) - occurrence of word \( i \) in a document (image) \( d \);
- \( n_d \) - total number of words in a document (image) \( d \);
- \( n_d \) - number of documents (images in the database) that contain the word \( i \);
- \( N \) - number of documents (images in the database)

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-IDF as \( t_0 = \frac{1}{8} \). 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 \( t_0 \) 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:

- 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