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.

1

Preparation: Watch 5 Min Video



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 from our eves. For a that the sensory, brain, point retinal imp to visue visual, perception, cortex upon retinal, cerebral cortes project eye, cell, optical and W nerve, image origin of there is Hubel, Wiesel course of Sual impulses alond us cell layers of the optics and Wiesel have been able to demon that the message about the image fallin he retina undergoes a step-wise analys system of nerve cells stored in column this system each cell has its spe function and is responsible for a specidetail 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 Minisurplus would be created jump in China, trade, 🔪 exports // 18% rise in are urplus, commerce likelv has long exports, imports, US lirlv help uan, bank, domestic ın. Beiji but foreign, increase, of says 1 China the trade, value country a boost domestic staved within the course ed the value of the yuan against the do 2.1% in July and permitted it to trad in a narrow band, but the US wants the to be allowed to trade freely. However, has made it clear that it will take its tin tread carefully before allowing the yua rise further in value.

5

Looking for Similar Papers

	••• < > m	0	🗎 ipb.uni-bonn.de	¢	西二十二十	
	Co IPB Paper Reposito	ry		Dis	Abstract Comments	
	Q					
		Sort by ‡ dat	te \$author \$conference	≑ year		
	Dense Planar-Iner	tial SLAM with Structural Constrain	nts		hslao2017licra	
	M. Hsiao, E. Westman In Proc. of the IEEE In 5/26/2018 SLAM LMs	, M. Kaess II. Conf. on Robotics & Automation (ICRA), pping I Sensor Fusion	, 2018	1		
	12 P					
	In this work, we devi hand-held RGB-D is odornetry (VO) estin optimized together v odornetry estimation information (e.g. text	elop a novel dense planarinertial SLAM (I ensor and an inertial measurement unit (nation and tightly-coupled with the planar ith the planar landmarks in a global fact using both RGB-D and IMU data, our sy ureless walls) temporarily, Modeling plane with of elem SI AM describters. Manaret	DPI-SLAM) system to reconstruct (IMU). The preintegrated IMU m r measurements in a full SLAM or graph using incremental smor stem can keep track of the poes s and IMU states in the fully prot	dense 3D models of large indoor e assurements are loosely-coupled w framework. The poses, velocities, whing and mapping with the Bayes s of the sensors even without suffic abilistic global optimization reduces are one of the sensors are an even without suffic	wironments using a th the dense visual and IMU blases are Tree (ISAM2). With ent planes or visual the drift that distorts and and the DB	
"find occu	simil rrence	ar pape es of ce	rs by f rtain v	irst co vords a	unting t and seco	:he onc
ιειι		cument		511111a	Counts) .
	This work proposes i features similar to O research has shown the OF manifold, In a description of the O considerable increase	a novel deep network architecture to solve plical Flow (OF) fields starting from seque how to find linear approximations of this a iddition, we propose to learn the latent spi F input. We call this novel architecture L a in performances with respect to baseline	the camera Ego-Motion estimation noces of images. This OF can be space. We propose to use an Aut soc jointly with the estimation tasi latent Space Visual Odometry (I s, while the number of parameters	n problem. A motion estimation netw described by a lower dimensional la o-Encoder network to find a non-line , so that the learned OF features be .S-VO). The experiments show tha of the estimation network only sligh	ork generally learns ent space. Previous ser representation of come a more robust LS-VO achieves a dy increases.	

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] 7

Visual words = independent features



[Image courtesy: Fei-Fei Li] 8

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

dictionary ("codebook")



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



[Image courtesy: Fei-Fei Li] 10

- 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





Overview: Input Image



Overview: Extract Features



Overview: Visual Words



Overview: No Pixel Values



Overview: Word Occurrences



Images to Histograms



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



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





Feature Descriptors are Points in a High-Dimensional Space



[Image courtesy: Fei-Fei Li] 23

Group Similar Descriptors



Clusters of Descriptors from Data Forms the Dictionary





25

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 $oldsymbol{x}^t \in \mathcal{X}$ 1 if $\|\boldsymbol{x}^t - \boldsymbol{m}_i\| = \min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\|$ 0 otherwise $b_i^t \leftarrow \mathbf{k}$ For all $\boldsymbol{m}_i, i = 1, \dots, k$ $\boldsymbol{m}_i \leftarrow \sum_t b_i^t \boldsymbol{x}^t / \sum_t b_i^t$ Until \boldsymbol{m}_i converge Re-compute the cluster Assign each data point to the closest means using the current cluster cluster memberships

K-Means Example



31

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 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

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



- 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)



45

Computing the TF-IDF (2)









1 log 4 0 きんのち ŋ 4 log 4 5 60 4 0 0 t, 2 69 3 0.14 1 Log 43 0.04 0.07 t2 2 log 3 C log 43 0 1 6g 4 0 1 log 4 1 log 4 0 tz 1 Rog 4 0 0 0 q log 7 0 Of Log 4 ty - glog y 0 { log 4 0.23 € 65 7 O 0.4 Zlog 4 Elog 4 t 5 glog 1 0 0 04 4 0.3 02 0.2 0,1 0.1

Reweighted Histograms



47

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:

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

We use the cosine distance

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

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

- 4 images
- Image 0 and image 3 are similar







Images have a zero distance to themselves



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:

$$\begin{aligned} ||\mathbf{x} - \mathbf{y}||^2 &= (\mathbf{x} - \mathbf{y})^\top (\mathbf{x} - \mathbf{y}) \\ &= \mathbf{x}^\top \mathbf{x} - 2\mathbf{x}^\top \mathbf{y} + \mathbf{y}^\top \mathbf{y} \\ \text{as } ||\mathbf{x}|| &= ||\mathbf{y}|| = 1 \\ ||\mathbf{x} - \mathbf{y}||^2 &= 2 - 2\mathbf{x}^\top \mathbf{y} = 2 - 2\cos\theta \\ &= 2 \operatorname{d}_{\cos}(\mathbf{x}, \mathbf{y}) \end{aligned}$$

Comparison of Distance Metrics









Comparison of Distance Metrics



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

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/bl ob/master/bag_of_visual_words.ipynb

••		O in_simple,	english/bag_of_visual_ X +	-								
)→	G	ŵ	C A https://github.c	om/ovysotska/in_simple,	english/blob/	master/bag_of_visu	… ⊚	☆	II/		8	Ξ
		TF-IDF	weighting									
		Let's assum histograms. them as simi	e we already have a bag In the example below, ima liar. Every reweighted word	of visual words for our age 0 and image 3 are s d in the histogram can be I ₁ :	images and imilar to each e computed us = $\frac{n_{ed}}{2} \log \frac{N}{n}$	we have 4 images that other. Thus, we expect sing the TF-IDF formula	t are repr the mate given by	esen/ hing	ted through in algorithm to re	nage sport		
		where			"d "Y							
		• <i>n_{id}</i> - oc	curance of word / in a doc	sument (image) d;								
		 n_d total n_l numb 	number of words in a doc ser of documents (images)	ument d; in the database) that cor	tain the word	1						
		 N numb 	ber of documents (images	in the database)								
					_, i	3110	2	Ţ	1210	0		
		n_{id}	52100									
		n_d	8	6		7			4			
		n.		•								
		n_i	4	3	4	1	1					
		In the summer	ala akawa imaga A saata	ing E bits O give 1 and		allow as assessed to add	The less	ana b	an in total Q .			

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 occurances. With this information, we can compute the TF $\frac{k_{id}}{\kappa_{d}} = \frac{2}{5}$. 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

- 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/bl ob/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/pap ers/sivic03.pdf

 TF-IDF information: https://en.wikipedia.org/wiki/Tf%E2%80%93idf

- 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/bl ob/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/pap ers/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

Small Project

Task Description

 Task: Realize a visual place recognition system using BoVW

Input:

- Database of images
- Query image(s)

Output:

- The most similar 10 images to the query image
- Implementation in C++

Hints

- Read/write features in **binary** files for loading/saving the descriptor values
- Test k-means with tiny 2D examples
- k-means without FLANN will be slow
- FLANN = Fast approximate NN search
- FLANN is an approximation and it is non-deterministic (output varies)
- Dictionary size to start with: 1000
- Visualize results by writing simple html files and display them with your browser



. . .

Download:

https://uni-bonn.sciebo.de/s/c2d0a1ebbe575fdba2a35a8033f1e2ab

Freiburg dataset

- gps_info.txt (GPS w/ timestamps)
- image-timestamps.txt (image timestamps)
- imageCompressedCam0_0000000.png
- imageCompressedCam0_000NNNN.png

Data Example





1337850707.108278
1337850707.558294
1337850708.058266
1337850708.508274
1337850709.008243
1337850709.458249
1337850709.958227
1337850710.408230
1337850710.908209
1337850711.358219
1337850711.858189
1337850712.358196

-0.103	GPS RAW	005	time=1337850706.695046.utctime=[3x1]
{9.11.23}.lat=48	3.014315.lon=7.832350.	gual=1.sats=8.h	dop=1.3
0.798	GPS RAW	aps	time=1337850707.596694.utctime=[3x1]
{9.11.24}.lat=48	3.014315.lon=7.832350.	gual=1.sats=8.h	dop=1.3
1.700	GPS RAW	qps	time=1337850708.498274.utctime=[3x1]
{9,11,25},lat=48	3.014315, lon=7.832350,	gual=1,sats=8,h	dop=1.3
2.902	GPS_RAW	qps	time=1337850709.700428.utctime=[3x1]
{9,11,26},lat=48	3.014315,lon=7.832350,	qual=1,sats=8,h	dop=1.3
3.704	GPS RAW	gps	time=1337850710.502017,utctime=[3x1]
{9,11,27},lat=48	3.014315,lon=7.832350,	qual=1,sats=8,h	dop=1.3
4.506	GPS_RAW	gps	time=1337850711.303786,utctime=[3x1]
{9,11,28},lat=48	3.014315,lon=7.832350,	qual=1,sats=8,h	dop=1.3
5.908	GPS_RAW	gps	time=1337850712.706173,utctime=[3x1]
{9,11,29},lat=48	.014315,lon=7.832350,	qual=1,sats=8,h	dop=1.3
6.710	GPS_RAW	gps	time=1337850713.507768,utctime=[3x1]
{9,11,30},lat=48	.014315,lon=7.832350,	qual=1,sats=8,h	dop=1.3
7.514	GPS_RAW	gps	time=1337850714.312165,utctime=[3x1]
{9,11,31},lat=48	3.014315,lon=7.832350,	qual=1,sats=8,h	dop=1.3
8.920	GPS_RAW	gps	time=1337850715.718731,utctime=[3x1]
{9,11,32},lat=48	3.014315,lon=7.832350,	qual=1,sats=8,h	dop=1.3
9.723	GPS_RAW	gps	time=1337850716.520946,utctime=[3x1]
{9,11,33},lat=48	3.014315,lon=7.832350,	qual=1,sats=8,h	dop=1.3
10.524	GPS_RAW	gps	time=1337850717.322499,utctime=[3x1]
{9,11,34},lat=48	3.014315,lon=7.832350,	qual=1,sats=8,h	dop=1.3
11.927	GPS_RAW	gps	time=1337850718.724799,utctime=[3x1]
{9,11,35},lat=48	3.014315,lon=7.832350,	qual=1,sats=8,h	dop=1.3
C 2 2 2 2	CDC DUU		

Next Steps

- 1. Read the Jupyter notebook by Vysotska
- 2. Read "Video Google: A Text Retrieval Approach to Object Matching in Videos" by Sivic and Zisserman
- 3. Identify the key components to implemennt
- 4. Identify dependencies as well as inputs and outputs between components
- 5. Create a schedule and assign tasks
- 6. Go!

Rules

- Team work in teams of two students
- Code all components yourself

Two exceptions:

1. Use OpenCV only for loading/displaying images and for extracting SIFT features

 If your nearest neighbor queries are too slow, use approximate NN techniques (FLANN - Fast Approximate Nearest Neighbor Search in OpenCV 2.4+)