Computer Vision - Lecture 15

Indexing and Visual Vocabularies

18.12.2014

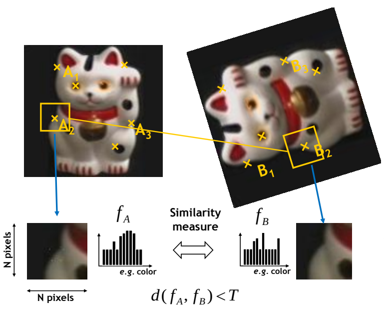
Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de

leibe@vision.rwth-aachen.de

Course Outline

- Image Processing Basics
- Segmentation & Grouping
- Object Recognition
- Object Categorization I
  - Sliding Window based Object Detection
- Local Features & Matching
  - Local Features - Detection and Description
  - Recognition with Local Features
  - Indexing & Visual Vocabularies
- Object Categorization II
  - Bag-of-Words Approaches & Part-based Approaches
- 3D Reconstruction

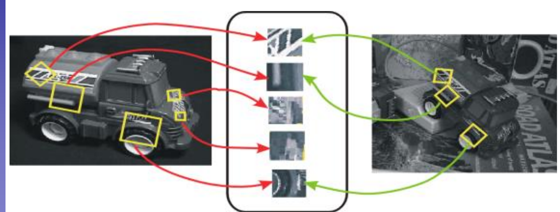
Recap: Local Feature Matching Outline



1. Find a set of distinctive key-points
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors

Recap: Recognition with Local Features

- Image content is transformed into local features that are invariant to translation, rotation, and scale
- Goal: Verify if they belong to a consistent configuration

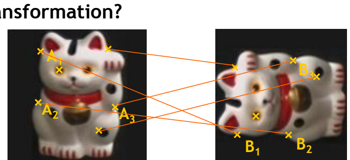


Local Features, e.g. SIFT

Slide credit: David Lowe

Recap: Fitting an Affine Transformation

- Assuming we know the correspondences, how do we get the transformation?



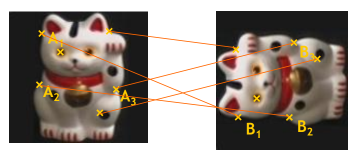
(x_i, y_i) (x_i', y_i')

$$\begin{bmatrix}x_i'\\y_i'\end{bmatrix}=\begin{bmatrix}m_1&m_2\\m_3&m_4\end{bmatrix}\begin{bmatrix}x_i\\y_i\end{bmatrix}+\begin{bmatrix}t_1\\t_2\end{bmatrix}$$

$$\begin{bmatrix} & & & \cdots & & \\ x_i & y_i & 0 & 0 & 1 & 0\\ 0 & 0 & x_i & y_i & 0 & 1\\ & & & \cdots & & \end{bmatrix}\begin{bmatrix}m_1\\m_2\\m_3\\m_4\\t_1\\t_2\end{bmatrix}=\begin{bmatrix}\cdots\\x_i'\\y_i'\\\cdots\end{bmatrix}$$

Recap: Fitting a Homography

- Estimating the transformation



$\mathbf{x}_{A_1} \leftrightarrow \mathbf{x}_{B_1}$
$\mathbf{x}_{A_2} \leftrightarrow \mathbf{x}_{B_2}$
$\mathbf{x}_{A_3} \leftrightarrow \mathbf{x}_{B_3}$
$\vdots$

Homogenous coordinates

$$\begin{bmatrix}x'\\y'\\z'\end{bmatrix}=\begin{bmatrix}h_{11}&h_{12}&h_{13}\\h_{21}&h_{22}&h_{23}\\h_{31}&h_{32}&1\end{bmatrix}\begin{bmatrix}x\\y\\1\end{bmatrix}$$

Image coordinates

$$\begin{bmatrix}x''\\y''\\1\end{bmatrix}=\frac{1}{z'}\begin{bmatrix}x'\\y'\\z'\end{bmatrix}$$

Matrix notation

$$x' = Hx$$

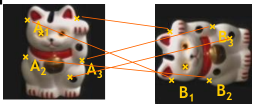
$$x'' = \frac{1}{z'}x'$$

$$x_{A_1} = \frac{h_{11}x_{B_1}+h_{12}y_{B_1}+h_{13}}{h_{31}x_{B_1}+h_{32}y_{B_1}+1} \qquad y_{A_1} = \frac{h_{21}x_{B_1}+h_{22}y_{B_1}+h_{23}}{h_{31}x_{B_1}+h_{32}y_{B_1}+1}$$

Slide credit: Krystian Mikolajczyk

Recap: Fitting a Homography

- **Estimating the transformation**



$$\boxed{\begin{aligned} h_{11}x_{B_1} + h_{12}y_{B_1} + h_{13} - x_{A_1}h_{31}x_{B_1} - x_{A_1}h_{32}y_{B_1} - x_{A_1} &= 0 \\ h_{21}x_{B_1} + h_{22}y_{B_1} + h_{23} - y_{A_1}h_{31}x_{B_1} - y_{A_1}h_{32}y_{B_1} - y_{A_1} &= 0 \end{aligned}}$$

$\mathbf{x}_{A_1} \leftrightarrow \mathbf{x}_{B_1}$
$\mathbf{x}_{A_2} \leftrightarrow \mathbf{x}_{B_2}$
$\mathbf{x}_{A_3} \leftrightarrow \mathbf{x}_{B_3}$
$\vdots$

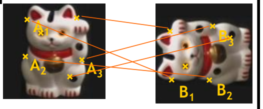
$$\begin{bmatrix} x_{B_1} & y_{B_1} & 1 & 0 & 0 & 0 & -x_{A_1}x_{B_1} & -x_{A_1}y_{B_1} & -x_{A_1} \\ 0 & 0 & 0 & x_{B_1} & y_{B_1} & 1 & -y_{A_1}x_{B_1} & -y_{A_1}y_{B_1} & -y_{A_1} \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \end{bmatrix} \cdot \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ . \end{bmatrix}$$

$$Ah = 0$$

Slide credit: Krystian Mikolajczyk

Recap: Fitting a Homography

- **Estimating the transformation**
- **Solution:**
  - Null-space vector of A
  - Corresponds to smallest eigenvector



$$Ah = 0$$

$\mathbf{x}_{A_1} \leftrightarrow \mathbf{x}_{B_1}$
$\mathbf{x}_{A_2} \leftrightarrow \mathbf{x}_{B_2}$
$\mathbf{x}_{A_3} \leftrightarrow \mathbf{x}_{B_3}$
$\vdots$

SVD

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T = \mathbf{U}\begin{bmatrix} d_{11} & \cdots & d_{19} \\ \vdots & \ddots & \vdots \\ d_{91} & \cdots & d_{99} \end{bmatrix}\begin{bmatrix} v_{11} & \cdots & \boxed{v_{19}} \\ \vdots & \ddots & \vdots \\ v_{91} & \cdots & v_{99} \end{bmatrix}^T$$

$$\boxed{\mathbf{h} = \frac{[v_{19}, \cdots, v_{99}]}{v_{99}}}$$

Minimizes least square error

Slide credit: Krystian Mikolajczyk

Recap: Object Recognition by Alignment

- **Assumption**
  - Known object, rigid transformation compared to model image
  - ⇒ *If we can find evidence for such a transformation, we have recognized the object.*
- **You learned methods for**
  - Fitting an ***affine transformation*** from ≥ 3 correspondences
  - Fitting a ***homography*** from ≥ 4 correspondences

Affine: solve a system $At = b$

Homography: solve a system $Ah = 0$

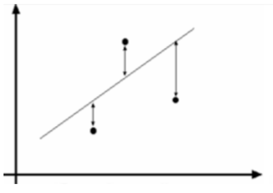
- **Correspondences may be noisy and may contain outliers**
  - ⇒ Need to use robust methods that can filter out outliers

Topics of This Lecture

- **Dealing with Outliers**
  - RANSAC
  - Generalized Hough Transform
- **Indexing with Local Features**
  - Inverted file index
  - Visual Vocabularies
- **Bag-of-Words Model**
  - Use for image classification

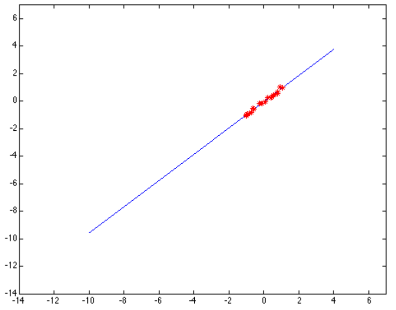
Example: Least-Squares Line Fitting

- **Assuming all the points that belong to a particular line are known**



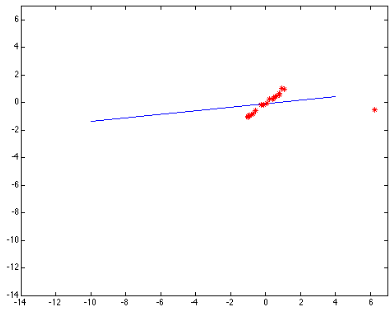
Source: Forsyth & Ponce

Outliers Affect Least-Squares Fit



Source: Forsyth & Ponce

Outliers Affect Least-Squares Fit



Strategy 1: RANSAC [Fischler81]

- **RAN**dom **SA**mple **C**onsensus
- Approach: we want to avoid the impact of outliers, so let's look for "inliers", and use only those.
- Intuition: if an outlier is chosen to compute the current fit, then the resulting line won't have much support from rest of the points.

RANSAC

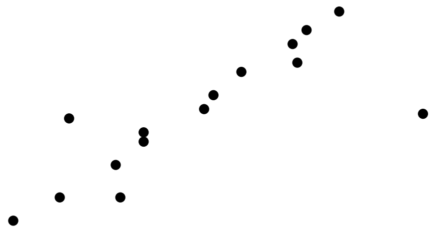
RANSAC loop:

1. Randomly select a *seed group* of points on which to base transformation estimate (e.g., a group of matches)
2. Compute transformation from seed group
3. Find *inliers* to this transformation
4. If the number of inliers is sufficiently large, re-compute least-squares estimate of transformation on all of the inliers

- Keep the transformation with the largest number of inliers

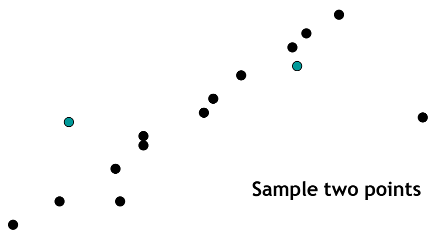
RANSAC Line Fitting Example

- Task: Estimate the best line
  - *How many points do we need to estimate the line?*



RANSAC Line Fitting Example

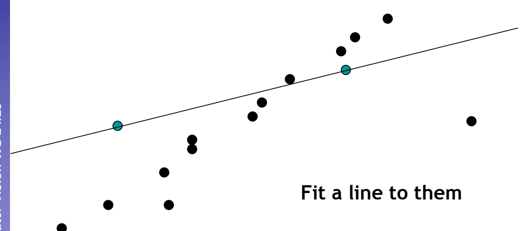
- Task: Estimate the best line



Sample two points

RANSAC Line Fitting Example

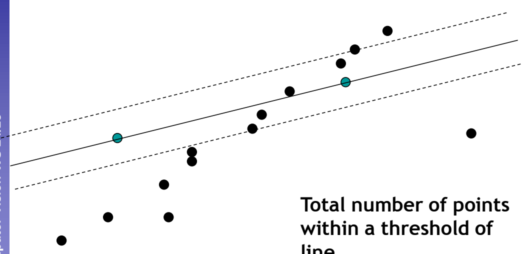
- Task: Estimate the best line



Fit a line to them

RANSAC Line Fitting Example

- Task: Estimate the best line



Total number of points within a threshold of line.

Computer Vision WS 14/15

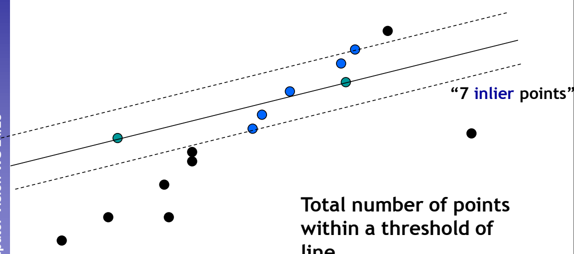
Slide credit: Jinxiang Chai

B. Leibe

21

RANSAC Line Fitting Example

- Task: Estimate the best line



Total number of points within a threshold of line.

Computer Vision WS 14/15

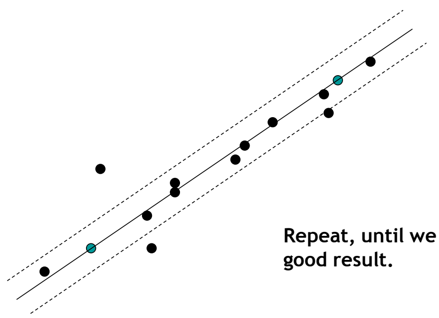
Slide credit: Jinxiang Chai

B. Leibe

22

RANSAC Line Fitting Example

- Task: Estimate the best line



Repeat, until we get a good result.

Computer Vision WS 14/15

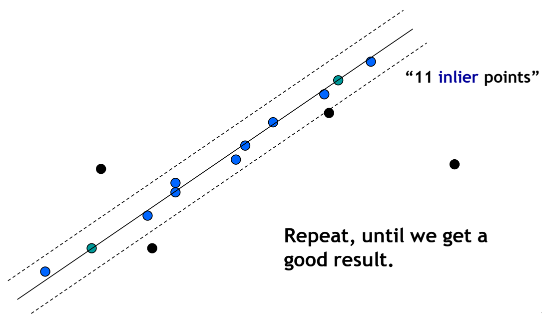
Slide credit: Jinxiang Chai

B. Leibe

23

RANSAC Line Fitting Example

- Task: Estimate the best line



Repeat, until we get a good result.

Computer Vision WS 14/15

Slide credit: Jinxiang Chai

B. Leibe

24

RANSAC: How many samples?

- How many samples are needed?
  - Suppose w is fraction of inliers (points from line).
  - n points needed to define hypothesis (2 for lines)
  - k samples chosen.
- Prob. that a single sample of n points is correct: w^n
- Prob. that all k samples fail is: $(1-w^n)^k$

⇒ Choose k high enough to keep this below desired failure rate.

Computer Vision WS 14/15

Slide credit: David Lowe

B. Leibe

25

RANSAC: Computed k (p=0.99)

Sample size	Proportion of outliers						
n	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Computer Vision WS 14/15

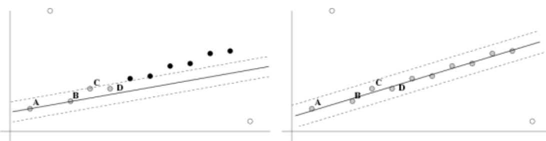
Slide credit: David Lowe

B. Leibe

26

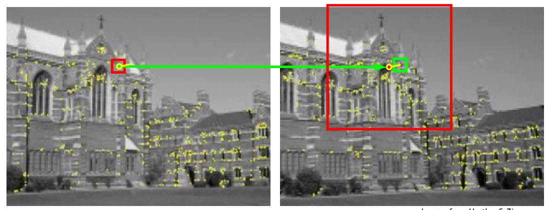
After RANSAC

- RANSAC divides data into inliers and outliers and yields estimate computed from minimal set of inliers.
- Improve this initial estimate with estimation over all inliers (e.g. with standard least-squares minimization).
- But this may change inliers, so alternate fitting with re-classification as inlier/outlier.



Example: Finding Feature Matches

- Find best stereo match within a square search window (here 300 pixels²)
- Global transformation model: epipolar geometry

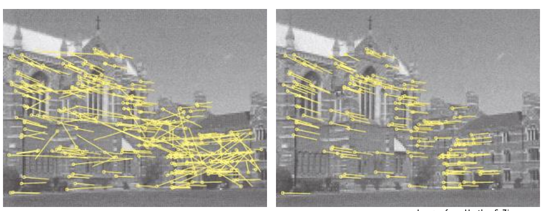


Images from Hartley & Zisserman

Example: Finding Feature Matches

- Find best stereo match within a square search window (here 300 pixels²)
- Global transformation model: epipolar geometry

before RANSAC | after RANSAC



Images from Hartley & Zisserman

Problem with RANSAC

- In many practical situations, the percentage of outliers (incorrect putative matches) is often very high (90% or above).
- Alternative strategy: Generalized Hough Transform

Strategy 2: Generalized Hough Transform

- Suppose our features are scale- and rotation-invariant
  - Then a single feature match provides an alignment hypothesis (translation, scale, orientation).

model

Strategy 2: Generalized Hough Transform

- Suppose our features are scale- and rotation-invariant
  - Then a single feature match provides an alignment hypothesis (translation, scale, orientation).
  - Of course, a hypothesis from a single match is unreliable.
  - Solution: let each match vote for its hypothesis in a Hough space with very coarse bins.

model

Pose Clustering and Verification with SIFT

- To detect instances of objects from a model base:

1. Index descriptors
   - Distinctive features narrow down possible matches

Pose Clustering and Verification with SIFT

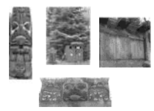
- To detect instances of objects from a model base:

1. Index descriptors
   - Distinctive features narrow down possible matches
2. Generalized Hough transform to vote for poses
   - Keypoints have record of parameters relative to model coordinate system
3. Affine fit to check for agreement between model and image features
   - Fit and verify using features from Hough bins with 3+ votes

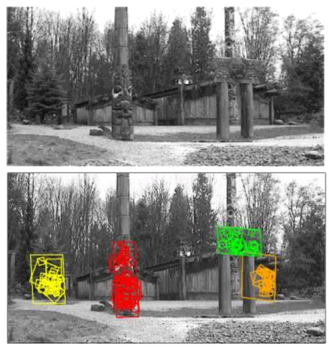
Object Recognition Results

Background subtract for model boundaries | Objects recognized | Recognition in spite of occlusion

Location Recognition



Training



[Lowe, IJCV'04]

Topics of This Lecture

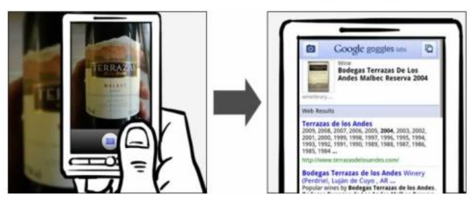
- Dealing with Outliers
  - RANSAC
  - Generalized Hough Transform
- **Indexing with Local Features**
  - Inverted file index
  - Visual Vocabularies
- Bag-of-Words Model
  - Use for image classification

Application: Mobile Visual Search

Google goggles labs

Google Goggles in Action

Click the icons below to see the different ways Google Goggles can be used.



- Take photos of objects as queries for visual search

Large-Scale Image Matching Problem

Database with thousands (millions) of images

- How can we perform this matching step efficiently?

Indexing Local Features

- Each patch / region has a descriptor, which is a point in some high-dimensional feature space (e.g., SIFT)

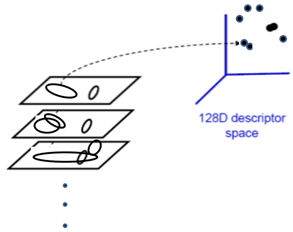
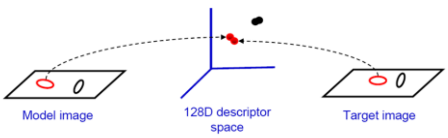


Figure credit: A. Zisserman

Indexing Local Features

- When we see close points in feature space, we have similar descriptors, which indicates similar local content.



- This is of interest for many applications
  - E.g. Image matching,
  - E.g. Retrieving images of similar objects,
  - E.g. Object recognition, categorization, 3d Reconstruction,…

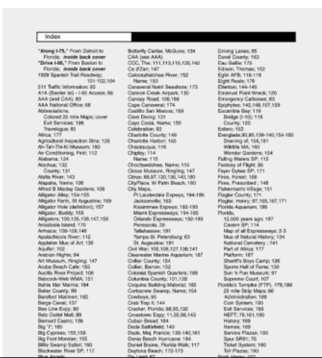
Figure credit: A. Zisserman

Indexing Local Features

- With potentially thousands of features per image, and hundreds to millions of images to search, how to efficiently find those that are relevant to a new image?
- Low-dimensional descriptors (e.g. through PCA):
  - Can use standard efficient data structures for nearest neighbor search
- High-dimensional descriptors
  - Approximate nearest neighbor search methods more practical
- Inverted file indexing schemes

Slide credit: Kristen Grauman

Indexing Local Features: Inverted File Index



- For text documents, an efficient way to find all *pages* on which a *word* occurs is to use an index…
- We want to find all *images* in which a *feature* occurs.
- To use this idea, we'll need to map our features to "visual words".

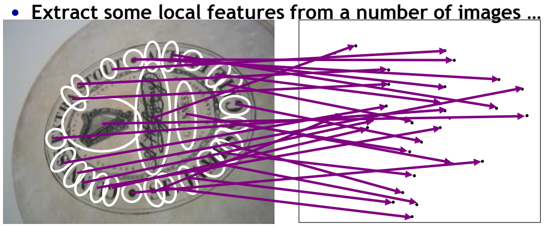
Text Retrieval vs. Image Search

- What makes the problems similar, different?

Slide credit: Kristen Grauman

Visual Words: Main Idea

- Extract some local features from a number of images …

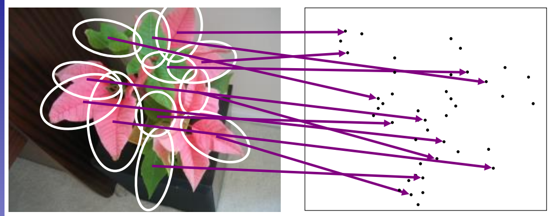


Computer Vision WS 14/15

Slide credit: David Nister

B. Leibe

Visual Words: Main Idea

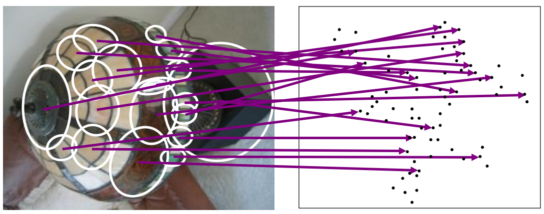


Computer Vision WS 14/15

Slide credit: David Nister

B. Leibe

Visual Words: Main Idea

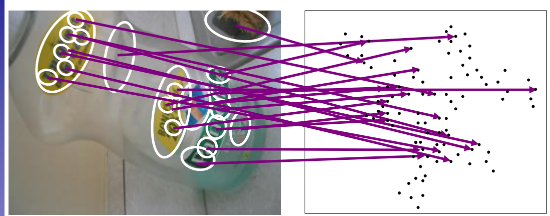


Computer Vision WS 14/15

Slide credit: David Nister

B. Leibe

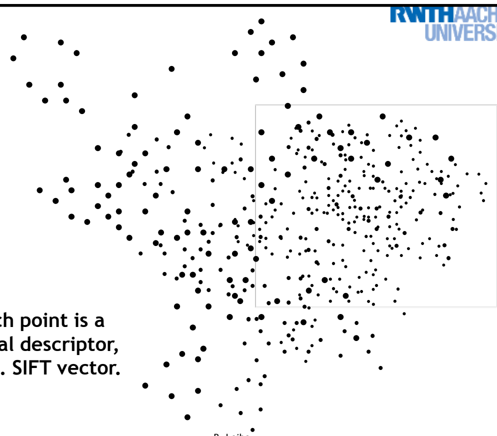
Visual Words: Main Idea



Computer Vision WS 14/15

Slide credit: David Nister

B. Leibe

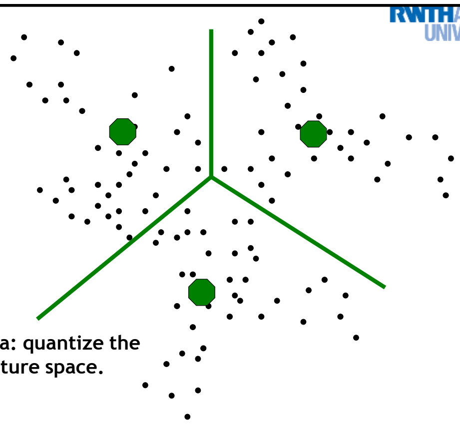


Each point is a local descriptor, e.g. SIFT vector.

Computer Vision WS 14/15

Slide credit: David Nister

B. Leibe



Idea: quantize the feature space.

Computer Vision WS 14/15

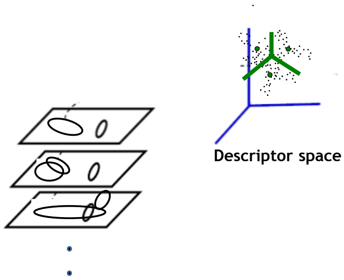
Slide credit: David Nister

B. Leibe

Indexing with Visual Words

Map high-dimensional descriptors to tokens/words by quantizing the feature space

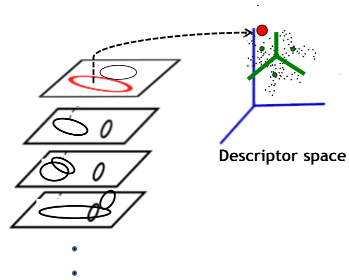
- Quantize via clustering, let cluster centers be the prototype "words"



Indexing with Visual Words

Map high-dimensional descriptors to tokens/words by quantizing the feature space

- Determine which word to assign to each new image region by finding the closest cluster center.



Visual Words

- Example: each group visual word

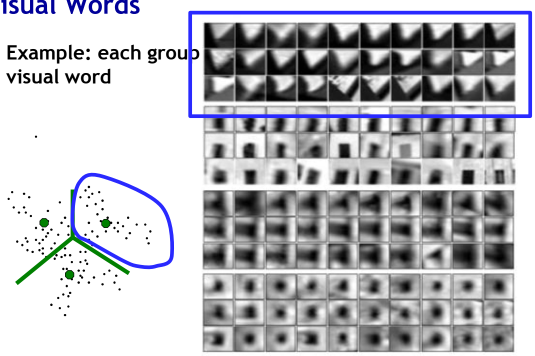
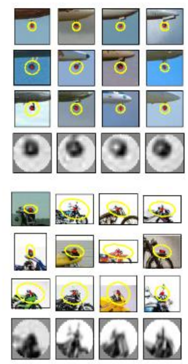


Figure from Sivic & Zisserman, ICCV 2003

Visual Words

- Often used for describing scenes and objects for the sake of indexing or classification.



Sivic & Zisserman 2003; Csurka, Bray, Dance, & Fan 2004; many others.

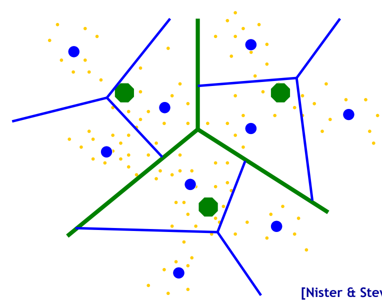
Inverted File for Images of Visual Words

Word number	List of image numbers
1	5,10, ...
2	10,...
...	...

When will this give us a significant gain in efficiency?

Example: Recognition with Vocabulary Tree

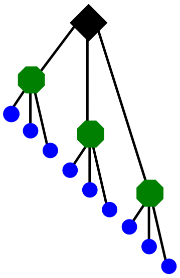
- Tree construction:



[Nister & Stewenius, CVPR'06]

Vocabulary Tree

- Training: Filling the tree

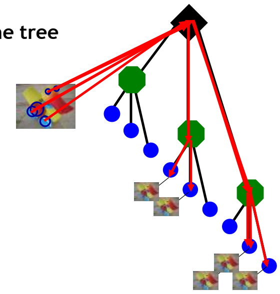


[Nister & Stewenius, CVPR'06]

RWTH AACHEN UNIVERSITY

Vocabulary Tree

- Training: Filling the tree

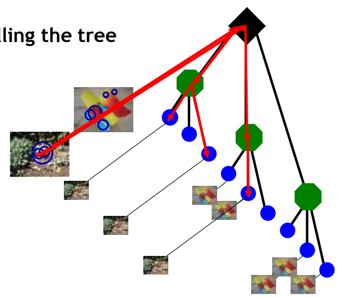


[Nister & Stewenius, CVPR'06]

RWTH AACHEN UNIVERSITY

Vocabulary Tree

- Training: Filling the tree

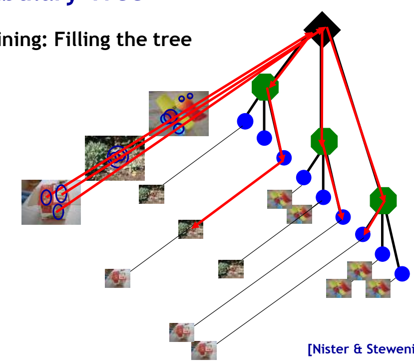


[Nister & Stewenius, CVPR'06]

RWTH AACHEN UNIVERSITY

Vocabulary Tree

- Training: Filling the tree

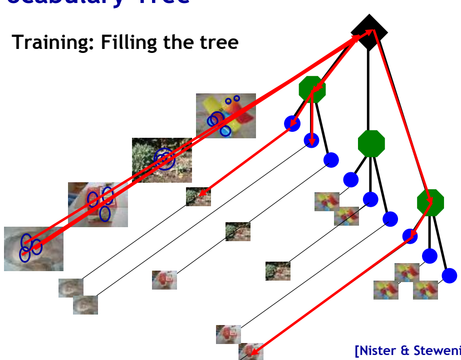


[Nister & Stewenius, CVPR'06]

RWTH AACHEN UNIVERSITY

Vocabulary Tree

- Training: Filling the tree

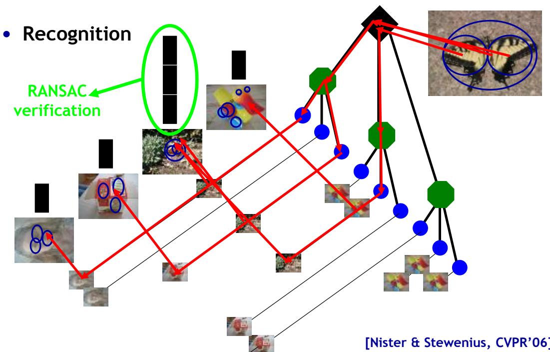


[Nister & Stewenius, CVPR'06]

RWTH AACHEN UNIVERSITY

Vocabulary Tree

- Recognition



[Nister & Stewenius, CVPR'06]

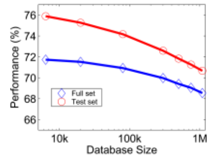
Quiz Questions

- What is the computational advantage of the hierarchical representation vs. a flat vocabulary?
- What dangers does such a representation carry?

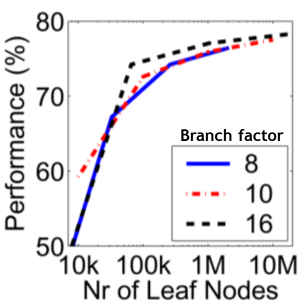
Vocabulary Tree: Performance

- Evaluated on large databases
  - Indexing with up to 1M images
- Online recognition for database of 50,000 CD covers
  - Retrieval in ~1s (in 2006)
- Experimental finding that large vocabularies can be beneficial for recognition

[Nister & Stewenius, CVPR'06]



Vocabulary Size



- Larger vocabularies can be advantageous…
- But what happens when the vocabulary gets too large?
  - Efficiency?
  - Robustness?

tf-idf Weighting

- Term frequency – inverse document frequency
- Describe frame by frequency of each word within it, downweight words that appear often in the database
- (Standard weighting for text retrieval)

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

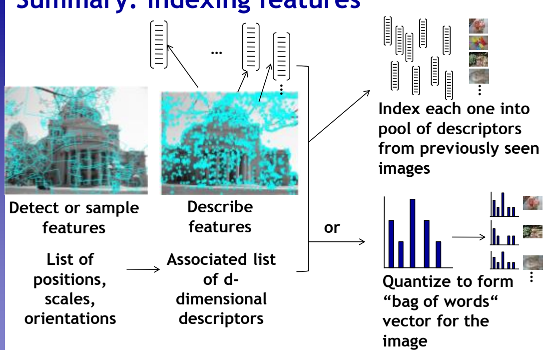
Number of occurrences of word i in document d

Number of words in document d

Total number of documents in database

Number of occurrences of word i in whole database

Summary: Indexing features



Application for Content Based Img Retrieval

- What if query of interest is a portion of a frame?

Visually defined query

"Groundhog Day" [Rammis, 1993]

"Find this clock"

"Find this place"

Video Google System

1. Collect all words within query re
2. Inverted file index to find releva
3. Compare word counts
4. Spatial verification

Sivic & Zisserman, ICCV 2003

- Demo online at : http://www.robots.ox.ac.uk/~vgg/rese

Query region

Retrieved frames

Collecting Words Within a Query Region

- Example: Friends

Query region: pull out only the SIFT descriptors whose positions are within the polygon

Example Results

Query

raw nn 1sim=0.56697 raw nn 2sim=0.56163 raw nn 5sim=0.54917

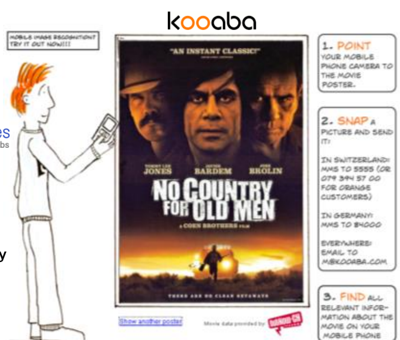
More Results

Query

Retrieved shots

Applications: Specific Object Recognition

- Commercial services coming out:
  kooaba
  Google goggles labs

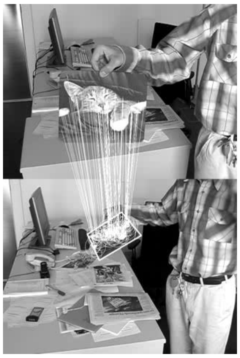


(~20M images indexed)

Works well for mostly planar objects:
- Movie posters,
- Book covers,
- CD/DVD covers,
- Video games,
- ...

Applications: Aachen Tourist Guide

Applications: Fast Image Registration



Applications: Mobile Augmented Reality

D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, D. Schmalstieg, Pose Tracking from Natural Features on Mobile Phones. In *ISMAR 2008*.

References and Further Reading

- More details on RANSAC can be found in Chapter 4.7 of
  - R. Hartley, A. Zisserman
    Multiple View Geometry in Computer Vision
    2nd Ed., Cambridge Univ. Press, 2004
- Details about the Hough transform for object recognition can be found in
  - D. Lowe, Distinctive image features from scale-invariant keypoints, *IJCV* 60(2), pp. 91-110, 2004
- Details about the Video Google system can be found in
  - *J. Sivic, A. Zisserman*, Video Google: A Text Retrieval Approach to Object Matching in Videos, ICCV'03, 2003.