Scalable Recognition with a Vocabulary Tree

Srikumar Ramalingam

School of Computing
University of Utah

Presentation Outline

Main paper to be discussed

- David Nister and Henrik Stewenius,Scalable Recognition with a Vocabulary Tree, CVPR 2006.

Matching Local Features

Image 1

Image 2

- To generate candidate matches, find patches that have the most similar appearance (e.g., lowest SSD)
- Simplest approach: compare them all, take the closest (or closest k, or within a thresholded distance)

[0]Source: Kristen Grauman

Matching Local Features

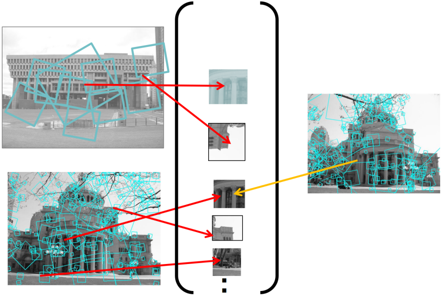

Image 1


Image 2

- In stereo case, may constrain by proximity if we make assumptions on max disparities.

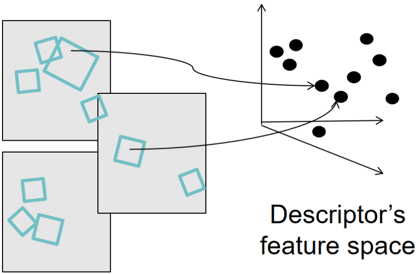
[0]Source: Kristen Grauman

Indexing local features



[0] Source: Kristen Grauman

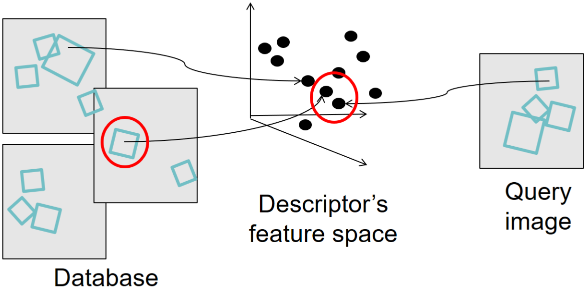
Indexing local features



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

[0]Source: Kristen Grauman

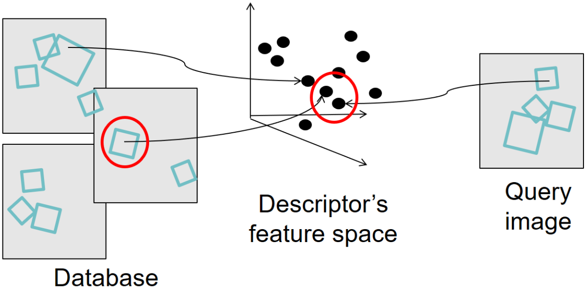
Indexing local features



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

[0] Source: Kristen Grauman

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?

0 Source: Kristen Grauman

Problem Statement

Scalable Recognition with a Vocabulary Tree

- An image matching scheme that scales efficiently to a large number of objects is presented.
- Robust indexing of local image descriptors with respect to background clutter and occlusion.
- The local region descriptors are hierarchically quantized in a vocabulary tree.

Presentation Outline

Bag of features

Collection of features or parts reveal the underlying object.

Bag of features

Collection of features or parts reveal the underlying object.

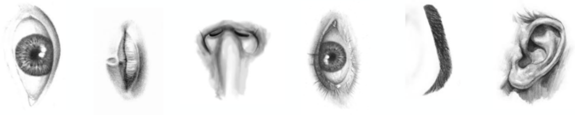
Bag of features

Collection of features or parts reveal the underlying object.

Bag of features

[0] Source: Kris Kitani

Bag of features



spatial information of local features
can be ignored for object recognition (i.e., verification)

[0] Source: Kris Kitani

Bag of features

CalTech6 dataset

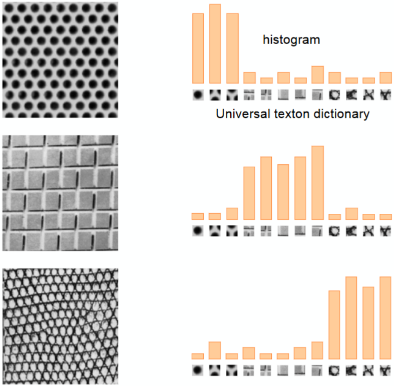
class	bag of features Zhang et al. (2005)	bag of features Willamowski et al. (2004)	Parts-and-shape model Fergus et al. (2003)
airplanes	**98.8**	97.1	90.2
cars (rear)	98.3	**98.6**	90.3
cars (side)	**95.0**	87.3	88.5
faces	**100**	99.3	96.4
motorbikes	**98.5**	98.0	92.5
spotted cats	**97.0**	—	90.0

Works pretty well for image-level classification

Csurka et al. (2004), Willamowski et al. (2005), Grauman & Darrell (2005), Sivic et al. (2003, 2005)

[0]Source: Kris Kitani

Bag of features: texture classification



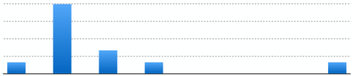
Julesz, 1981
Mori, Belongie and Malik, 2001

[0]Source: Kris Kitani

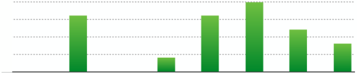
Bag of features

Vector Space Model

G. Salton. 'Mathematics and Information Retrieval' Journal of Documentation, 1979



1	6	2	1	0	0	0	1
Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor



0	4	0	1	4	5	3	2
Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor

http://www.fodey.com/generators/newspaper/snippet.asp

[0]Source: Kris Kitani

Bag of features

A document (datapoint) is a vector of counts over each word (feature)

$$\boldsymbol{v}_d = [n(w_{1,d}) \;\; n(w_{2,d}) \;\; \cdots \;\; n(w_{T,d})]$$

$n(\cdot)$ counts the number of occurrences

just a histogram over words

What is the similarity between two documents?

[0] Source: Kris Kitani

Bag of features

A document (datapoint) is a vector of counts over each word (feature)

$$\boldsymbol{v}_d = [n(w_{1,d}) \quad n(w_{2,d}) \quad \cdots \quad n(w_{T,d})]$$

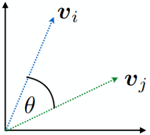
$n(\cdot)$ counts the number of occurrences

just a histogram over words

What is the similarity between two documents?

Use any distance you want but the cosine distance is fast.

$$d(\boldsymbol{v}_i, \boldsymbol{v}_j) = \cos\theta = \frac{\boldsymbol{v}_i \cdot \boldsymbol{v}_j}{\|\boldsymbol{v}_i\|\|\boldsymbol{v}_j\|}$$



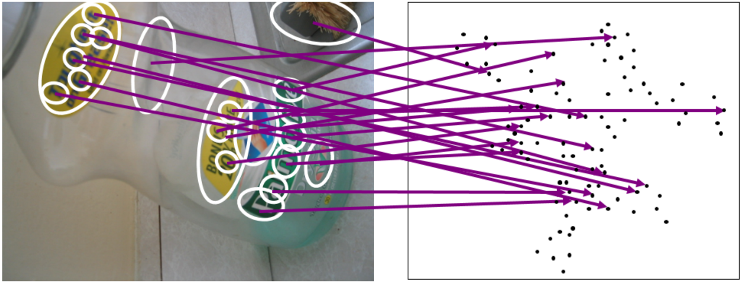
[0] Source: Kris Kitani

Text Retrieval vs. Image Search

- What makes the two problems different?

[0] Source: Kristen Grauman

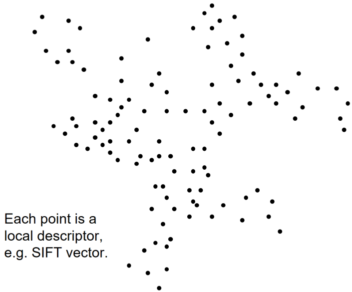
Visual Words: Main Idea



---

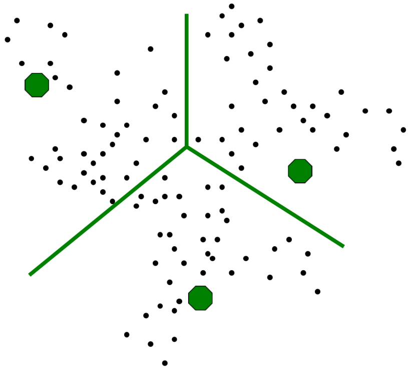
[0] Source: David Nister

Visual Words: Main Idea



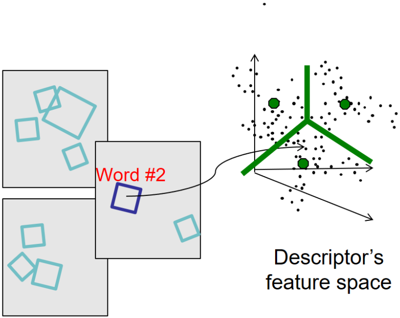
[0] Source: David Nister

Visual Words: Main Idea



[0] Source: David Nister

Visual Words: Main Idea

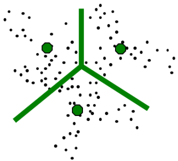


- Quantize via clustering, let cluster centers be the prototype "words"
- Determine which word to assign to each new image region by finding the closest cluster center.

[0] Source: David Nister

Visual Words: Main Idea

- Example: each group of patches belongs to the same visual word



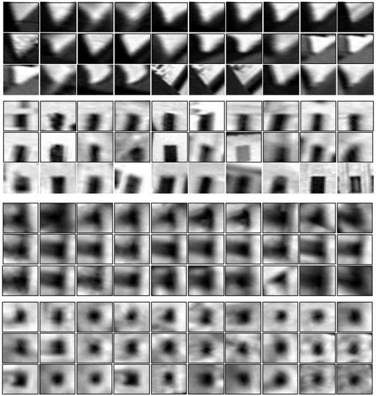
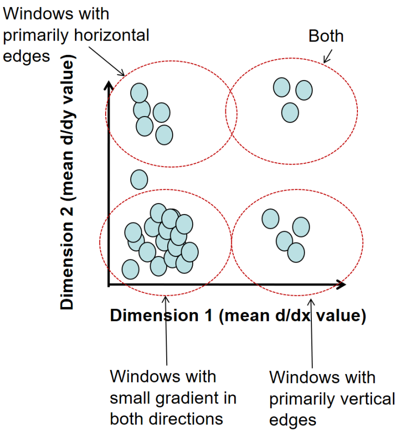


Figure from Sivic & Zisserman, ICCV 2003

[0] Source: Kristen Grauman

Recall: Texture representation example



	mean d/dx value	mean d/dy value
Win. #1	4	10
Win.#2	18	7
⋮		
Win.#9	20	20

⋮

statistics to summarize patterns in small windows

[0] Source: Kristen Grauman

Visual Vocabulary Information

- Sampling strategy: where to extract features?
- Clustering / quantization algorithm
- What corpus provides features (universal vocabulary?)
- Vocabulary size, number of words

[0] Source: Kristen Grauman

Inverted file index

Word #	Image #
1	3
2 …	
7	1, 2
8	3
9	
10 …	
91	2

…

- Database images are loaded into the index mapping words to image numbers

[0] Source: Kristen Grauman

Inverted file index

New query image

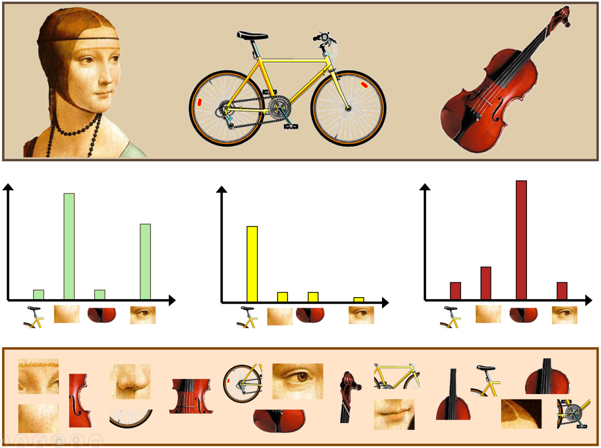
Word #	Image #
1	3
2 ...	
7	1, 2
8	3
9	
10 ...	
91	2

...

- New query image is mapped to indices of database images that share a word.

[0] Source: Kristen Grauman

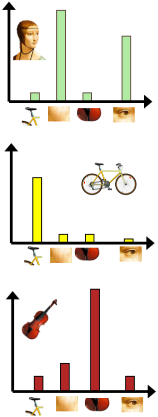
Visual Words: Main Idea



[0]Source: Kristen Grauman

Visual Words: Main Idea

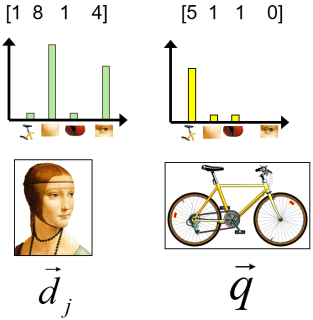
- Summarize entire image based on its distribution (histogram) of word occurrences.
- Analogous to bag of words representation commonly used for documents.



[0]Source: Kristen Grauman

Comparing bag of words

- Rank frames by normalized scalar product between their (possibly weighted) occurrence counts---*nearest neighbor* search for similar images.



$$sim(d_j, q) = \frac{\langle d_j, q \rangle}{\|d_j\| \|q\|}$$

$$= \frac{\sum_{i=1}^{V} d_j(i) * q(i)}{\sqrt{\sum_{i=1}^{V} d_j(i)^2} * \sqrt{\sum_{i=1}^{V} q(i)^2}}$$

for vocabulary of V words

[0] Source: Kristen Grauman

tf-idf weighting

- **T**erm **f**requency – **i**nverse **d**ocument **f**requency
- 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 → n_{id}

Number of words in document d → n_d

Total number of documents in database → N

Number of documen[ts] word i occurs in, in whole database → n_i

[0] Source: Kristen Grauman

Visual Words: Main Idea

Visually defined query

"Groundhog Day" [Rammis, 1993]

"Find this clock"

"Find this place"

Slide from Andrew Zisserman
Sivic & Zisserman, ICCV 2003

Visual Words: Main Idea

Example

retrieved shots

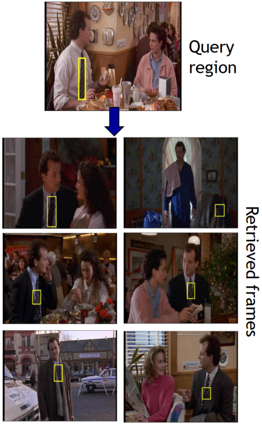
Visual Words: Main Idea

Video Google System

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

Sivic & Zisserman, ICCV 2003

- **Demo online at :** http://www.robots.ox.ac.uk/~vgg/research/vgoogle/index.html



K. Grauman, B. Leibe

Image Retrieval Performance

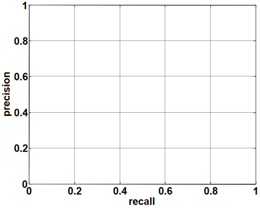
Scoring retrieval quality

Query

Database size: 10 images
Relevant (total): 5 images

Results (ordered):

precision = #relevant / #returned
recall = #relevant / #total relevant



Slide credit: Ondrej Chum

Image Retrieval Performance

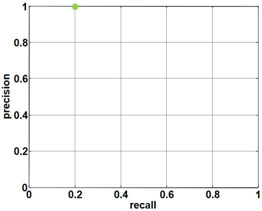
Scoring retrieval quality

Query

Database size: 10 images
Relevant (total): 5 images

Results (ordered):

precision = #relevant / #returned
recall = #relevant / #total relevant



Slide credit: Ondrej Chum

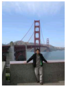
Image Retrieval Performance

Scoring retrieval quality

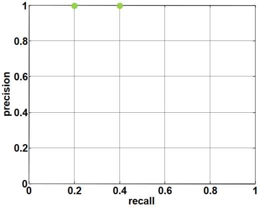
Query

Database size: 10 images
Relevant (total): 5 images

Results (ordered):



precision = #relevant / #returned
recall = #relevant / #total relevant



Slide credit: Ondrej Chum

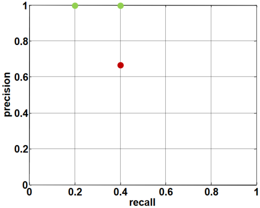
Image Retrieval Performance

Scoring retrieval quality


Query

Database size: 10 images
Relevant (total): 5 images

precision = #relevant / #returned
recall = #relevant / #total relevant



Results (ordered):

Slide credit: Ondrej Chum

Image Retrieval Performance

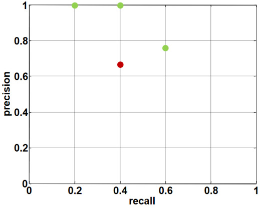
Scoring retrieval quality

Query

Database size: 10 images
Relevant (total): 5 images

Results (ordered):

precision = #relevant / #returned
recall = #relevant / #total relevant



Slide credit: Ondrej Chum

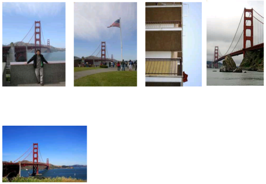
Image Retrieval Performance

Scoring retrieval quality

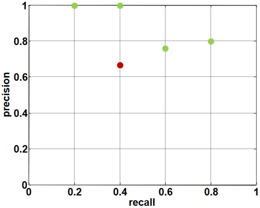
Query

Database size: 10 images
Relevant (total): 5 images

Results (ordered):



precision = #relevant / #returned
recall = #relevant / #total relevant



Slide credit: Ondrej Chum

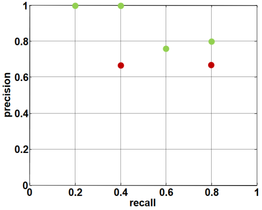
Image Retrieval Performance

Scoring retrieval quality

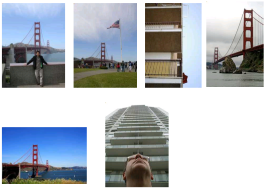
Query

Database size: 10 images
Relevant (total): 5 images

precision = #relevant / #returned
recall = #relevant / #total relevant



Results (ordered):



Slide credit: Ondrej Chum

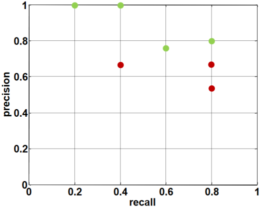
Image Retrieval Performance

Scoring retrieval quality

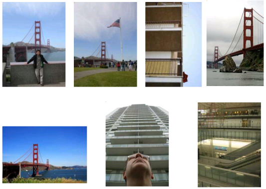
Query

Database size: 10 images
Relevant (total): 5 images

precision = #relevant / #returned
recall = #relevant / #total relevant



Results (ordered):



Slide credit: Ondrej Chum

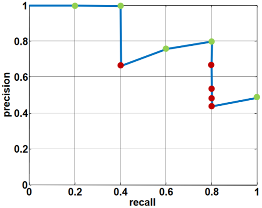
Image Retrieval Performance

Scoring retrieval quality

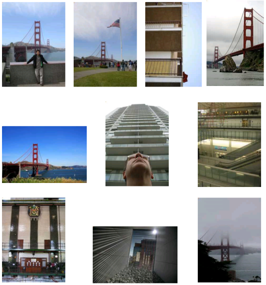
Query

Database size: 10 images
Relevant (total): 5 images

precision = #relevant / #returned
recall = #relevant / #total relevant



Results (ordered):



Slide credit: Ondrej Chum

Standard Bag of Words Pipeline

1. Extract features
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary
4. Represent images by frequencies of “visual words”

[0] Source: Kris Kitani

Standard Bag of Words Pipeline

1. **Extract features**
2. Learn "visual vocabulary"
3. Quantize features using visual vocabulary
4. Represent images by frequencies of "visual words"

[0]Source: Kris Kitani

Standard Bag of Words Pipeline

Scalable Recognition with a Vocabulary Tree

Srikumar Ramalingam

Problem Statement

Bag of Features

Building the Vocabulary Tree

1. Extract features
2. **Learn "visual vocabulary"**

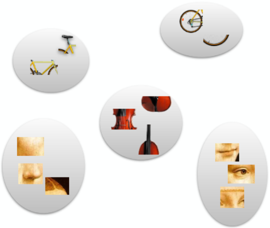
3. Quantize features using visual vocabulary
4. Represent images by frequencies of "visual words"

---

[0] Source: Kris Kitani

Standard Bag of Words Pipeline

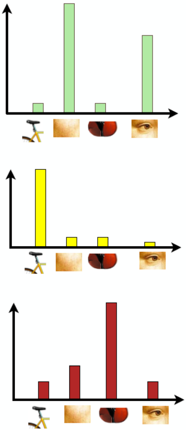
1. Extract features
2. Learn "visual vocabulary"
3. **Quantize features using visual vocabulary**
4. Represent images by frequencies of "visual words"



[0]Source: Kris Kitani

Standard Bag of Words Pipeline

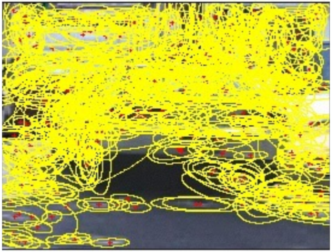
1. Extract features
2. Learn "visual vocabulary"
3. Quantize features using visual vocabulary
4. **Represent images by frequencies of "visual words"**



[0]Source: Kris Kitani

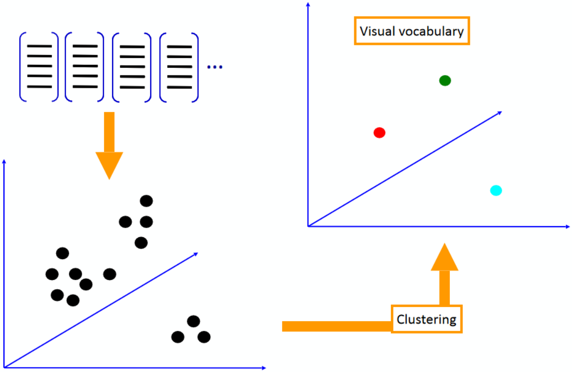
Feature Extraction

- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
- Interest point detector
  - Csurka et al. 2004
  - Fei-Fei & Perona, 2005
  - Sivic et al. 2005
- Other methods
  - Random sampling (Vidal-Naquet & Ullman, 2002)
  - Segmentation-based patches (Barnard et al. 2003)



[0]Source: Kris Kitani

Visual Vocabulary



[0]Source: Kris Kitani

K-Means Clustering

K-means Clustering

Given k:

1. Select initial centroids at random.

2. Assign each object to the cluster with the nearest centroid.

3. Compute each centroid as the mean of the objects assigned to it.

4. Repeat previous 2 steps until no change.

[0] Source: Kris Kitani

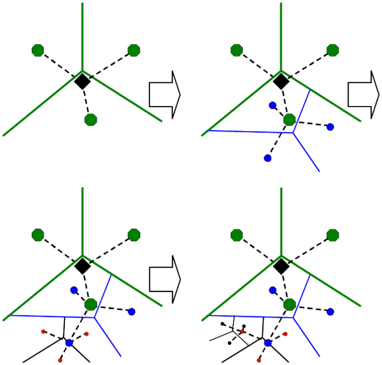
Clustering and Vector Quantization

- Clustering is a common method for learning a visual vocabulary or codebook
  - Unsupervised learning process
  - Each cluster center produced by k-means becomes a codevector
  - Codebook can be learned on separate training set
  - Provided the training set is sufficiently representative, the codebook will be "universal"
- The codebook is used for quantizing features
  - A *vector quantizer* takes a feature vector and maps it to the index of the nearest codevector in a codebook
  - Codebook = visual vocabulary
  - Codevector = visual word

[0] Source: Kris Kitani

Presentation Outline

Vocabulary Tree



- An illustration of the process of building the vocabulary tree. The hierarchical quantization is defined at each level by k centers (in this case k = 3) and their Voronoi regions.

Building the Vocabulary Tree

- The vocabulary tree is built by hierarchical k-means clustering.
- Descriptor vectors are used in the unsupervised training.
- First, an initial k-means process is run to define k cluster centers.
- The training data is then partitioned into k groups, where each group consists of the descriptor vectors closest to a particular cluster center.
- The same process is then recursively applied to each group of descriptor vectors, recursively defining quantization cells by splitting each quantization cell into k new parts.

Vocabulary Tree

Scalable Recognition with a Vocabulary Tree

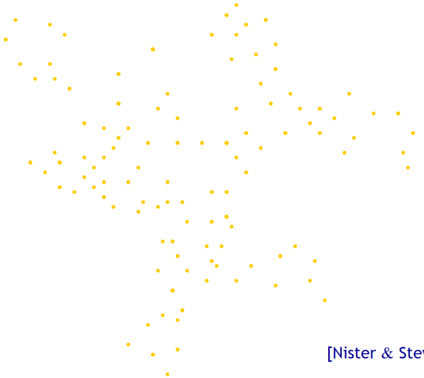
Srikumar Ramalingam

Problem Statement

Bag of Features

Building the Vocabulary Tree

- **Tree construction:**

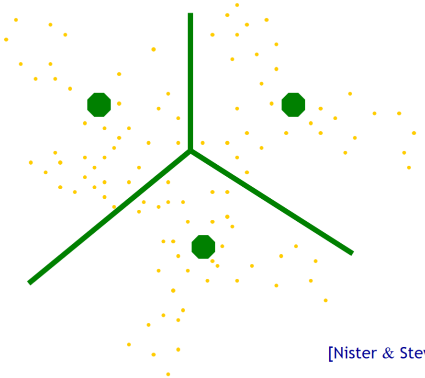


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

- **Tree construction:**

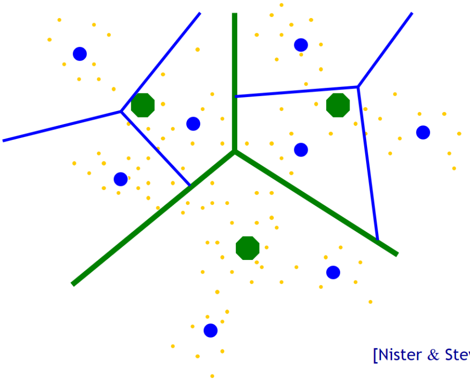


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

- **Tree construction:**

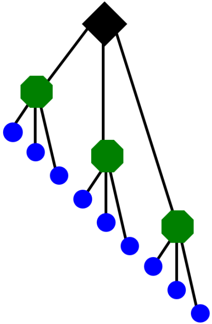


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

- **Training: Filling the tree**



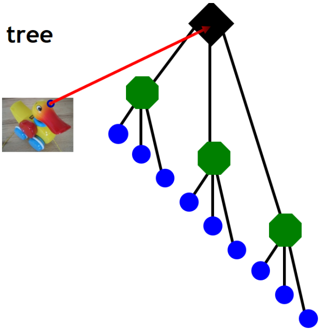
[Nister & Stewenius, CVPR'06]

K. Grauman, B. Leibe

Slide credit: David Nister

Vocabulary Tree

- **Training: Filling the tree**



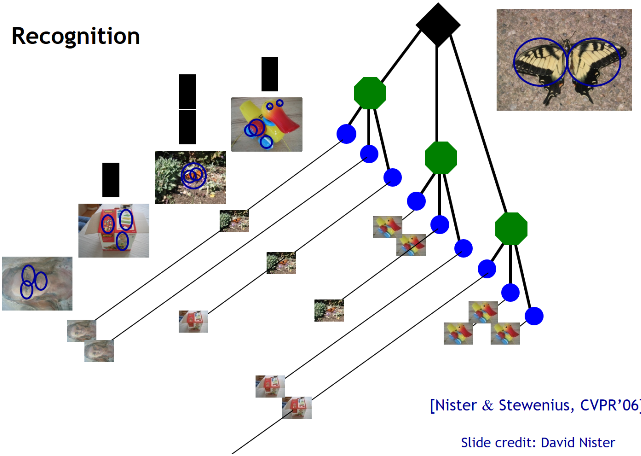
[Nister & Stewenius, CVPR'06]

K. Grauman, B. Leibe

Slide credit: David Nister

Vocabulary Tree

- **Recognition**

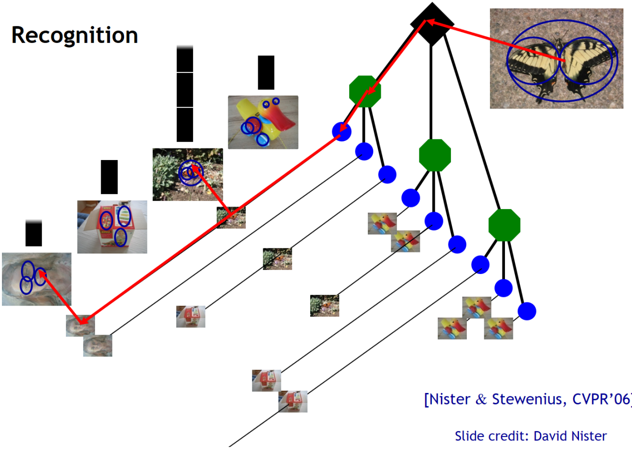


[Nister & Stewenius, CVPR’06]

Slide credit: David Nister

Vocabulary Tree

- **Recognition**

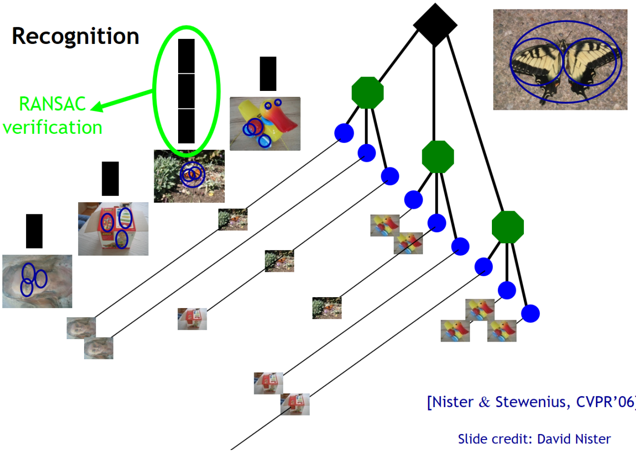


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

- **Recognition**



[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

$$q_i = n_i w_i$$ Query vector elements

$$d_i = m_i w_i$$ Database vector elements

Distance between query and the database vector:

$$s(q,d) = \left\| \frac{q}{\| q \|} - \frac{d}{\| d \|} \right\|$$

Acknowledgments

Some presentation slides are adapted from David Lowe's landmark paper, Kristen Grauman, Andrew Zisserman, Joseph Sivic, wikipedia.org, and Utkarsh Sinha (aishack.in)