Image Matching

Srikumar Ramalingam

School of Computing
University of Utah
Presentation Outline

1. Problem Statement
2. Bag of Features
3. Building the Vocabulary Tree
Main paper to be discussed

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

Source: Kristen Grauman
Matching Local Features

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)

Source: Kristen Grauman
In stereo case, may constrain by proximity if we make assumptions on max disparities.

Source: Kristen Grauman
Indexing local features

Source: Kristen Grauman
Indexing local features

0Source: Kristen Grauman
Indexing local features

Source: Kristen Grauman
Indexing local features

Source: Kristen Grauman
Indexing local features

Source: Kristen Grauman
Indexing local features

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)

Source: Kristen Grauman
When we see close points in feature space, we have similar descriptors, which indicates similar local content.

Source: Kristen Grauman
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?

Source: Kristen Grauman
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

1. Problem Statement

2. Bag of Features

3. Building the Vocabulary Tree
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

Some local features are very informative

An object as

A collection of local features

- deals well with occlusion
- scale invariant
- rotation invariant

Source: Kris Kitani
spatial information of local features can be ignored for object recognition (i.e., verification)
Bag of features

CalTech6 dataset

<table>
<thead>
<tr>
<th>class</th>
<th>bag of features</th>
<th>bag of features</th>
<th>Parts-and-shape model</th>
</tr>
</thead>
<tbody>
<tr>
<td>airplanes</td>
<td>98.8</td>
<td>97.1</td>
<td>90.2</td>
</tr>
<tr>
<td>cars (rear)</td>
<td>98.3</td>
<td>98.6</td>
<td>90.3</td>
</tr>
<tr>
<td>cars (side)</td>
<td>95.0</td>
<td>87.3</td>
<td>88.5</td>
</tr>
<tr>
<td>faces</td>
<td>100</td>
<td>99.3</td>
<td>96.4</td>
</tr>
<tr>
<td>motorbikes</td>
<td>98.5</td>
<td>98.0</td>
<td>92.5</td>
</tr>
<tr>
<td>spotted cats</td>
<td>97.0</td>
<td>—</td>
<td>90.0</td>
</tr>
</tbody>
</table>

Works pretty well for image-level classification

Csurka et al. (2004), Willamowski et al. (2005), Grauman & Darrell (2005), Sivic et al. (2003, 2005)
Bag of features: texture classification

Source: Kris Kitani
Vector Space Model


- Bag of features
- Building the Vocabulary Tree

Source: Kris Kitani
A document (datapoint) is a vector of counts over each word (feature)

\[ \mathbf{v}_d = [n(w_1,d), n(w_2,d), \ldots, n(w_T,d)] \]

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

What is the similarity between two documents?

Source: Kris Kitani
A document (datapoint) is a vector of counts over each word (feature)

$$v_d = [n(w_{1,d}) \ n(w_{2,d}) \ \cdots \ n(w_{T,d})]$$

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

What is the similarity between two documents?

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

$$d(v_i, v_j) = \cos \theta$$

$$= \frac{v_i \cdot v_j}{\|v_i\| \|v_j\|}$$

Source: Kris Kitani
Text Retrieval vs. Image Search

- What makes the two problems different?

0Source: Kristen Grauman
• Extract some local features from a number of images …

  e.g., SIFT descriptor space: each point is 128-dimensional

Source: David Nister
Visual Words: Main Idea

• Extract some local features from a number of images …

e.g., SIFT descriptor space: each point is 128-dimensional

Source: David Nister
Visual Words: Main Idea

- Extract some local features from a number of images ...

  e.g., SIFT descriptor space: each point is 128-dimensional

  Source: David Nister
Visual Words: Main Idea

- Extract some local features from a number of images ...

![Diagram]

e.g., SIFT descriptor space: each point is 128-dimensional

Source: David Nister
Visual Words: Main Idea

Source: David Nister
Visual Words: Main Idea

Source: David Nister
Visual Words: Main Idea

Source: David Nister
Visual Words: Main Idea

Source: David Nister
Each point is a local descriptor, e.g. SIFT vector.

Source: David Nister
Visual Words: Main Idea

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.

Source: David Nister
Visual Words: Main Idea

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

Figure from Sivic & Zisserman, ICCV 2003

Source: Kristen Grauman
Recall: Texture representation example

Source: Kristen Grauman
Visual Vocabulary Information

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

Source: Kristen Grauman
• Database images are loaded into the index mapping words to image numbers
Inverted file index

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

Source: Kristen Grauman
Visual Words: Main Idea

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

\[\text{Source: Kristen Grauman}\]
Comparing bag of words

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

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

\[
= \frac{\sum_{i=1}^{V} d_j(i) \cdot q(i)}{\sqrt{\sum_{i=1}^{V} d_j(i)^2} \cdot \sqrt{\sum_{i=1}^{V} q(i)^2}}
\]

for vocabulary of \( V \) words

Source: Kristen Grauman
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)

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

\begin{align*}
\text{Number of occurrences of word} \quad & \text{Total number of documents in database} \\
\text{in document d} \quad & \text{Number of documents in } \text{database} \\
\text{Number of words in} \quad & \text{Number of document } \text{word i occurs in, in} \\
\text{document d} \quad & \text{whole database}
\end{align*}

Source: Kristen Grauman
Visual Words: Main Idea

Visually defined query

“Find this clock”

“Find this place”

“Groundhog Day” [Rammis, 1993]

Slide from Andrew Zisserman
Sivic & Zisserman, ICCV 2003
Visual Words: Main Idea

Example

retrieved shots

Start frame 52907 | Key frame 53026 | End frame 53028

Start frame 54342 | Key frame 54376 | End frame 54644

Start frame 51770 | Key frame 52251 | End frame 52348

Start frame 54079 | Key frame 54201 | End frame 54201

Start frame 38909 | Key frame 39126 | End frame 39300

Start frame 40760 | Key frame 40826 | End frame 41049
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
Scoring retrieval quality

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

\[
\text{precision} = \frac{\#\text{relevant}}{\#\text{returned}} \\
\text{recall} = \frac{\#\text{relevant}}{\#\text{total relevant}}
\]

Results (ordered):

Slide credit: Ondrej Chum
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
Scoring retrieval quality

Query

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

Results (ordered):

\[
\text{precision} = \frac{\text{#relevant}}{\text{#returned}} \\
\text{recall} = \frac{\text{#relevant}}{\text{#total relevant}}
\]

Slide credit: Ondrej Chum
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
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
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
Scoring retrieval quality

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

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

Results (ordered):

Slide credit: Ondrej Chum
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
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”

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”

*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”

*Source: Kris Kitani*
1. Extract features

2. Learn “visual vocabulary”

3. **Quantize features using visual vocabulary**

4. Represent images by frequencies of “visual words”

---

Source: Kris Kitani
1. Extract features

2. Learn “visual vocabulary”

3. Quantize features using visual vocabulary

4. Represent images by frequencies of “visual words”

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)

Source: Kris Kitani
Feature Extraction

Problem Statement

Bag of Features

Building the Vocabulary Tree

Source: Kris Kitani

Image Matching

Srikumar Ramalingam

Feature Extraction

Detect patches

Compute SIFT descriptor

Source: Kris Kitani

Normalize patch

[Lowe'99]

[Mikojaczyk and Schmid ’02]

[Mata, Chum, Urban & Pajdla, ’02]

[Sivic & Zisserman, ’03]
Feature Extraction

0Source: Kris Kitani
Visual Vocabulary

$^0$Source: Kris Kitani
Visual Vocabulary

0Source: Kris Kitani
Visual Vocabulary

Source: Kris Kitani
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.

Source: Kris Kitani
Visual Vocabulary

Source: Kris Kitani
Visual Vocabulary

0Source: 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

*Source: Kris Kitani*
Presentation Outline

1. Problem Statement
2. Bag of Features
3. Building the 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

- 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
• 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

- 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
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
Training: Filling the tree

[Nister & Stewenius, CVPR’06]

K. Grauman, B. Leibe

Slide credit: David Nister
Vocabulary Tree

- Training: Filling the tree

K. Grauman, B. Leibe

[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
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

- 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

- Training: Filling the tree

[Nister & Stewenius, CVPR'06]

K. Grauman, B. Leibe

Slide credit: David Nister
- **Training:** Filling the tree

[Nister & Stewenius, CVPR’06]

K. Grauman, B. Leibe

Slide credit: David Nister
• **Training: Filling the tree**

[Nister & Stewenius, CVPR’06]

K. Grauman, B. Leibe
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

- 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

- Recognition

RANSAC verification

[Nister & Stewenius, CVPR’06]

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