Outline

1. Local invariant features (C. Schmid)

2. Matching and recognition with local features (J. Sivic)

3. Large scale visual search (J. Sivic)

4. Very large scale visual indexing (C. Schmid)

Practical session – Instance-level recognition and search
Outline

Efficient visual search
  Approximate nearest neighbour matching
  Bag-of-visual-words representation
  Efficient visual search and extensions
  Beyond bag-of-visual-words representations
Example: Two images again

1000+ descriptors per image
Match regions between frames using SIFT descriptors and spatial consistency

Multiple regions overcome problem of partial occlusion
Approach - review

1. Establish tentative (or putative) correspondence based on local appearance of individual features (now)

2. Verify matches based on semi-local / global geometric relations (You have just seen this).
What about multiple images?

• So far, we have seen successful matching of a query image to a single target image using local features.

• How to generalize this strategy to multiple target images with reasonable complexity?

  • $10, 10^2, 10^3, \ldots, 10^7, \ldots 10^{10}, \ldots$ images?
History of “large scale” visual search with local regions

<table>
<thead>
<tr>
<th>Research Team</th>
<th>Image Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schmid and Mohr ’97</td>
<td>1k images</td>
</tr>
<tr>
<td>Sivic and Zisserman’03</td>
<td>5k images</td>
</tr>
<tr>
<td>Nister and Stewenius’06</td>
<td>50k images (1M)</td>
</tr>
<tr>
<td>Philbin et al.’07</td>
<td>100k images</td>
</tr>
<tr>
<td>Chum et al.’07 + Jegou et al.’07</td>
<td>1M images</td>
</tr>
<tr>
<td>Chum et al.’08</td>
<td>5M images</td>
</tr>
<tr>
<td>Jegou et al. ’09</td>
<td>10M images</td>
</tr>
<tr>
<td>Jegou et al. ’10 and ’12</td>
<td>100M images</td>
</tr>
</tbody>
</table>

All on a single machine in ~ 1 second
Two strategies

1. Efficient approximate nearest neighbour search on local feature descriptors.

2. Quantize descriptors into a “visual vocabulary” and use efficient techniques from text retrieval.  
(Bag-of-words representation)
1. Compute local features in each image independently (Part 1)
2. “Label” each feature by a descriptor vector based on its intensity (Part 1)
3. Finding corresponding features is transformed to finding nearest neighbour vectors
4. Rank matched images by number of (tentatively) corresponding regions
5. Verify top ranked images based on spatial consistency (Part 2)
Finding nearest neighbour vectors

Establish correspondences between object model image and images in the database by **nearest neighbour matching** on SIFT vectors.

Solve following problem for all feature vectors, \( x_j \in \mathcal{R}^{128} \), in the query image:

\[
\forall j \quad NN(j) = \arg \min_i ||x_i - x_j||
\]

where, \( x_i \in \mathcal{R}^{128} \), are features from all the database images.
Quick look at the complexity of the NN-search

N … images
M … regions per image (~1000)
D … dimension of the descriptor (~128)

Exhaustive linear search: \( O(M N D) \)

Example:
• Matching two images (N=1), each having 1000 SIFT descriptors
  Nearest neighbors search: 0.4 s (2 GHz CPU, implementation in C)
• Memory footprint: 1000 * 128 = 128kB / image

<table>
<thead>
<tr>
<th># of images</th>
<th>CPU time</th>
<th>Memory req.</th>
</tr>
</thead>
<tbody>
<tr>
<td>N = 1,000</td>
<td>~7min</td>
<td>(~100MB)</td>
</tr>
<tr>
<td>N = 10,000</td>
<td>~1h7min</td>
<td>(~ 1GB)</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N = (10^7)</td>
<td>~115 days</td>
<td>(~ 1TB)</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

All images on Facebook:
N = \(10^{10}\) … ~300 years (~ 1PB)
Finding *approximate* nearest neighbour vectors

- Approximate method is not guaranteed to find the nearest neighbour.

- Can be much faster, but at the cost of missing some nearest matches
Approximate nearest neighbor search

Best-Bin First (BBF), a variant of k-d trees that uses priority queue to examine most promising branches first
[Beis & Lowe, CVPR 1997]
Extended to multiple randomized trees in :
[Muja & Lowe, 2009]

Locality-Sensitive Hashing (LSH), a randomized hashing technique using hash functions that map similar points to the same bin, with high probability
[Indyk & Motwani, 1998]

Can reduce the complexity of the search, e.g. $O(\log N)$ for k-d tree.
But at the cost of missing some nearest matches.

Adapted from K. Grauman, B. Leibe
Comparison of approximate NN-search methods


FAST APPROXIMATE NEAREST NEIGHBORS WITH AUTOMATIC ALGORITHM CONFIGURATION

Marius Muja, David G. Lowe

Computer Science Department, University of British Columbia, Vancouver, B.C., Canada
mariusm@cs.ubc.ca, lowe@cs.ubc.ca

Keywords: nearest-neighbors search, randomized kd-trees, hierarchical k-means tree, clustering.

Abstract: For many computer vision problems, the most time consuming component consists of nearest neighbor matching in high-dimensional spaces. There are no known exact algorithms for solving these high-dimensional problems that are faster than linear search. Approximate algorithms are known to provide large speedups with only minor loss in accuracy, but many such algorithms have been published with only minimal guidance on selecting an algorithm and its parameters for any given problem. In this paper, we describe a system that answers the question, “What is the fastest approximate nearest-neighbor algorithm for my data?” Our system will take any given dataset and desired degree of precision and use these to automatically determine the best algorithm and parameter values. We also describe a new algorithm that applies priority search on hierarchical k-means trees, which we have found to provide the best known performance on many datasets. After testing a range of alternatives, we have found that multiple randomized k-d trees provide the best performance for other
Comparison of approximate NN-search methods

Dataset: 100K SIFT descriptors

Code for all methods available online, see Muja&Lowe’09
Approximate nearest neighbour search (references)


ANN - search (references continued)


T. Norouzi and D. Fleet, Cartesian k-means., CVPR, 2013

See also next lecture by C. Schmid

and tutorial at CVPR’13 by H. Jegou: https://sites.google.com/site/lsvr13
So far …

• Linear exhaustive search can be prohibitively expensive for large image collections

• Answer (so far): approximate NN search methods
  • Randomized KD-trees
  • Locality sensitive hashing

• However, memory footprint can be still high.
  Example: \( N = 10^7 \) images, \( 10^{10} \) SIFT features with 128B per feature \( \rightarrow \) 1TB of memory

Look how text-based search engines (Google) index documents – inverted files.
Indexing text with inverted files

Document collection:

- **d1**
  - common
  - people
  - common
  - people

- **d2**
  - sculpture

- **d3**
  - sculpture
  - common

- **d4**
  - common
  - common
  - people
  - people
  - common

Inverted file:

<table>
<thead>
<tr>
<th>Term</th>
<th>List of hits (occurrences in documents)</th>
</tr>
</thead>
<tbody>
<tr>
<td>People</td>
<td>[d1:hit hit hit], [d4:hit hit] …</td>
</tr>
<tr>
<td>Common</td>
<td>[d1:hit hit], [d3: hit], [d4: hit hit hit] …</td>
</tr>
<tr>
<td>Sculpture</td>
<td>[d2:hit], [d3: hit hit hit] …</td>
</tr>
</tbody>
</table>
Vector quantize descriptors
- Compute SIFT features from a subset of images
- K-means clustering (need to choose K)

[Sivic and Zisserman, ICCV 2003]
Visual words

Example: each group of patches belongs to the same visual word
Samples of visual words (clusters on SIFT descriptors):

More specific example
Samples of visual words (clusters on SIFT descriptors):

More specific example
Visual words

• First explored for texture and material representations

• Texton = cluster center of filter responses over collection of images

• Describe textures and materials based on distribution of prototypical texture elements.

Visual words: quantize descriptor space
Sivic and Zisserman, ICCV 2003

Nearest neighbour matching
- expensive to do for all frames
Nearest neighbour matching
- expensive to do for all frames

Vector quantize descriptors

Visual words: quantize descriptor space
Sivic and Zisserman, ICCV 2003
Visual words: quantize descriptor space

Sivic and Zisserman, ICCV 2003

Nearest neighbour matching
- expensive to do for all frames

Vector quantize descriptors
Nearest neighbour matching
- expensive to do for all frames

Vector quantize descriptors

Visual words: quantize descriptor space
Sivic and Zisserman, ICCV 2003
Vector quantize the descriptor space (SIFT)

The same visual word
Representation: bag of (visual) words

Visual words are ‘iconic’ image patches or fragments
• represent their frequency of occurrence
• but not their position
Offline: Assign visual words and compute histograms for each image

Detect patches → Normalize patch → Compute SIFT descriptor → Find nearest cluster center

Represent image as a sparse histogram of visual word occurrences
Offline: create an index

• For fast search, store a “posting list” for the dataset
• This maps visual word occurrences to the images they occur in
  (i.e. like the “book index”)
At run time

• User specifies a query region
• Generate a short-list of images using visual words in the region

1. Accumulate all visual words within the query region
2. Use “book index” to find other frames with these words
3. Compute similarity for images that share at least one word
At run time

• Score each image by the (weighted) number of common visual words (tentative correspondences)

• Worst case complexity is linear in the number of images $N$

• In practice, it is linear in the length of the lists ($<< N$)
Strategy I: Efficient approximate NN search

1. Compute local features in each image independently (offline)
2. “Label” each feature by a descriptor vector based on its intensity (offline)
3. Finding corresponding features is transformed to finding nearest neighbour vectors
4. Rank matched images by number of (tentatively) corresponding regions
5. Verify top ranked images based on spatial consistency (The first part of this lecture)
Strategy II: Match histograms of visual words

1. Compute affine covariant regions in each frame independently (offline)
2. “Label” each region by a vector of descriptors based on its intensity (offline)
3. **Build histograms of visual words by descriptor quantization** (offline)
4. **Rank retrieved frames by matching vis. word histograms using inverted files.**
5. Verify retrieved frame based on spatial consistency (the first part of the lecture).
Overview of the retrieval system

query image

[Lowe04, Mikolajczyk07]
Hessian-Affine regions + SIFT descriptors

Set of SIFT descriptors

[Sivic03, Philbin07]
Clustered and quantized to visual words

sparse frequency vector

Results

1
2
3
4
5

[Lowe04, Philbin07]

Inverted file

Querying

Geometric verification

Ranked short-list of images

tf-idf weighting
Visual words: discussion I.

Efficiency – cost of quantization

• Need to still assign each local descriptor to one of the cluster centers. Could be prohibitive for large vocabularies (K=1M).

• Approximate NN-search still needed
  • e.g. randomized k-d trees [Muja\&Lowe 2009]

• True also for building the vocabulary
  • approximate k-means [Philbin et al. 2007]
  • Reduce k-means cost from $O(NK)$ to $O(N \log K)$
  • Can scale to very large $K$. 
Visual words: discussion II.

• Need to determine the size of the vocabulary, $K$.

• Other algorithms for building vocabularies, e.g. agglomerative clustering / mean-shift, but typically more expensive.

• Supervised quantization?
Also give examples of images / descriptors which should and should not match.

Visual search using local regions (references)


J. Sivic, A. Zisserman, Text retrieval approach to object matching in videos, ICCV, 2003


J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, CVPR, 2007


H. Jegou, M. Douze, C. Schmid, Hamming embedding and weak geometric consistency for large scale image search, ECCV’2008


Visual search using local regions (references)


H. Jégou, M. Douze, C. Schmid and P. Pérez, Aggregating local descriptors into a compact image representation, CVPR 2010


O. Chum, A. Mikulik, M. Perdoch, J. Matas, Total recall II: Query expansion revisited, CVPR 2011


And see the next lecture by C. Schmid
Efficient visual search for objects and places

Oxford Buildings Search - demo

http://www.robots.ox.ac.uk/~vgg/research/oxbuildings/index.html
Oxford buildings dataset

- Automatically crawled from Flickr

- Consists of:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Resolution</th>
<th># images</th>
<th># features</th>
<th>Descriptor size</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>1024 × 768</td>
<td>5,062</td>
<td>16,334,970</td>
<td>1.9 GB</td>
</tr>
<tr>
<td>ii</td>
<td>1024 × 768</td>
<td>99,782</td>
<td>277,770,833</td>
<td>33.1 GB</td>
</tr>
<tr>
<td>iii</td>
<td>500 × 333</td>
<td>1,040,801</td>
<td>1,186,469,709</td>
<td>141.4 GB</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>1,145,645</td>
<td>1,480,575,512</td>
<td>176.4 GB</td>
</tr>
</tbody>
</table>
Oxford buildings dataset

- Landmarks plus queries used for evaluation

- Ground truth obtained for 11 landmarks

- Evaluate performance by mean Average Precision
Measuring retrieval performance: Precision - Recall

- **Precision:** % of returned images that are relevant
- **Recall:** % of relevant images that are returned

![Precision-Recall Diagram](image-url)
Average Precision

- A good AP score requires both high recall and high precision
- Application-independent

Performance measured by mean Average Precision (mAP) over 55 queries on 100K or 1.1M image datasets
Mean Average Precision variation with vocabulary size

<table>
<thead>
<tr>
<th>vocab size</th>
<th>bag of words</th>
<th>spatial</th>
</tr>
</thead>
<tbody>
<tr>
<td>50K</td>
<td>0.473</td>
<td>0.599</td>
</tr>
<tr>
<td>100K</td>
<td>0.535</td>
<td>0.597</td>
</tr>
<tr>
<td>250K</td>
<td>0.598</td>
<td>0.633</td>
</tr>
<tr>
<td>500K</td>
<td>0.606</td>
<td>0.642</td>
</tr>
<tr>
<td>750K</td>
<td>0.609</td>
<td>0.630</td>
</tr>
<tr>
<td>1M</td>
<td>0.618</td>
<td>0.645</td>
</tr>
<tr>
<td>1.25M</td>
<td>0.602</td>
<td>0.625</td>
</tr>
</tbody>
</table>

![Graph showing mAP variation with vocabulary size](image)
• high precision at low recall (like google)
• variation in performance over query
• none retrieve all instances
Obtaining visual words is like a sensor measuring the image.

“Noise” in the measurement process means that some visual words are missing or incorrect, e.g. due to:

- Missed detections
- Changes beyond built in invariance
- Quantization effects

Consequence: Visual word in query is missing in target image.
Query Expansion in text

In text:
• Reissue top n responses as queries
• Pseudo/blind relevance feedback
• Danger of topic drift

In vision:
• Reissue spatially verified image regions as queries
Query Expansion: Text

Original query: Hubble Telescope Achievements

Query expansion: Select top 20 terms from top 20 documents according to tf-idf

Added terms: Telescope, hubble, space, nasa, ultraviolet, shuttle, mirror, telescopes, earth, discovery, orbit, flaw, scientists, launch, stars, universe, mirrors, light, optical, species

Example from: Jimmy Lin, University of Maryland
Automatic query expansion

Visual word representations of two images of the same object may differ (due to e.g. detection/quantization noise) resulting in missed returns

Initial returns may be used to add new relevant visual words to the query

Strong spatial model prevents ‘drift’ by discarding false positives

[Chum, Philbin, Sivic, Isard, Zisserman, ICCV’07; Chum, Mikulik, Perdoch, Matas, CVPR’11]
Visual query expansion - overview

1. Original query

2. Initial retrieval set

3. Spatial verification

4. New enhanced query

5. Additional retrieved images
Query Expansion

Query Image

Originally retrieved image

Originally not retrieved
Query Expansion
Query Expansion
Query Expansion
Demo
Quantization errors

Typically, quantization has a significant impact on the final performance of the system [Sivic03, Nister06, Philbin07]

Quantization errors split features that should be grouped together and confuse features that should be separated.
Overcoming quantization errors

• Soft-assign each descriptor to multiple cluster centers
[Philbin et al. 2008, Van Gemert et al. 2008]

Learning a vocabulary to overcome quantization errors
[Mikulik et al. ECCV 2010, Philbin et al. ECCV 2010]

See also next lecture.
Other recent work

Learning a vocabulary to overcome quantization errors
[Mikulik et al. ECCV 2010, Philbin et al. ECCV 2010]

Large scale image clustering [Chum et al. CVPR 2009, Philbin et al. IJCV 2010, Li et al., ECCV 2008]

Matching in structured datasets (3D landmarks or street-view images)
What objects/scenes local regions do not work on?
What objects/scenes local regions do not work on?

E.g. texture-less objects, objects defined by shape, deformable objects, wiry objects.
What next?

Visual search for texture-less, wiry, deformable and 3D objects..
Example:
Smooth object retrieval using a bag of boundaries by Arandjelovic and Zisserman, ICCV 2011
Category-level visual search [See later lectures.]

Query

same category

See also e.g. [Torresani et al. ECCV 2010]
What next?

Match objects across large changes of appearance
Examples: non-photographic depictions, degradation over time, change of season, …
Example: Painting-to-3D model alignment via discriminative visual elements

Inputs: paintings, drawings, historical photographs, reference 3D model

Output: recovered artist/camera viewpoints

[Aubry, Russell, Sivic, to appear in TOG 2013]
Why do this?

There are many non-photographic depictions of our world

Ultimate goal: to reason about these depictions
Applications

New ways to access archives for archeology, history or architecture

Example: evolution of a particular place over time

See also [WhatWasThere.com] with historical imagery manually aligned to a map.
Difficulty in finding correspondences

Color, geometry, illumination, shading, shadows and texture may be rendered by the artist in a realistic, but “non physical” manner

• 121 putative matches total across 563 photographs using SIFT matching
• 0 correct putative matches
Difficulty in finding correspondences

Local feature matching using SIFT:

Figure from [A. Shrivastava, T. Malisiewicz, A. Gupta, A. Efros
Data-driven Visual Similarity for Cross-domain Image Matching
SIGGRAPH Asia 2011]

See also:
[Hauagge & Snavely CVPR 2012]
[Chum & Matas CVPR 2006]
[Russell, Sivic, Ponce, Dessalles 2011]
How to match a painting to a 3D model?
I. Use 3D model to synthesize a similar view

Synthesize ~10,000 viewpoints for an architectural site

See also: [Irschara et al. CVPR 2009], [Baatz et al. ECCV 2012]
I. Use 3D model to synthesize a similar view

See also: [Irschara et al. CVPR 2009], [Baatz et al. ECCV 2012]
II. Matching as discriminative classification

1. Represent query region $q$ using HOG descriptor

2. Train a linear classifier $f(x) = w^Tx + b$ using $q$ as a positive example and large number of negatives

See detection lecture by A. Zisserman
See also: Exemplar SVM by [Malisiewicz et al., ICCV’11], [Shrivastava et al.’11]
II. Matching as discriminative classification

1. Represent query region $q$ using HOG descriptor
2. Train a linear classifier $f(x) = w^T x + b$ using $q$ as a positive example and large number of negatives
II. Matching as discriminative classification

1. Represent image region using HOG descriptor $x$
2. Train a linear classifier $f(x) = w^T x + b$
3. Find best match in the painting maximizing the classification score $f(x)$
II. Matching as discriminative classification

Discriminative visual element: trained classifier \( f(x) = w^T x + b \)

How to choose discriminative visual elements representing architectural site?

See also [Doersch et al. SIGGRAPH 2012] [Singh et al. ECCV 2012], [Juneja et al. CVPR 2013]
Algorithm outline

Offline:
1. Sample virtual viewpoints from 3D site
2. Learn discriminative visual elements from rendered views

Given painting:
3. Obtain element detections on the painting
4. Keep only matches consistent with a single view (RANSAC)
5. Optional: fine viewpoint alignment
 Offline: Learn a “vocabulary” of discriminative visual elements

• Train classifiers for all candidate regions in synthesized views
  – Can be done efficiently, see [Gharbi et al. 2012; Hariharan et al. 2012 ]
• Score each classifier by its training error.
• Keep only the top N most discriminative visual elements.

Note: Can be thought of as a generalization of local feature detection.
Offline: Learn a “vocabulary” of discriminative visual elements

- Back-project learnt discriminative elements onto the 3D model

See also [Doersch et al. SIGGRAPH 2012] [Singh et al. ECCV 2012], [Juneja et al. CVPR 2013]
Given a painting:
Obtain visual element detections and verify matches with RANSAC
Example II.
Experiments

• 3D architectural sites
  – Venice (PMVS reconstruction from “Rome in a day” photographs)
  – Venice (3D CAD model)
  – Trevi Fountain (3D CAD model)
  – Notre Dame of Paris (3D CAD model)

• “Test queries”
  – 50 historical photographs
  – 150 paintings/drawings
Results: historical photographs
Results: paintings and drawings
SATURDAY 18 SEPTEMBER

SAN MARCO
Standing in line from 9.28 to 9.49. Plenty time for sketching!
Challenging examples

Scene distortion

Drawing errors

Different scene
Failures

- Extreme change in depiction styles (smeared watercolor)
- Part of the architectural site not covered by 3D model
- Extreme geometric distortion
Viewing frusta in 3D
Fly-through video
Outline

1. Local invariant features (C. Schmid)

2. Matching and recognition with local features (J. Sivic)

3. Efficient visual search (J. Sivic)

4. Very large scale visual indexing (C. Schmid)

Practical session – Instance-level recognition and search