Instance-level recognition III. Visual search: extensions and applications

Josef Sivic
http://www.di.ens.fr/~josef
INRIA, WILLOW, ENS/INRIA/CNRS UMR 8548
Laboratoire d’Informatique, Ecole Normale Supérieure, Paris

Announcements

Class web-page:
http://www.di.ens.fr/willow/teaching/recvis10/

Email list: Please add your name and email.

Assignment 1 deadline was extended to
Next Tuesday, Nov 2\textsuperscript{nd} 2010!

Assignment 2: Stitching photo-mosaics
http://www.di.ens.fr/willow/teaching/recvis10/assignment2/
is due next Tuesday, Nov 2\textsuperscript{nd} 2010
Lecture plan

Lecture 2:
• Local invariant features (C. Schmid)

Lecture 3:
• Camera geometry – review (J. Ponce)
• Correspondence, matching and recognition with local features, efficient visual search (J. Sivic)

Lecture 4: (C. Schmid):
• Very large scale visual indexing
• Bag-of-feature models for category-level recognition

Lecture 5 (today):
• Sparse coding and dictionary learning (J. Ponce)
• Visual search – extensions and applications (J. Sivic)
• Category-level localization (J. Sivic)
1. Review: Large-scale recognition with local features
1000+ descriptors per image
Match regions between frames using SIFT descriptors and spatial consistency

Multiple regions overcome problem of partial occlusion
Fast descriptor search

Complexity

- $O(nd)$ for $n$ features and $d$ dimensions
- Linear in the number of features / images

Speed up individual descriptor vector search

- kd-trees (k dim. tree), approximate nearest neighbor search
- K-means tree
- Locality sensitive hashing (LSH)
Visual words: main idea

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.

K. Grauman, B. Leibe
Bag-of-features / Bag-of-visual-words [Sivic&Zisserman’03]

“visual words”:
- 1 “word” (index) per local descriptor
- only images ids in inverted file
=> 8 GB fits!

[Harris-Hessian-Laplace regions + SIFT descriptors]

Set of SIFT descriptors

Bag-of-features processing + tf-idf weighting

sparse frequency vector

centroids (visual words)

Inverted file

querying

ranked image short-list

Re-ranked list

Geometric verification

[Chum & al. 2007]
Beyond visual words: Hamming Embedding

Representation of a descriptor \( x \)

- Vector-quantized to \( q(x) \) as in standard BOF
- Short binary vector \( b(x) \) for an additional localization in the Voronoi cell

Two descriptors \( x \) and \( y \) match iff

\[
f_{HE}(x, y) = \begin{cases} 
(tf-idf(q(x)))^2 & \text{if } q(x) = q(y) \\
0 & \text{and } h(b(x), b(y)) \leq h_t \\
\end{cases}
\]

where \( h(a,b) \) Hamming distance
Recent approaches for very large scale indexing

Hessian-Affine regions + SIFT descriptors

Set of SIFT descriptors

Bag-of-features processing + tf-idf weighting

sparse frequency vector

Vector compression

Vector search

ranked image short-list

Re-ranked list

Geometric verification
VLAD: vector of locally aggregated descriptors

- Simplification of Fisher kernels

- Learning: a vector quantizer (k-means)
  - output: $k$ centroids (visual words): $c_1, \ldots, c_i, \ldots c_k$
  - centroid $c_i$ has dimension $d$

- For a given image
  - assign each descriptor to closest center $c_i$
  - accumulate (sum) descriptors per cell
    \[ v_i := v_i + (x - c_i) \]

- VLAD (dimension $D = k \times d$)

- The vector is L2-normalized
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


H. Jégou, M. Douze, C. Schmid and P. Pérez, Aggregating local descriptors into a compact image representation, CVPR’2010
Efficient visual search for objects and places

Oxford Buildings Search - demo

http://www.robots.ox.ac.uk/~vgg/research/oxbuildings/index.html
Example
Search results 1 to 20 of 104844

1
ID: oxc1_hertford_000011
Score: 1816.000000
Putative: 2325
Inliers: 1816
Hypothesis: 1.000000 0.000000 0.000015 0.000000 1.000000 0.00031
Detail

2
ID: oxc1_all_souls_000075
Score: 352.000000
Putative: 645
Inliers: 352
Hypothesis: 1.162245 0.041211 -70.414459 -0.012913 1.146417 91.276093
Detail

3
ID: oxc1_hertford_000064
Score: 278.000000
Putative: 527
Inliers: 278
Hypothesis: 0.923686 0.026134 169.954620 -0.041703 0.937558 97.962112
Detail
2. Visual search - extensions

- Query expansion
- Pre-computing matching graph
- Overcoming quantization errors
- Retrieval in structured databases
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
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]
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
Query Expansion

New expanded query is formed as

- the average of visual word vectors of spatially verified returns
- only inliers are considered
- regions are back-projected to the original query image
Efficient visual search for objects and places

Oxford Buildings Search - demo

http://www.robots.ox.ac.uk/~vgg/research/oxbuildings/index.html
Query Expansion

Query image

Originally retrieved

Retrieved only after expansion
Query image

Original results (good)

Expanded results (better)
Pre-compute query expansion?

- Query expansion works well, however, at an additional cost at the query time.

- Can we offline pre-process the database and pre-compute the query expansion?

Solution: Compute and build a matching graph.
Matching graph

Build a ‘matching graph’ over all the images in the dataset

Each image is a node and a link represents two images having some object in common

Instead of expanding the query, traverse links of this graph

[Chum et al. 2008, Philbin et al. IJCV 2010, Turcot and Lowe 2009]
Example:
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

- Query expansion. [Chum et al. 2007]
- Soft-assignment. [Philbin et al. 2008]
- Hamming embedding / VLAD [Jegou & Schmid ’08, ‘10]

Overcome errors **given** a quantization.
Have cost in terms of space and/or time complexity at query-time.
Descriptor learning for efficient retrieval

The aim of this work is to reduce these errors at source, by learning a projection function that actively reduces this error:

\[ T(x; W) \quad T : \mathbb{R}^D \rightarrow \mathbb{R}^M \]

\[ d_W(x, y) = \|T(x; W) - T(y; W)\|_2 \]

- \( T \) can be linear or non-linear and we can choose to keep the descriptor dimensionality the same or reduce it
- After this projection, use the same visual words architecture

[Philbin, Isard, Sivic, Zisserman, ECCV 2010]
Descriptor learning for efficient retrieval

- No additional query-time cost over BOW

- For particular object retrieval, we can leverage the spatial consistency between object instances to automatically generate large amounts of training data (matched / non matched point pairs)

Confusion and splitting vs. No confusion or splitting
Descriptor learning for efficient retrieval

Choose form of \( T(x; W) \):

- Can be linear: \( T(x; W) = Wx \)

- Or non-linear (DBN-style formulation):

\[
T(x; W_1, W_2, W_3, h_0, h_1, h_2) = \\
W_3\sigma(W_2\sigma(W_1\sigma(x + h_0) + h_1) + h_2)
\]

Non-linear model gives better results.
Results: Spatial Verification

Quantized 128-D SIFT descriptors (K=1M)
Results: Spatial Verification

- Quantized 128-D SIFT descriptors (K=1M)
  - 26 inliers
  - 38 inliers
  - 49 inliers

- Raw 128-D SIFT
  - 48 inliers
  - 61 inliers
  - 114 inliers
Results: Spatial Verification

26 inliers

38 inliers

49 inliers

Quantized 128-D SIFT descriptors (K=1M)

37 inliers

56 inliers

64 inliers

Quantized 32-D learnt descriptors (K=1M)

48 inliers

61 inliers

114 inliers

Raw 128-D SIFT
## Results: Baseline to State of the Art

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean Average Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Baseline Method K = 10K</td>
<td>0.389</td>
</tr>
<tr>
<td>2. Large Vocabulary K=1M</td>
<td>0.618</td>
</tr>
<tr>
<td>3. Spatial Re-ranking</td>
<td>0.653</td>
</tr>
<tr>
<td>4. Soft Assignment (SA) Learnt descriptors</td>
<td>0.731</td>
</tr>
<tr>
<td>5. Query Expansion (QE)</td>
<td>0.801</td>
</tr>
<tr>
<td>6. SA &amp; QE</td>
<td>0.825</td>
</tr>
</tbody>
</table>
Place recognition: retrieval in a **structured** (on a map) database

[Knopp, Sivic, Pajdla, ECCV 2010]
Correctly recognized examples
More correctly recognized examples

<table>
<thead>
<tr>
<th>Query</th>
<th>Top ranked image</th>
<th>Query</th>
<th>Top ranked image</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Query Image" /></td>
<td><img src="image2" alt="Top ranked image" /></td>
<td><img src="image3" alt="Query Image" /></td>
<td><img src="image4" alt="Top ranked image" /></td>
</tr>
</tbody>
</table>

- **Top ranked image**
  - ![Top ranked image](image5)
  - ![Top ranked image](image6)

- **Query**
  - ![Query Image](image7)
  - ![Query Image](image8)
Quantitative evaluation

- 200 challenging test queries downloaded from Panoramio
- ~17,000 geotagged images downloaded from Google Street View

<table>
<thead>
<tr>
<th>Method</th>
<th>% correct initial retrieval</th>
<th>% correct with spatial verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Baseline place recognition</td>
<td>20.96</td>
<td>29.34</td>
</tr>
<tr>
<td>b. Query expansion</td>
<td>26.35</td>
<td>41.92</td>
</tr>
<tr>
<td>c. Confuser suppression</td>
<td>29.94</td>
<td>37.72</td>
</tr>
<tr>
<td>d. Confuser suppression + Query expansion</td>
<td>32.93</td>
<td>47.90</td>
</tr>
</tbody>
</table>

*Table 1. Percentage of correctly localized test queries for different place recognition approaches.*
Other recent work

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

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

Very large scale retrieval -- towards 1 billion images
  [Jegou et al. CVPR 2010] Last lecture!

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.
3. Example applications of large scale visual search and matching
Sony Aibo (Evolution Robotics)

SIFT usage
- Recognize docking station
- Communicate with visual cards

Other uses
- Place recognition
- Loop closure in SLAM
Application: Internet-based inpainting

Photo-editing using images of the same place

[Whyte, Sivic and Zisserman, 2009]
Mobile tourist guide

- Self-localization
- Object/building recognition
- Photo/video augmentation

[Quack, Leibe, Van Gool, CIVR’08]
Web Demo: Movie Poster Recognition

50’000 movie posters indexed

Query-by-image from mobile phone available in Switzerland


K. Grauman, B. Leibe
Image Auto-Annotation

Left: Wikipedia image
Right: closest match from Flickr

[Quack CIVR’08]
Visual search in your pocket

Google Goggles
Use pictures to search the web. Watch a video
but it doesn't work well yet on things like food, cars, plants, or animals.
Building Rome in a Day – or –

matching and 3D reconstruction in large unstructured datasets.

Goal: Build a 3D model of a city from a large collection of images downloaded from the Internet

Use a cluster with 500 CPU cores.

http://grail.cs.washington.edu/rome/
Creative Commons / Attribution-NonCommercial-NoDerivs License

(Or, browse popular tags)

Here are the 130 most recent licensed photos:

From: ncismell
From: mugey1274
From: darren * dip* paine
From: darren * dip* paine
From: dizz

From: alpha zone
From: 331
From: mugey1274
From: ololoko
Photo Tourism overview

Input photographs → Scene reconstruction → Photo Explorer

- Relative camera positions and orientations
- Point cloud
- Sparse correspondence
Photo Tourism overview

Input photographs → Scene reconstruction → Photo Explorer

Slide: N. Snavely
Scene reconstruction

Automatically estimate

- position, orientation, and focal length of cameras
- 3D positions of feature points
Feature detection

Detect features using SIFT [Lowe, IJCV 2004]
Feature detection

Detect features using SIFT [Lowe, IJCV 2004]
Feature detection

Detect features using SIFT [Lowe, IJCV 2004]
Feature matching

Complexity of matching:

Unfortunately, even with a well optimized implementation of the matching procedure described above, it is not practical to match all pairs of images in our corpus. For a corpus of 100,000 images, this translates into 5,000,000,000 pairwise comparisons, which with 500 cores operating at 10 image pairs per second per core would require about 11.5 days to match. Furthermore, this does not even take into account the network transfers required for all cores to have access to all the SIFT feature data for all images.

From Agarwal et al. “Building Rome in a Day”, ICCV’09
Feature matching

Obtain candidate pairs of images to match using visual vocabulary matching based on k-means tree

Figure: N. Snavely
Feature matching

Match features between candidate pairs using K-d trees built on SIFT descriptors.

Figure: N. Snavely
Feature matching

Refine matching using RANSAC [Fischler & Bolles 1987] to estimate fundamental matrices between pairs

Slide: N. Snavely
Structure from motion (R. Keriven’s class)

minimize $f(R, T, P)$
Example of the final 3D point cloud and cameras

57,845 downloaded images, 11,868 registered images. This video: 4,619 images.