Large Scale Image Retrieval


*Presented by:*

Hatem Alismail
halismai@cs.cmu.edu
04/12/12
Large Scale Image Retrieval

Find these landmarks
... In these images

Image from: http://www.robots.ox.ac.uk/~vgg/research/oxbuildings/index.html
Large Scale Image Retrieval

Challenge
How to find a query image “correctly” and “efficiently” in very large databases (billions of images)?
Large Scale Image Retrieval: Difficulties

Scale

Viewpoint

Lighting

Occlusion

Image from: Instance-level Recognition, Cordelia Schmid & Josef Sivic
Large Scale Image Retrieval: Difficulties

• Very Large image datasets, billions of photos
  – In 2008, Facebook had 10 Billion photos
  – In 2010, 5 billion photos on Flicker, 3000+ per minute
  – YouTube http://www.youtube.com/t/press_statistics
    • 60 hours of video every minute (1 hr/s)
      – Say at 10fps, 36K images per second, in 10 minutes we have 1,296,000,000 images
Bag of Words (BOW)

• Successful method in document/text retrieval
  – Represent documents as histograms of word occurrence frequencies [Salton83]

• Can be done efficiently for large datasets, with the help of:
  – Inverted files
  – \textit{tf-idf weighting}
    • \textit{Term Frequency Inverse Document Frequency}

• Introduced for object retrieval in Computer Vision by [Sivic03]
  – VideoGoogle \url{http://www.robots.ox.ac.uk/~vgg/research/vgoogle/}
    • Check out the link for a demo
**BOW [Sivic03]**

- **Query image**
  - Hessain-Affine regions + SIFT descriptors
    - [Mikolajczyk & Schmid 04]
    - [Lowe 04]

  → **Set of SIFT descriptors**
  - **centroids** (visual words)

  → **Bag-of-features processing + tf-idf weighting**

  → **[Nister & al 04, Chum & al 07]** sparse frequency vector

  → **Inverted file**

  → **querying**

  → **Re-ranked list**

  → **Geometric verification**

  → **ranked image short-list**

- “visual words”:
  - 1 “word” (index) per local descriptor
  - only images ids in inverted file

*Slide from: Instance-level Recognition, Cordelia Schmid & Josef Sivic*
Efficiency: Inverted Index

- Mapping between words and on which page(s) their occur
  - We can efficiently find all pages/docs associated with a word

- Efficient, but does not solve the problem completely
  - Visual words might occur in many images
    - Confusers [REF], cooc-sets [REF]
  - Large dictionary size ~1M
    - Good for Inverted Index File

Image from: http://votebits.com/do-you-know-how-web-search-engines-work/inverted-index/
tf-idf

- Term Frequency – Inverse Document Frequency
- Indicates the importance of a word
- TF := # of times a word appears in the doc
  - Could have many variations
- IDF := total number of docs / docs containing the term
  - Many variations are possible
- tf-idf = TF * IDF

See e.g. http://pyevolve.sourceforge.net/wordpress/?p=1589
BOW

201 matches

240 matches

image from: Instance-level Recognition, Cordelia Schmid & Josef Sivic
Example from: http://www.robots.ox.ac.uk/~vgg/research/vgoogle/
Query

3rd ranked result

Example from: http://www.robots.ox.ac.uk/~vgg/research/vgoogle/
Original selection from frame 49325

Query

Matched region from frame 116675

3rd ranked result

Example from: http://www.robots.ox.ac.uk/~vgg/research/vgoogle/
How to fix this?

Image from: A. Zisserman, Scalable Object Retrieval
Spatial Verification

Spatially consistent, OK

Too few inliers, NOT OK

Image from: A. Zisserman, Scalable Object Retrieval
Spatial Verification

• Use local matches to verify geometry

• Rank the results based on the number of inliers

• Two images of the same scene are related via Epipolar Geometry
  – Arbitrary Perspective (8 DOF)
  – Fundamental matrix (7 DOF)
  – Affine (5 DOF)
  – Homography (4 DOF)
  – Isotropic Scaling only (1 DOF)
  – ...
Spatial Verification

- Many noisy matches
- **RANSAC**
  - Select a minimal set of points
  - Fit a model
  - Test inlier
  - Repeat until done
- Does not scale for high DOF model over large scale datasets
  - Need as small minimal set as possible
Faster RANSAC: LO-RANSAC [Chum04]

- Wrong assumption in RANSAC:
  - Every model computed from an uncontaminated set of samples is consistent with all inliers

Figure from [Chum04]
Faster RANSAC: LO-RANSAC [Chum04]

- Select minimal set and fit model
- Test for inlier support
- If model is best-so-far
  - Refine locally and find more inliers
  - best_model = refined_model
  - best_inliers = refined_inliers
- Stop when number of inliers is good enough
LO-RANSAC: Usage in Large Scale

- If we can use a very simple model (1-3 DOF) in vanilla RANSAC, we can refine this model to a higher DOF in the “LO” part

- Need to get good initializations for the simple models
Spatial Verification: Speed up via Exploiting Affine Invariant regions

Image from [Philbin07]

<table>
<thead>
<tr>
<th>Transformation</th>
<th>dof</th>
<th>Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation + isotropic scale</td>
<td>3</td>
<td>$\begin{bmatrix} a &amp; 0 &amp; t_x \ 0 &amp; a &amp; t_y \ 0 &amp; 0 &amp; 1 \end{bmatrix}$</td>
</tr>
<tr>
<td>translation + anisotropic scale</td>
<td>4</td>
<td>$\begin{bmatrix} a &amp; 0 &amp; t_x \ 0 &amp; b &amp; t_y \ 0 &amp; 0 &amp; 1 \end{bmatrix}$</td>
</tr>
<tr>
<td>translation + vertical shear</td>
<td>5</td>
<td>$\begin{bmatrix} a &amp; 0 &amp; t_x \ b &amp; c &amp; t_y \ 0 &amp; 0 &amp; 1 \end{bmatrix}$</td>
</tr>
</tbody>
</table>

(a) Table 6. (a) The three affine sub-groups compared in the spatial re-ranking. (b) Computing $H$ as $H_2^{-1} H_1$, preserving “upness” for the 5 dof case.

Figure from [Philbin07]
(3DOF) Translation + isotropic scale

\[
\begin{pmatrix}
  a & 0 & t_x \\
  0 & a & t_y
\end{pmatrix}
\]

Single correspondence between region, centroid for t and regions scale for a - models:
- change in camera zoom / distance from scene
(4DOF) Translation + anisotropic scale

\[
\begin{pmatrix}
a & 0 & t_x \\
0 & b & t_y \\
0 & 0 & 1 \\
\end{pmatrix}
\]

Single correspondence between region, centroid for \( t \) and regions scale along x-axis for \( a \) and along y-axis for \( b \)

- models:
  - change in camera zoom / distance from scene,
  - And foreshortening
Single correspondence between region, find a transformation for every region such that the region is a unit circle, such that the y-direction is maintained.

- models:
  - change in camera zoom / distance from scene, anisotropic scaling
  - maintains vertical structure
Fast Spatial Verification

- RANSAC with single point correspondence is very fast
- Can use all single point hypothesis (deterministic)
- LO-RANSAC optimized full 6DOF Affine
- **Differences between using 3/4/5 DOF for initialization are negligible**
  - Optimization for a full 6DOF improves the initialization equally well
  - Time needed to upgrade the model from 3/4/5 $\leftrightarrow$ 6?
Spatial Verification

<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><strong>0.618</strong></td>
<td><strong>0.645</strong></td>
</tr>
<tr>
<td>1.25M</td>
<td>0.602</td>
<td>0.625</td>
</tr>
</tbody>
</table>

Table 5. Examining the effect of vocabulary size on performance for the 5K dataset. Each vocabulary is trained using AKM on all 16.7M descriptors. There is a performance peak at 1 million words. The spatial verification consistently improves performance.
Spatial Verification

- Works very well
- Including the spatial information in the index to reduce computation at the ranking stage? [Philbin07]
Large illumination/viewpoint change. BOW + spatial verification cannot find the image. Will return a wrong list instead.

Can we fix this?
Query Expansion

We can get some easy images (verify with spatial matching)
Query Expansion

Other easy images can be used to find “harder ones”. Re-issue the verified images as new queries
Query Expansion

Slide adapted from A. Zisserman, Scalable Object Retrieval
Query Expansion

• From text retrieval
  – Reissue top N retrieved documents as new queries
  – Blind relevance feedback
  – Topic drift

• Vision
  – Reissue **spatially verified** image regions as new queries
  – **Query Expansion will not work if the expanded queries are wrong**
    • *In a sense, QE depends on spatial verification*
Query Expansion

Slide adapted from A. Zisserman, Scalable Object Retrieval
Query Expansion

Slide adapted from A. Zisserman, Scalable Object Retrieval
Query Expansion [Chum07]

- Transitive Closure Expansion
  - Store spatially verified images in a priority queue

- Average Query Expansion
  - Average verified results of the original

\[ d_{avg} = \frac{1}{m + 1} \left( d_0 + \sum_{i=1}^{m} d_i \right) \]

- Recursive Average Query Expansion

- Multiple Image Resolution
  - Average Query using verified images with features of within a limit close to the query’s resolution
Query Expansion [Chum07]

- **Transitive Closure Expansion**
  - Store spatially verified images in a priority queue

- **Average Query Expansion**
  - Average verified results of the original

- **Recursive Average Query Expansion**

- **Multiple Image Resolution**
  - Average Query using verified images with features of within a limit close to the query's resolution

\[
\bar{d}_{avg} = \frac{1}{m + 1} \left( d_0 + \sum_{i=1}^{m} d_i \right)
\]

Simple and works well
• Improving Query Expansion
• *tf-idf* might fail sometimes due to confuser words
  – Method to detect the failure
• Improving blind relevance feedback
  – Incremental spatial re-ranking
• Incorporating Context
  – Search outside the (bounding) box
tf-idf failure detection [Chum11]

Object words have a high probability of being observed in image containing the object

\[ P(w|O) \gg P(w) \]

Confuser words, are the set of correlated words, such that

\[ P(w|C) \gg P(w) \]

1- Find the confuser words (words that appear too much)

\[ W_c = \{ w : \frac{P(w|S)}{P(w)} > r_0 \} \]

2- Remove W_c from the query

Implicit assumption: we are not looking for the confuser words
tf-idf failure detection [Chum11]

How to know that tf-idf has failed?

Compute an estimated “Quality” of matching: sum of inlier ratios over acceptable results

Set of acceptable results

$$A_Q = \{ X | X \in S_Q \& I_Q(X) > I_0 \& \frac{I_Q(X)}{T_Q(X)} > \epsilon_0 \}$$

<table>
<thead>
<tr>
<th>Image</th>
<th># of consistent matches</th>
<th># of correspondences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$$\rho(Q) = \sum_{X \in A_Q} \frac{I_Q(X)}{T_Q(X)}$$

Tune the thresholds, $\epsilon_0$ & $\rho_0$
Incremental Spatial Re-ranking [Chum11]

- Improving the blind relevance feedback

- Do the spatial verification for all previously verified images, not only the initial query

- Back project onto the original query and add visual words that pass the spatial verification

- Rank the result using the number of consistent visual words found
Figure 2. The process of context learning. Left column: the original query. Other columns: feature patches back-projected into the spatial context from 2, 5, 10 and 20 spatially verified images.
Features that pass iSP are just added back to the feature model even if they do not fall into the query's bounding box.

If the feature is geometrically verified from other images, it is activated.

Works if the query does not cover the whole object to begin with.

What if we just expand the query's box with a some amount for every image? (i.e. no learning)
The algorithm (by design) will remove potentially “useful confusers”

Or

Add real context...
## Total Recall II: Results

<table>
<thead>
<tr>
<th></th>
<th>I. w/o QE</th>
<th></th>
<th>II. avg QE</th>
<th></th>
<th>III. ctx QE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SP</td>
<td>iSP</td>
<td>SP</td>
<td>iSP</td>
<td>SP</td>
<td>iSP</td>
</tr>
<tr>
<td>Oxford 5k</td>
<td>0.616</td>
<td>0.741</td>
<td>0.785</td>
<td>0.825</td>
<td>0.781</td>
<td>0.827</td>
</tr>
<tr>
<td>Oxford 105k</td>
<td>0.553</td>
<td>0.649</td>
<td>0.725</td>
<td>0.761</td>
<td>0.731</td>
<td>0.767</td>
</tr>
<tr>
<td>Paris 6k</td>
<td>0.617</td>
<td>0.679</td>
<td>0.720</td>
<td>0.772</td>
<td>0.753</td>
<td>0.805</td>
</tr>
<tr>
<td>Paris 106k</td>
<td>0.508</td>
<td>0.556</td>
<td>0.627</td>
<td>0.687</td>
<td>0.653</td>
<td>0.710</td>
</tr>
</tbody>
</table>

Table 4. Comparison of image retrieval methods with standard (SP) and incremental spatial re-ranking (iSP).

**SP** spatial verification re-ranking, no query expansion  
**iSP** incremental spatial re-ranking, no query expansion  
**SP + avg QE** (query expansion by averaging)  
**iSP + avg QE** (ditto with iSP)  
**SP + ctx QE** query expansion with context learning  
**iSP + ctx QE** (ditto with iSP)
Second Paper
Learning Query-dependent Prefilters for Scalable Image Retrieval
Overview

- Use the BOW to find similar object *classes* rather than object *instances*, doing this *efficiently* to scale up for billions of images
Overview

- To be able to scale we need efficient algorithms to return a small enough *short list* of images that we can later process.
- Main idea: Form phrases from visual words, for which a disjunction of conjunctions of these words will reduce the search space.
- Example, Google *(your mileage may vary +your world)*
  - monkey (380,000,000)
  - monkey + cat (71,900,000)
  - monkey + cat + fire (35,900,000)
  - monkey + cat + fire + science (14,400,000)
  - monkey + cat + fire + science +haemophilus (200,000)
  - etc... could not break it yet
Learning Prefilters

- Search engines are optimized for OR of AND's queries
- We can generate many of these filters (OR of AND's)
  - We need to find the best ones
- For each query, select OR of AND's filter that maximizes the training set recall, subject to a bound on the repose set size
OR's of AND's

• Search engines are very good at handling OR's of AND's
  – Use the inverted index file for efficient implementation

• Learn boolean functions from class histograms
  – Decision stumps: “word $w$ occurs more/less than $t$ times”
    \[
    C_s(h) := (h_{w_s} > t_s)
    \]
  – Two images $(i,j)$ are similar:
    \[
    C_s(i, j) := C_s(h_i) \land C_s(h_j)
    \]
  – A phrase is defined using a set of stumps AND'd together
    \[
    P_S(h) := \bigwedge_{s \in S} C_s(h_i)
    \]
  – Images are similar:
    \[
    P_S(i, j) := P_S(h_i) \land P_S(h_j)
    \]
Classifier

- Defined as an OR of AND's as specified by a subset of the stumps:

\[ Q_\Sigma(h_i, h_j) = \bigvee_{S \in \Sigma} P_S(h_i, h_j) \]

\[ = \bigvee_{S \in \Sigma} \bigwedge_{s \in S} C_s(h_i, h_j) \]

- Learn the phrase pool

- Learn query phrases at run time subject to bounded response set
Bounding the classifier at run time

- Assume we have the pool of phrases
- We have $N$ images with their histogram representations
- We have $M$ labeled pairs indicating similarity (positive)
- We know the set of phrases that exist in the current query
- Goal: select a useful subset of the phrases in query, s.t.:
  - The response set size is bounded by some threshold
  - Performance on the training set is good
- The problem can be converted into integer programming and solved with SAT solvers
- The filter we want to apply to the database

$$\bigvee_{S \in B} P_S(h)$$
Phrases we want [i.e. indicate image (i,j) are similar]

Maximize the union of these boxes (for each query). Call this the positive region.
The Phrase Pool
The Phrase Pool
The Phrase Pool
The Phrase Pool
The Phrase Pool
Results

Figure from [Torresani09]
Results

- Performs reasonably well on never seen object classes
  - Appears to be learning something useful
- Good recall with a smaller response set size

Figure from [Torresani09]
Conclusions

- **BOW is great, but it needs to be adapted more to images**
  - Spatial Verification (SV)
    - Incremental transform estimation
  - Phrases

- **Query Expansion (QE) boosts recall**
  - Depends on correct initial retrievals from SV
  - Making sure that the images used to re-query are good
    - Could improve performance if users rate relevance of results
  - Many methods for QE, average QE works well

- **There is room for improvements by including context**
References


