Content Based Image Search

Mark Desnoyer
Text Search - TF-IDF

\[ \text{Normalized Term Count} \times \text{Inverse Document Frequency} = \text{Document Descriptor} \]

Corpus
Text Search - Query

Document Rankings = Query^T Corpus
Current Image Search

- On the web uses context around image
  - Words around it
  - Words in the alt tag
- Those words are treated as a document
- Same as normal text search

- But we want pictures, not text!

Query: Horse
Searching With Pictures

- How about searching with pictures instead
Using Visual Words For Search

- Use visual words paradigm we've seen before
- Can use all the text search machinery we already have
- But, we're searching with pictures now
The Players


SIFT Features

- Succinct descriptors
- Scale invariant
- Robust to changes in lighting, viewpoint, blur etc.
- Therefore, normally used in this search context
Bag of Words First Step - Build a Dictionary

- Must be big to be expressive enough to differentiate objects
- So, cluster SIFT features
- Each cluster is a word in the dictionary
- But K-means clustering 10M+ descriptors is $O(NK)$
  - Hierarchical K-means (Nister)
  - Approximate K-means (Chum)
Vocabulary Tree (Nister)
Vocabulary Tree (Nister)
Vocabulary Tree (Nister)
Vocabulary Tree (Nister)

Copyright David Nistér, Henrik Stewénius
Vocabulary Tree (Nister)
Approximate Nearest Neighbour (Chum)

- Most of the time in K-Means is spent doing Nearest Neighbour
- Nearest Neighbour can be approximated using kd-trees
- $O(N \log K)$ vs. $O(NK)$
Another Problem - Synonyms

Visual Polysemy. Single visual word occurring on different (but locally similar) parts on different object categories.

Visual Synonyms. Two different visual words representing a similar part of an object (wheel of a motorbike).
Video Google

- Search for recurring objects in a movie
- Synonyms suppressed by enforcing consistency in time
- Stop list used to throw out words that are too common
Photo Tourism Overview

Input photographs

Scene reconstruction

Relative camera positions and orientations
Point cloud
Sparse correspondence

Photo Explorer

Copyright Noah Snavely
Photo Tourism Scene Reconstruction

- Automatically estimate
  - position, orientation, and focal length of cameras
  - 3D positions of feature points

Feature detection

Pairwise feature matching

Correspondence estimation

Incremental structure from motion

Copyright Noah Snavely
Photo Tourism

Demo
Photo Tourism Limitations

- Matching is only performed between pairs of images
- Does not scale to large datasets
Chum et al. Experiment

- 5K labeled images of sights at Oxford
- 1M Flickr images from popular tags (distractors)
- Dictionary built from Oxford Images
  - 16M descriptions -> 1M word dictionary
- Query for landmarks and calculate PR curve using different forms of query expansion
Oxford Buildings Dataset

- Landmarks plus queries used for evaluation

- Ground truth obtained for 11 landmarks over 5062 images

- Evaluate performance by mean Average Precision
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
Beyond Bag of Words

- Can we use the **position** and **shape** of the underlying features to improve retrieval quality?

- Both images have lots of matches – which is correct?
Beyond Bag of Words

- We can enforce **spatial consistency** between the query and each result to improve retrieval quality!

Lots of spatially consistent matches – **correct result**

Few spatially consistent matches – **incorrect result**

Copyright James Philbin
Beyond Bag of Words

- Extra bonus – gives us localization of the object
Estimating Spatial Correspondences

1. Test each correspondence
Estimating Spatial Correspondences

2. Compute a (restricted) affine transformation (5 dof)
Estimating Spatial Correspondences

3. Score by number of consistent matches

Use RANSAC on full affine transformation (6 dof)
Text Query Expansion

- In text search, some words are similar, but they are different in the dictionary
  - e.g. gray and grey

- Improve results by expanding the query to include similar words
  - e.g. "grey goose" -> "grey goose gray"

- Similar words are found by clustering on document data

- At query time, relevant clusters are found and pulled in

- False positives add a lot of noise to the results
Image Query Expansion - Baseline

Query Result

Average Top M Results

Re-Query

Corpus
Transitive Closure
Average Query Expansion
Recursive Average Query Expansion

Query Result

Average Top M Spatially Consistent Results

Re-Query

Corpus
Multiple Image Resolution Expansion
Query Expansion

Query image

Originally retrieved

 Retrieved only after expansion
Demo

http://arthur.robots.ox.ac.uk:8080/search/?id=oxc1_hertford_000011
Results - PR Curves Before & After Expansion
Results - Effect Of Distractors

- My distractors: 9K images from searches like "building", "cathedral", "library", "historic", "spire" etc.

<table>
<thead>
<tr>
<th>Ground truth</th>
<th>OK</th>
<th>Junk</th>
<th>(\text{Oxford + Flickr1 dataset})</th>
<th>(\text{Oxford + Flickr1 + Flickr2 dataset})</th>
<th>(\text{Oxford + mine})</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Souls</td>
<td>78</td>
<td>111</td>
<td>ori, qeb, trc, avg, rec, sca</td>
<td>ori, qeb, trc, avg, rec, sca</td>
<td>78.1</td>
</tr>
<tr>
<td>Ashmolean</td>
<td>25</td>
<td>31</td>
<td>53.8, 52.4, 44.2, 63.9, 74.5, 71.2</td>
<td>40.1, 39.4, 39.6, 55.5, 67.6, 60.0</td>
<td>54.8</td>
</tr>
<tr>
<td>Balliol</td>
<td>12</td>
<td>18</td>
<td>50.4, 52.4, 44.2, 63.9, 74.5, 71.2</td>
<td>40.1, 39.4, 39.6, 55.5, 67.6, 60.0</td>
<td>53.0</td>
</tr>
<tr>
<td>Bodleian</td>
<td>24</td>
<td>30</td>
<td>42.3, 47.4, 49.3, 57.6, 48.6, 53.3</td>
<td>32.3, 36.9, 43.5, 46.8, 43.8, 44.9</td>
<td>38.2</td>
</tr>
<tr>
<td>Christ Church</td>
<td>78</td>
<td>133</td>
<td>53.7, 36.3, 56.2, 63.1, 63.3, 63.1</td>
<td>52.6, 18.9, 55.2, 61.0, 57.4, 57.7</td>
<td>53.6</td>
</tr>
<tr>
<td>Cornmarket</td>
<td>9</td>
<td>13</td>
<td>54.1, 60.4, 58.2, 74.7, 74.9, 83.1</td>
<td>42.2, 53.4, 56.0, 65.2, 68.1, 74.9</td>
<td>83.4</td>
</tr>
<tr>
<td>Hertford</td>
<td>24</td>
<td>31</td>
<td>69.8, 74.4, 77.4, 89.9, 90.3, 97.9</td>
<td>64.7, 70.7, 75.8, 87.7, 87.7, 94.9</td>
<td>53.6</td>
</tr>
<tr>
<td>Keble</td>
<td>7</td>
<td>11</td>
<td>79.3, 59.6, 64.1, 90.2, 100, 97.2</td>
<td>55.0, 15.6, 57.3, 67.4, 65.8, 65.0</td>
<td>42.8</td>
</tr>
<tr>
<td>Magdalen</td>
<td>54</td>
<td>103</td>
<td>9.5, 6.9, 25.2, 28.3, 41.5, 33.2</td>
<td>5.4, 0.2, 16.9, 15.7, 31.3, 26.1</td>
<td>10.3</td>
</tr>
<tr>
<td>Pitt Rivers</td>
<td>7</td>
<td>9</td>
<td>100, 100, 100, 100, 100, 100</td>
<td>100, 90.2, 100, 100, 100, 100</td>
<td>40.2</td>
</tr>
<tr>
<td>Radcliffe Cam.</td>
<td>221</td>
<td>348</td>
<td>50.5, 59.7, 88.0, 71.3, 73.4, 91.9</td>
<td>44.2, 56.8, 86.8, 70.5, 72.5, 91.3</td>
<td>82.1</td>
</tr>
<tr>
<td>Total</td>
<td>539</td>
<td>838</td>
<td>55.0, 52.9, 63.5, 71.1, 75.2, 78.2</td>
<td>46.5, 40.5, 59.7, 63.1, 67.0, 69.6</td>
<td>64.7</td>
</tr>
</tbody>
</table>