Paper Presentation

Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval

02/12/2015

-Bhavin Modi

Outline

- Image description
- Scaling up visual vocabularies
- Search
- Spatial Verification
- Query expansion
- Methods
- Experiments
- Conclusion

General Approach

- Given a query region, search the corpus and retrieve a set of image regions that match the query object. Use bag-of-visual-words retrieval together with spatial verification.
- Combine the retrieved regions, along with the original query, to form a richer latent model of the object of interest.
- Improve query expansion by…
  - …(1) using spatial constraints to suppress false positives, and
  - …(2) constructing a latent feature model of the query object
- Re-query the corpus using this expanded model to retrieve an expanded set of matching regions.
- Repeat the process as necessary, alternating between model refinement and re-querying.

When do (images of) objects match?

- Two requirements:
  - “patches” (parts) correspond
  - Configuration (spatial layout) corresponds
- Can we use retrieval mechanisms from text retrieval?
  - Need a visual analogy of a textual word.
- Success of text retrieval
  - efficient
  - scalable
  - high precision

Slide credits: [5]

Object Retrieval

- Find multi-scale Hessian interest points and fit an affine invariant region to each using the semi-local second moment matrix.
- On average, there are 3,300 regions detected on an image of size 1024 × 768.
- For each of these affine regions, we compute 128-dimensional SIFT descriptors.

Bag of visual word

- Quantize the visual descriptor to index the image

- Representation is a sparse histogram for each image
- Similarity measure is L2 distance between tf-idf weighted histograms

Slide credits: [5]

Datasets

Dataset	Number of images	Number of features
Oxford	5,062	16,334,970
Flickr1	99,782	277,770,833
Flickr2	1,040,801	1,186,469,709
Total	1,145,645	1,480,575,512

Table 1. The number of descriptors for each dataset.

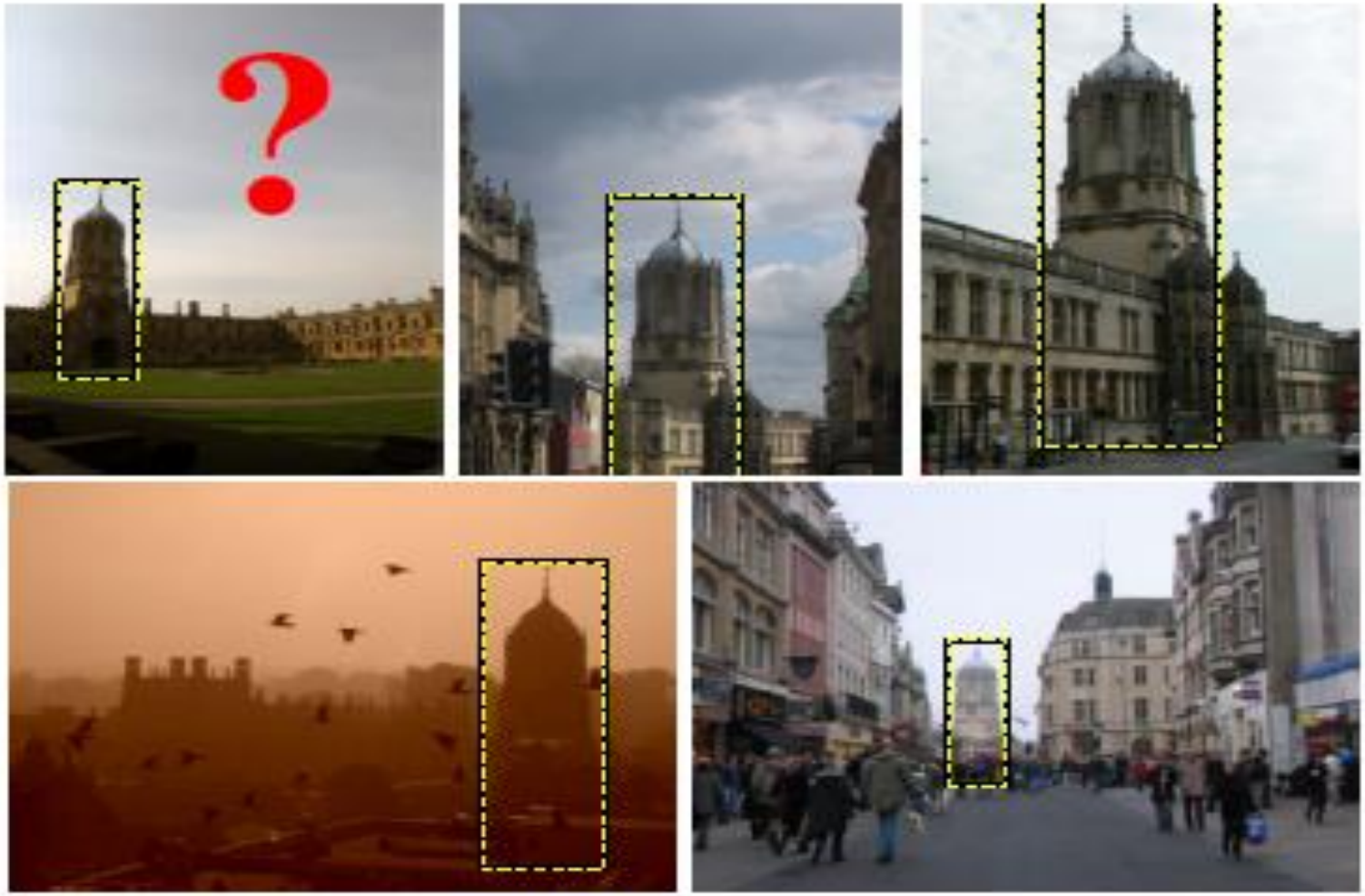


Figure 1. A sample of challenging results returned by our method in answer to a visual query for the *Tom Tower, Christ Church College, Oxford* (top left), which weren't found by a simple bag-of-visual-words method. This query was performed on a large dataset of 1,145,645 images.

Train the dictionary

- K-means is far too slow for our needs
  - D~128, N~20M+, K~1M
- Approximate k-means (AKM)
  - Reduce the number of candidates of nearest cluster heads
  - Approximate nearest neighbor
    - Use multiple, randomized k-d trees for search
    - Points nearby in the space can be found by backtracking around the tree some small number of steps

Slide credits: [5]

Scaling up Visual Vocabulary

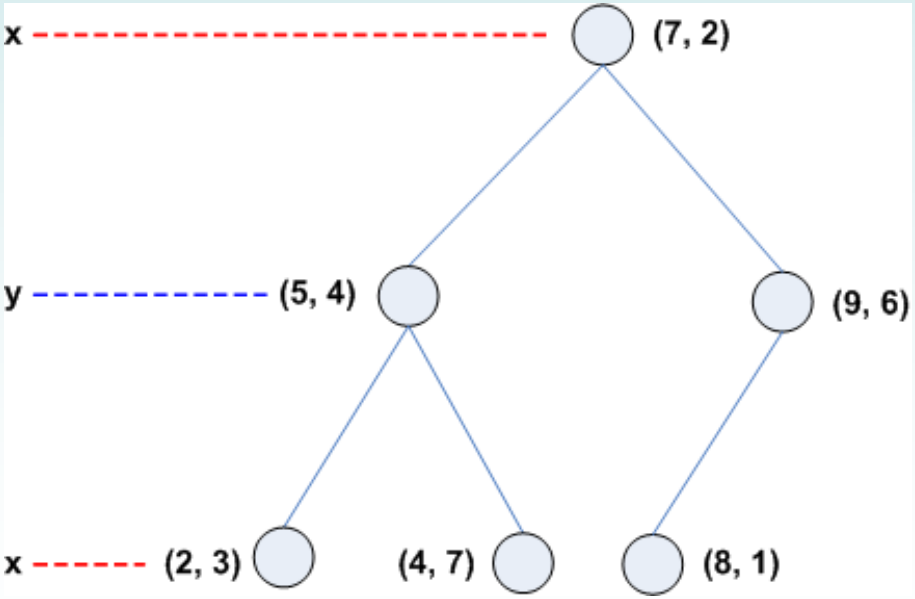
- A visual vocabulary of 1M words is generated using an approximate K-means clustering method based on randomized trees.
- This produces visual vocabularies which perform as well as those generated by exact Kmeans at a fraction of the computational cost.
- Each visual descriptor is assigned, via approximate nearest neighbour search, to a single cluster centre, giving a standard bag-of visual-words model.
- These quantized visual features are then used to index the images for the search engine.

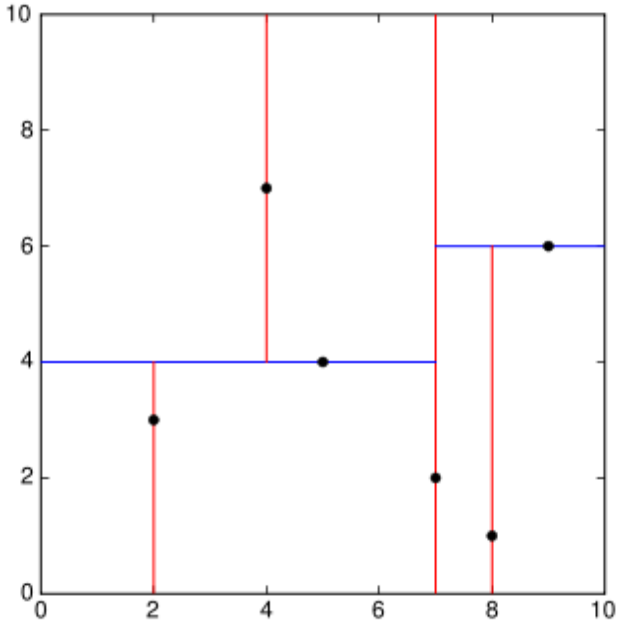
K-d Trees

- A ***k*-d tree** (short for *k-dimensional* tree) is a space-partitioning data structure for organizing points in a *k*-dimensional space.
- *k*-d trees are a useful data structure for several applications, such as searches involving a multidimensional search key (e.g. range searches and nearest neighbour searches).
- *k*-d trees are a special case of binary space partitioning trees in which every node is a k-dimensional point.

AKM

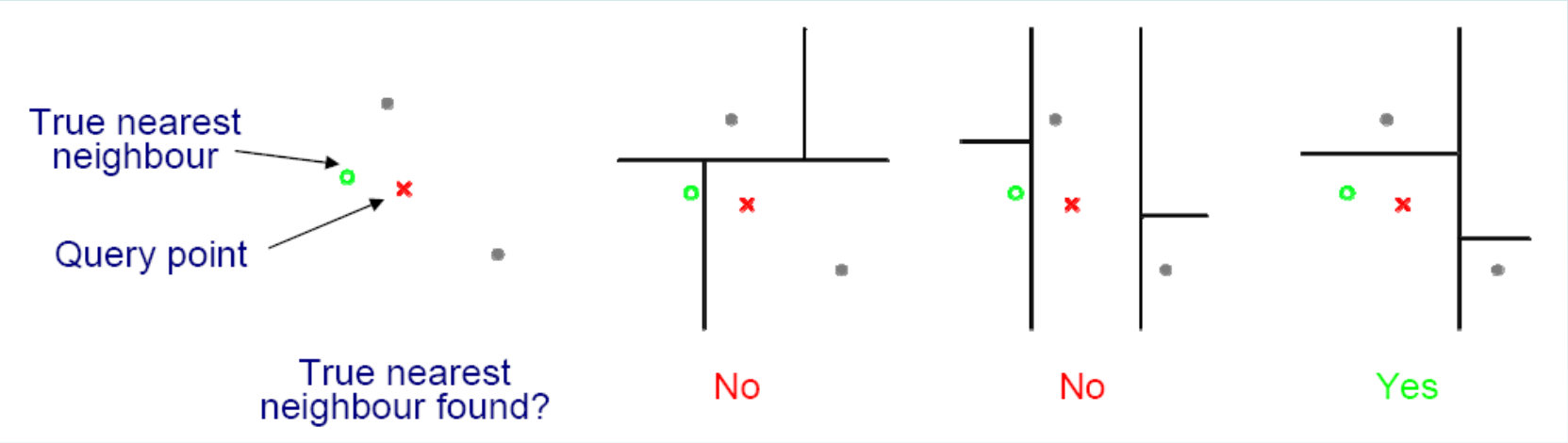
- Point List = [(2,3), (5,4), (9,6), (4,7), (8,1), (7,2)]
- Sorted List = [(2,3), (4,7), (5,4), (7,2), (8,1),(9,6)]





AKM

- Multiple randomized trees increase the chances of finding nearby points



- Original K-means complexity = O(N K)
- Approximate K-means complexity = O(N log K)
- This means we can scale to very large K

Search

- Search proceeds by calculating the similarity between the query vector and each document vector (Stored Vocabulary).
- We use the standard tf-idf weighting scheme, which down-weights the contribution that commonly occurring, and therefore less discriminative, words make to the relevance score.
- It maps individual words to the documents in which they occur.
- The memory usage problem is overcome by storing the inverted file in a space-efficient binary-packed structure.

Matching a query region

- Stage 1: generate a short list of possible frames using bag of visual word representation
  - Accumulate all visual words within the query region
  - Use “book index” to find other frames
  - Compute similarity for frames
  - tf-idf ranked list of all the frames in dataset

Slide credits: [5]

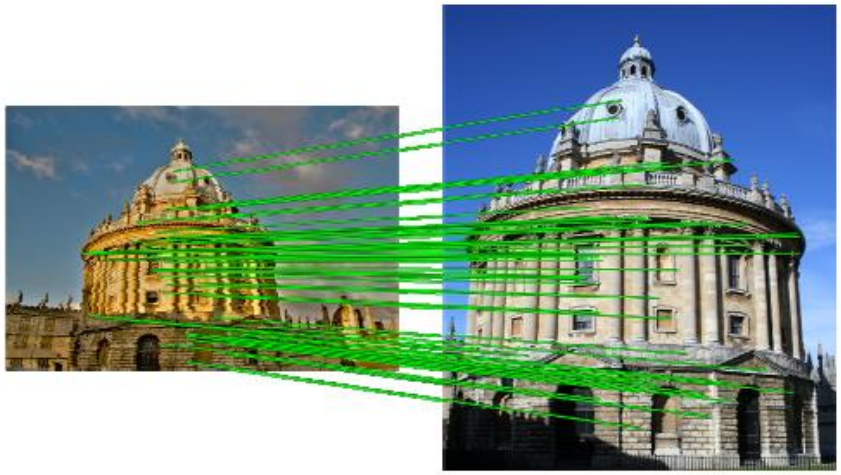
Response Set

- Images from the corpus that contain a large number of visual words in common with the query region
- May subsequently be ranked using spatial information to ensure that…
  - …(1) response and query contain similar features, and
  - …(2) features occur in compatible spatial configurations

Slide credits: Brandon Smith [6]

Beyond Bag of Words

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



Many spatially consistent matches – correct result

Few spatially consistent matches – incorrect result

Spatial Verification

- It is vital for query-expansion that we do not expand using false positives, or use features which occur in the result image, but not in the object of interest.
- So the authors use hypothesize and verify procedure to estimate homography between query and target.

Spatial Verification

- Usage
  - Re-ranking the top ranked results
- Procedure
  1. Estimate a transformation for each target image
  2. Refine the estimations
     - Reduce the errors due to outliers
       - RANSAC
         - » RANdom SAmple Consensus
  3. Re-rank target images
     - Scoring target images to the sum of the idf value for the inlier words
     - Verified images above unverified images

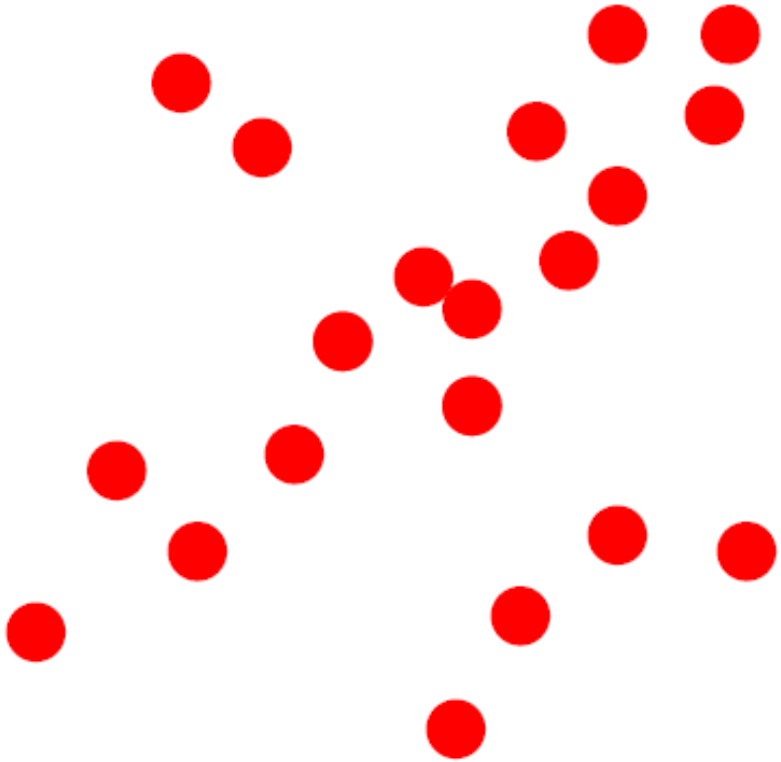
Spatial Verification

- Each interest point has an affine invariant semi-local region associated with it and we use this extra information to hypothesize transformations using single correspondences.
- RANSAC-like scoring mechanism is used to select the hypothesis with the greatest number of inliers.
- The spatial verification is applied up to a maximum of the top 1000 results returned from the search engine.
- Results with more than 20 inliers reliably contain the object being sought for. Such results are spatially verified.

RANSAC

(RANdom SAmple Consensus) :

Fischler & Bolles in '81.



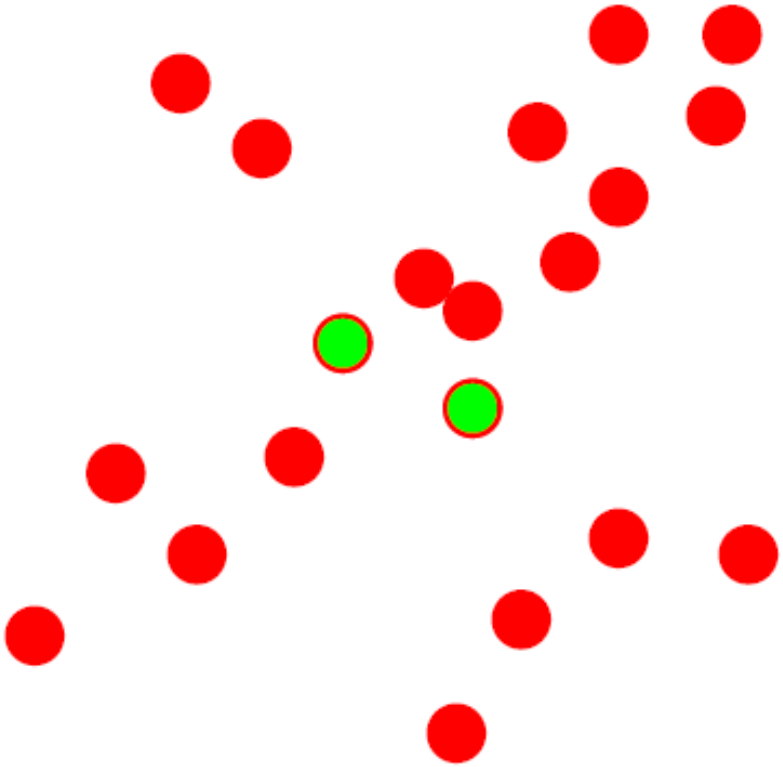
Algorithm:

1. **Sample** (randomly) the number of points required to fit the model
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

RANSAC

Line fitting example



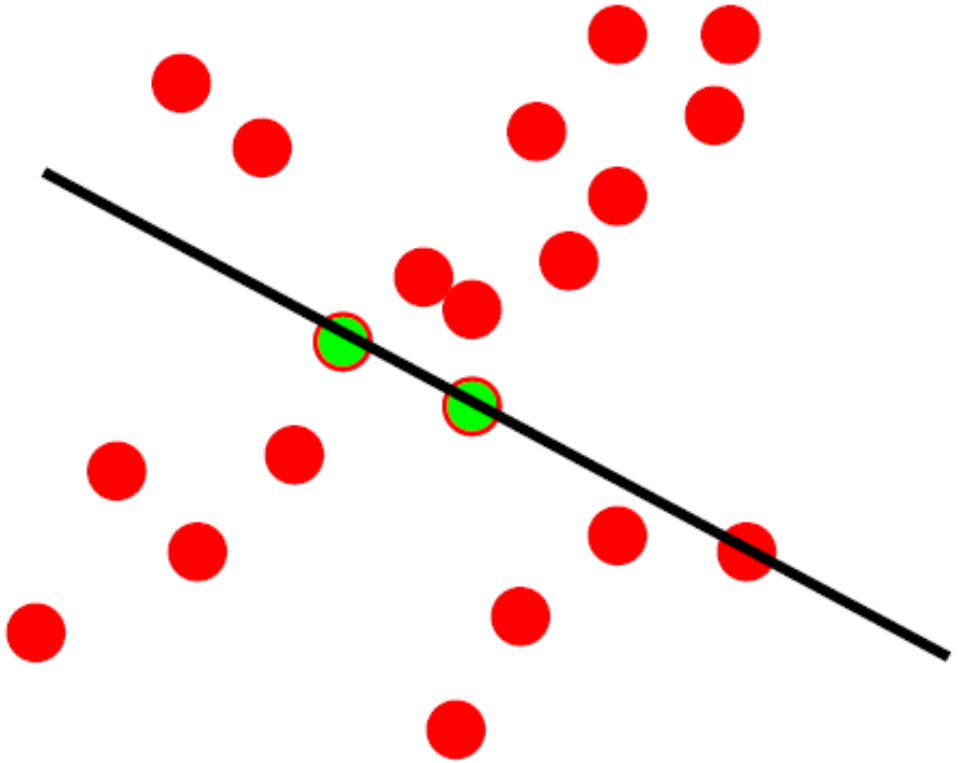
Algorithm:

1. **Sample** (randomly) the number of points required to fit the model (#=2)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

RANSAC

Line fitting example



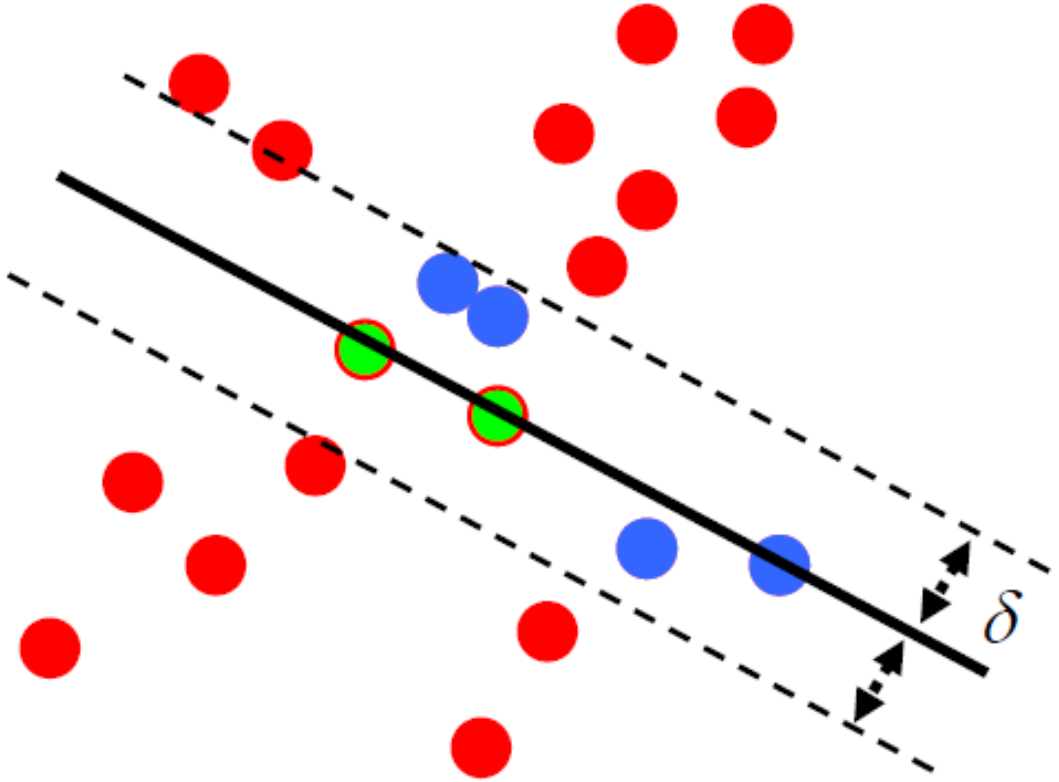
Algorithm:

1. **Sample** (randomly) the number of points required to fit the model (#=2)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

RANSAC

Line fitting example



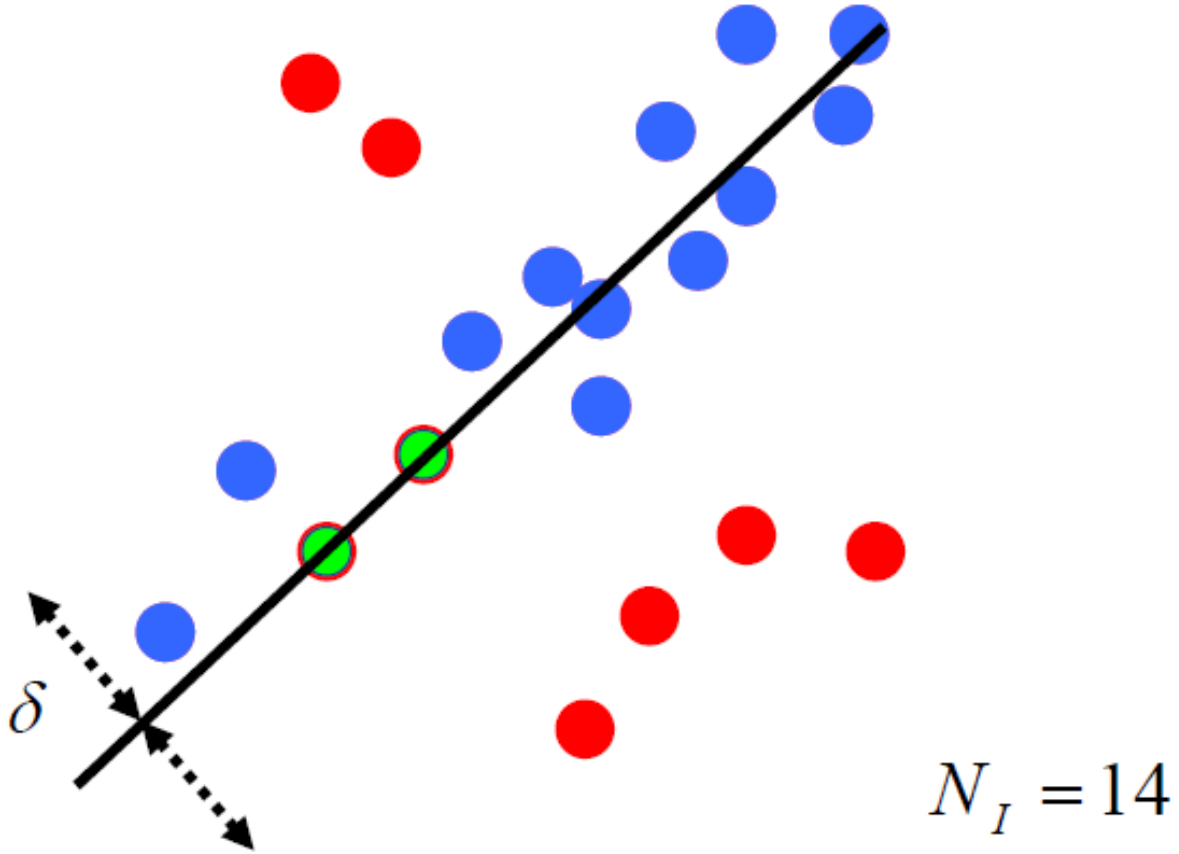
$$N_I = 6$$

Algorithm:

1. **Sample** (randomly) the number of points required to fit the model (#=2)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

RANSAC



Algorithm:

1. **Sample** (randomly) the number of points required to fit the model (#=2)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

Query Expansion

- Well-known technique in text-based information retrieval

(1) Generate a query
(2) Obtain an original response set
(3) Use a number of high ranked documents to generate a new query
(4) Repeat from (2)…

Slide credits: Brandon Smith [6]

Generative Model

- Extract a *generative model* of an object from the query region
- Form the *response set* from images in the corpus
- A spatial configuration of visual words extracted from the query region, plus…
- A “background” distribution of words that encodes the overall frequency statistics of the corpus.

Slide credits: Brandon Smith [6]

Latent Model

- Keep the form fixed (still a configuration of visual words)
- Enrich the object model with additional information from the corpus
- Refer to this as a *latent model*
- Well suited to this problem domain:

  (1) Spatial structure can be used to avoid false positives.

  (2) The baseline image search *without* query expansion suffers more from false negatives than most text retrieval systems.

Methods for Computing Latent Object Models

To address two issues:

(i) how far should this sequence extend – should a new latent model be built from the returns of Q1 and another query issued, etc?

(ii) (ii) how should the ranked lists returned from Q0,Q1, … be combined?

- ❑ Query expansion baseline
- ❑ Transitive closure expansion
- ❑ Average query expansion
- ❑ Recursive average query expansion
- ❑ Multiple image resolution expansion

Slide credits: Brandon Smith [6]

Query expansion baseline

1. Find top 5 (unverified) results from original query
2. Average the term-frequency vectors
3. Requery once
4. Append these results

Slide credits: [5]

Transitive closure expansion

1. Create a priority queue based on number of inliers
2. Get top image in the queue
3. Find region corresponding to original query
4. Use this region to issue a new query
5. Add new *verified* results to queue
6. Repeat until queue is empty

Slide credits: [5]

Average query expansion

1. Obtain top (m < 50) *verified* results of original query
2. Construct new query using average of these results

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

3. Requery once
4. Append these results

Slide credits: [5]

Recursive average query expansion

1. Improvement of average query expansion method
2. Recursively generate queries from all verified results returned so far
3. Stop once 30 verified images are found or once no new images can be positively verified

Slide credits: [5]

Multiple image resolution expansion

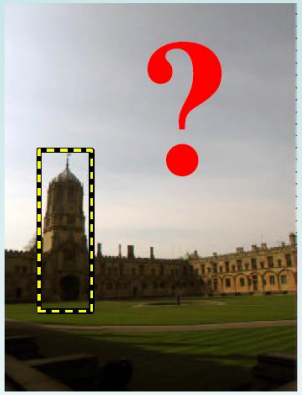
1. For each verified result of original query calculate relative change in resolution required to project verified region onto query region
2. Place results in 3 bands:

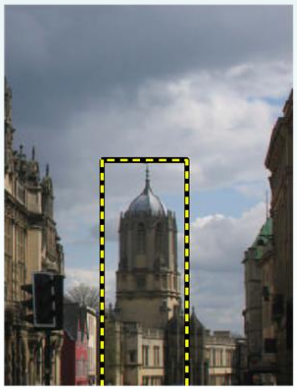
   (0, 4/5) (2/3, 3/2) (5/4, infinity)

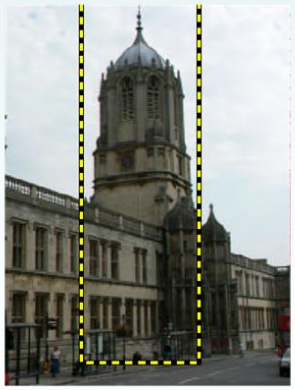
3. Construct average query for each band
4. Execute independent queries
5. Merge results
   1. Verified images from first query first, then…
   2. Expanded queries in order of number of inliers

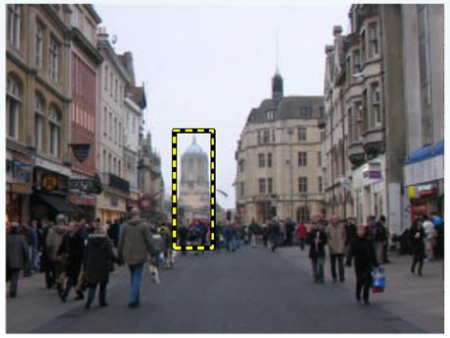
Slide credits: [5]

Query Expansion









Slide credits: Brandon Smith [6]

Experiment

- Datasets
  - Oxford buildings dataset
    - ~5,000 high resolution (1024x768) images
    - Crawled from Flickr using 11 landmarks as queries

Slide credits: Brandon Smith [6]

Datasets

- Ground truth labels
  - *Good* – a nice, clear picture of landmark
  - *OK* – more than 25% of object visible
  - *Bad* – object not present
  - *Junk* – less than 25% of object visible

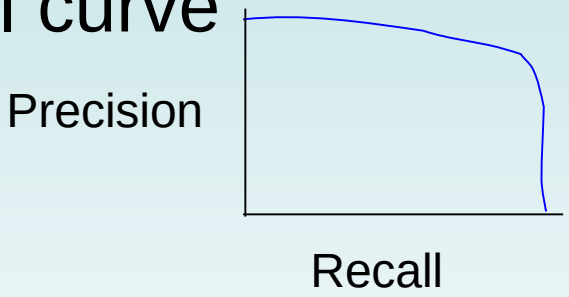
– Flickr1 dataset
  - ~100k high resolution images
  - Crawled from Flickr using 145 most popular tags

– Flickr2 dataset
  - ~1M medium resolution (500 x 333) images
  - Crawled from Flickr using 450 most popular tags

Slide credits: Brandon Smith [6]

Evaluation Procedure

- Compute Average Precision (AP) score for each of the 5 queries for a landmark
  - Area under the precision-recall curve
    - Precision = RPI / TNIR
    - Recall = RPI / TNPC



RPI = retrieved positive images

TNIR = total number of images retrieved

TNPC = total number of positives in the corpus

- Average these to obtain a Mean Average Precision (MAP) for the landmark
- Use two databases
  - D1: Oxford + Flickr1 datasets (~100k)
  - D2: D1 + Flickr1 datasets (~1M)

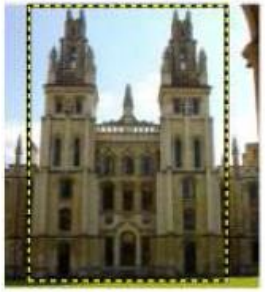
Slide credits: Brandon Smith [6]

Method and Dataset Comparison

	Ground truth		*Oxford* + *Flickr1* dataset						*Oxford* + *Flickr1* + *Flickr2* dataset					
	OK	Junk	ori	qeb	trc	avg	rec	sca	ori	qeb	trc	avg	rec	sca
All Souls	78	111	41.9	49.7	85.0	76.1	85.9	**94.1**	32.8	36.9	80.5	66.3	73.9	**84.9**
Ashmolean	25	31	53.8	35.4	51.4	66.4	74.6	**75.7**	41.8	25.9	45.4	57.6	**68.2**	65.5
Balliol	12	18	50.4	52.4	44.2	63.9	**74.5**	71.2	40.1	39.4	39.6	55.5	**67.6**	60.0
Bodleian	24	30	42.3	47.4	49.3	**57.6**	48.6	53.3	32.3	36.9	43.5	**46.8**	43.8	44.9
Christ Church	78	133	53.7	36.3	56.2	63.1	**63.3**	63.1	52.6	18.9	55.2	**61.0**	57.4	57.7
Cornmarket	9	13	54.1	60.4	58.2	74.7	74.9	**83.1**	42.2	53.4	56.0	65.2	68.1	**74.9**
Hertford	24	31	69.8	74.4	77.4	89.9	90.3	**97.9**	64.7	70.7	75.8	87.7	87.7	**94.9**
Keble	7	11	79.3	59.6	64.1	90.2	**100**	97.2	55.0	15.6	57.3	**67.4**	65.8	65.0
Magdalen	54	103	9.5	6.9	25.2	28.3	**41.5**	33.2	5.4	0.2	16.9	15.7	**31.3**	26.1
Pitt Rivers	7	9	**100**	**100**	**100**	**100**	**100**	**100**	**100**	90.2	**100**	**100**	**100**	**100**
Radcliffe Cam.	221	348	50.5	59.7	88.0	71.3	73.4	**91.9**	44.2	56.8	86.8	70.5	72.5	**91.3**
Total	539	838	55.0	52.9	63.5	71.1	75.2	**78.2**	46.5	40.5	59.7	63.1	67.0	**69.6**

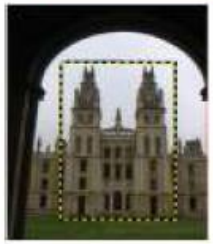
Table 2. Summary of ground truth, and the relative performance of the different expansion methods.

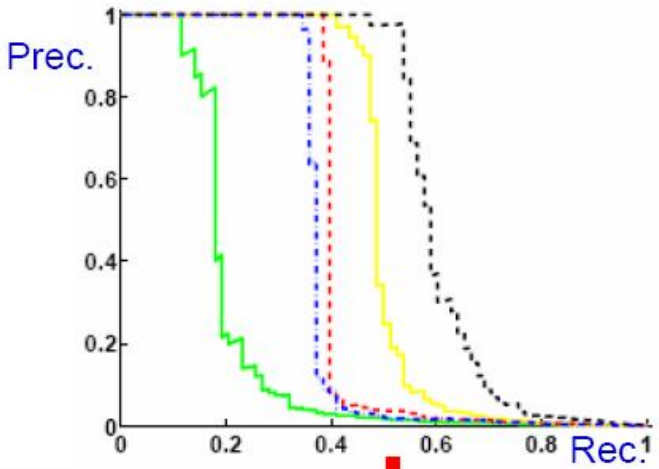
ori = original query
qeb = query expansion baseline
trc = transitive closure expansion
avg = average query expansion
rec = recursive average query expansion
sca = multiple image resolution expansion



Query image

Original results (good)

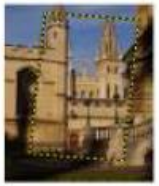



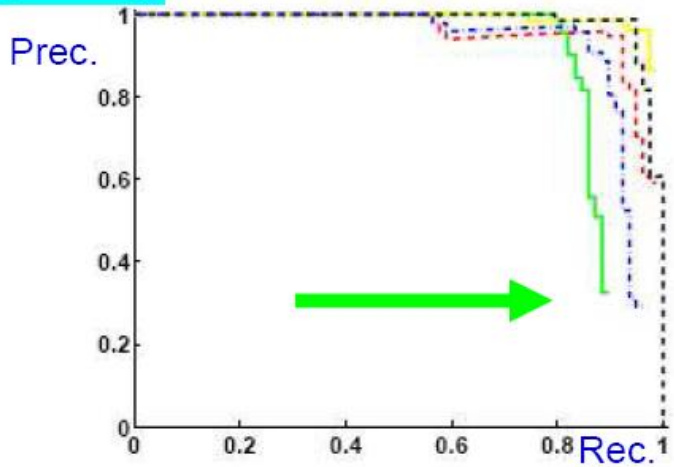


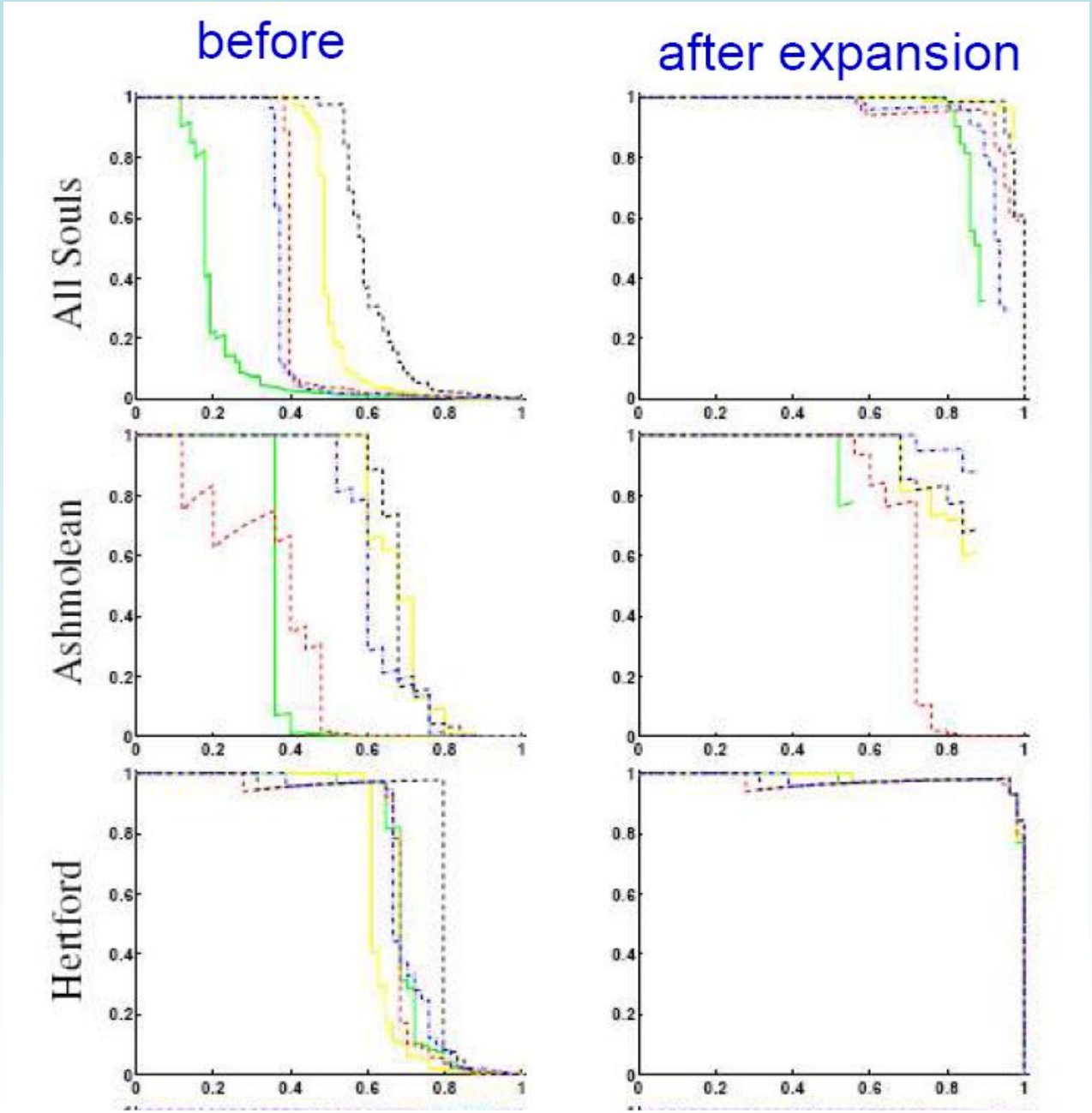
Expanded results (better)

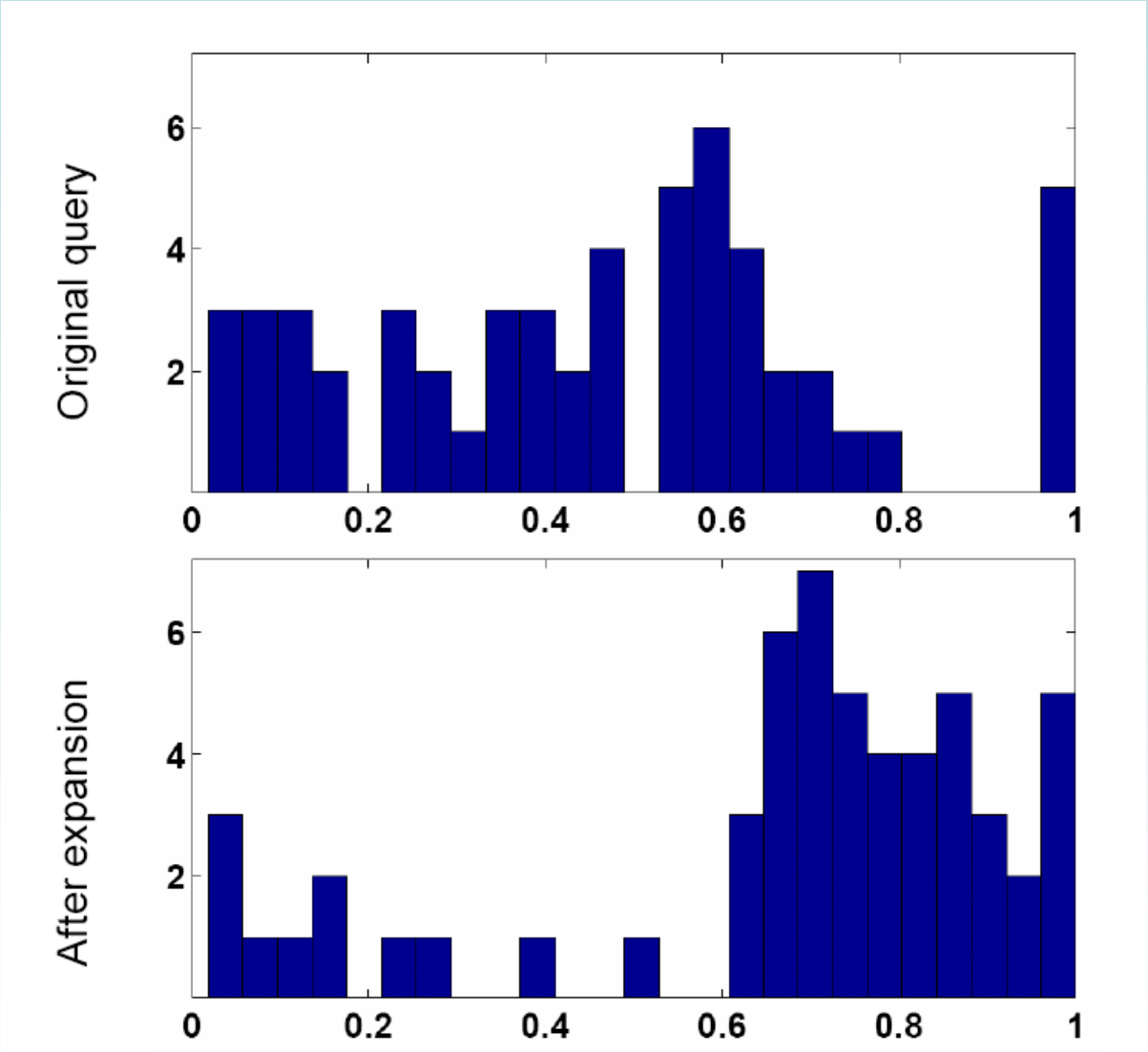







before
after expansion
All Souls
Ashmolean
Hertford



Average Precision histogram for 55 queries

Retrieval Performance

Figure 5. Some false positive images for Magdalen Tower query. The tower shown is actually part of Merton College chapel.

Retrieval Performance

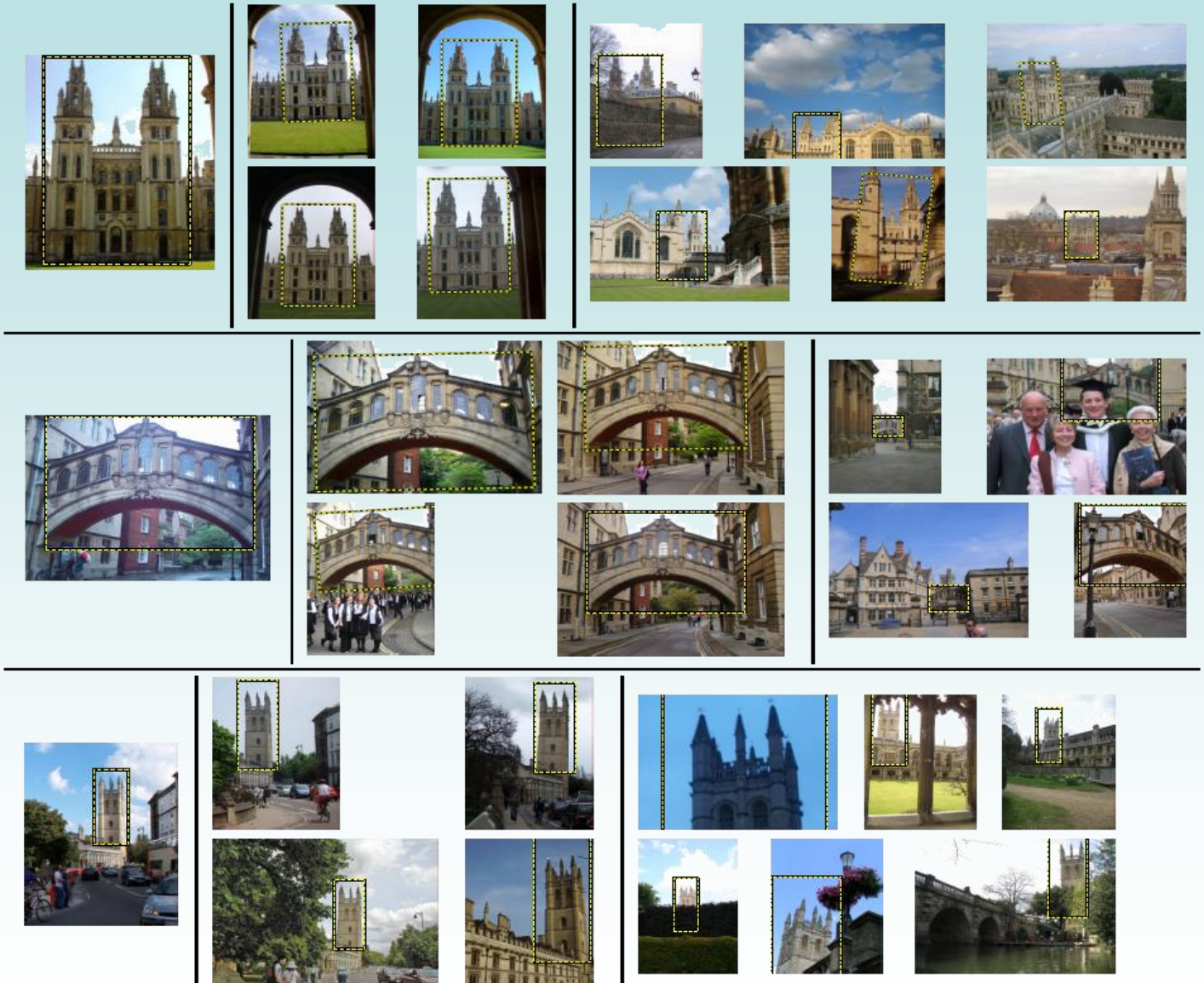


Figure 6. Demonstrating the performance of the method on a number of different queries. The image to the left shows the original query image. The four images in the middle show the first four results returned by the original query before query expansion. The images to the right show true positive images returned after query expansion which were not found from the bag-of-words method.

Conclusion

- Have successfully ported methods from text retrieval to the visual domain:
  - Visual words enable posting lists for efficient retrieval of specific objects
  - Spatial re-ranking improves precision
  - Query expansion improves recall, without drift
- Outstanding problems:
  - Include spatial information into index
  - Universal vocabularies

Slide credits: Brandon Smith [6]

Future Work

Total Recall 2: The authors introduce three extensions to automatic query expansion:

(i) A method capable of preventing tf-idf failure caused by the presence of sets of correlated features (confusers)

(ii) An improved spatial verification and re-ranking step that incrementally builds a statistical model of the query object and

(iii) Learning relevant spatial context to boost retrieval performance.

World-Scale Mining of Objects and Events From Community Photo Collections

About

- Approach for mining images of objects (such as touristic sights) from community photo collections in an unsupervised fashion using geotagged photos from Flickr and creating a grid of geospatial tiles.
- Several modalities, including visual, textual and spatial proximity are use of clustering these images.
- A final verification step uses the contents (including images) from the selected Wikipedia article to verify the cluster-article assignment.
- We demonstrate this approach on several urban areas, densely covering an area of over 700 square kilometres and mining over 200,000 photos, making it probably the largest experiment of its kind to date.

Motivation

Deal with a crucial but often neglected building block towards Internet-scale image retrieval:

The automated collection of a high quality image database with correct annotations.

For each mined item, a textual description is automatically derived in an unsupervised manner. The resulting “cleaned” image database for the mined objects and events is of far higher quality than the original data and facilitates

a variety of applications.

For example, the mined structure can be used for:

- Automated annotation of photos uploaded to community collections
- For retrieval and browsing of landmark buildings
- Automatic 3D reconstruction of sights
- For mobile phone tourist guide applications.

Approach

- Gathering the geotagged data from the WWW
- Clustering to group images of the same object/event
- Classification of clusters into objects or events
- Frequent itemset mining to derive cluster labels
- Unsupervised linking to Wikipedia and verification of those links

Gathering The Data

- Divide the earth's surface into square tiles T_k of about 200m side length. A tile centre is set every 100m.
- Query the Flickr API with the tile's centre coordinates and bounding box to obtain all geotagged photos for that area.
- The majority of tiles (about 52′000) were empty. The remaining tiles contained on average 10 and a maximum of 3750 photos.
- For each photo downloaded, the associated metadata, namely the textual descriptions (tags, title, description), user-id, and timestamps is also obtained. These are considered to allow for classifying each cluster candidate into event or object types.

Figure 1: Tiles over Paris. The size of a tile is marked in red. Note the overlap of 50% (100m).

Name	# tiles	#photos	area (km^2)
Munich	18'228	24'069	184.99
Oxford	2'112	7'431	22.05
Paris	12'532	87'452	127.57
Pisa	723	1'950	7.78
Prague	11'110	28'872	113.22
Rome	14'397	48'750	146.38
St. Petersburg	3'400	2'573	35.18
Tallinn	890	1'350	9.51
Venice	449	7'708	4.92
Zurich	5'663	12'602	58.15
Total	**69'504**	**222'757**	**709.74**

Table 1: Urban areas processed in this paper and the number of tiles and photos per area.

Photo Clustering

Visual Features and Similarity:

- Extract the visual features from each photo using 64-dimensional SURF feature vectors.
- Find matching features by calculating the nearest neighbour (NN) in Euclidean distance between all feature pairs, followed by a verification with the 2nd nearest neighbour criterion.

To find object candidates from the matching features the homography mappings for each matched image pair {i,j} are calculated,

$$Hx^i_n = x^j_n \text{ , } n \epsilon 2\ 1\ .\ .\ .\ 4$$

H is estimated using RANSAC discussed in the previous paper.

Photo Clustering

Test Features And Similarity:

Three sources for text meta-data were considered for each photo downloaded from flickr: tags, title, and description. These three text fields are combined into a single text per photo for further processing stages.

- The first stage consists of a stoplist.
- As with the visual features, the pairwise text similarities between the documents (photos) are calculated. A vector space model with term weighting of the following form is applied:

$$w_{i,j} = L_{i,j} * G_i * N_j$$

Different form the general tf-idf ranking approach.

Photo Clustering

Clustering Approach:

- For each tile T_k, hierarchical agglomerative clustering to the distance matrix of each modality is applied.
- Using different linking criteria for cluster merging allows to create different kinds of clusters. The linkage methods are:

$$\text{single-link:} \quad d_{AB} = \min_{i \in A, j \in B} d_{ij}$$

$$\text{complete-link:} \quad d_{AB} = \max_{i \in A, j \in B} d_{ij}$$

$$\text{average-link:} \quad d_{AB} = \frac{1}{n_i n_j} \sum_{i \in A, j \in B} d_{ij}$$

where A and B are the clusters to merge, and i and j index their n_i and n_j elements, respectively

Photo Clustering

- Singlelink clustering adds images to a cluster as long as they yield a good match to at least one cluster member.
- In contrast, complete-link clustering enforces that a new image matches to all cluster members.
- Average-link clustering, takes a compromise between those two extremes and provides clusters that still prefer views of the same object, while allowing more flexibility in viewpoint shifts.

Approach Used in the Paper

First identify distinct objects or landmark buildings through complete- or average-link clusters and later find out which of them are located close to each other by their membership in the same single-link cluster.

(a) (b) (c) (d)

Figure 4: Clusters around the Louvre: (a) shows single-link clusters, the photos of the cluster marked in yellow are located as shown in (b). (c) shows complete-link clusters for the same area, again with the photos of the yellow cluster in (d). (Only clusters with at least 4 elements are shown).

Labelling Clusters

Classifying into objects and events:

- "object" is defined as any rigid physical item with a fixed position, including landmark buildings, statues, etc.
- "events", occasions that took place at a specific time and location, for instance concerts, parties, etc.

Rely on the collected metadata for the photos in each cluster to include two new features,

$$f_1 = |D|$$
$$f_2 = \frac{|U|}{|N|}$$

where |D| is the number of days, |U| the number of users, and |N| the number of photos in the cluster. (Consider clusters with N>4)

Use **ID3** Decision Tree for training.

Precision achieved 88% for objects and 94% for events.

Linking to Wikipedia

- First finds relevant word combinations from the text associated with each cluster using a frequent **itemset mining algorithm**
- In the second step the resulting frequent combinations are then used to query Wikipedia.
- An image based matching step finally verifies that the links are indeed correct.

(a)

museum
museum louvre
carrousel
carrousel triomphe
carrousel triomphe arc
carrousel triomphe arc du
carrousel triomphe du
carrousel arc
carrousel arc du
carrousel du
triomphe
triomphe arc
triomphe arc du
triomphe du
arc
arc du

(b)

http://en.wikipedia.org/wiki/Arc_de_Triomphe_du_Ca.
http://en.wikipedia.org/wiki/Axe_historique
http://fr.wikipedia.org/wiki/Arc_de_Triomphe_du_Ca...
http://en.wikipedia.org/wiki/Quadriga
http://hu.wikipedia.org/wiki/Arc_de_Triomphe_du_Ca...
http://de.wikipedia.org/wiki/Arc_de_Triomphe_du_Ca...
http://nl.wikipedia.org/wiki/Arc_de_Triomphe_du_Ca...
http://it.wikipedia.org/wiki/Arc_de_Triomphe_du_Ca...
http://en.wikipedia.org/wiki/Triumphal_arch

(c)

(d)

(e)

(f) http://en.wikipedia.org/wiki/Arc_de_Triomphe_du_Carrousel

Figure 6: Matching clusters to Wikipedia articles. The text for the photos in a cluster (a) is mined for frequent word combinations (b), which are used to search Wikipedia for candidate URLs (c). Each image (d) of an article is in return matched to the images in the cluster. If a good match (e) can be found, the candidate link is selected (f).

Itemset Mining (Frequent Labels)

- Finding and trying all possible combinations would mean considering 2^N combinations of words, where N can easily be in the hundreds.
- The task consists of detecting rules in large numbers (millions) of customer transactions, where the rules describe the probability that a customer buys item(s) B, given that he has already item(s) A in his shopping basket.

$$supp(A) = \frac{|\{T \in D | A \subseteq T\}|}{|D|} \in [0, 1]$$

- An itemset A is called frequent in D if supp(A) >= smin , where smin is a threshold for the minimal support. (0.15, only top 15 are kept)
- The advantage of using itemset mining over other probabilistic method is its speed and scalability. Tens of thousands of word combinations can be processed in fractions of seconds.

Itemset Example

The support supp(X) of an itemset X is defined as the proportion of transactions in the data set which contain the itemset.

In the example database, the itemset {milk, bread, butter} has a support of 1/5=0.2 since it occurs in 20% of all transactions (1 out of 5 transactions).

Example database with 4 items and 5 transactions

transaction ID	milk	bread	butter	beer
1	1	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0

The set of items is

I={milk, bread, butter, beer}

and a small database containing the items (1 codes presence and 0 absence of an item in a transaction) is shown in the table. An example rule for the supermarket could be

{butter, bread} → **{milk}** meaning that butter and bread are bought, customers also buy milk.

Results

- Around 220,000 images downloaded from flickr, along with their metadata.
- Using single-link clustering, the mean precision obtained was 98%. (Results seem to be aimed at only for the area around Pantheon in Rome)
- Using only text, resulted in a precision of 60%.
- Able to separate Objects and events accurately with results mentioned previously.
- In total, 861 unique Wikipedia articles were verified by matching their images to the clusters as described above. The precision of this assignment was about 94%, i.e. 94% of the articles referred to a cluster which contained images of the article's correct subject.

Figure 3: Clusters found around the Pantheon and the number of photos contained in each. Note the automatic separation into indoor (a), outdoor (b), and panorama views (e), and the discovery of separate objects (c,d). Mean locations of the photos are shown on the map. (e) is estimated at about the same position as (b) and is therefore not drawn on the map.

Questions

- Disadvantages?
- What do you think was the most interesting aspect of the paper?

References

1. Total Recall: Ondrej Chum, James Philbin, Josef Sivic, Michael Isard and Andrew Zisserman, University of Oxford, ICCV 2007
2. Philbin, J., Chum, O., Isard, M., Sivic, J. and Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching, CVPR 2007
3. Chum, O., Mikulik, A., Perdoch, M. and Matas, J.: Total Recall II: Query Expansion Revisited, CVPR 2011
4. Till Quack, Bastian Leibe and Luc Van Gool, World-scale Mining of Objects and Events from Community Photo Collections
   CIVR 2008, Niagara Falls, Canada, 7.-9. July 2008
5. http://vc.cs.nthu.edu.tw/home/paper/codfiles/wthuang/200808191838/Total_Recall_automatic_query_expansion_with_a_generative_feature_model_for_object_retrieval.ppt.
6. http://pages.cs.wisc.edu/~lizhang/courses/cs766-2007f/syllabus/11-29-search/smith.ppt.
7. http://en.wikipedia.org/wiki/K-d_tree.
8. D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In Proc. CVPR, 2006.