Three things everyone should know to improve object retrieval

Relja Arandjelović and Andrew Zisserman
Visual Geometry Group, Department of Engineering Science, University of Oxford

Objectives

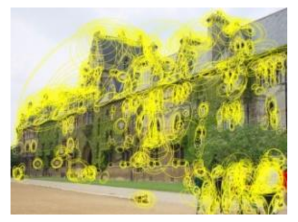
- Find all instances of an object in a large dataset
  - Do it instantly
  - Be robust to scale, viewpoint, lighting, partial occlusion

query image

[Lowe04, Mikolajczyk07]

Hessian-Affine regions +SIFT descriptors

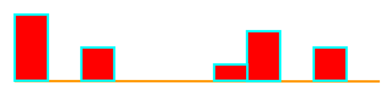
Set of SIFT descriptors



[Sivic03]

Visual words

sparse frequency vector + tfidf weighting



Inverted file

querying

ranked image short-list

[Lowe04, Philbin07]

Geometric verification

[Chum07]

Query expansion

1. RootSIFT

- **Not only specific to retrieval**
- **Everyone using SIFT can benefit**

- Hellinger or χ2 measures outperform Euclidean distance when comparing histograms, examples in image categorization, object and texture classification etc.
- SIFT is a histogram: can performance be boosted using a better distance measure?
- Hellinger kernel (Bhattacharyya's coefficient) for L1 normalized histograms x and y:

$$H(x, y) = \sum_{i=1}^{n} \sqrt{x_i y_i}$$

- Explicit feature map of x into x' :
  - L1 normalize x
  - element-wise square root x to give x'

$\left.\right\}$ **RootSIFT**

- Computing Euclidean distance in the feature map space is equivalent to Hellinger distance in the original space
- Extremely simple to implement and use:
  - One line to convert SIFT to RootSIFT: **rootsift= sqrt(sift / sum(sift));**
- Conversion from SIFT to RootSIFT can be done on-the-fly
  - No need to modify your favourite SIFT implementation
  - No need to re-compute stored SIFT descriptors for large image datasets
  - No added storage requirements
  - Applications throughout computer vision
    k-means, approximate nearest neighbour methods, soft-assignment to visual words, Fisher vector coding, PCA, descriptor learning, hashing methods, product quantization etc.

- **Superior to SIFT in every single setting**
  - Large scale object retrieval

Retrieval method	Oxford 5k	Oxford 105k	Paris 6k
SIFT: tf-idf ranking	0.636	0.515	0.647
SIFT: tf-idf with spatial reranking	0.672	0.581	0.657
Philbin *et.al.* 2010 descriptor learning	0.707	0.615	**0.689**
RootSIFT: tf-idf ranking	0.683	0.581	0.681
RootSIFT: tf-idf with spatial reranking	**0.720**	**0.642**	**0.689**

  - Image classification: (Using the evaluation package of [Chatfield11])

Descriptor (dense + PHOW)	PASCAL VOC 2007
SIFT	0.5530
RootSIFT	0.5614

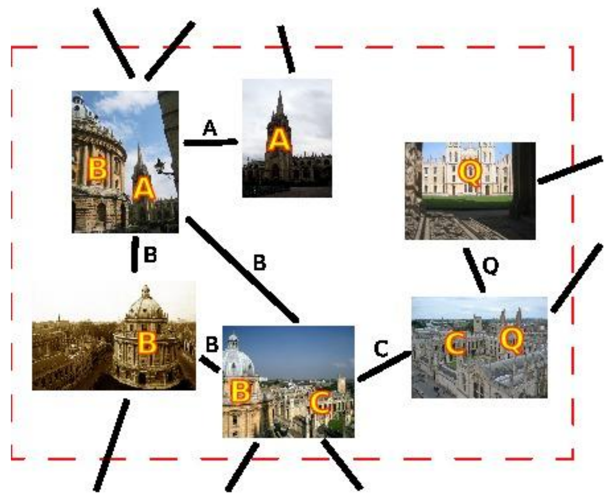
  - Repeatability under affine transformations

SIFT: 10 matches

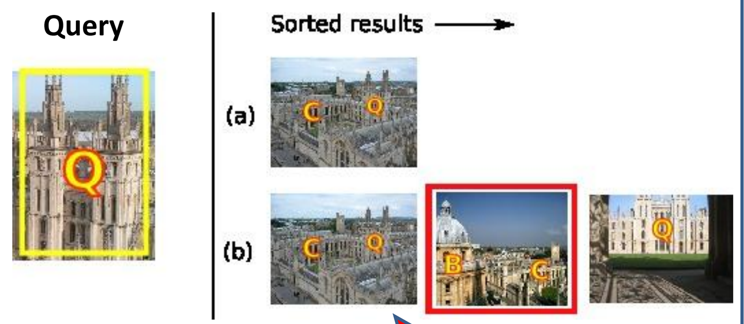
RootSIFT: 26 matches

2. Database-side feature augmentation

- Construct an image graph [Philbin08]
  - Nodes: images
  - Edges connect images containing the same object



- Obtain a better model for the database images [Turcot & Lowe 09] (AUG)
  - Augment database images with features from other images of the same object
  - Each image is augmented with all visual words from neighbouring images on graph



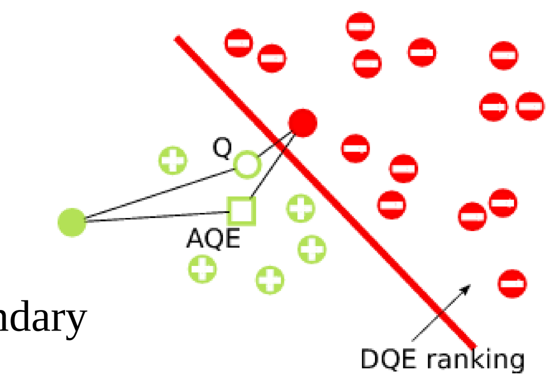
- Improves recall but precision is sacrificed
  - We propose spatial augmentation (SPAUG):
    - Only augment with *visible* visual words
    - 28% less features are augmented than in the original method

Retrieval method	Oxford 5k	Oxford 105k
tf-idf ranking	0.683	0.581
tf-idf with spatial reranking	0.720	0.642
AUG: tf-idf ranking	0.785	0.720
AUG: tf-idf with spatial reranking	0.827	0.759
Spatial AUG: tf-idf ranking	0.820	0.746
Spatial AUG: tf-idf with spatial reranking	**0.838**	**0.767**

* Uses RootSIFT

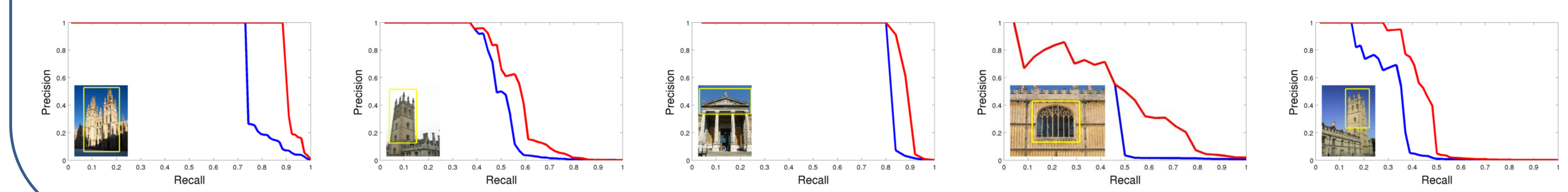
3. Discriminative query expansion

- Query expansion (QE)
  - BoW vectors from spatially verified regions are used to build a richer model for the query
  - The de facto standard: Average query expansion (AQE) [Chum07]:
    - Use the mean of the BoW vectors to re-query
- Discriminative query expansion (DQE):
  - Train a linear SVM classifier
  - Use query expanded BoW vectors as positive training data
  - Use low ranked images as negative training data
  - Rank images on their signed distance from the decision boundary



- DQE is efficient:
  - Ranking images using inverted index (as in average QE case)
  - Both operations are just scalar products between a vector and x
  - For average QE the vector is the average query idf-weighted BoW vector
  - For discriminative QE the vector is the learnt weight vector w
  - Training the linear SVM on the fly takes negligible amount of time (30ms on average)
- Significant boost in performance at no added cost, mAP on Oxford105k:

Retrieval method	SIFT	RootSIFT
tf-idf with spatial reranking	0.581	0.642
Chum *et.al.* 2007: Average Query Expansion (AQE)	0.726	0.756
Discriminative Query Expansion (DQE)	**0.752**	**0.781**

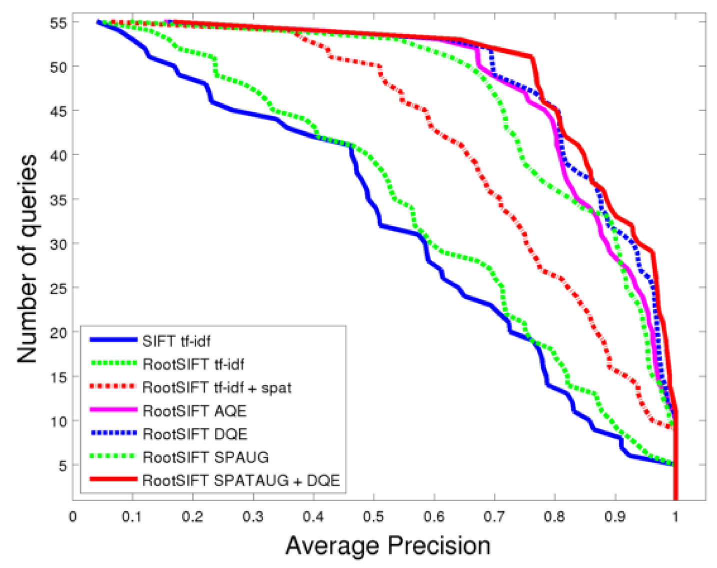


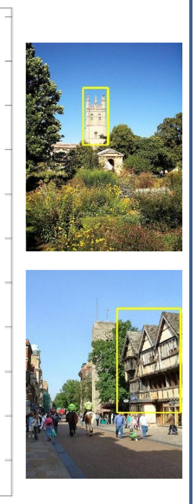
Results

- Combine all three improvements into one retrieval system
- New state of the art on all three datasets (without soft assignment!):

Oxford 5k	Oxford 105k	Paris 6k
0.929	0.891	0.910

- Quite close to total recall on Oxford 105k





Summary

1. RootSIFT:
   - Improves performance in every single experiment
   - Every system which uses SIFT is ready to use RootSIFT
   - Easy to implement, no added computational or storage cost
2. Database-side feature augmentation:
   - Useful for increasing recall
   - Our extension improves precision but increases storage cost
3. Discriminative query expansion:
   - Consistently outperforms average query expansion
   - At least as efficient as average QE, no reasons not to use it