CS688/WST665: Web-Scale Image Retrieval

Recent Image Retrieval Techniques

Sung-Eui Yoon

(윤성의)

Course URL:

http://sglab.kaist.ac.kr/~sungeui/IR

Today

- Go over some of recent image retrieval techniques

Video Google: A Text Retrieval Approach to Object Matching in Videos

Josef Sivic and Andrew Zisserman

Robotics Research Group, Department of Engineering Science

University of Oxford, United Kingdom

ICCV 03

Citation: over 1300 at 2011

Motivations

- Retrieve key frames and shots of a video containing a particular object
- Investigate whether a text retrieval approach can be successful for object recognition

Viewpoint Invariant Description

- **Extract image patches and compute a SIFT descriptor for each region**

Visual Vocabulary

- Quantize descriptor vectors into clusters, which are visual ‘word’ for text retrieval
  - Performed with K-means clustering

- Produce about 6K and 10K clusters for Shape adapted and Maximally Stable regions, respectively
  - Chosen empirically to maximize retrieval results

KAIST

Distance Function

- Use Mahalanobis distance as the distance function for clustering:

$$d(\vec{x},\vec{y}) = \sqrt{(\vec{x}-\vec{y})^T S^{-1} (\vec{x}-\vec{y})}.$$

  , where S is covariance matrix
  - If S is the identify matrix, it reduces to Euclidean distance
  - Decorrelate components of SIFT
- Instead, Euclidean distance may be used

Visual Indexing

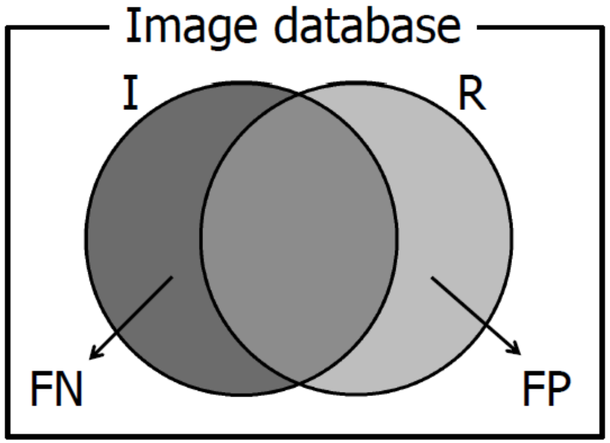
- **Each document is represented by k-vector** $(t_1, ..., t_i, ..., t_k)^\top$
- **Weighting by tf-idf**
  - **term frequency * log (inverse document frequency)**

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

  - **n_{id} : # of occurrences of word i in document d**
  - **n_d : total # of words in the document d**
  - **n_i : # of occurrences of term i in the whole database**
  - **N: # of documents in the whole database**
- **At the retrieval stage documents are ranked by their normalized scalar product between query vector V_q and V_d in database**

Video Google [Sivic et al. CVPR 2003]

- **mAP: mean average precision**



I : ground truth set

R : result set

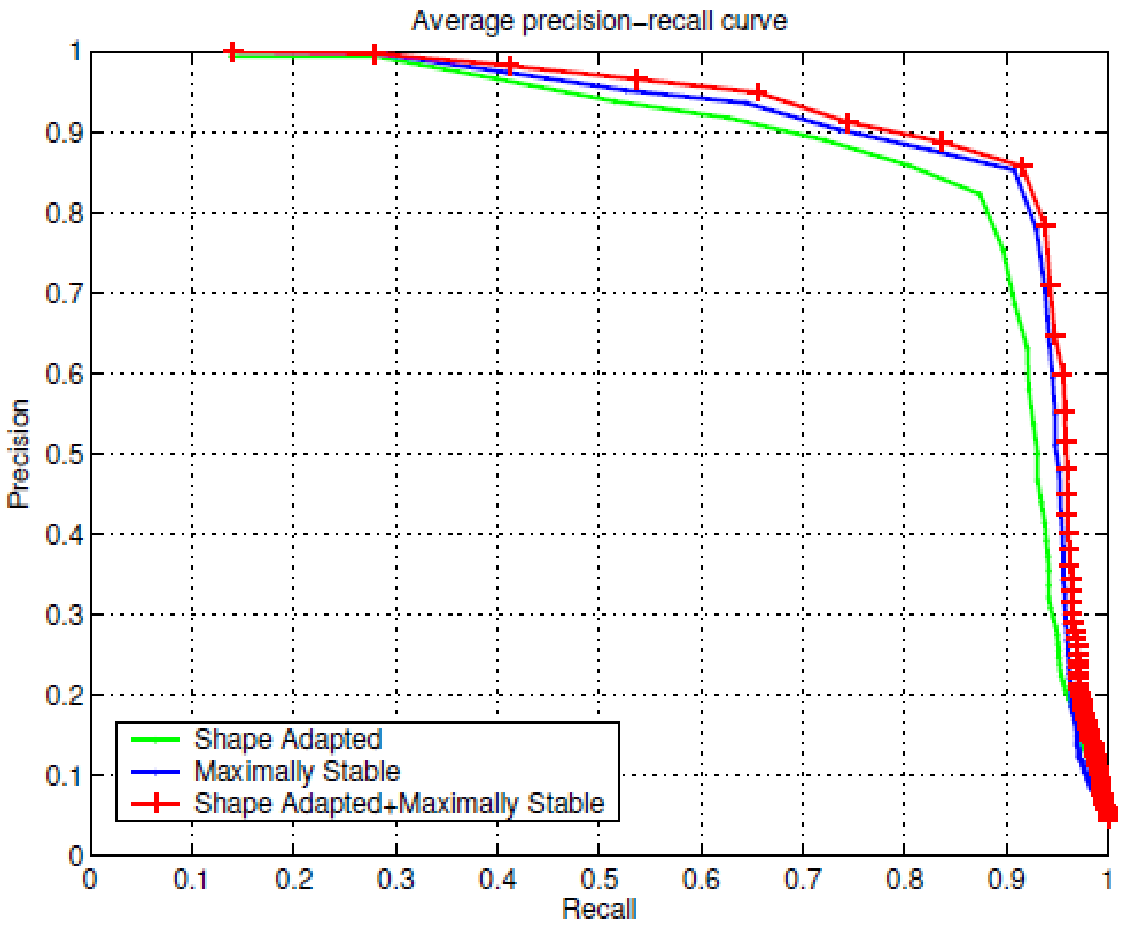
FN : false negative

FP : false positive

$$\text{Precision} = \frac{\#\text{of}\,(I \cap R)}{\#\text{of}\,(R)} \qquad \text{Recall} = \frac{\#\text{of}\,(I \cap R)}{\#\text{of}\,(I)}$$

Video Google [Sivic et al. CVPR 2003]

- **mAP: mean average precision**



Video Google [Sivic et al. CVPR 2003]

- **Performance highly depended on number of k(visual words) : not scalable**

Scalable Recognition with a Vocabulary Tree

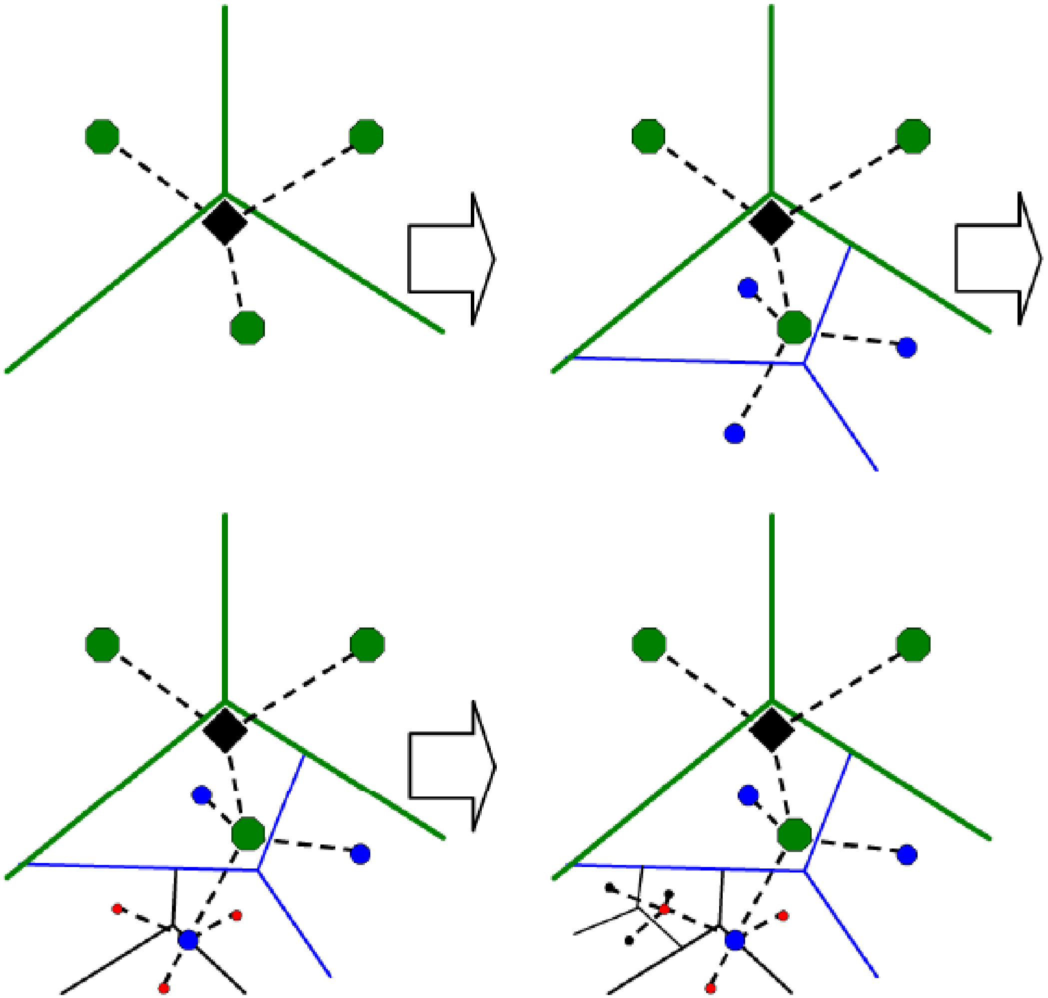
David Niter et al.

CVPR 2006

Citation: over 1000 at 2011

Vocabulary Tree [Nister et al. CVPR 06]

- **Hierarchical k-means clustering**



Vocabulary tree with branch factor 10

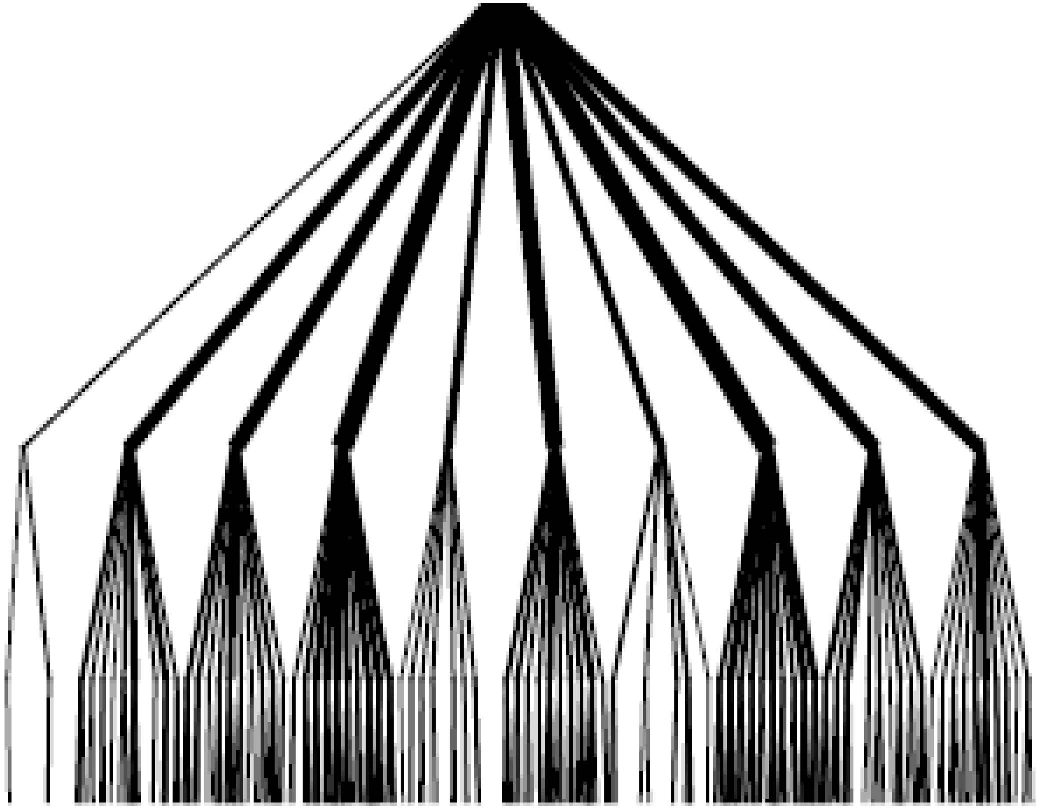


Figure 3. Three levels of a vocabulary tree with branch factor 10 populated to represent an image with 400 features.

Inverted File

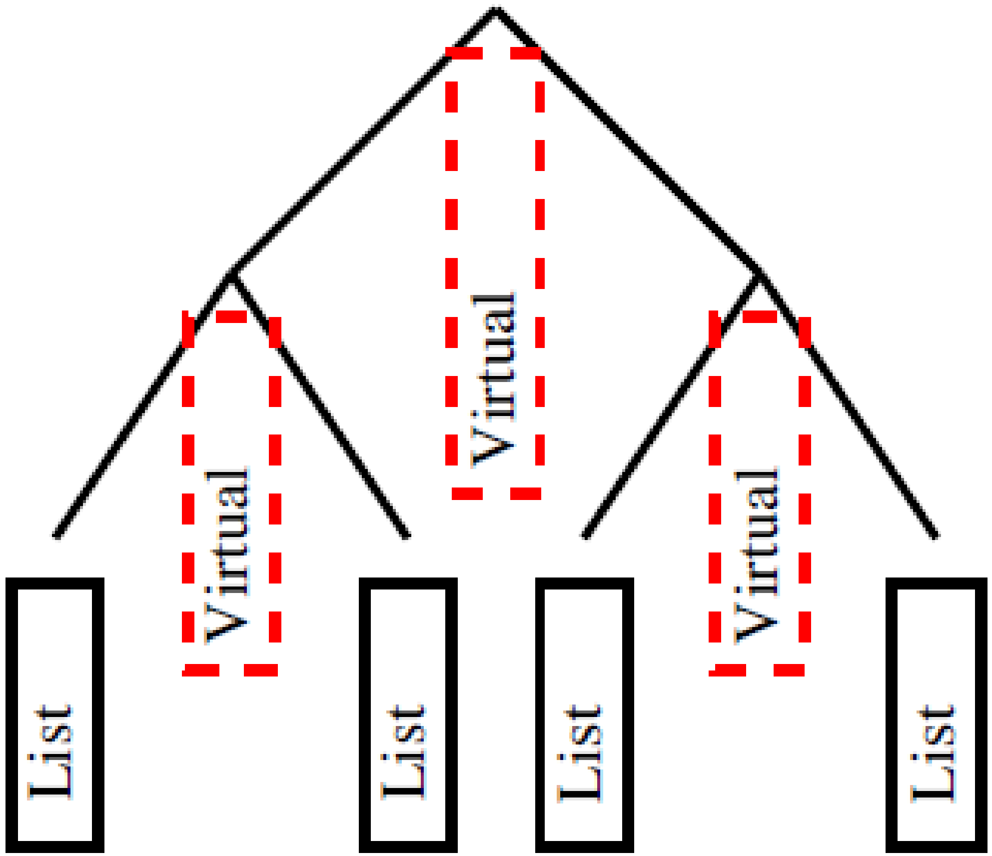
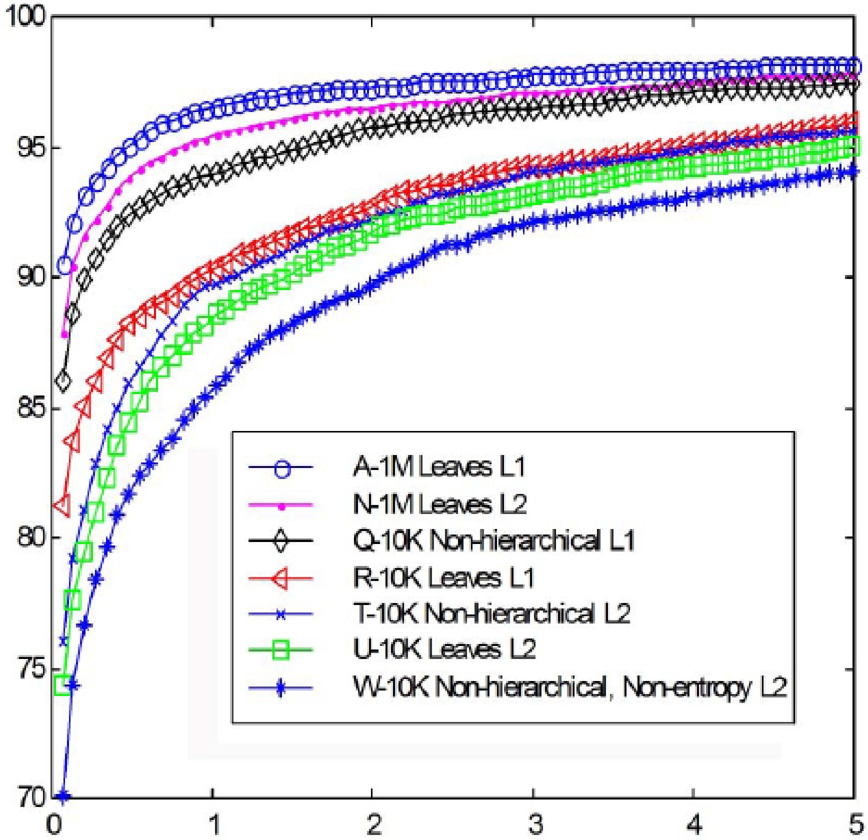


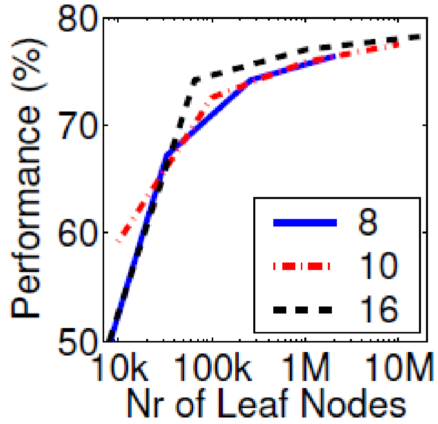
Figure 4. The database structure shown with two levels and a branch factor of two. The leaf nodes have explicit inverted files and the inner nodes have virtual inverted files that are computed as the concatenation of the inverted files of the leaf nodes.

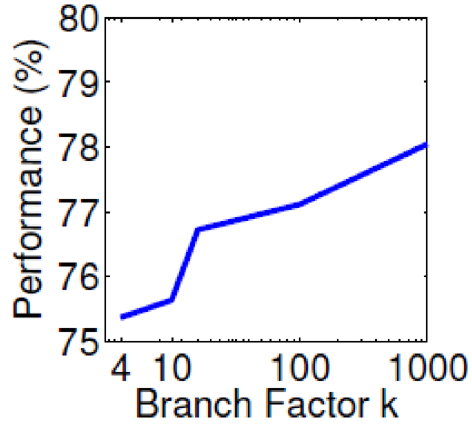
Retrieval Algorithm

- **Compute a histogram of visual words with SIFTs**
- **Identify images that contain words of the input query image**
  - **Can be done with the inverted file**
- **Sort images based on a similarity function**

Vocabulary Tree [Nister et al. CVPR 06]







- **On 8GB RAM machine(40000 images)queries took 1s, database creation took 2.5 days**

Vocabulary Tree

- **Benefits:**
  - **Allow faster image retrieval (and pre-computation)**
  - **Scales efficiently to a large number of images**
- **Problems:**
  - **Too much memory requirement**
  - **Quantization effects**

Object retrieval with large vocabularies and fast spatial matching

Philbin et al.

CVPR 2007

Citation: over 350 at 2011

KAIST

Approximating K-means

- Use a forest of 8 randomized k-d trees
  - Randomize splitting dimension among a set of the dimensions with highest variance
  - Randomly choose a point close to the median for split value
  - Helps to mitigate quantization effects
- Each tree is descending to leaf, distance from boundaries are recorded in a prior queue
  - Similar to best-bin-first search

Approximate K-means

- **Algorithmic complexity of a single k-means iteration**
  - **Reduces from O(NK) to O(NlogK), where N is the # of features**
  - **Achieved by multiple random kd-trees**
- **Find images with kd-trees too**
- **But using approximate K-means, performance is superior!**
  - **Due to reduction of quantization effect**

Spatial Re-Ranking with RANSAC

- Generate hypotheses with pairs of corresponding features
  - Assume a restricted transformation, since many images on the web are captured in particular ways (axis-aligned ways)
- Evaluate other pairs and measure errors
- Re-ranking images by scoring the # of inliers

Transformation	dof	Matrix
translation + isotropic scale	3	$\begin{bmatrix} a & 0 & t_x \\ 0 & a & t_y \end{bmatrix}$
translation + anisotropic scale	4	$\begin{bmatrix} a & 0 & t_x \\ 0 & b & t_y \end{bmatrix}$
translation + vertical shear	5	$\begin{bmatrix} a & 0 & t_x \\ b & c & t_y \end{bmatrix}$

(a)

Method / Rerank N	100	200	400	800
i 3dof	0.468	0.492	0.522	0.556
ii 4dof	0.465	0.490	0.521	0.555
iii 5dof	0.467	0.491	0.526	0.560

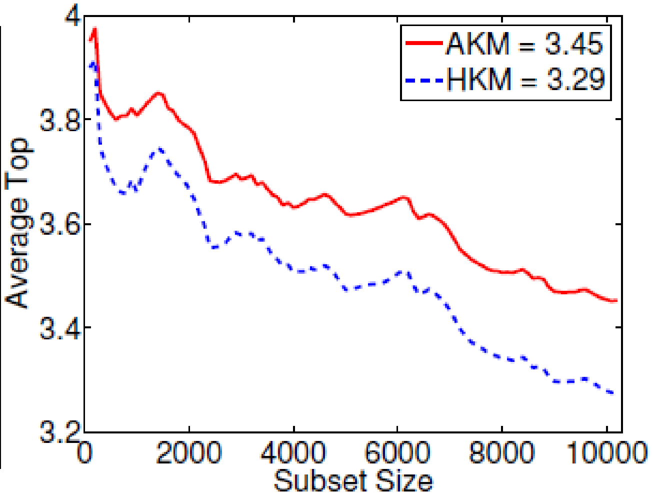
(b)

Method / Rerank N	100	200	400	800
i 3dof	0.644	0.650	0.652	0.655
ii 4dof	0.646	0.656	0.659	0.661
iii 5dof	0.648	0.657	0.660	0.664

Results

Clustering parameters		mAP	
# of descr.	Voc. size	k-means	AKM
800K	10K	0.355	0.358
1M	20K	0.384	0.385
5M	50K	0.464	0.453
16.7M	1M		0.618

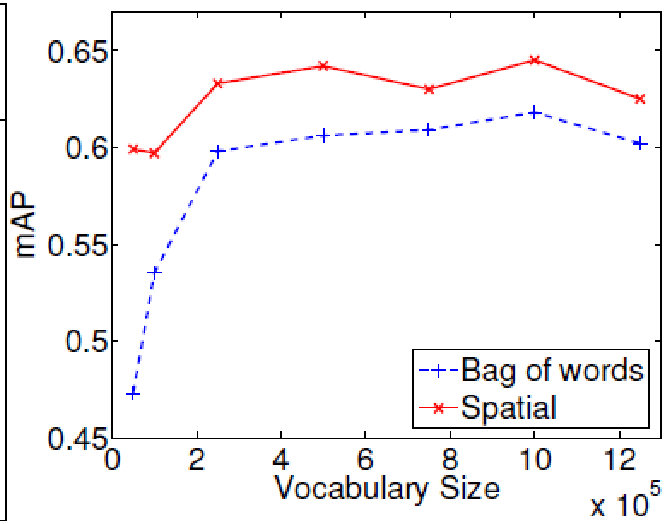
Method	Scoring Levels	Average Top
HKM	1	3.16
HKM	2	3.07
HKM	3	3.29
HKM	4	3.29
AKM		**3.45**



Results

Method	Dataset	mAP	
		Bag-of-words	Spatial
(a) HKM-1	5K	0.439	0.469
(b) HKM-2	5K	0.418	
(c) HKM-3	5K	0.372	
(d) HKM-4	5K	0.353	
(e) AKM	5K	0.618	0.647
(f) AKM	5K+100K	0.490	0.541
(g) AKM	5K+100K+1M	0.393	0.465

Vocab Size	Bag of words	Spatial
50K	0.473	0.599
100K	0.535	0.597
250K	0.598	0.633
500K	0.606	0.642
750K	0.609	0.630
1M	**0.618**	**0.645**
1.25M	0.602	0.625



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

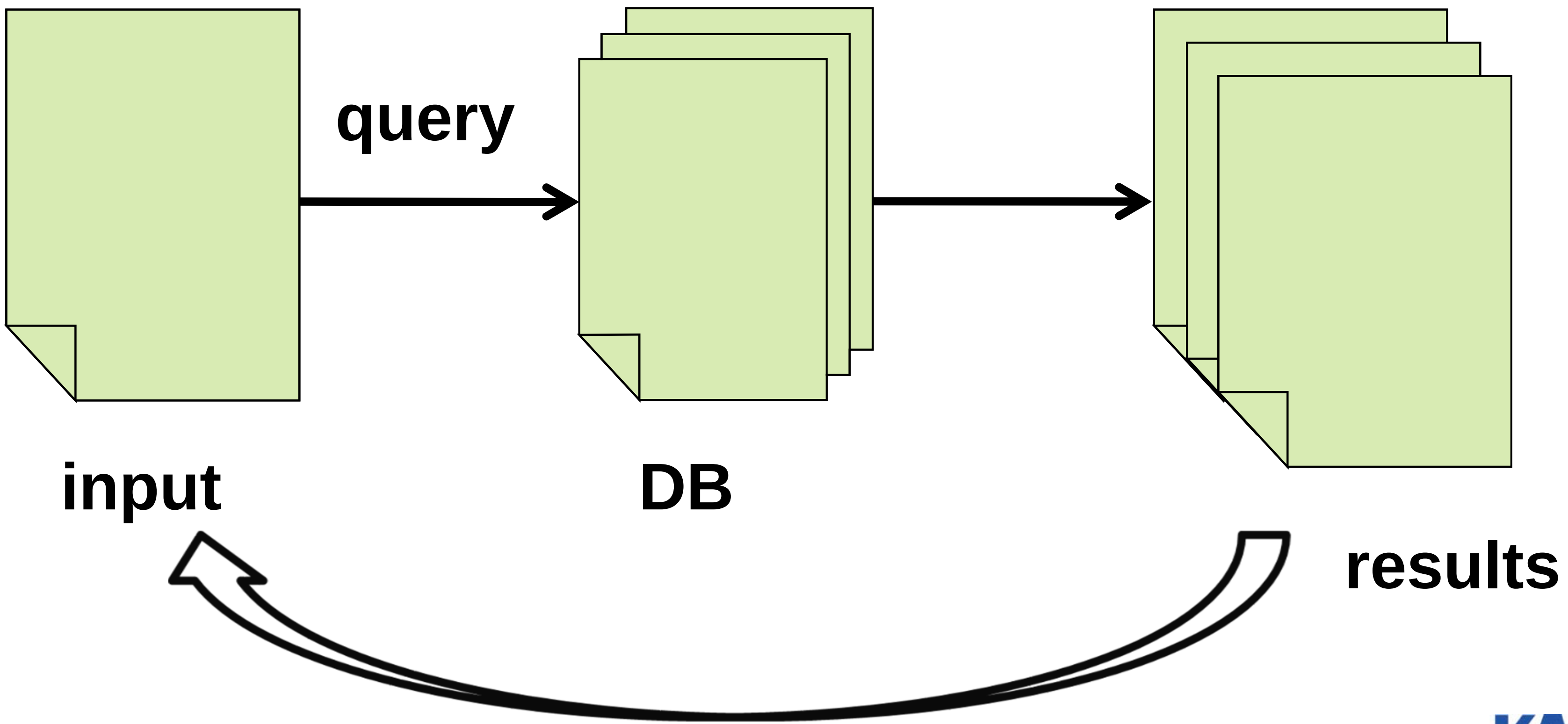
Chum et al.

ICCV 2007

Citation: over 150 at 2011

Query Expansion

- **Improve recall with re-querying combination of the original query and result with spatial verification**



Query Expansion

- Spatial verification
  - Similar with the technique used in [Philbin et al. 07]; Uses a RANSAC-like algorithm
  - Identify a set of images that are very similar to the original query image

BoW interpreted Probabilistically

- Extracts a generative model of an object from the query region
- Compute a response set that are likely to have been generated from the model
- The generative model
  - Spatial configuration of visual words with a background clutter

Generative Models

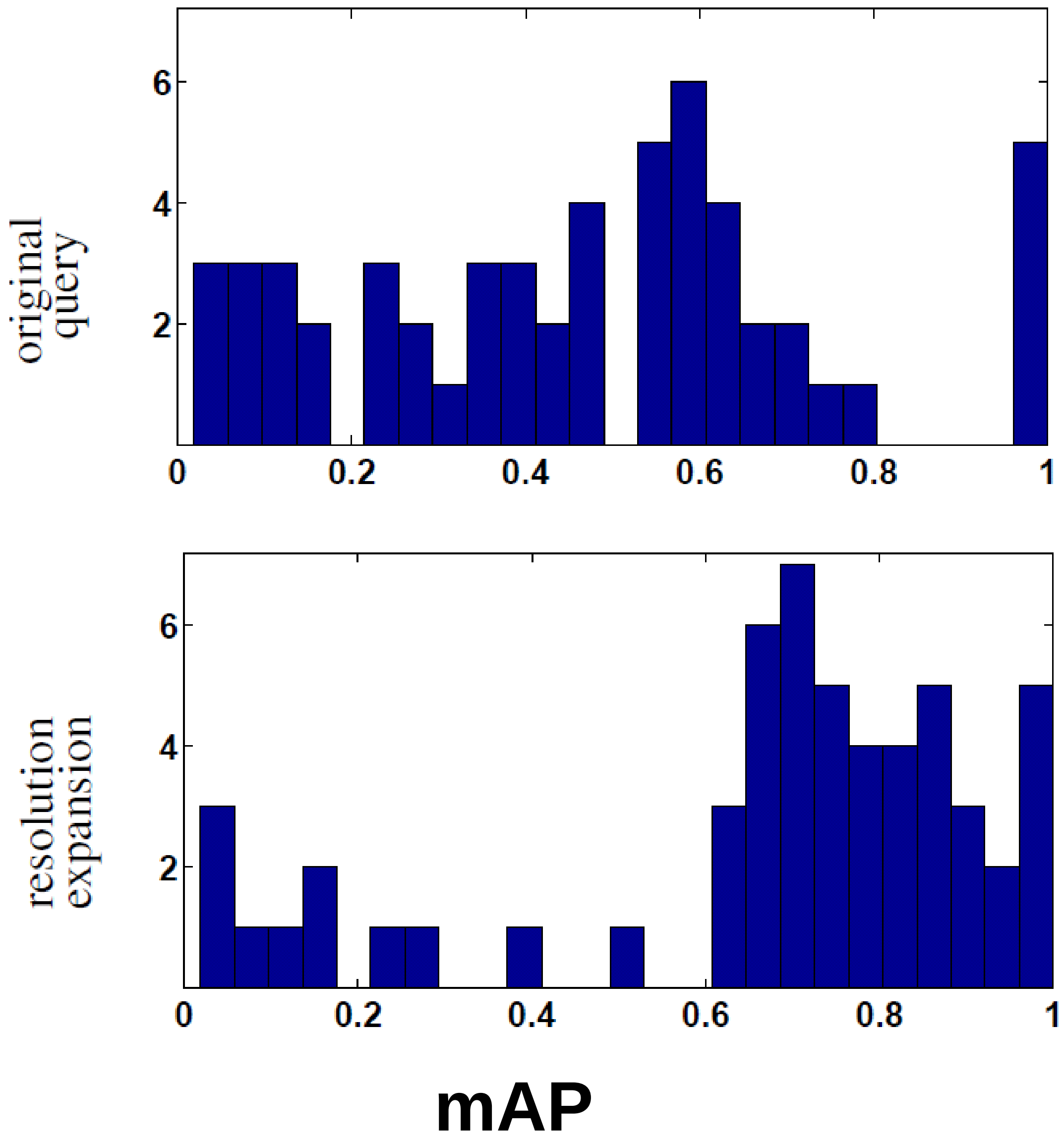
- **Query expansion baseline**
  - **Average term frequency vectors from the top 5 queries without verification**
- **Transitive closure expansion**
  - **A priority queue of verified images is keyed by # of inliers**
  - **Take the top image and query it as a new query**
- **Average query expansion**
  - **A new query is constructed by averaging the top 50 verified results (di is the term frequency vector of ith verified image)**

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

Generative Models

- **Multiple image resolution expansion**
  - **Consider images with different resolutions; higher resolutions give more detailed information**
  - **Use a resolution band with (0, 4/5), (2/3, 3/2), and (5/4, infinity)**
  - **Use averaged queries for each resolution band**
  - **Show the best result**

Results



original query
6
4
2
0
0.2
0.4
0.6
0.8
1
resolution expansion
6
4
2
0
0.2
0.4
0.6
0.8
1
mAP

Results

Original query

Top 4 images

Expanded results that were not identified by the original query

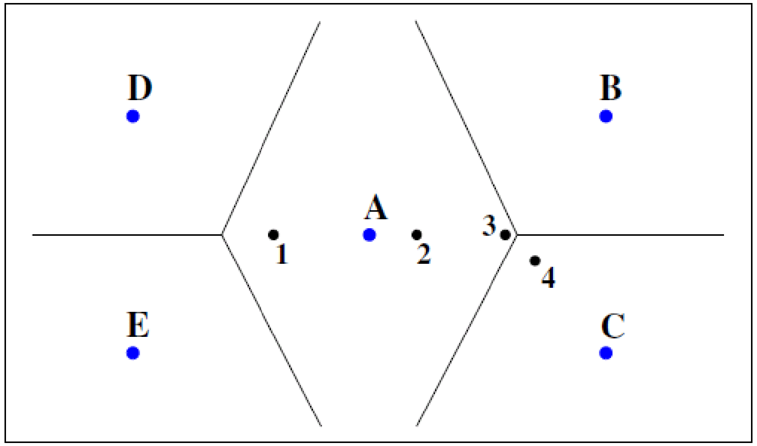
Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases

Philbin et al.

CVPR 2008

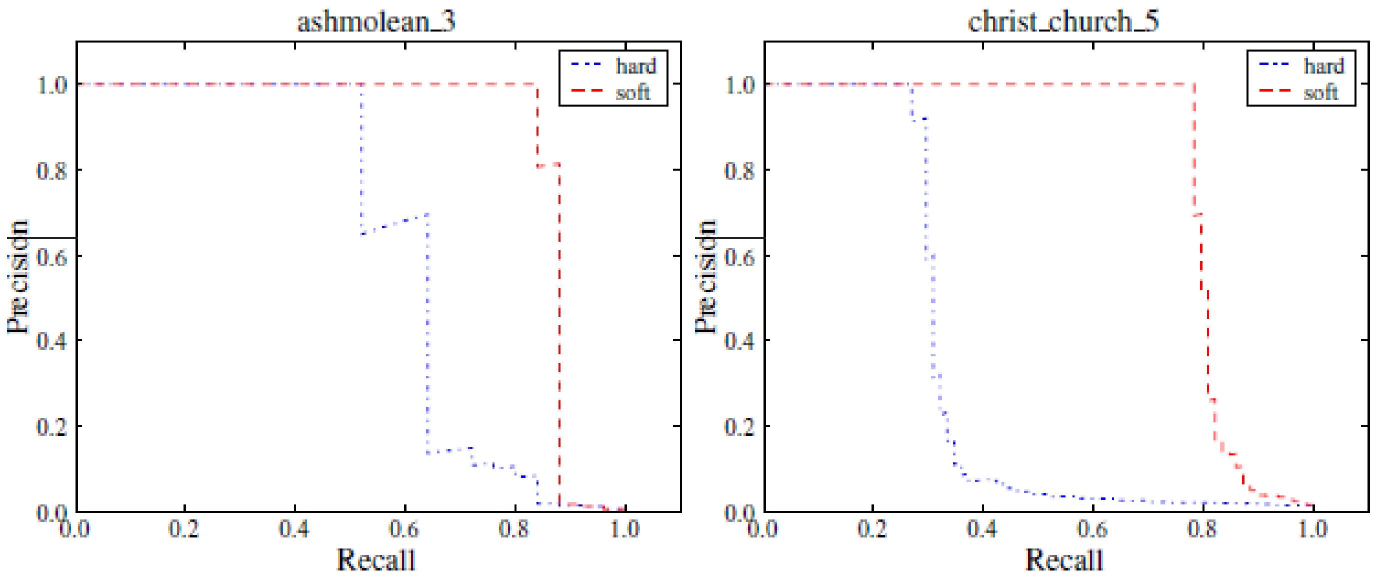
Citation: over 175 at 2011

Soft Quantization [Philbin et al. CVPR 08]



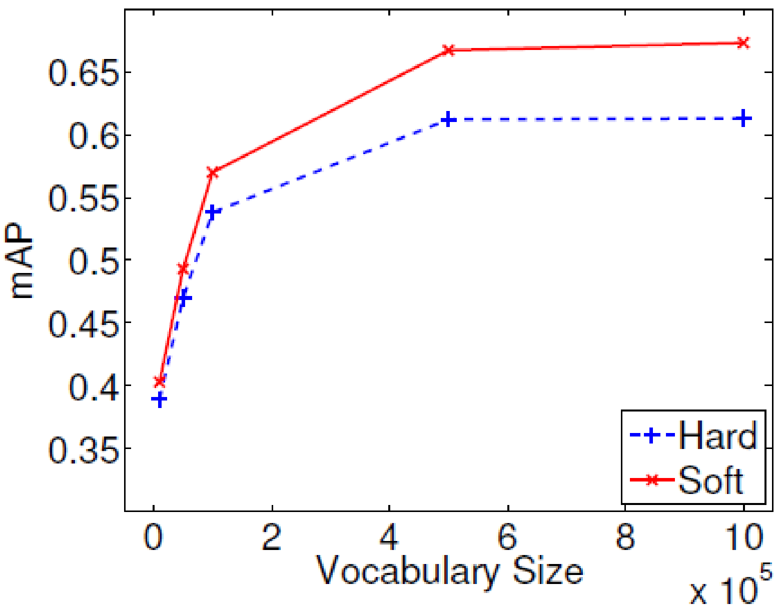
- **3 and 4 will be never matched in hard assignment**
- **No way of distinguishing 2 and 3 are closer than 1 and 2**
- **Soft assignment: use a weight vector**
  - **A weight to a cluster is assigned proportional to the distance between the descriptor and the center of the cluster**

Results



Method	Training data	
	Oxford	Paris
Fixed Quantization [18]	0.164	
HKM [14] (1 level)	0.422	0.401
HKM [14] (2 level)	0.410	0.340
Hard [15]	0.614	0.403
Soft	**0.673**	**0.494**

Effect of Vocabulary Size and Number of Images



- **For Oxford dataset with 1M vocabulary, hard assignment index costs 36MB and soft costs 108MB with compression**

KAIST

Next Time…

- **Nearest neighbor search using hashing**