Scalable Visual Object Retrieval

Andrew Zisserman

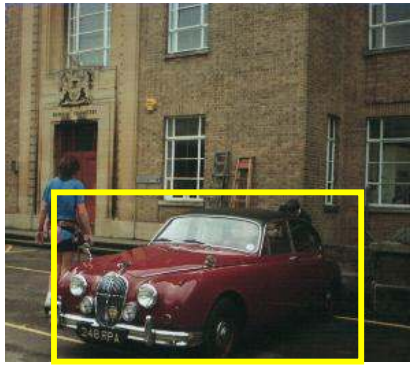
(work with Ondřej Chum, Michael Isard, James Philbin, Josef Sivic)

Visual Geometry Group
Dept of Engineering Science
University of Oxford

Query by visual example

Query: image/video clip ⇨ Retrieve: images/shots from archive

near duplicate



same object

same category

outline

In images and videos:

1. Retrieving specific objects
   - Use text analogy for efficient retrieval
2. Scaling up visual vocabularies
3. Query expansion to improve recall

Problem specification: particular object retrieval

Example: visual search in feature films

Visually defined query

"Groundhog Day" [Rammis, 1993]

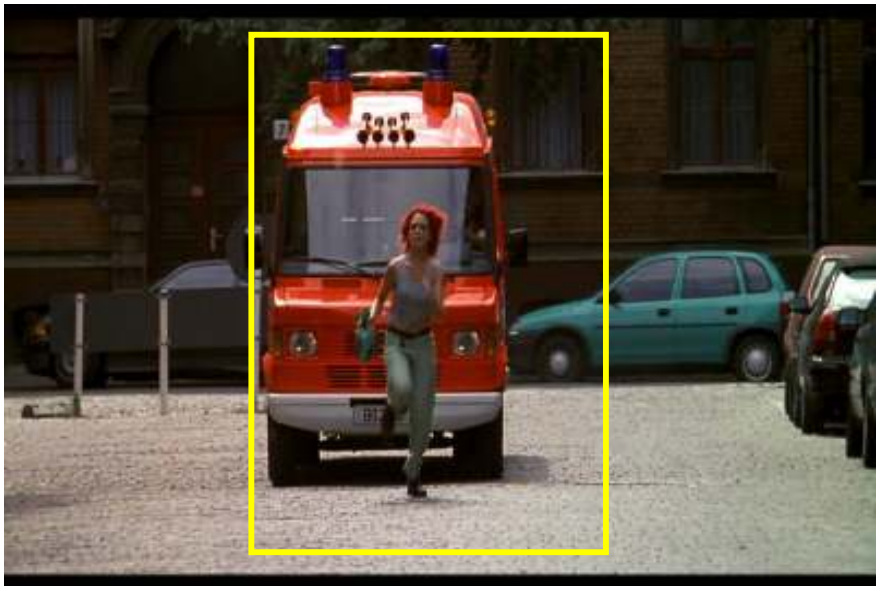
"Find this clock"

"Find this place"

Example

retrieved shots

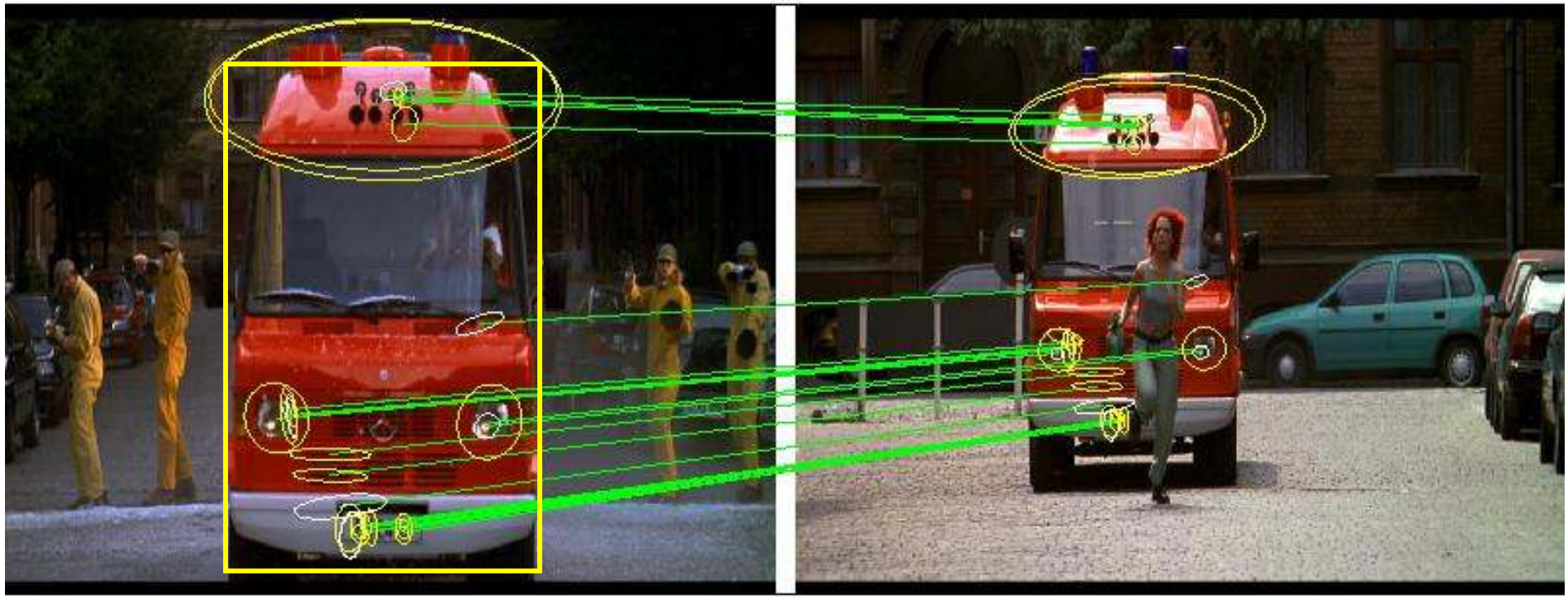
Particular objects, not entire images



Forced to face problems of:

- scale change,
- pose change,
- illumination change, and
- partial occlusion

When do (images of) objects match?



Two requirements:

1. “patches” (parts) correspond, and
2. Configuration (spatial layout) corresponds

Success of text retrieval

- efficient
- scalable
- high precision

Can we use retrieval mechanisms from text retrieval?

Need a visual analogy of a textual word.

Visual problem

- Retrieve key frames containing the same object

query

?

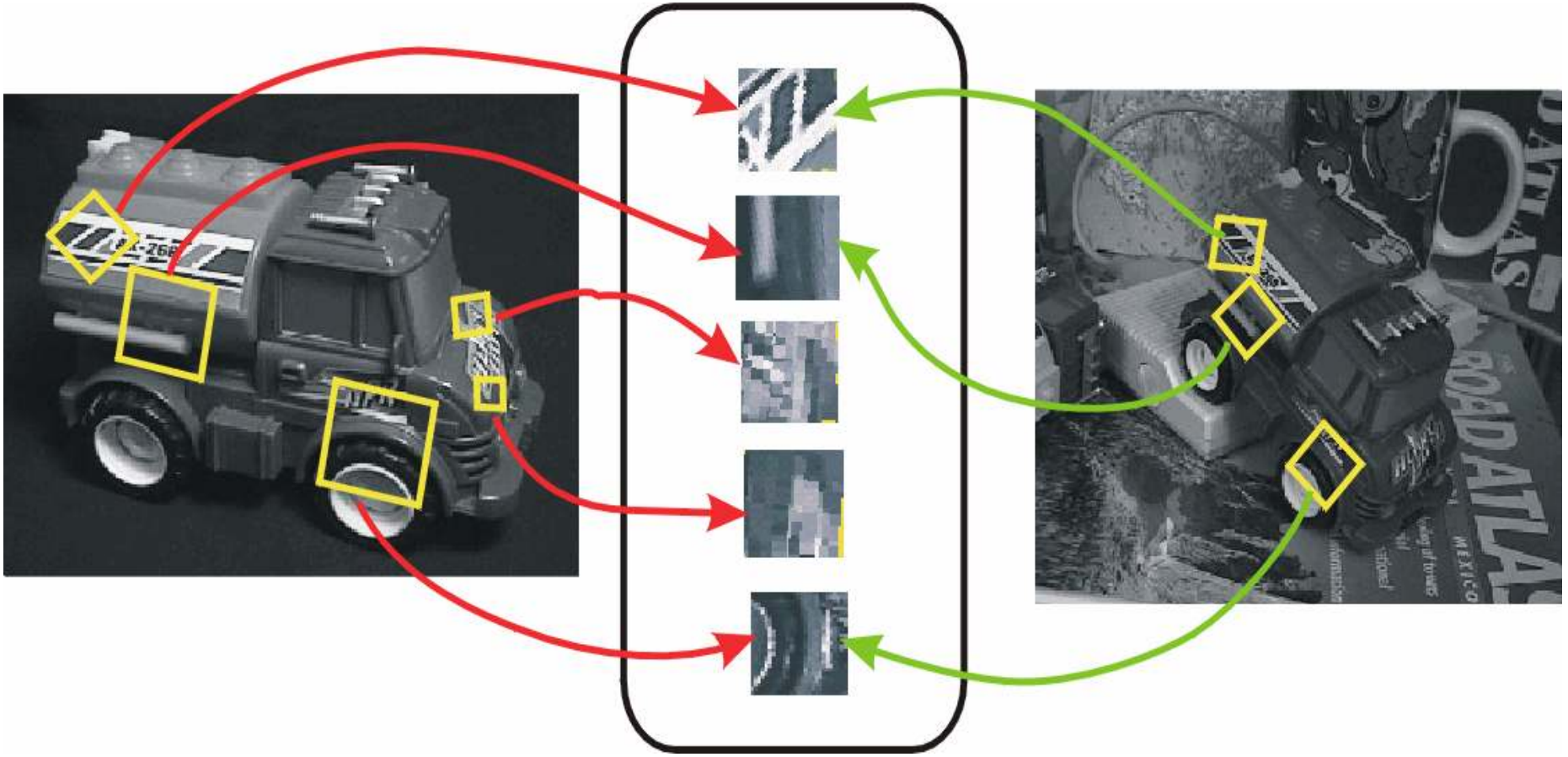
Approach

Determine regions (segmentation) and vector descriptors in each frame which are invariant to camera viewpoint changes

Match descriptors between frames using invariant vectors

Example of visual fragments

Image content is transformed into local fragments that are invariant to translation, rotation, scale, and other imaging parameters



- Fragments generalize over viewpoint and lighting

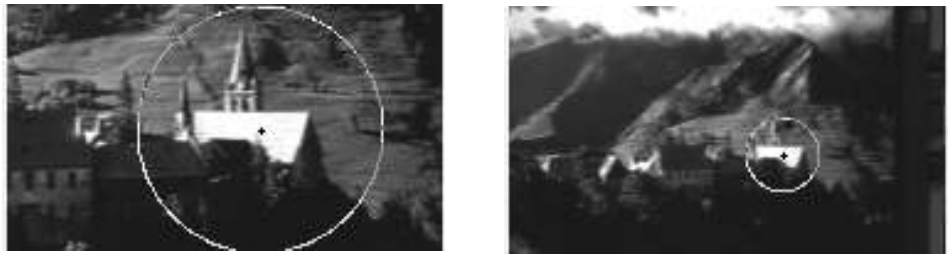
Lowe ICCV 1999

Scale invariance

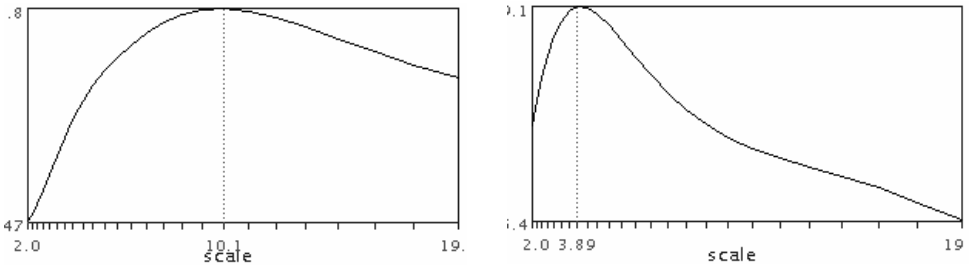
Mikolajczyk and Schmid ICCV 2001

Multi-scale extraction of Harris interest points

Selection of points at characteristic scale in scale space



Laplacian



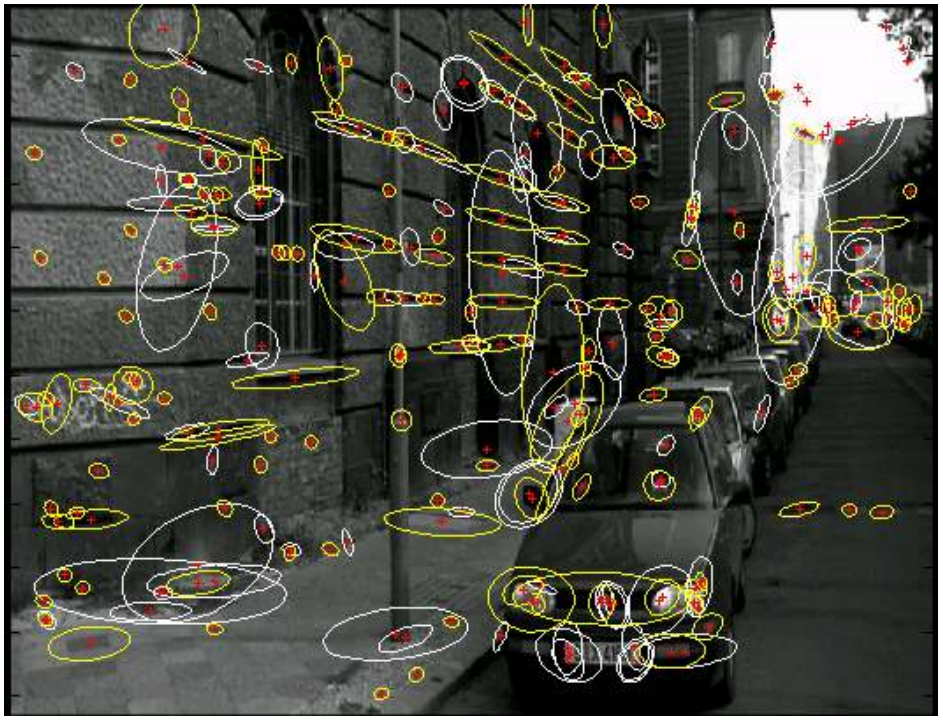
Chacteristic scale :

- maximum in scale space
- scale invariant

Viewpoint covariant segmentation

- Characteristic scales (size of region)
  - Lindeberg and Garding ECCV 1994
  - Lowe ICCV 1999
  - Mikolajczyk and Schmid ICCV 2001
- Affine covariance (shape of region)
  - Baumberg CVPR 2000
  - Matas et al BMVC 2002 — Maximally stable regions
  - Mikolajczyk and Schmid ECCV 2002
  - Schaffalitzky and Zisserman ECCV 2002 — Shape adapted regions "Harris affine"
  - Tuytelaars and Van Gool BMVC 2000

Example of affine covariant regions



1000+ regions per image

Harris-affine

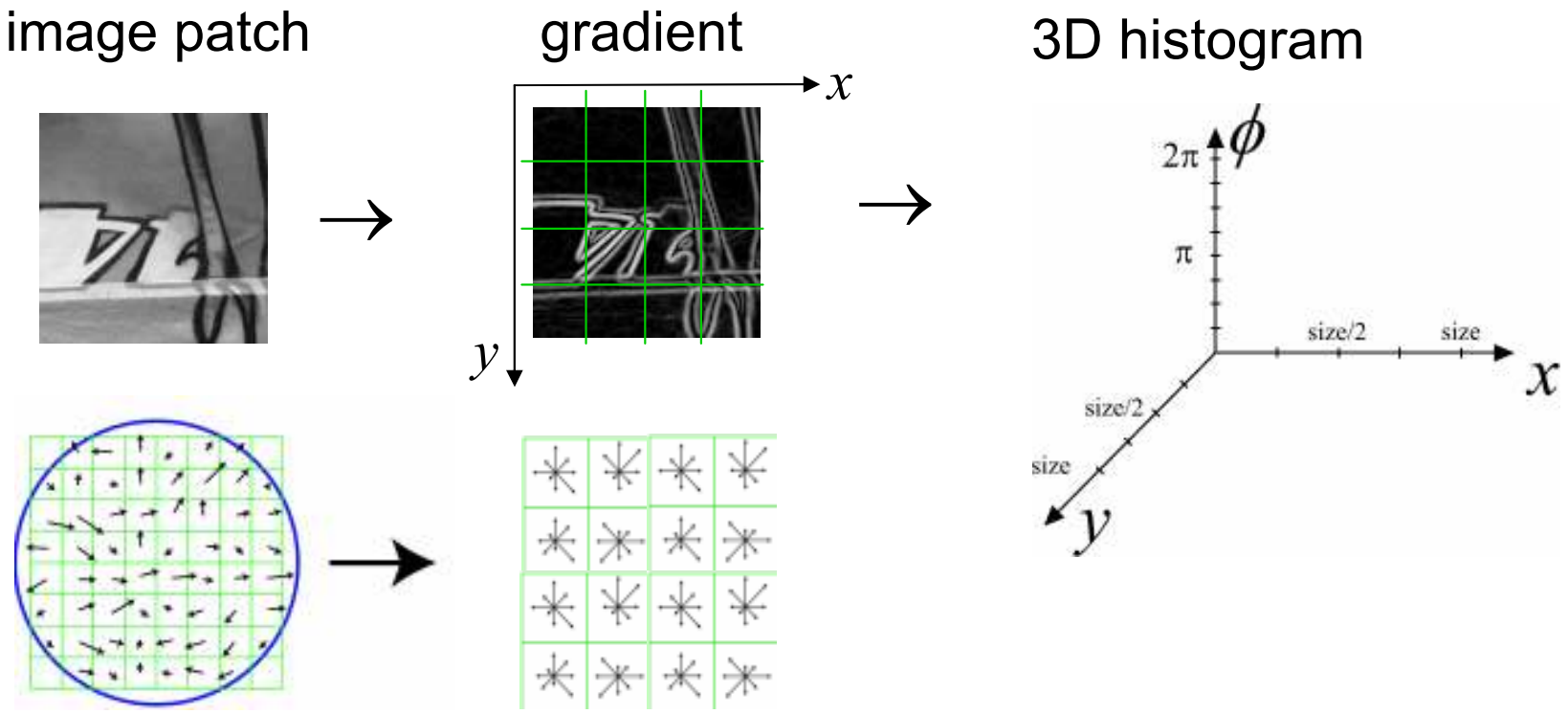
Maximally stable regions

- a region's size and shape are not fixed, but
- automatically adapts to the image intensity to cover the same physical surface
- i.e. pre-image is the same surface region

Represent each region by SIFT descriptor (128-vector) [Lowe 1999]

Descriptors – SIFT [Lowe'99]

distribution of the gradient over an image patch



4x4 location grid and 8 orientations (128 dimensions)

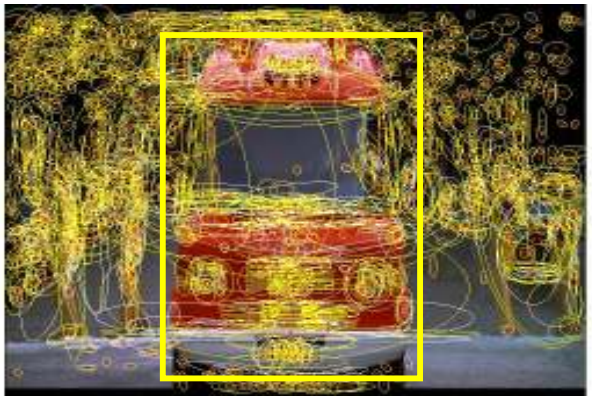
very good performance in image matching [Mikolaczyk and Schmid'03]

Example

In each frame independently

determine elliptical regions (segmentation covariant with camera viewpoint)

compute SIFT descriptor for each region [Lowe ‘99]



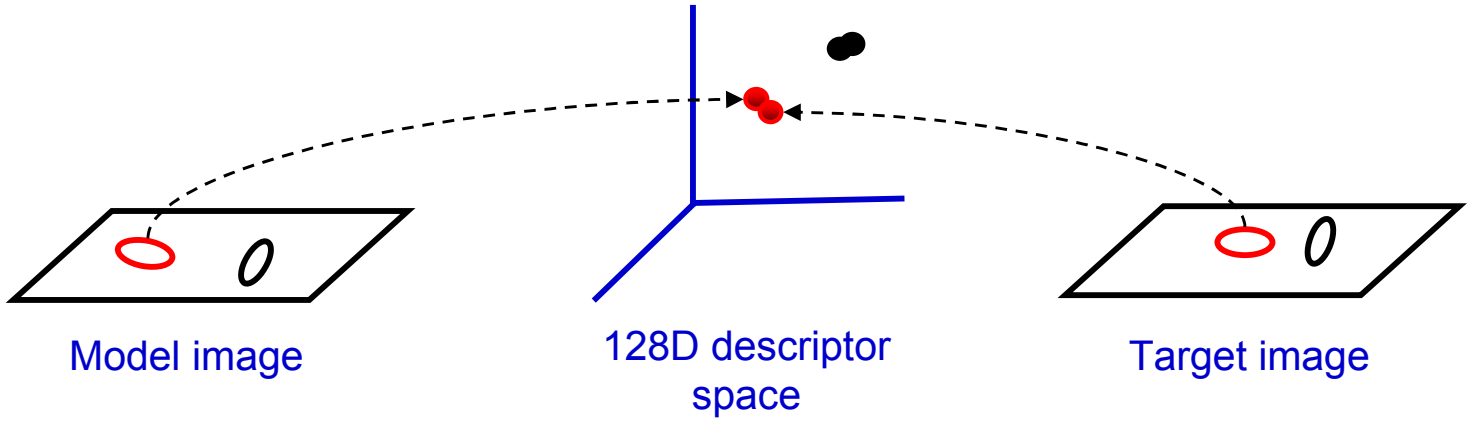
1000+ descriptors per frame

Harris-affine

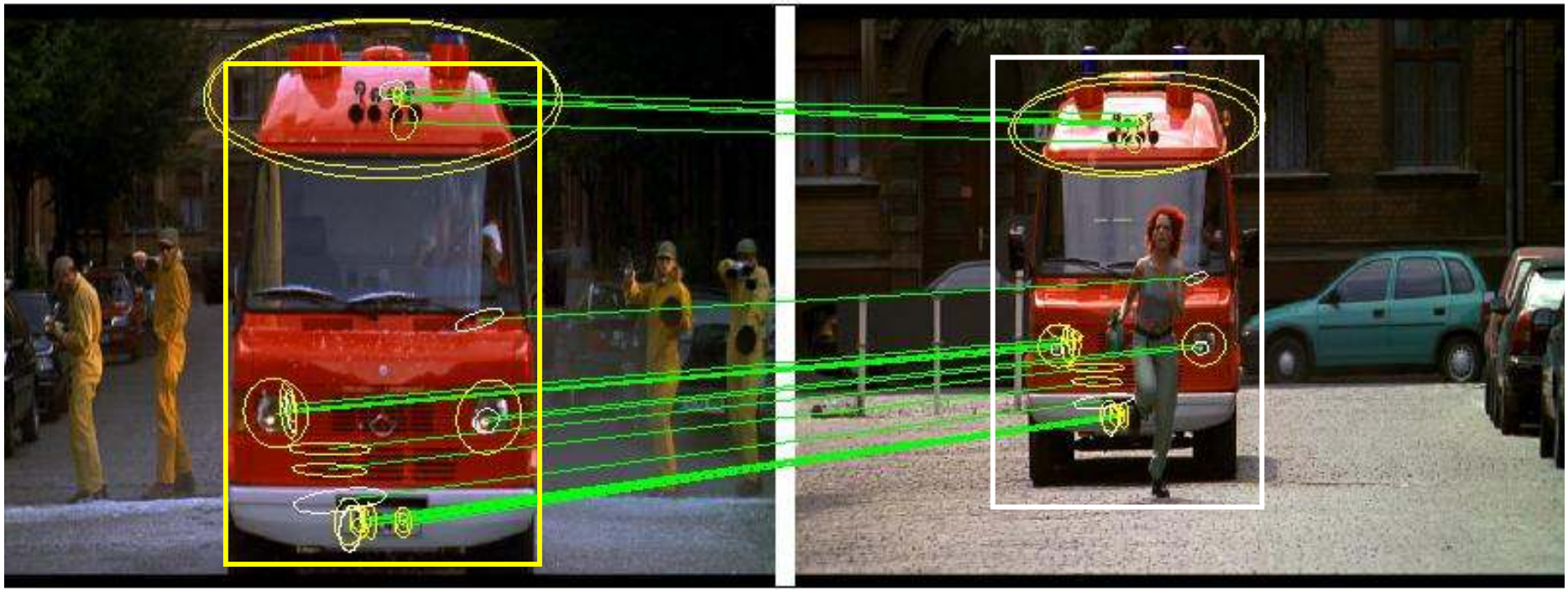
Maximally stable regions

Object recognition

Establish correspondences between object model image and target image by nearest neighbour matching on SIFT vectors



Match regions between frames using SIFT descriptors



- Multiple fragments overcomes problem of partial occlusion
- Transfer query box to localize object

Harris-affine

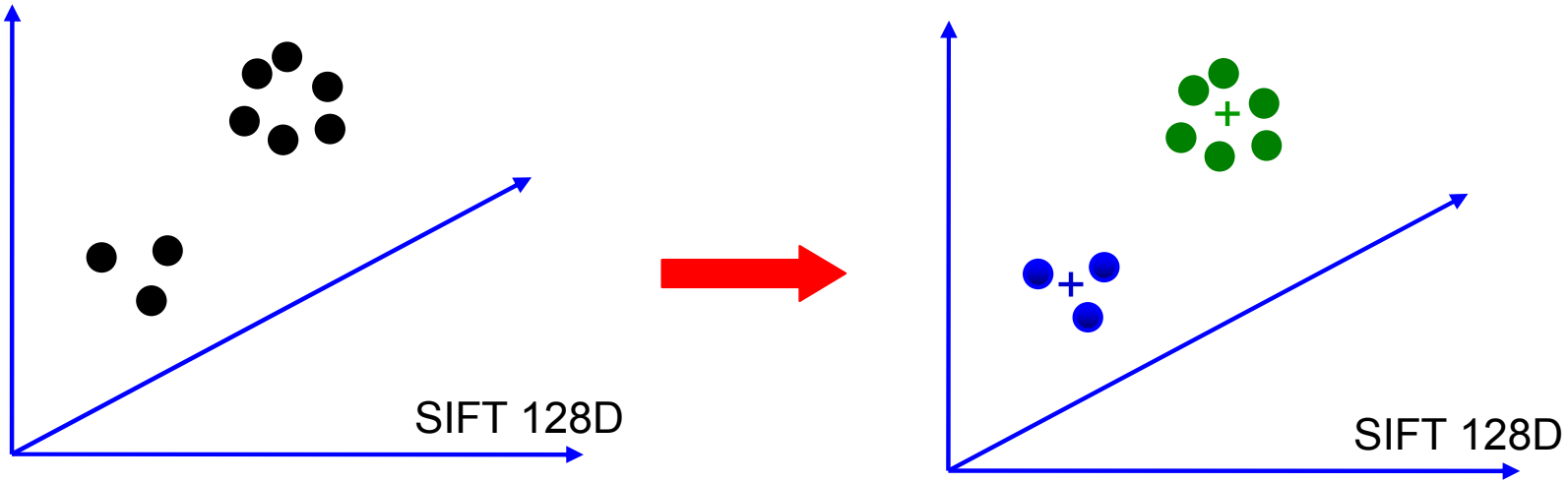
Maximally stable regions

Now, convert this approach to a text retrieval representation

Build a visual vocabulary for a movie

Vector quantize descriptors

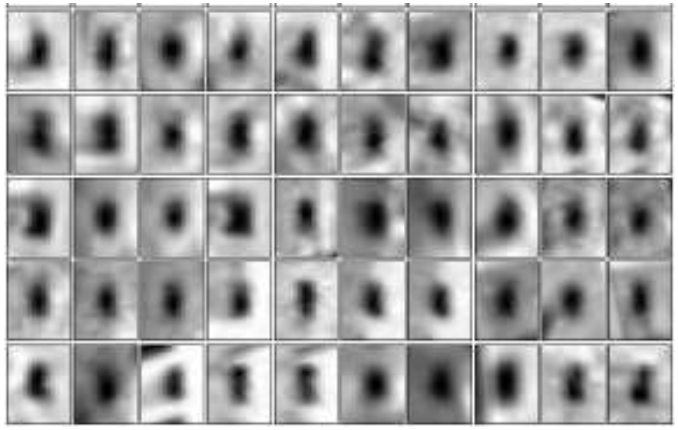
- k-means clustering

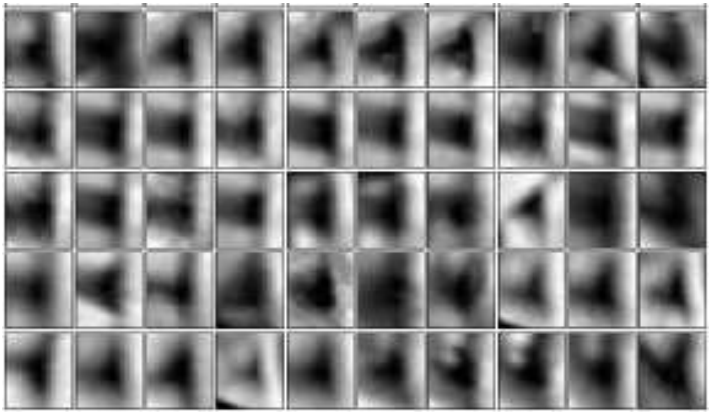


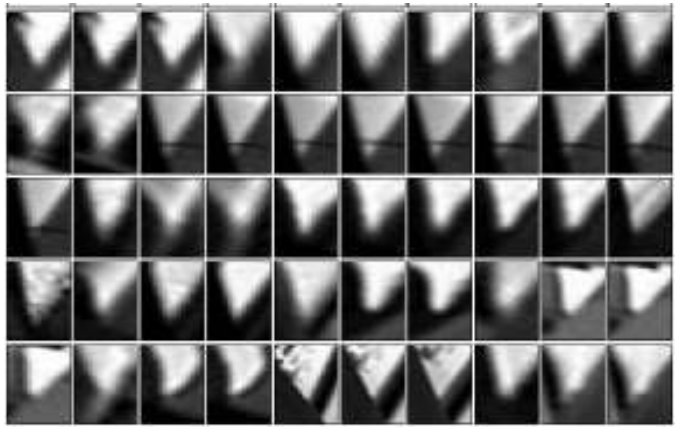
Implementation

- compute SIFT features on frames from 48 shots of the film
- 6K clusters for Shape Adapted regions
- 10K clusters for Maximally Stable regions

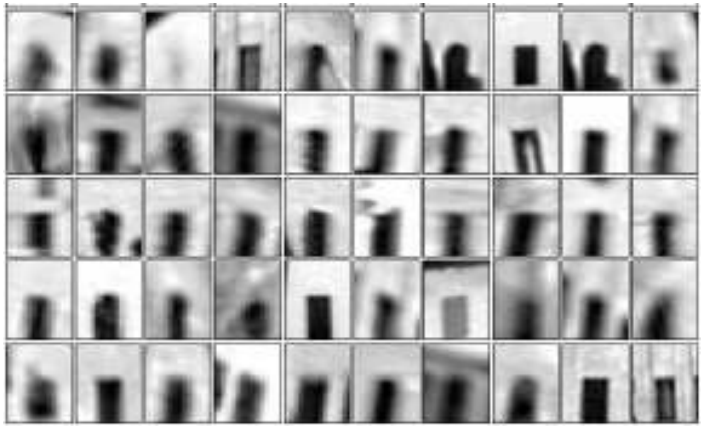
Samples of visual words (clusters on SIFT descriptors):







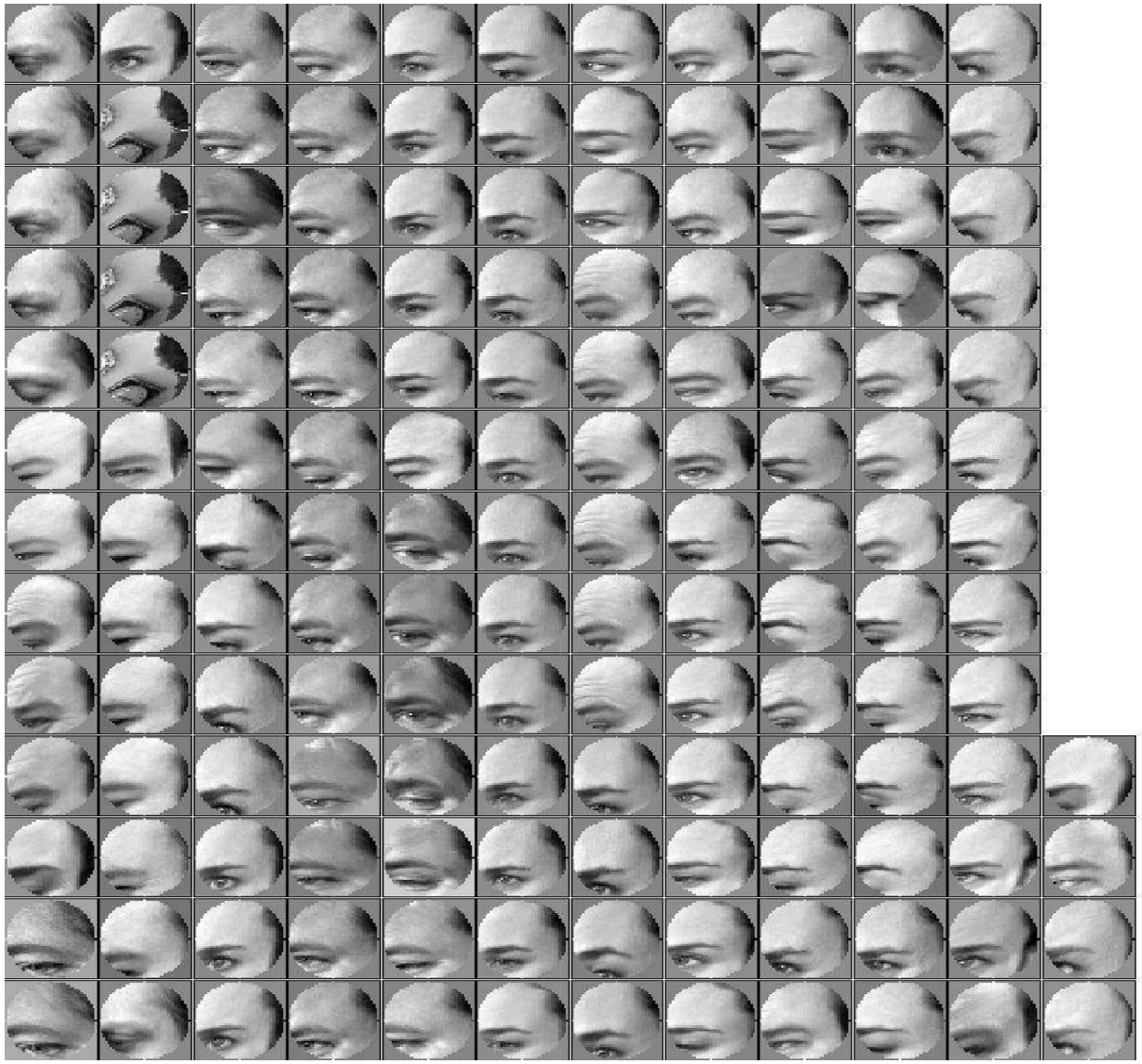
Shape adapted regions

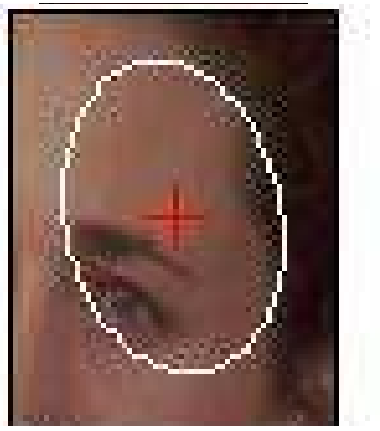


Maximally stable regions

generic examples – cf textons

Samples of visual words (clusters on SIFT descriptors):

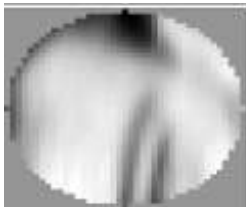




More specific example

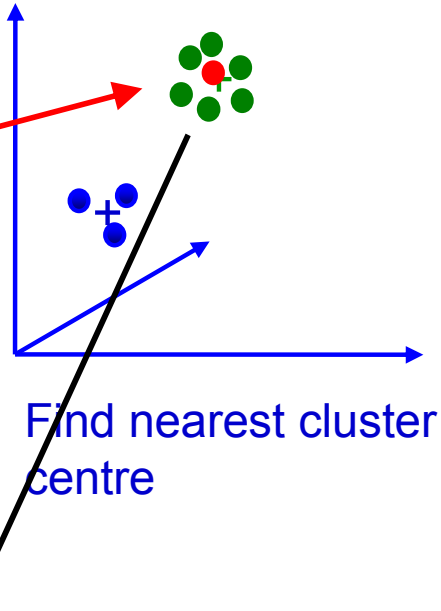
Assign visual words and compute histograms for each key frame in the video

Detect patches

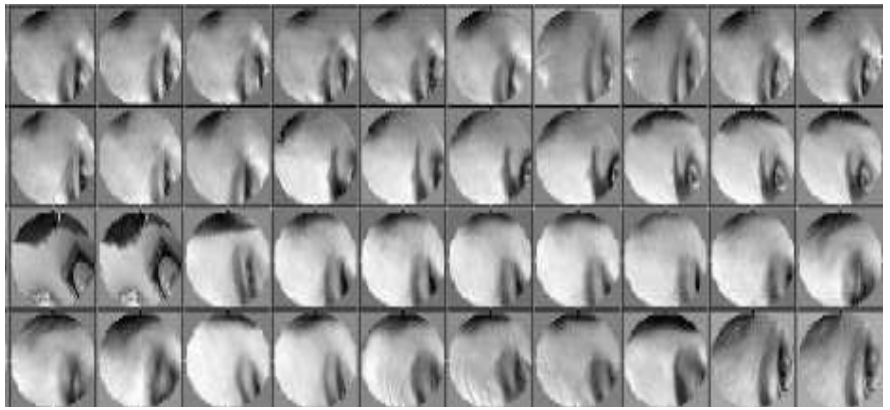


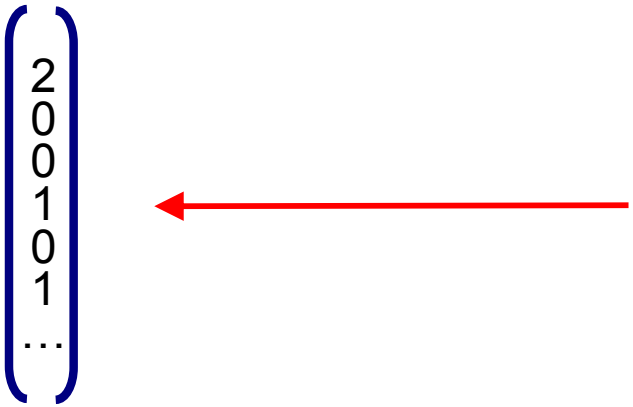
Normalize patch

Compute SIFT descriptor

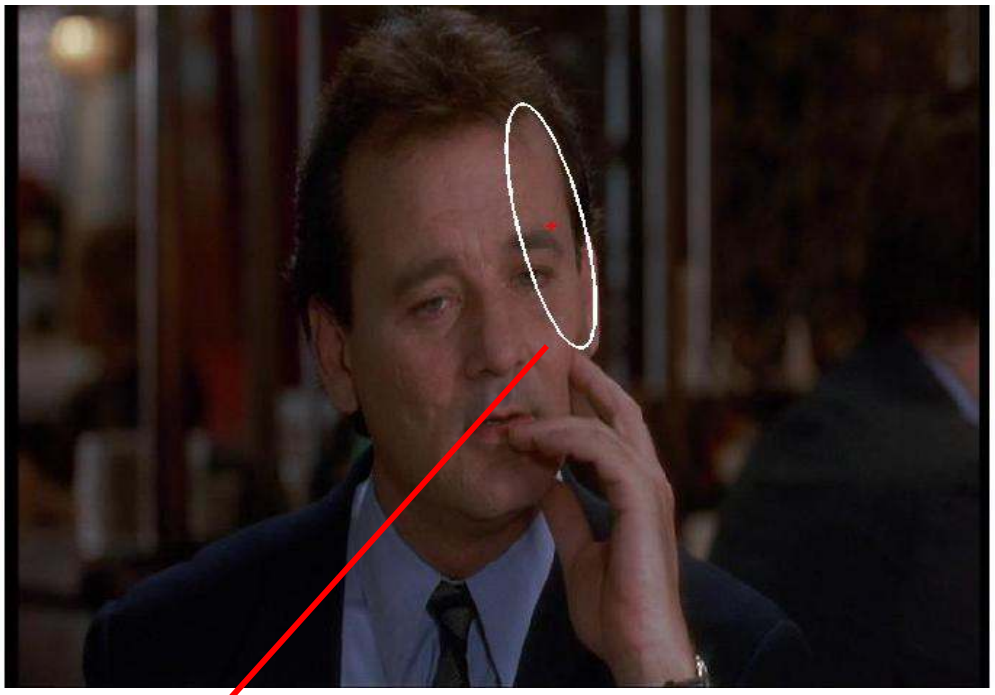


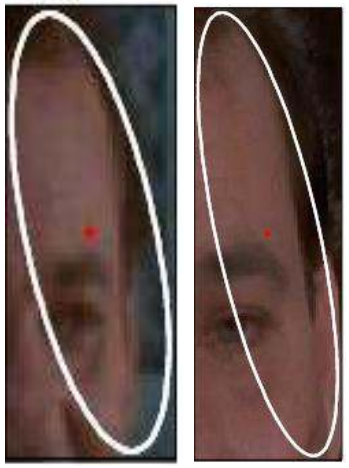
Find nearest cluster centre





Represent frame by sparse histogram of visual word occurrences



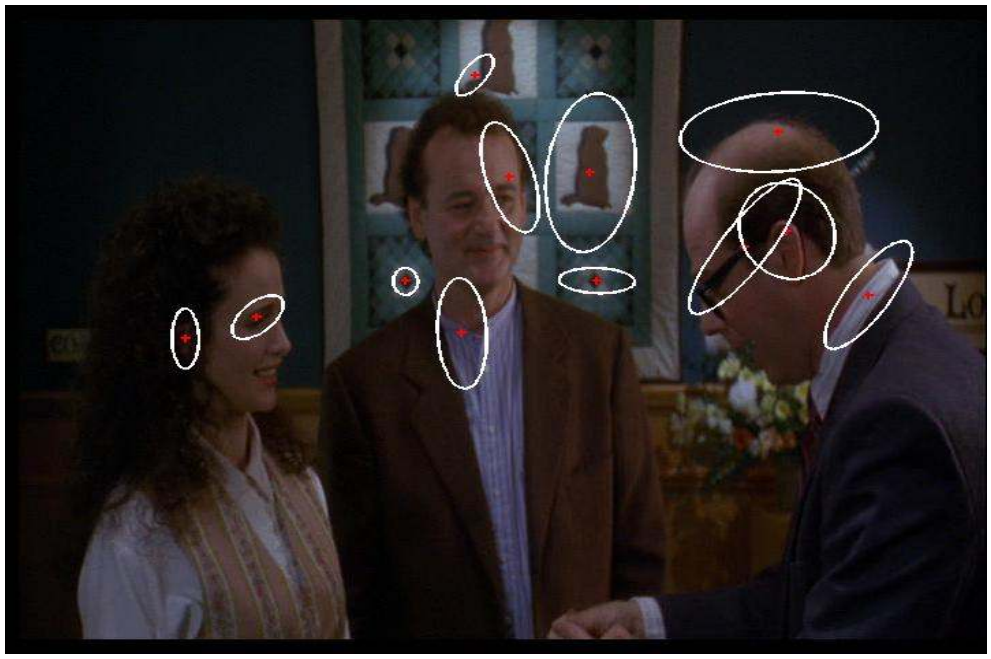


The same visual word

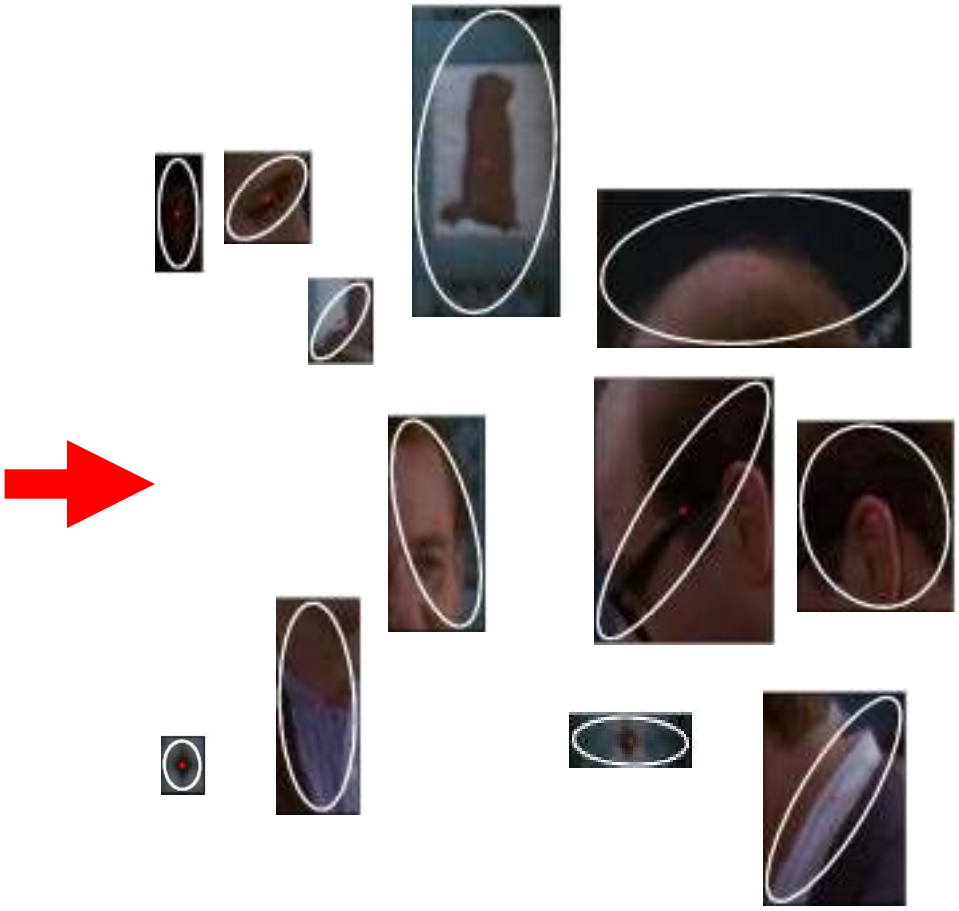
Representation: bag of (visual) words

Visual words are ‘iconic’ image patches or fragments

- represent the frequency of word occurrence
- but not their position



Image

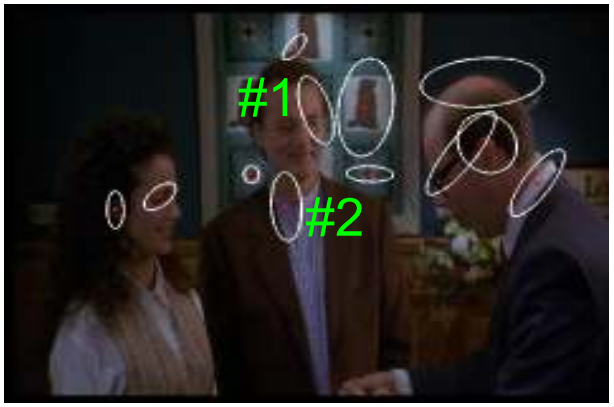


Collection of visual words

Search

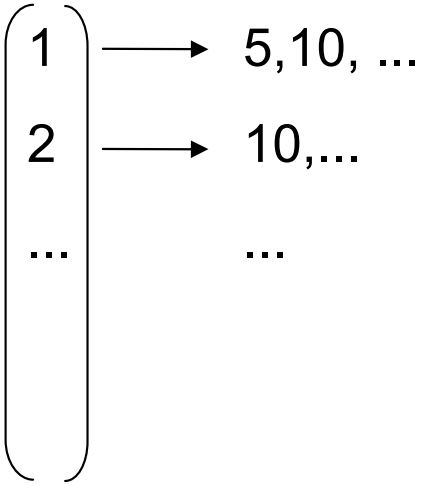
- For fast search, store a “posting list” for the dataset
- This maps word occurrences to the documents they occur in

frame #5



frame #10

Posting list



Films = common dataset

"Pretty Woman"

"Casablanca"

"Groundhog Day"

"Charade"

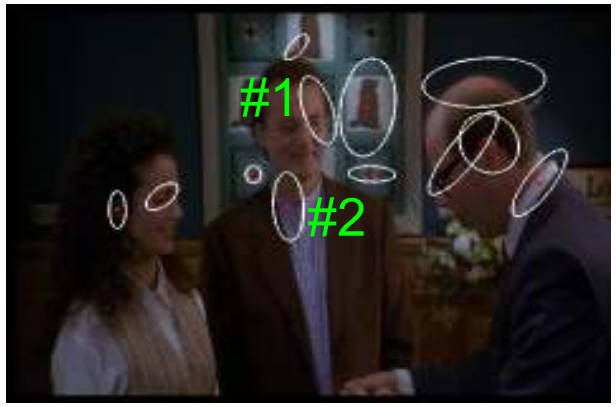
Video Google Demo

Matching a query region

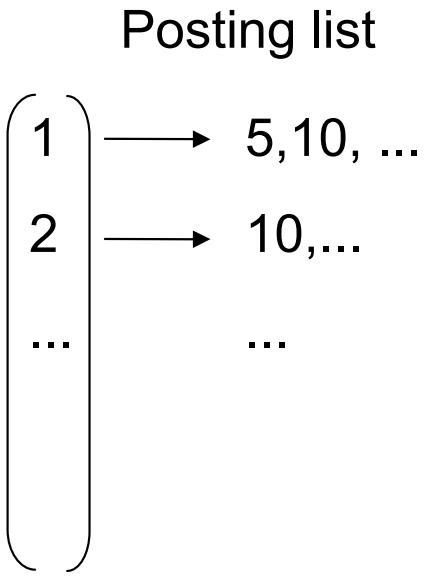
Stage 1: generate a short list of possible frames using bag of visual word representation:

1. Accumulate all visual words within the query region
2. Use “book index” to find other frames with these words
3. Compute similarity for frames which share at least one word

frame #5

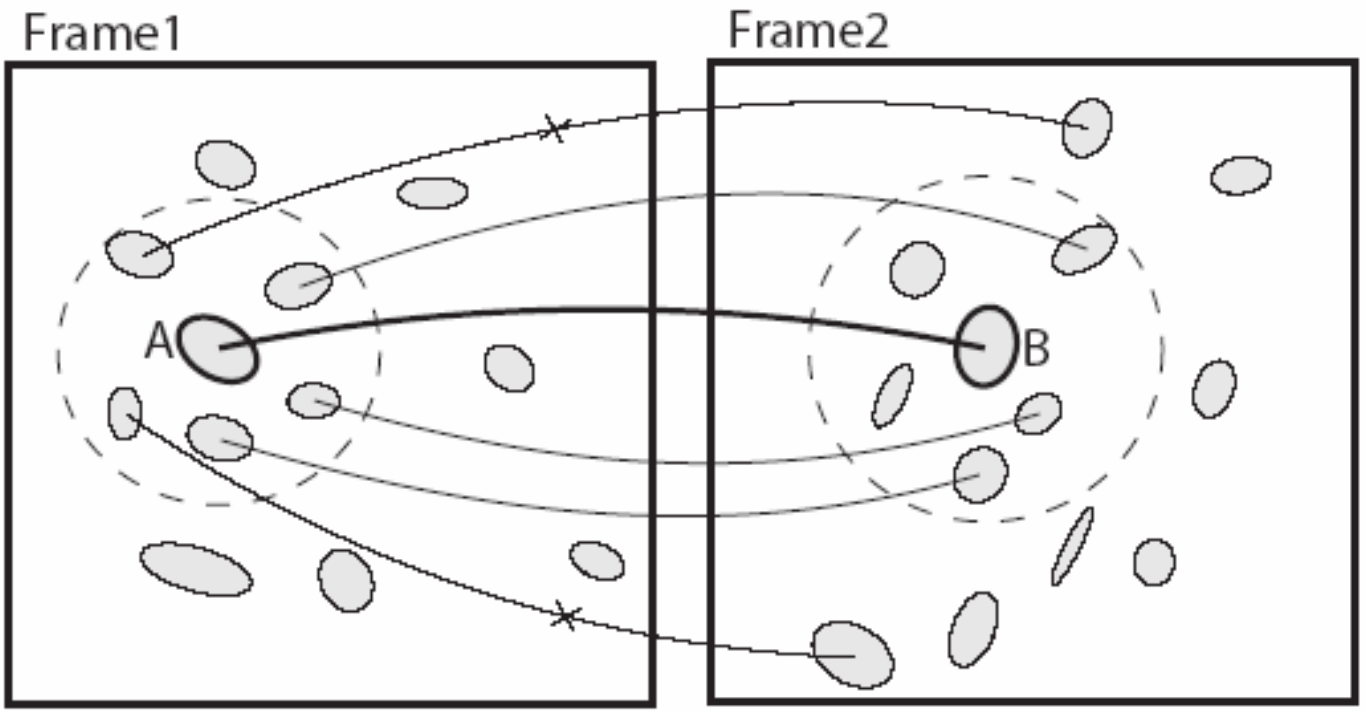


frame #10



- Generates a tf-idf ranked list of all the frames in dataset

Stage 2: re-rank short list on spatial consistency

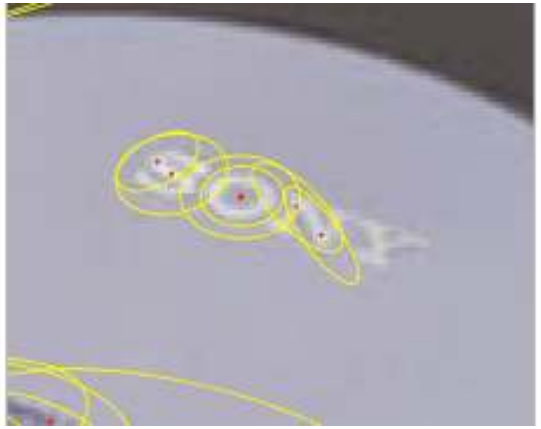


NB weak measure of spatial consistency

- Discard mismatches
  - require spatial agreement with the neighbouring matches
- Compute matching score
  - score each match with the number of agreement matches
  - accumulate the score from all matches
- Also matches define correspondence between target and query region

Example application I – product placement

Sony logo from Google image search on `Sony'



Retrieve shots from Groundhog Day

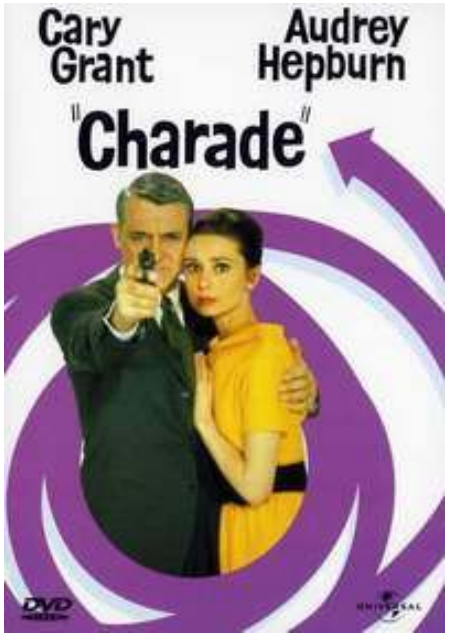
Retrieved shots in Groundhog Day for search on Sony logo

Example II - finding photos in a personal collection

Notre Dame from Google image search on `Notre Dame'

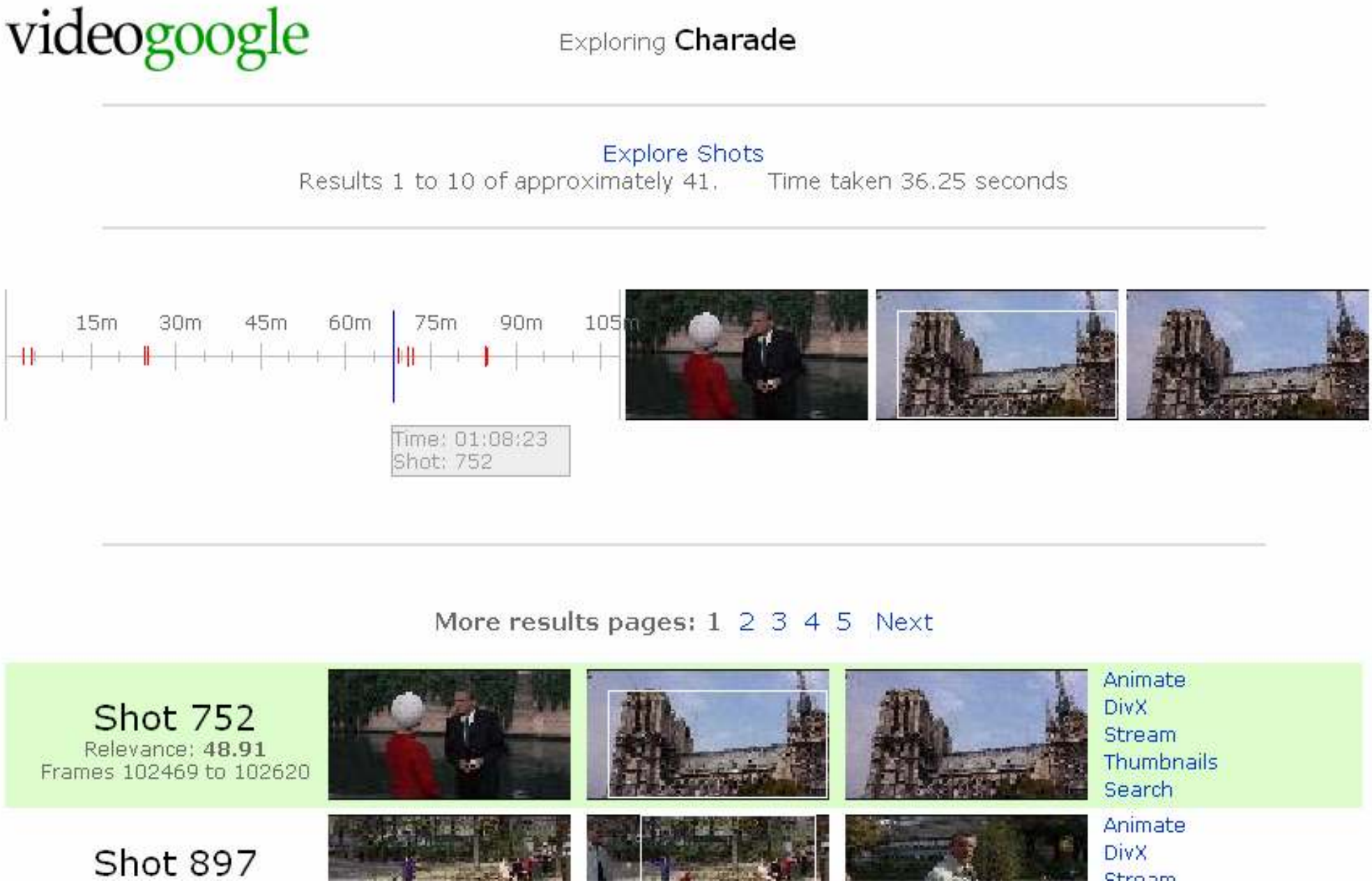
Query image

Charade (6,503 keyframes)



Retrieve shots from Charade

First (correctly) retrieved shot



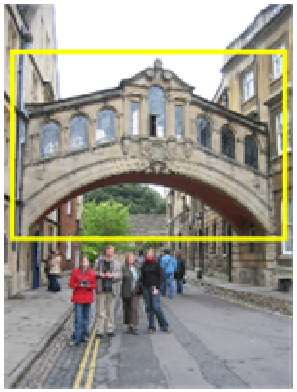
Viewpoint invariant matching

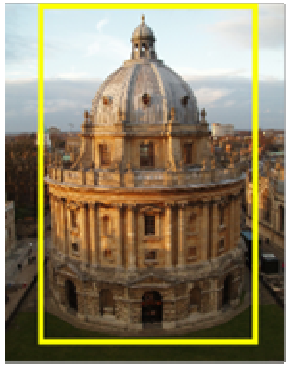
Query image

A keyframe from the matching shot

Part 2: Scaling up: the Oxford buildings dataset

Particular object search





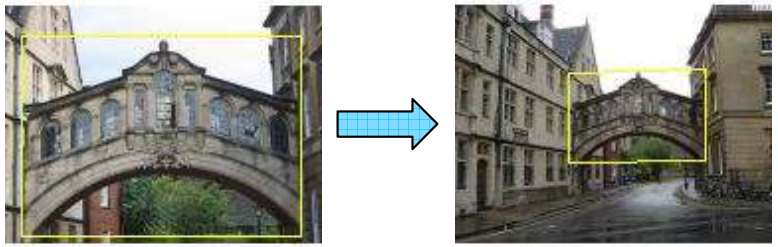
Find these landmarks

...in these images

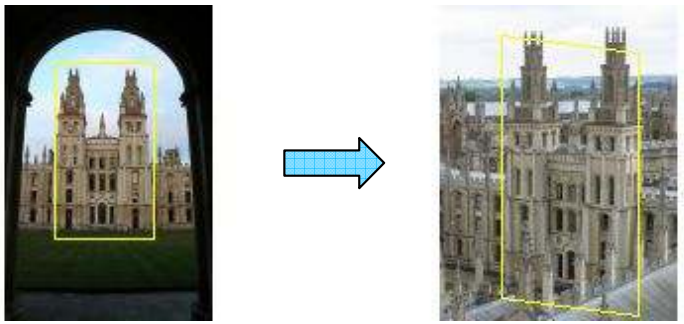
Particular Object Search

- Problem: find particular occurrences of an object in a very large dataset of images

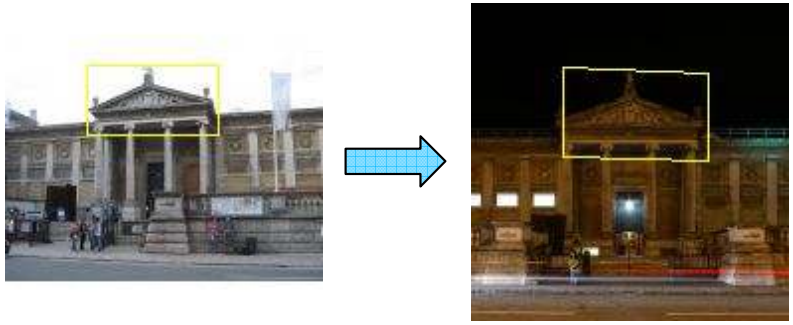
- Want to find the object despite possibly large changes in scale, viewpoint, lighting and partial occlusion



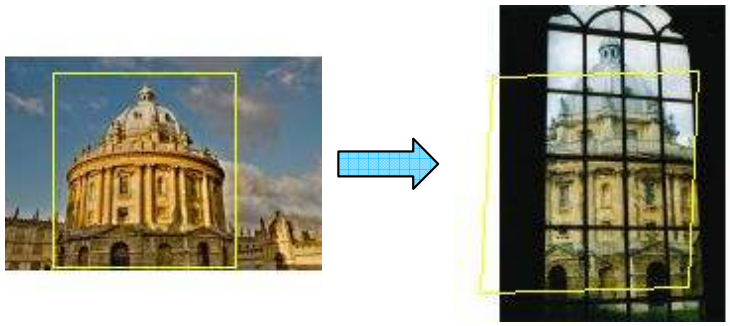
Scale



Viewpoint



Lighting



Occlusion

Representation & Similarity

- Text retrieval approach to visual search (“Video Google”)

Image

Detection + Description →

Sparse affine invariant regions descriptors (Hessian Affine + SIFT)

Quantize →

$$\begin{pmatrix} 2 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ \dots \end{pmatrix}$$

Sparse histogram of visual word occurrences

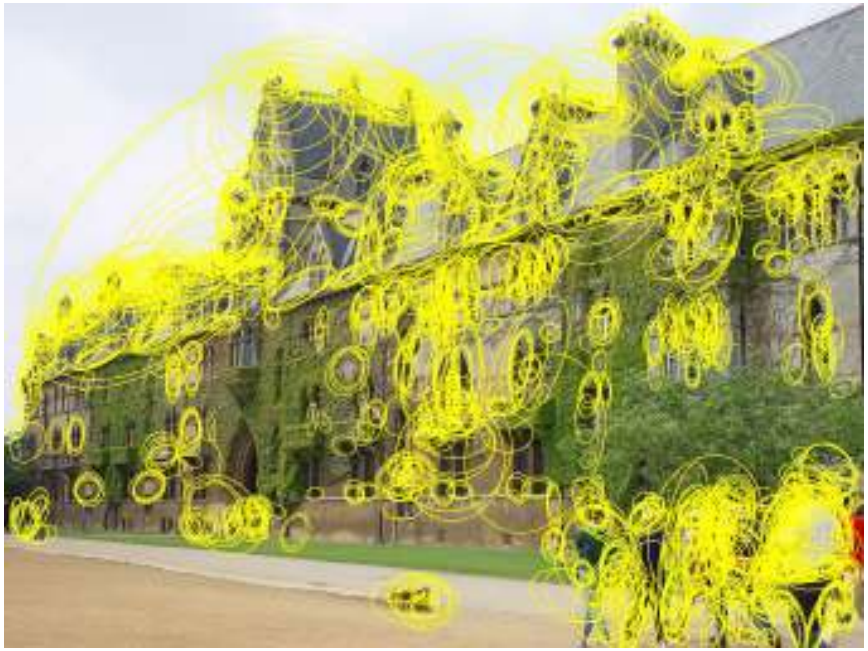
- Representation is a sparse histogram for each image

- Similarity measure is L_2 distance between tf-idf weighted histograms

Investigate …

Vocabulary size: number of visual words in range 10K to 1M

Use of spatial information to re-rank



Oxford buildings dataset

- Automatically crawled from flickr

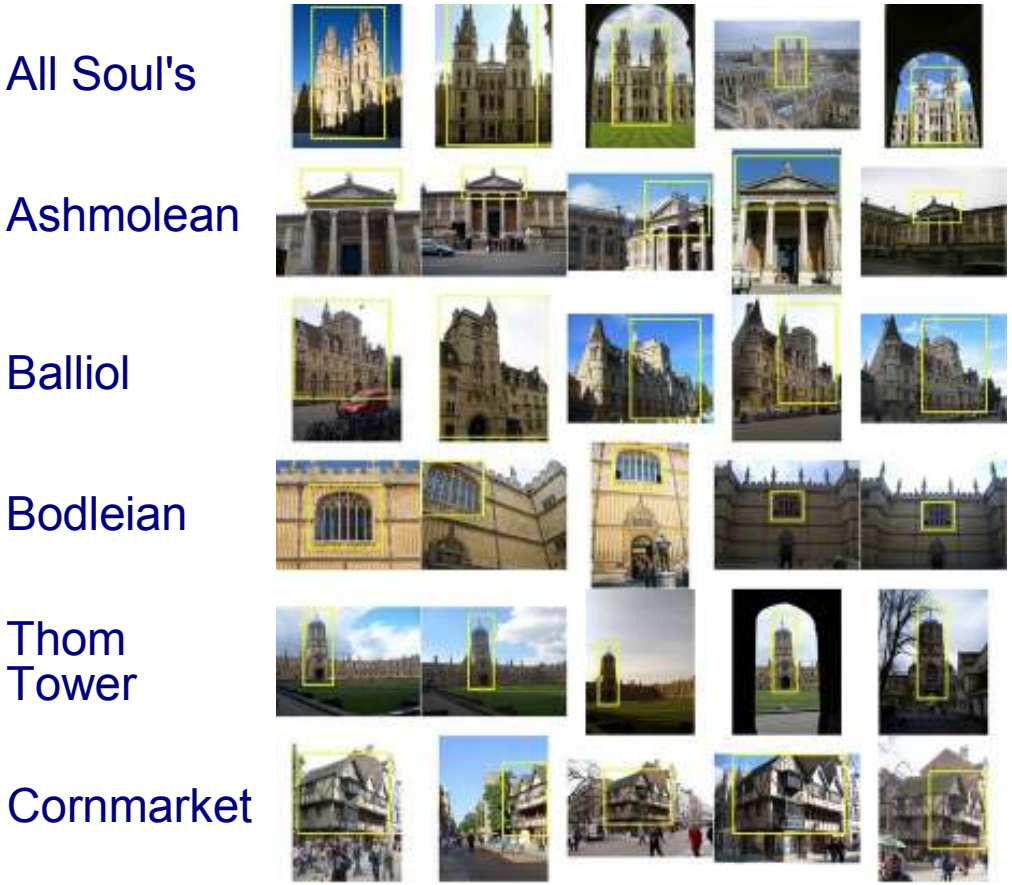
- Dataset (i) consists of 5062 images, crawled by searching for Oxford landmarks, e.g.

  - "Oxford Christ Church"
  - "Oxford Radcliffe camera"
  - "Oxford"

- High resolution images (1024 x 768)

Oxford buildings dataset

- Landmarks plus queries used for evaluation



- Ground truth obtained for 11 landmarks over 5062 images

- Performance measured by mean Average Precision (mAP) over 55 queries

Oxford buildings dataset

- Automatically crawled from flickr
- Consists of:

Dataset	Resolution	# images	# features	Descriptor size
i	1024×768	5,062	16,334,970	1.9 GB
ii	1024×768	99,782	277,770,833	33.1 GB
iii	500×333	1,040,801	1,186,469,709	141.4 GB
Total		1,145,645	1,480,575,512	176.4 GB

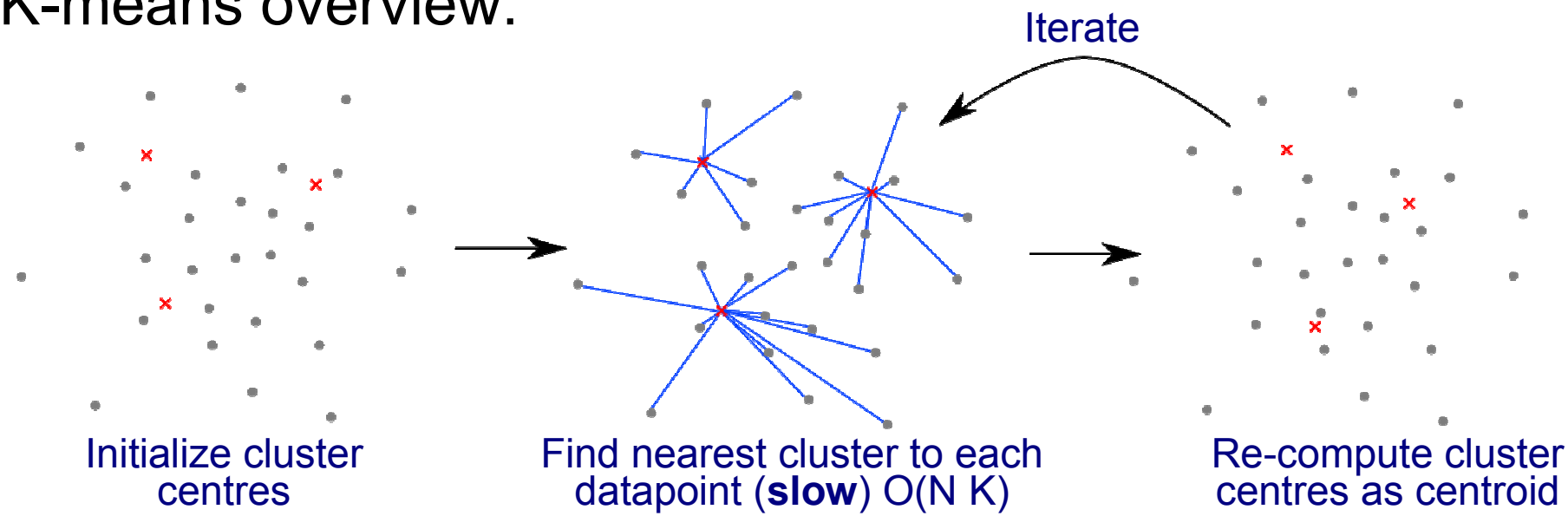
- Dataset (i) crawled by searching for Oxford landmarks
- Datasets (ii) and (iii) from other popular Flickr tags. Acts as additional distractors

Quantization / Clustering

- K-means usually seen as a quick + cheap method
- But far too slow for our needs – D~128, N~20M+, K~1M

K-means overview

- K-means overview:



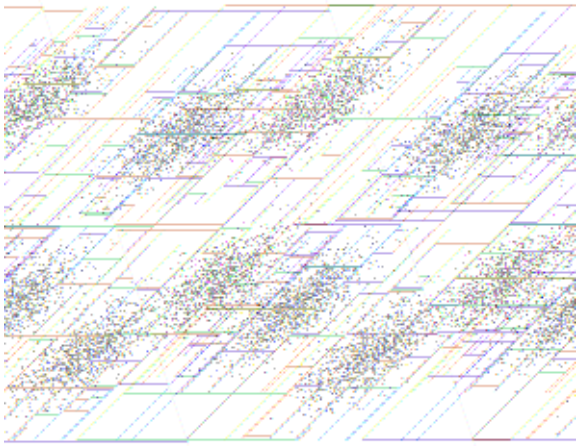
- K-means provably locally minimizes the sum of squared errors (SSE) between a cluster centre and its points

- Idea: nearest neighbour search is the bottleneck – use approximate nearest neighbour search

Approximate K-means

- Use multiple, randomized k-d trees for search

- A k-d tree hierarchically decomposes the descriptor space

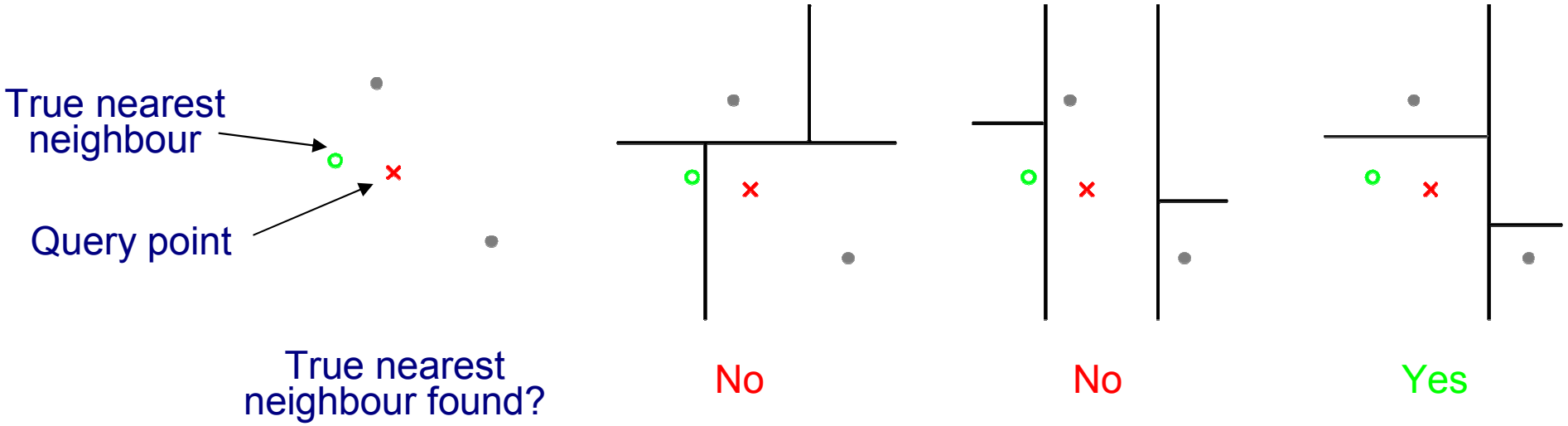


- Points nearby in the space can be found (hopefully) by backtracking around the tree some small number of steps

- Single tree works OK in low dimensions – not so well in high dimensions

Approximate K-means

- Multiple randomized trees increase the chances of finding nearby points



Approximate K-means

- Use the **best-bin first** strategy to determine which branch of the tree to examine next

- share this priority queue between multiple trees – searching multiple trees only slightly more expensive than searching one

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

Approximate K-means

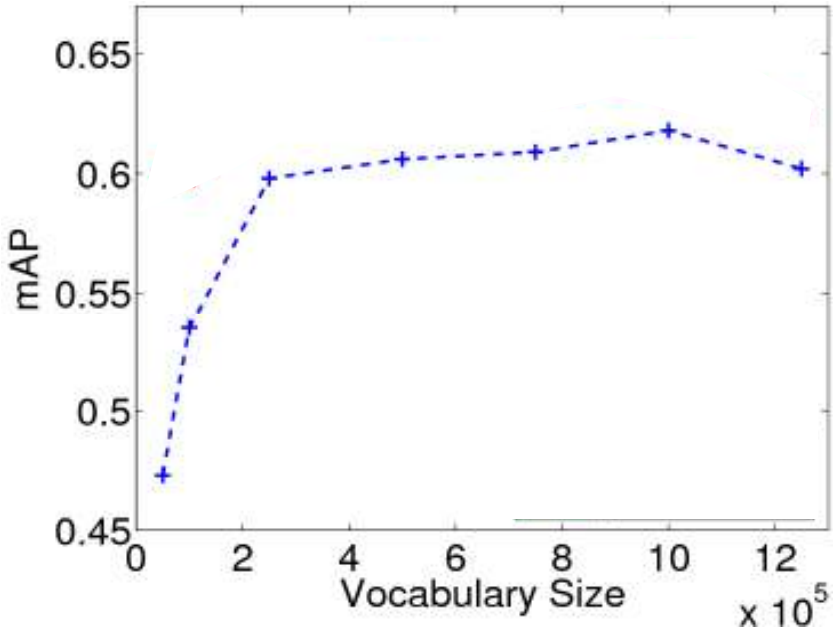
- How accurate is the approximate search?
- Performance on 5K image dataset for a random forest of 8 trees

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

- Allows much larger clusterings than would be feasible with standard K-means: N~17M points, K~1M
  - AKM – 8.3 cpu hours per iteration
  - Standard K-means - estimated 2650 cpu hours per iteration

Approximate K-means

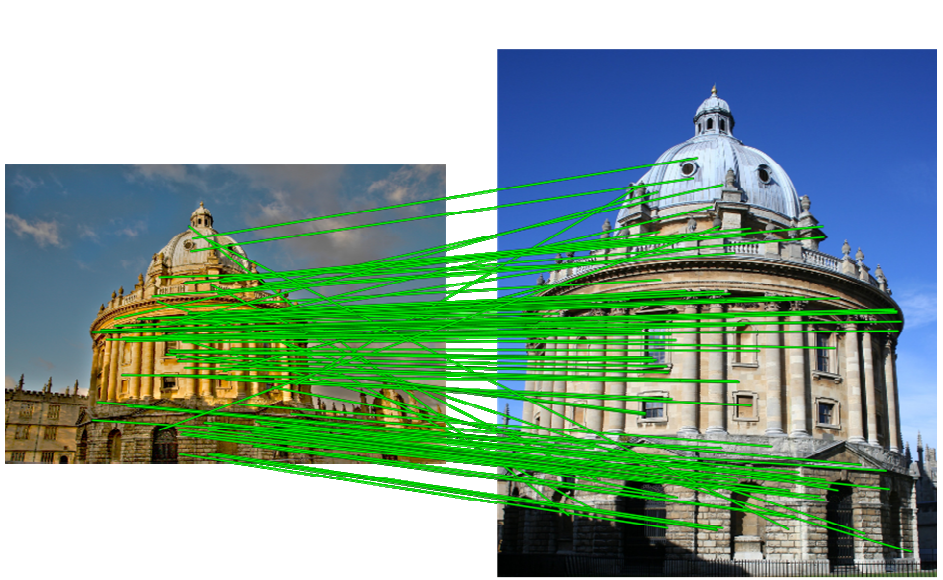
- Using large vocabularies gives a big boost in performance (peak @ 1M words)

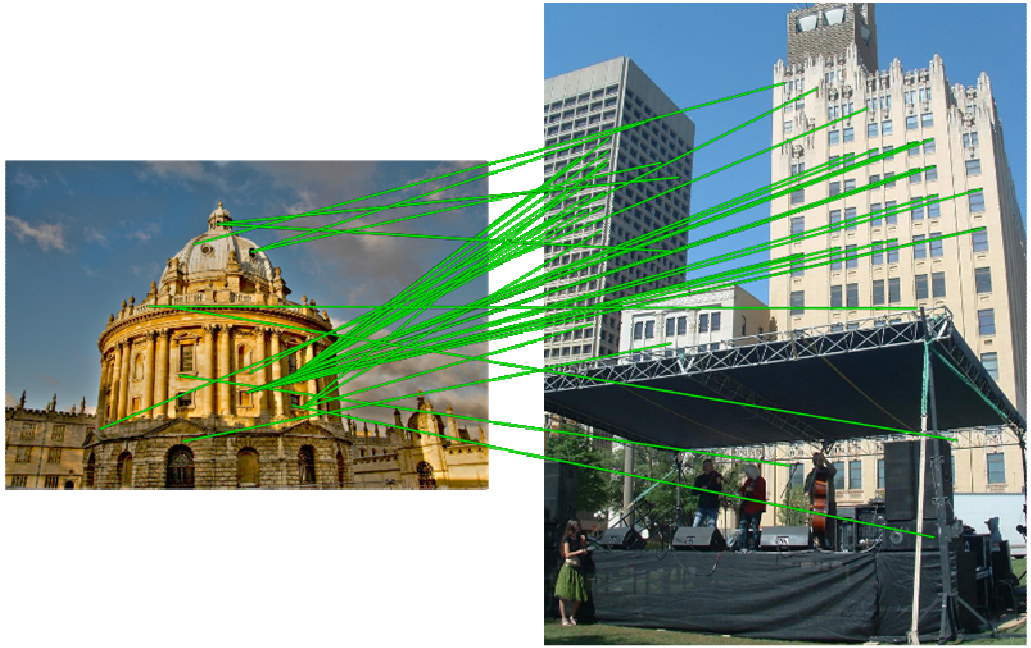


- More discriminative vocabularies give:
  - Better retrieval quality
  - Increased search speed – documents share less words, so fewer documents need to be scored

Beyond Bag of Words

- Use the **position** and **shape** of the underlying features to improve retrieval quality





- Both images have many matches – which is correct?

Beyond Bag of Words

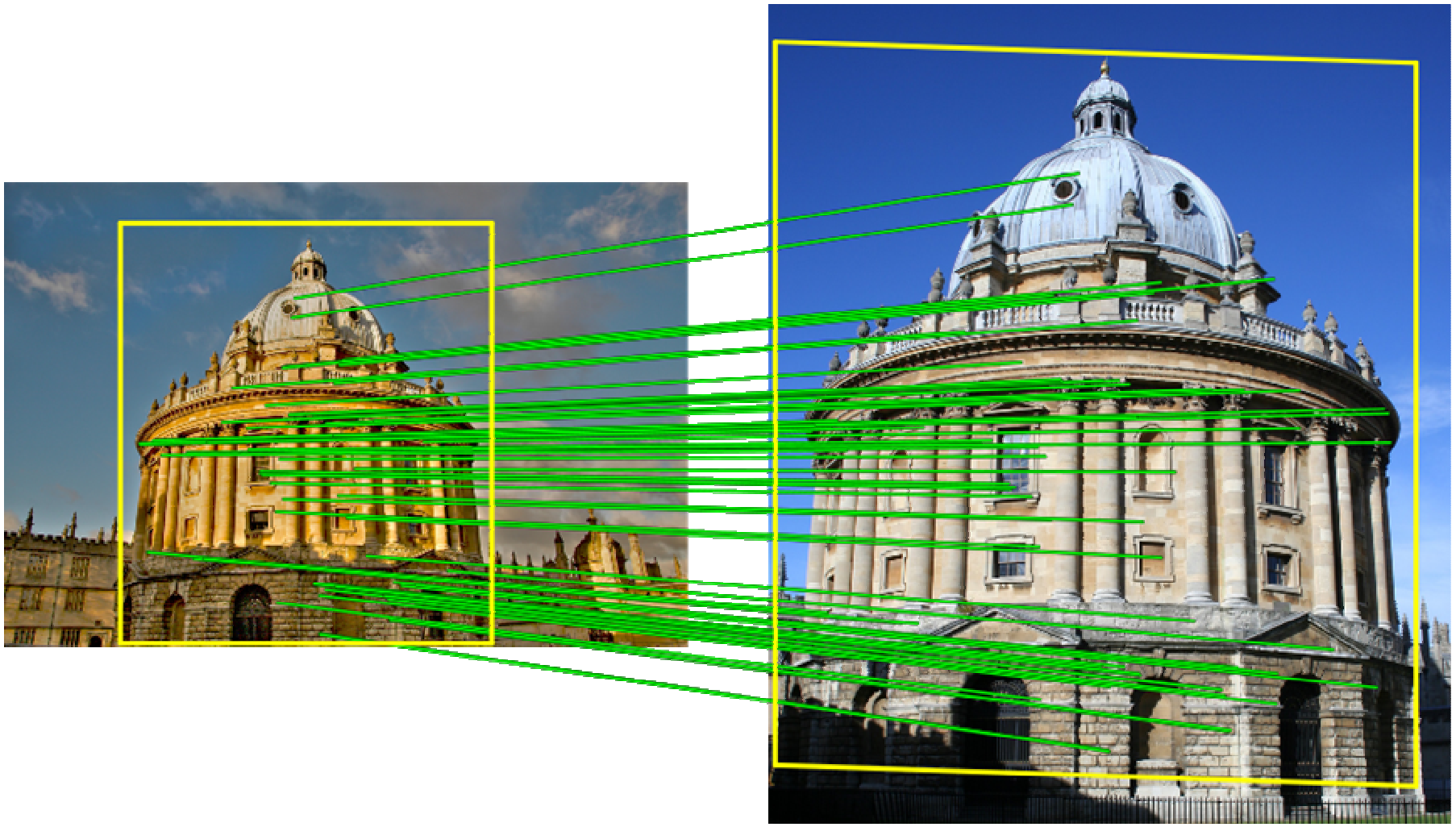
- We can measure **spatial consistency** between the query and each result to improve retrieval quality

Many spatially consistent matches – **correct result**

Few spatially consistent matches – **incorrect result**

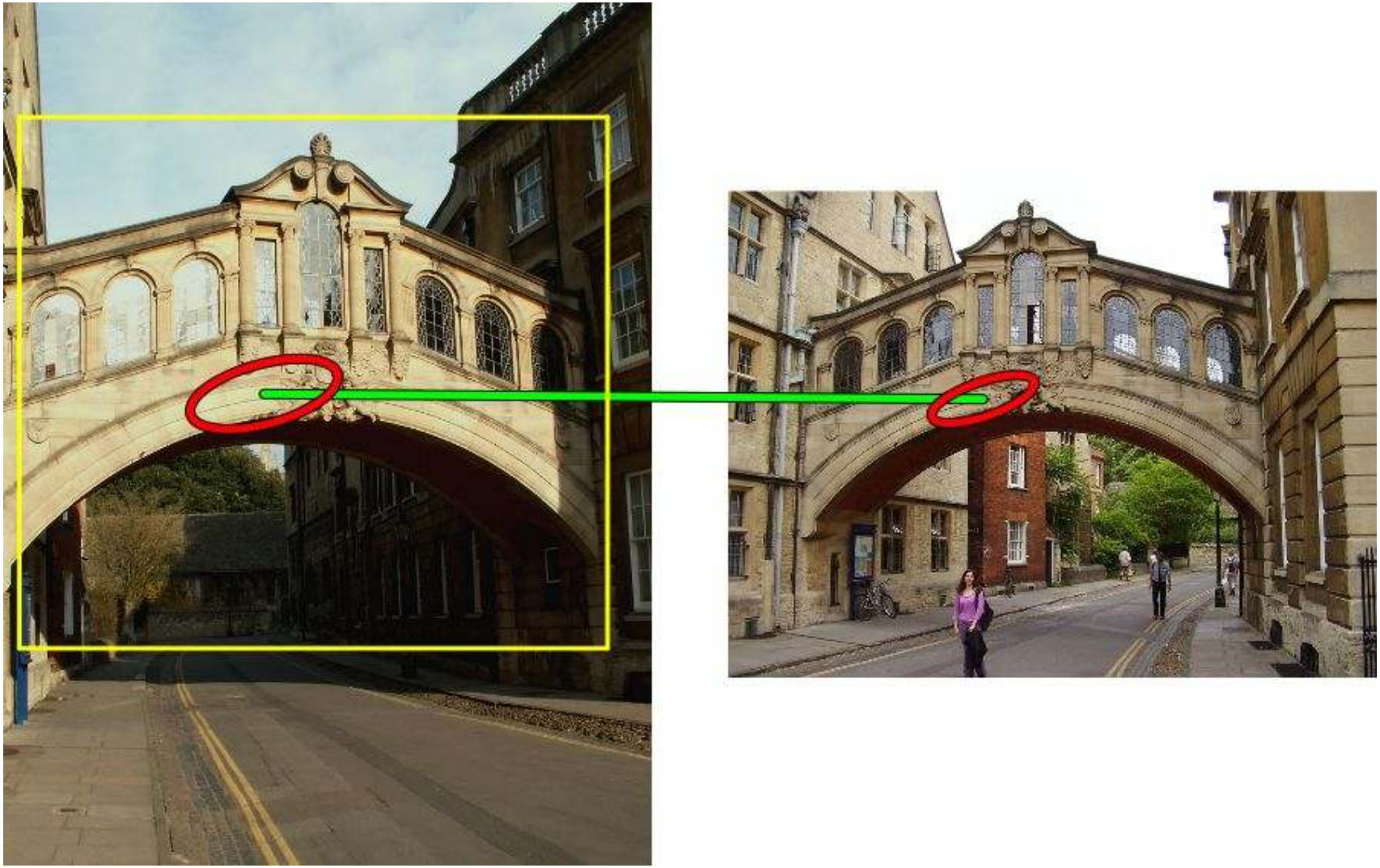
Beyond Bag of Words

- Extra bonus – gives **localization** of the object



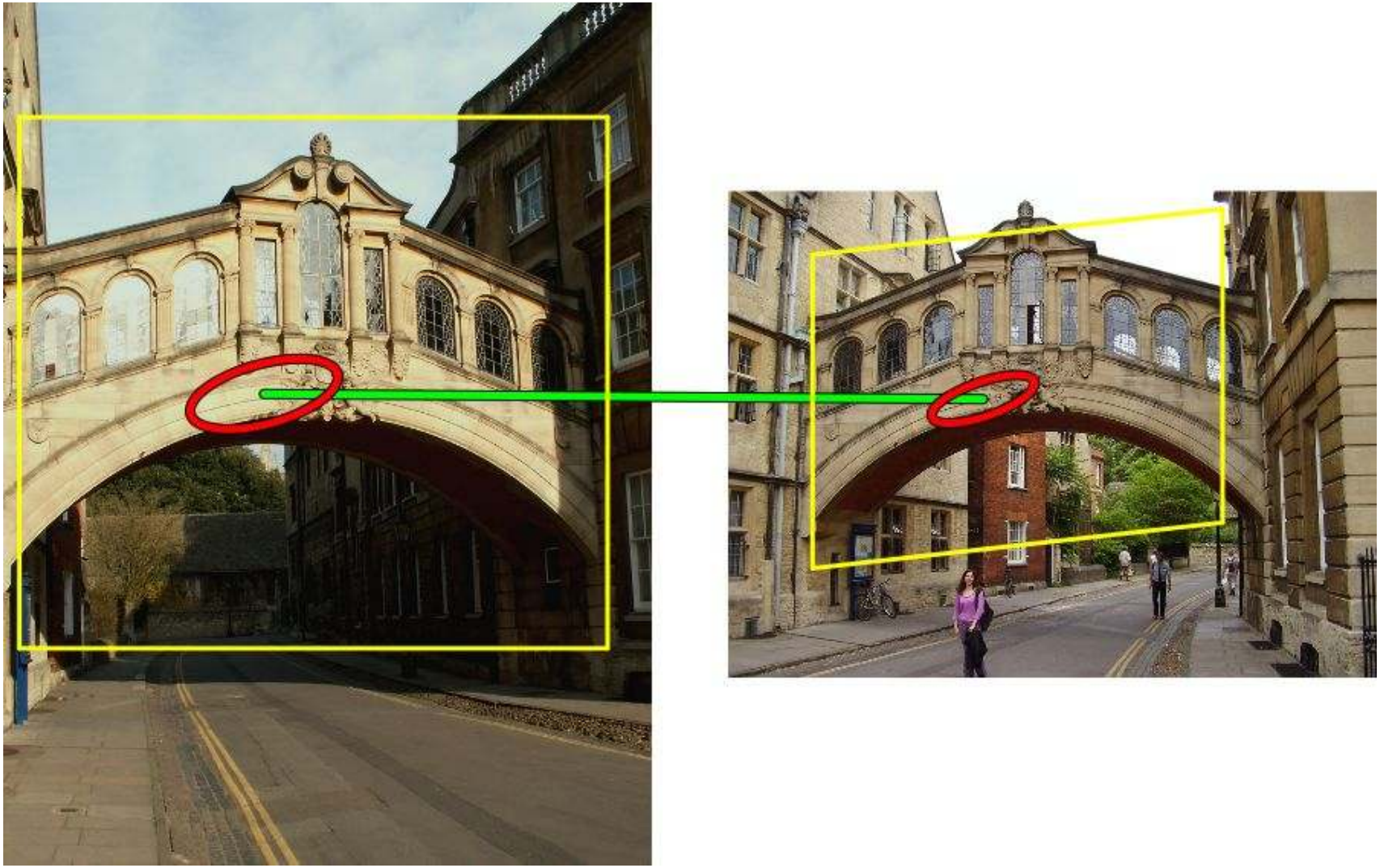
Estimating spatial correspondences

1. Test each correspondence



Estimating spatial correspondences

2. Compute a (restricted) affine transformation (5 dof)



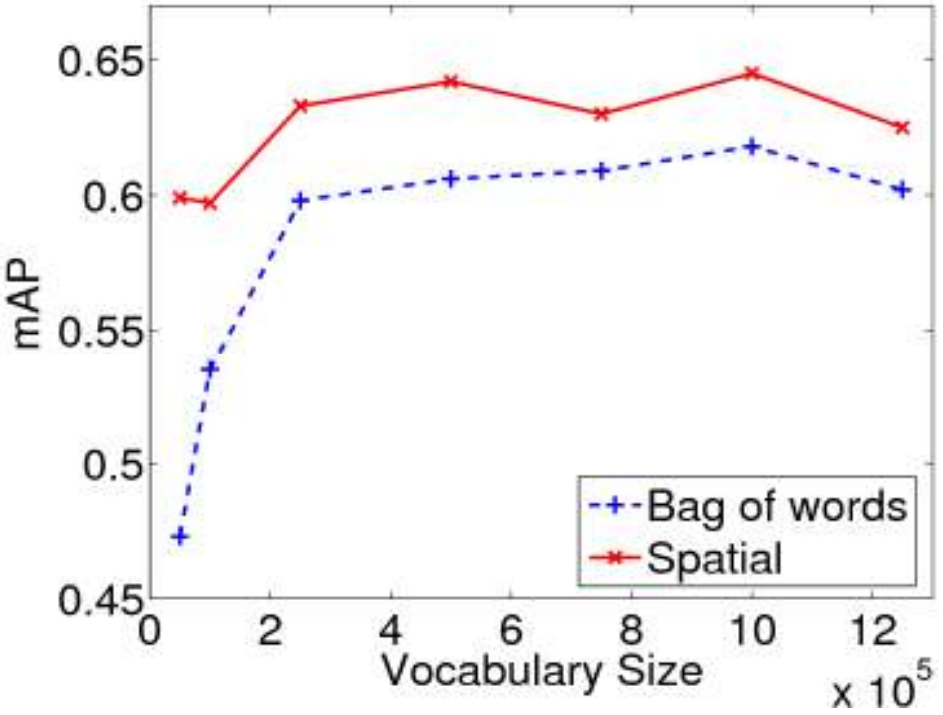
Estimating spatial correspondences

3. Score by number of consistent matches

Use RANSAC on full affine transformation (6 dof)

Mean Average Precision variation with vocabulary size

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




Query: ChristChurch3
Before re−ranking
After re−ranking
Precision
Recall
0
0.2
0.4
0.6
0.8
1

Example Results

Demo

Part 3: Query expansion

Query Expansion in text

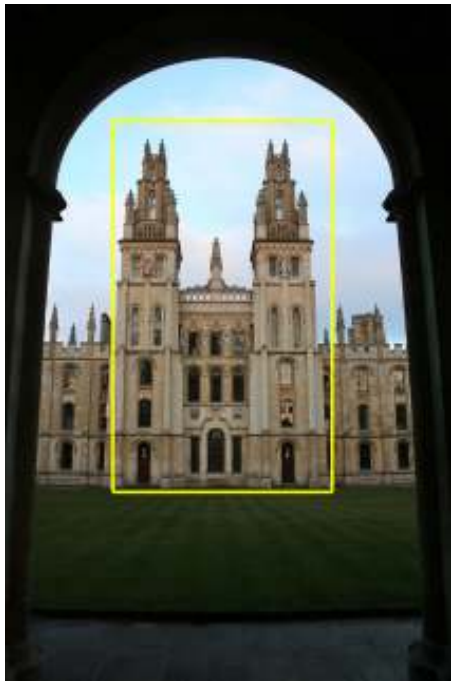
In text :

- Reissue top n responses as queries
- Pseudo/blind relevance feedback
- Danger of topic drift

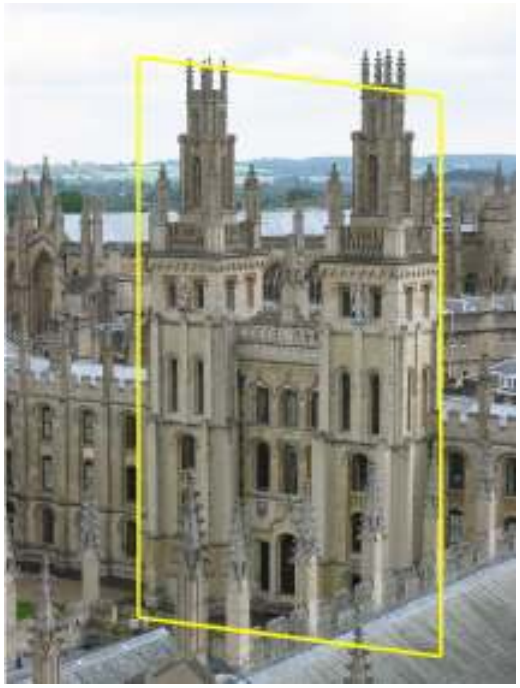
In vision:

- Reissue spatially verified image regions as queries

Query Expansion



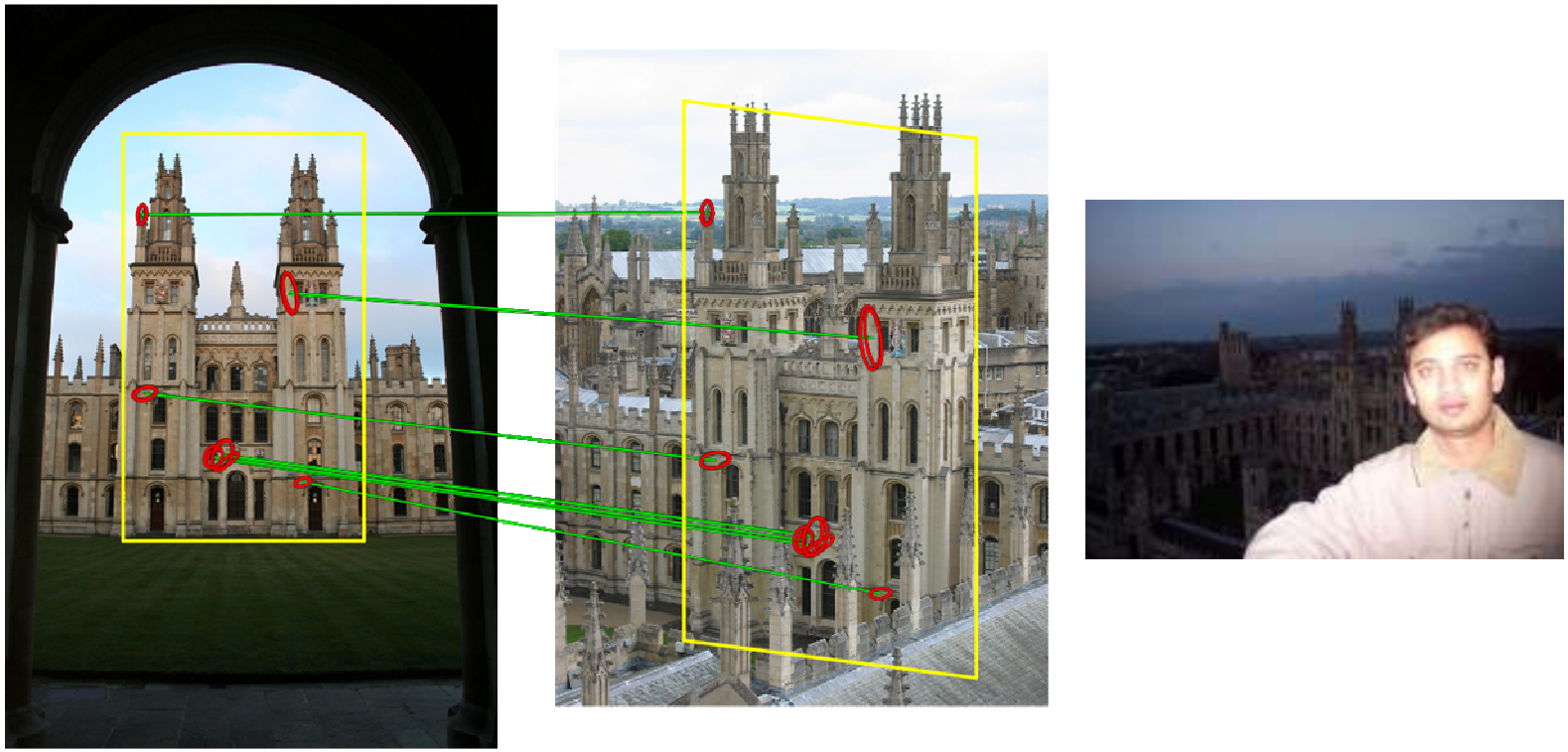
Query Image



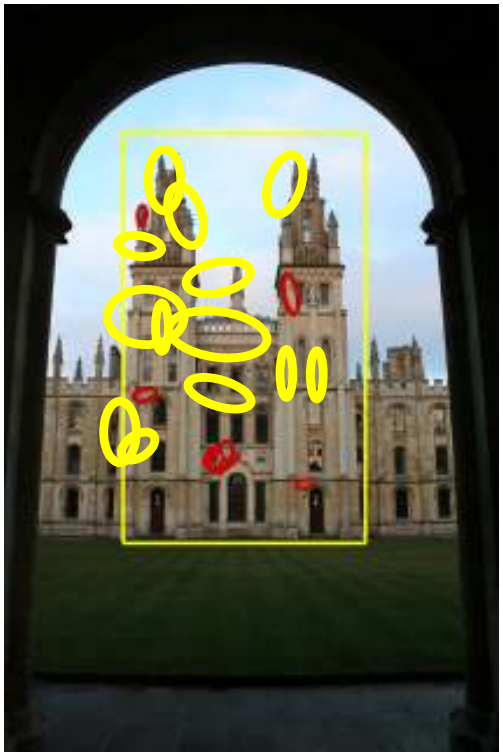
Originally retrieved image

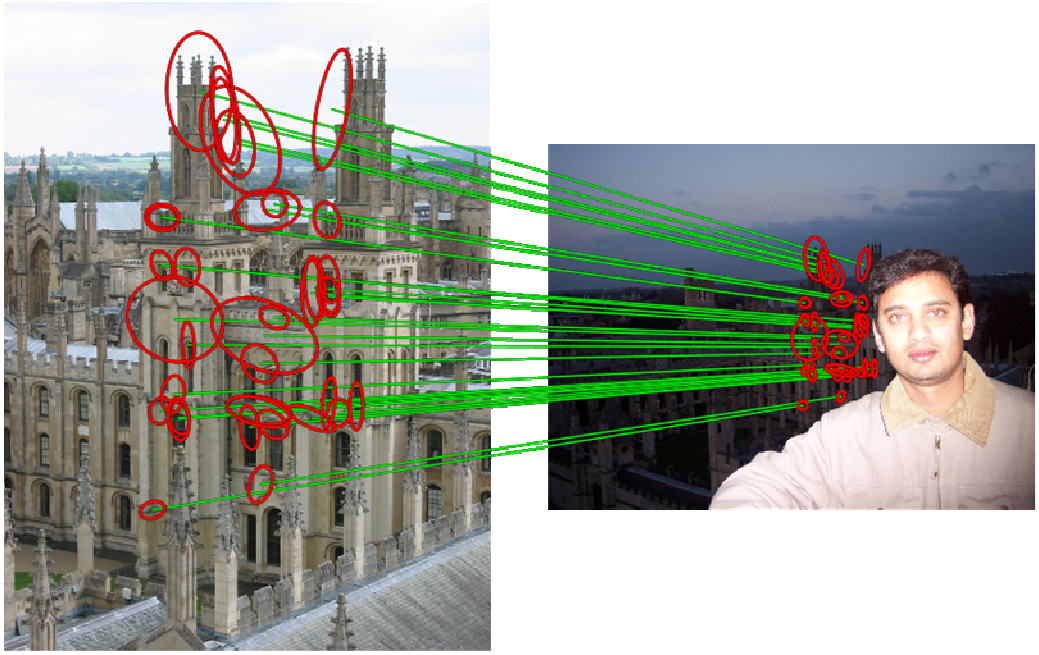
Originally not retrieved

Query Expansion

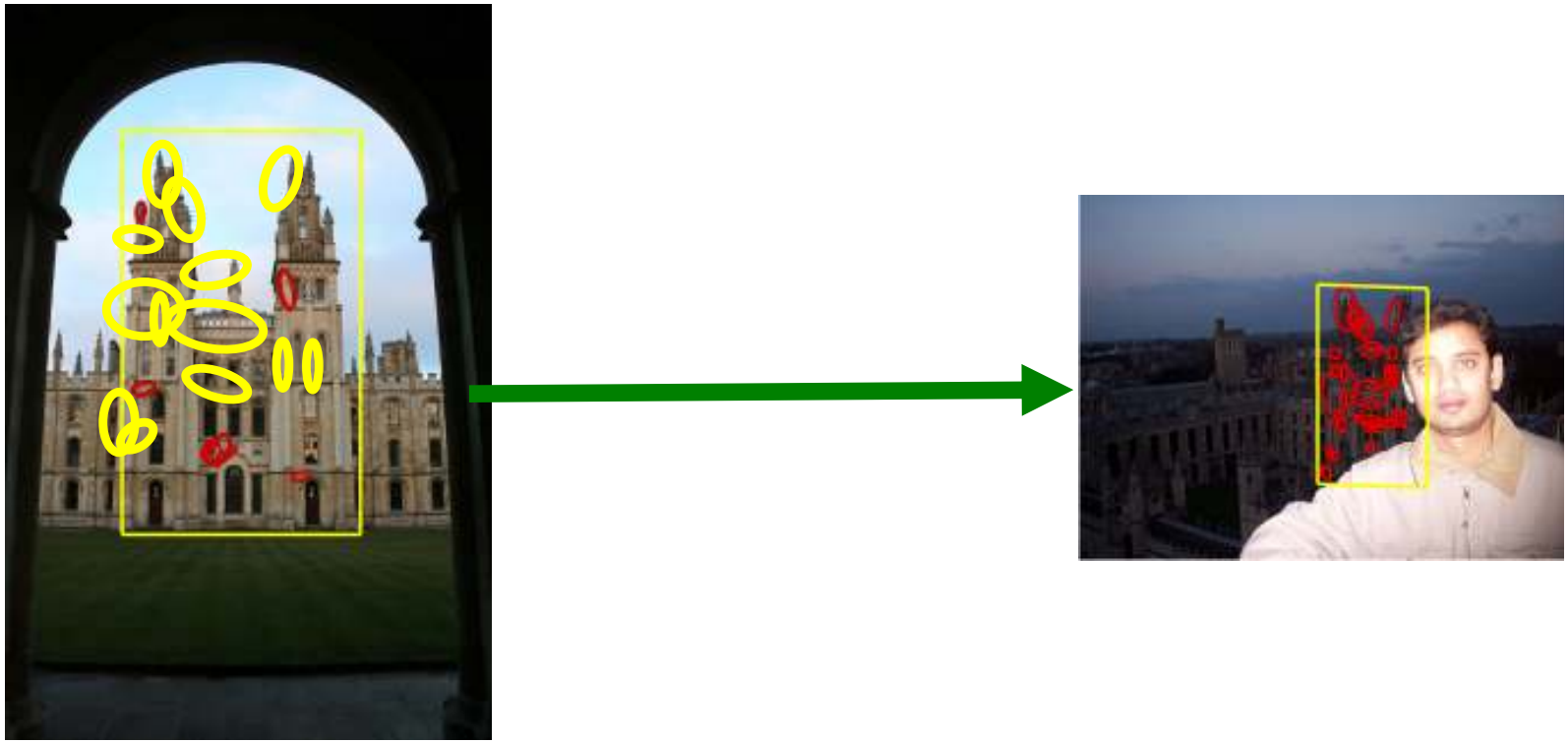


Query Expansion



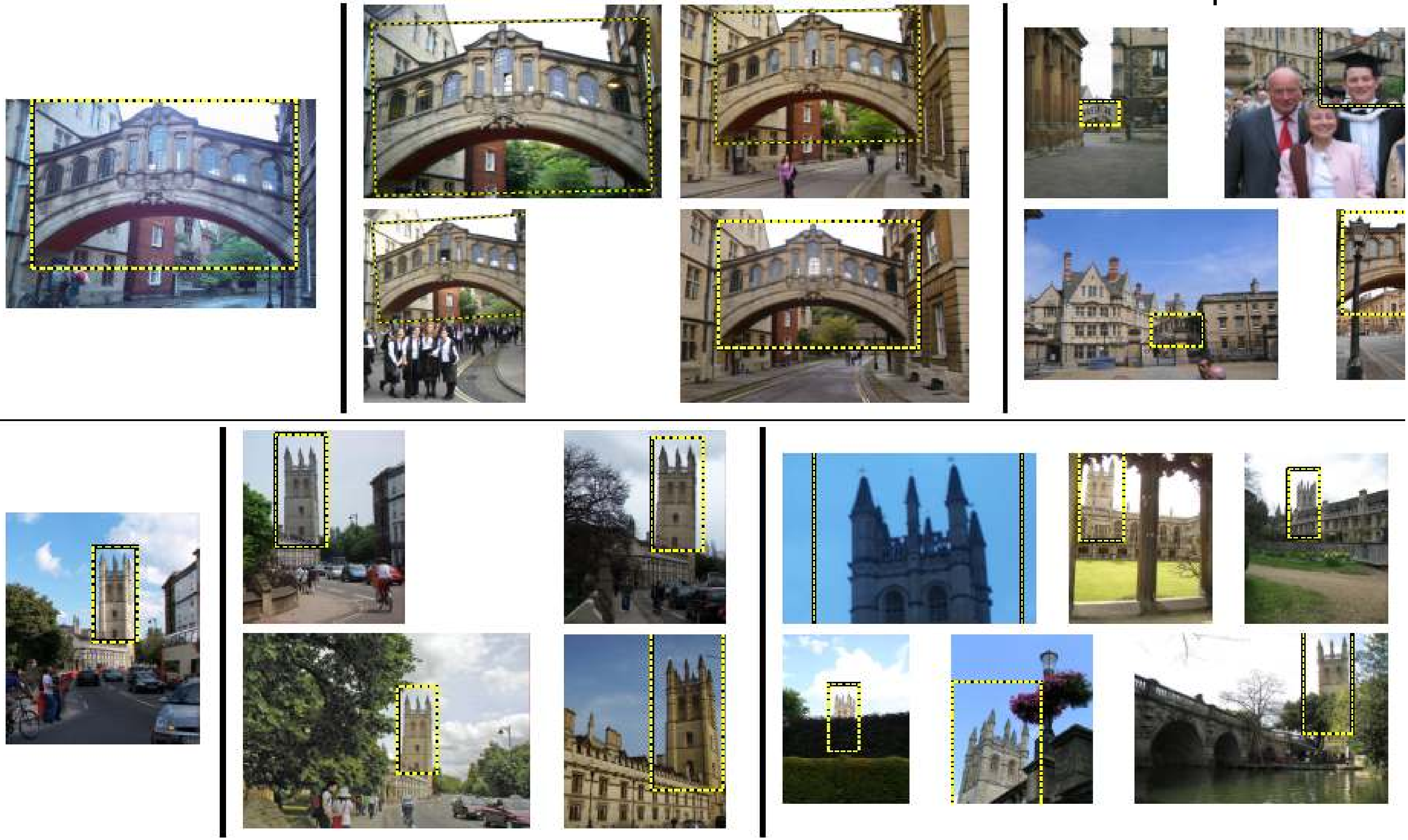


Query Expansion



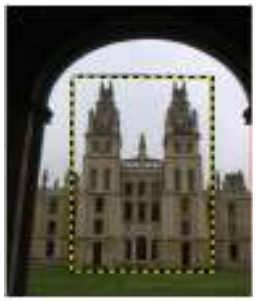
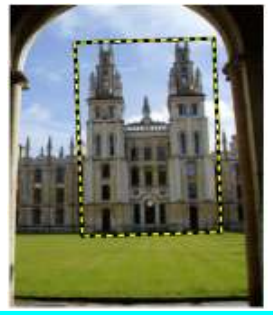
Query Expansion

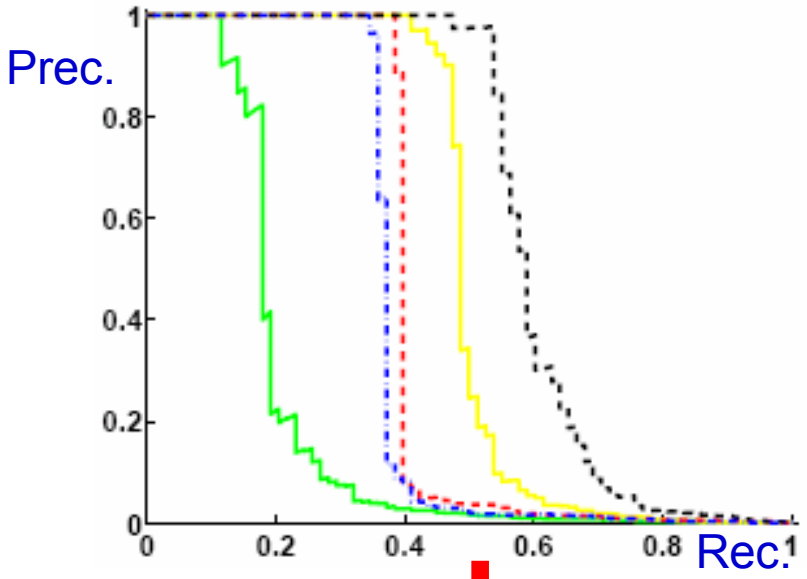
Query image | Originally retrieved | Retrieved only after expansion



Query image

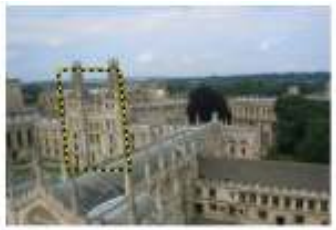
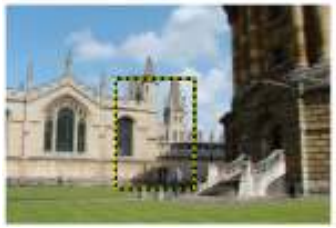
Original results (good)

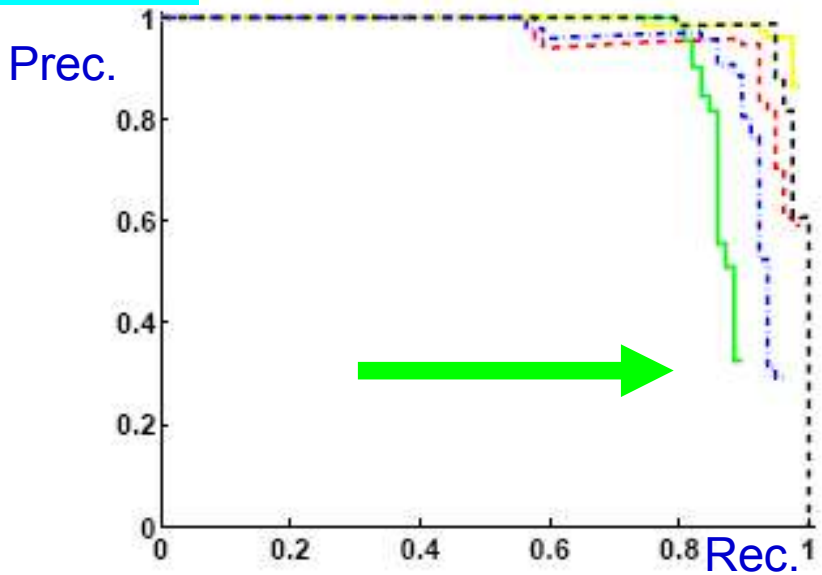




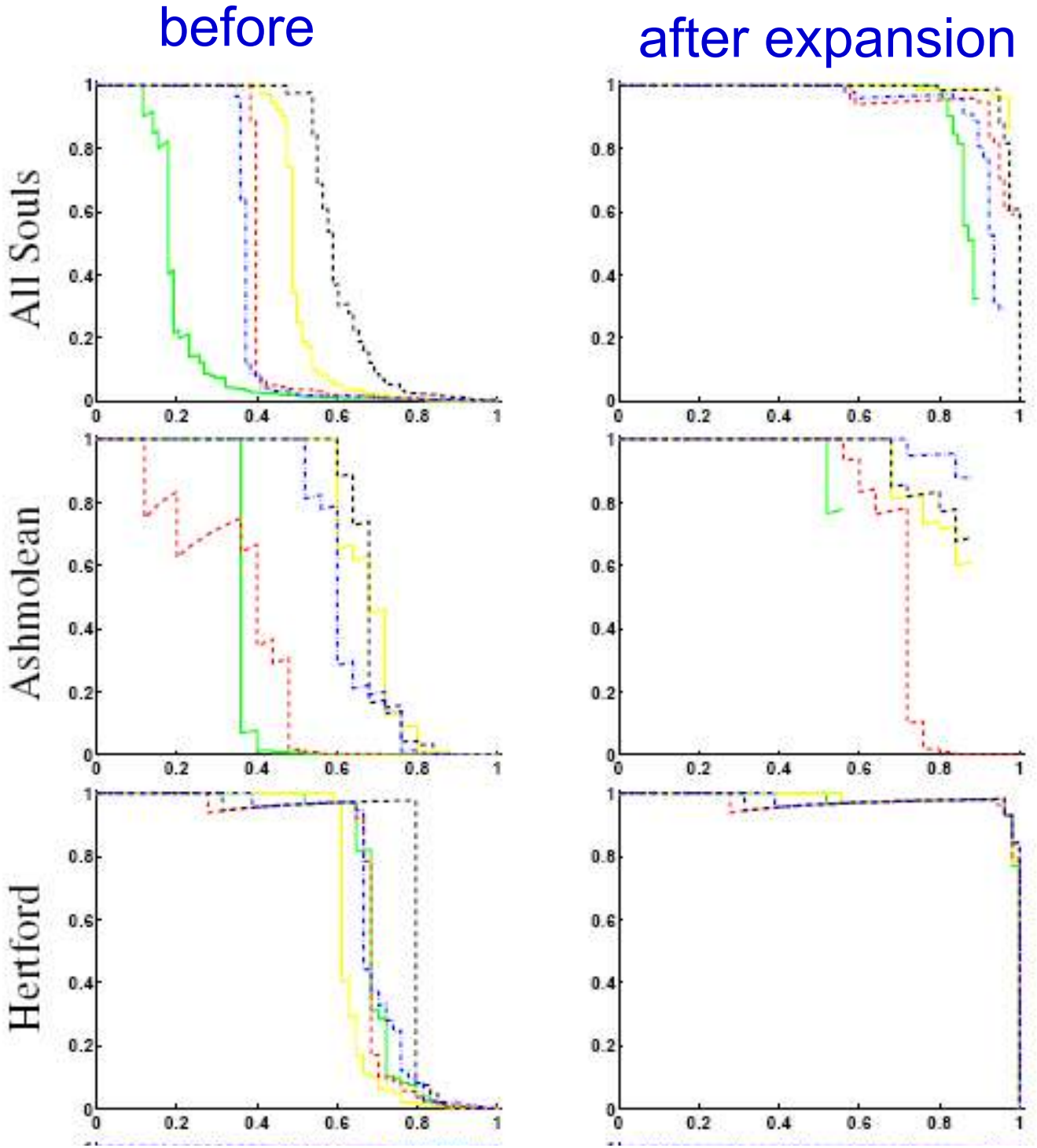
Expanded results (better)



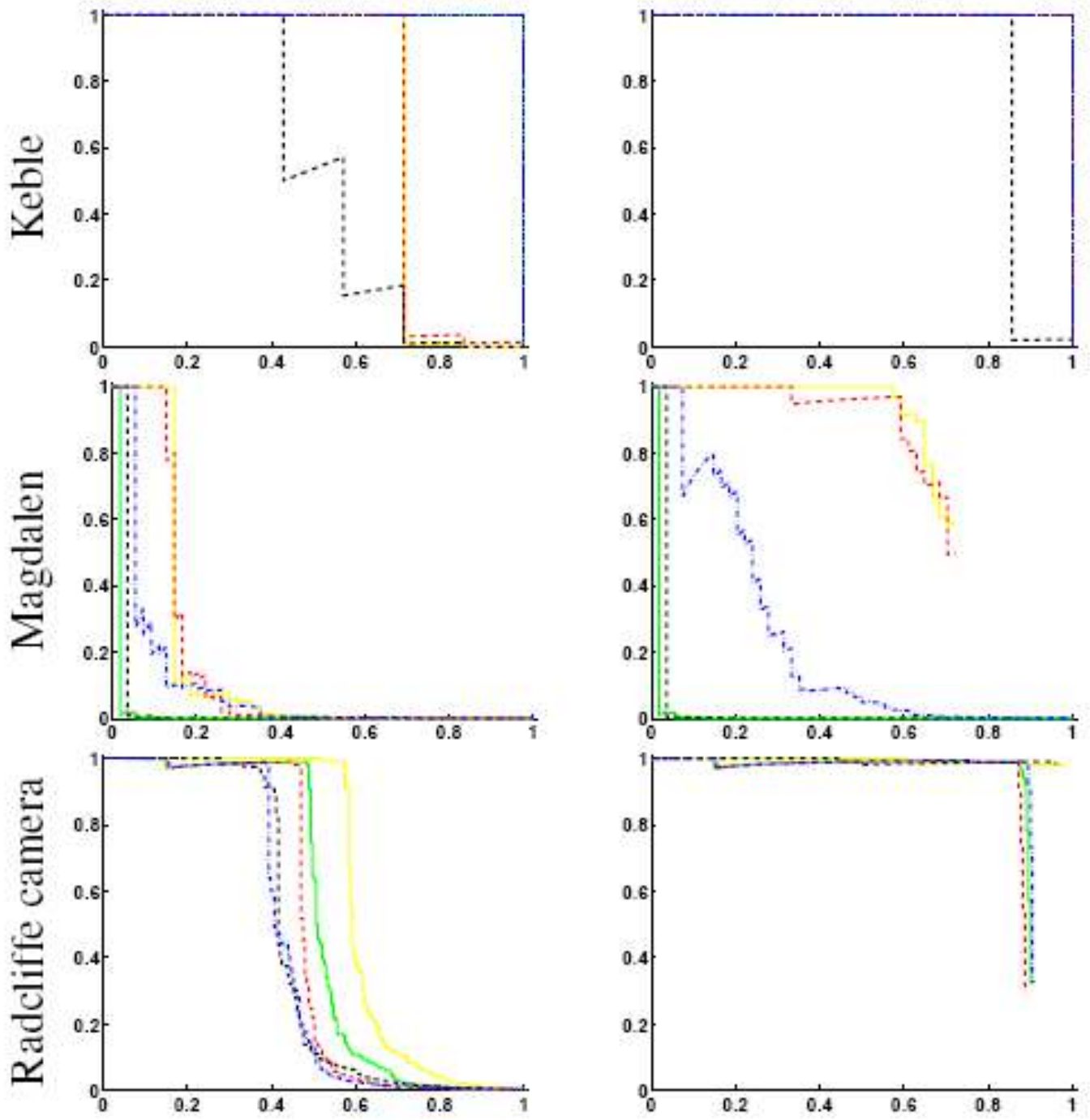


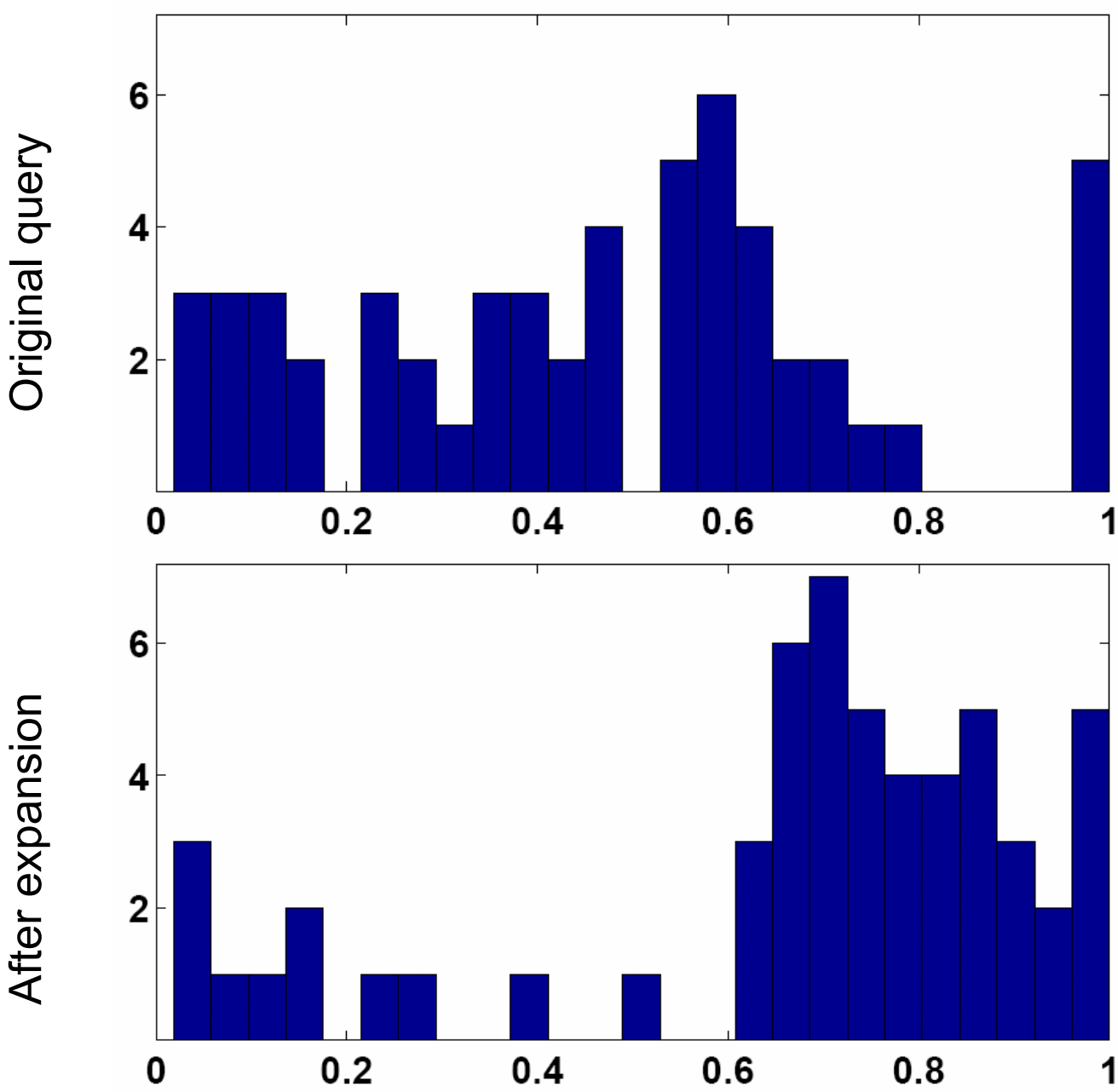
	Ground truth		*Oxford* + *Flickr1* dataset						*Oxford* + *Flickr1* + *Flickr2* dataset					
	OK	Junk	ori	qeb	trc	avg	rec	sca	ori	qeb	trc	avg	rec	sca
All Souls	78	111	41.9	49.7	85.0	76.1	85.9	**94.1**	32.8	36.9	80.5	66.3	73.9	**84.9**
Ashmolean	25	31	53.8	35.4	51.4	66.4	74.6	**75.7**	41.8	25.9	45.4	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**


before
after expansion
All Souls
Ashmolean
Hertford
0
0.2
0.4
0.6
0.8
1

before

after expansion




Original query
After expansion
0
2
4
6
0.2
0.4
0.6
0.8
1
Average Precision histogram for 55 queries

Demo

Summary and Extensions

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

Other examples of text methods ported to vision:

- Data mining – see Till Quack's talk
- Use of topic models, e.g. pLSA and LDA for object and scene categories

Papers and Demo

Sivic, J. and Zisserman, A.
Video Google: A Text Retrieval Approach to Object Matching in Videos
Proceedings of the International Conference on Computer Vision (2003)
http://www.robots.ox.ac.uk/~vgg/publications/papers/sivic03.pdf

Demo: http://www.robots.ox.ac.uk/~vgg/research/vgoogle/

Philbin, J., Chum, O., Isard, M., Sivic, J. and Zisserman, A.
Object retrieval with large vocabularies and fast spatial matching
Proceedings of the Conference on Computer Vision and Pattern Recognition(2007)
http://www.robots.ox.ac.uk/~vgg/publications/papers/philbin07.pdf

Chum, O., Philbin, J., Isard, M., Sivic, J. and Zisserman, A.
Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval
Proceedings of the International Conference on Computer Vision (2007)
http://www.robots.ox.ac.uk/~vgg/publications/papers/chum07b.pdf