Towards On-the-fly Large Scale Video Search

Andrew Zisserman

Relja Arandjelović, Ken Chatfield, Omkar Parkhi

Visual Geometry Group,
Dept of Engineering Science,
University of Oxford

The Vision

All visual material (images, video) should be searchable for anything

- people, object categories, scene categories, particular objects, human actions and interactions, activities …

and retrieved with high precision and high recall








The Problem

There exist large data sets of images and videos lacking almost any annotation (apart from the date), e.g.

- archive datasets
- personal photo and video collections

The Problem

There exist large data sets of images and videos with sufficient annotations to retrieve thousands of examples of a query, e.g.

- Photos on web pages – Google Image Search
- Flickr, Facebook

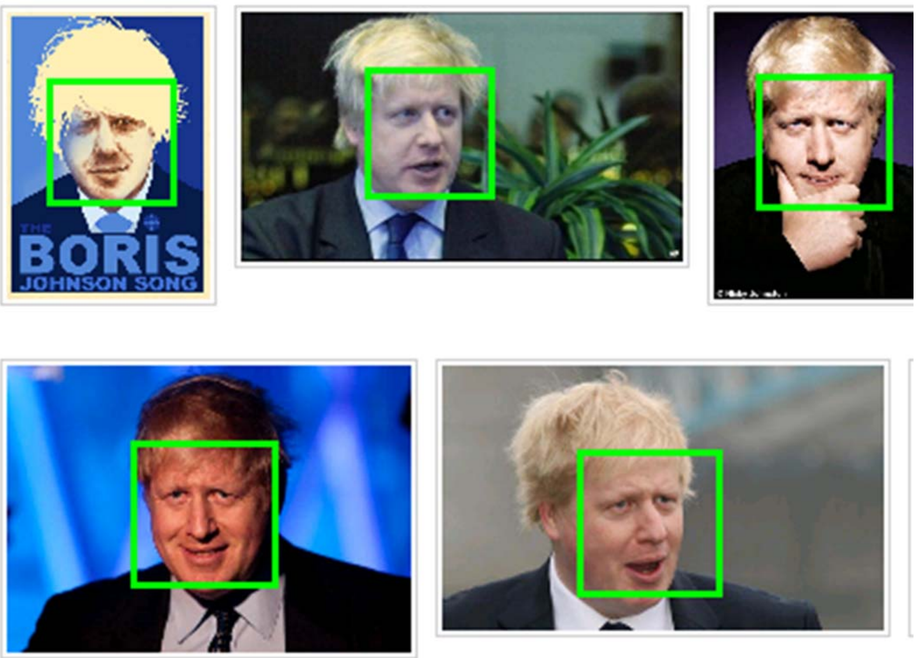
The Solution

To harness some of the information from annotation rich sources and use it to enable searching of annotation starved datasets:

- efficiently, and
- in a scalable manner

On-the-fly search for faces

Download images and detect faces



Person X classifier

ranked frames

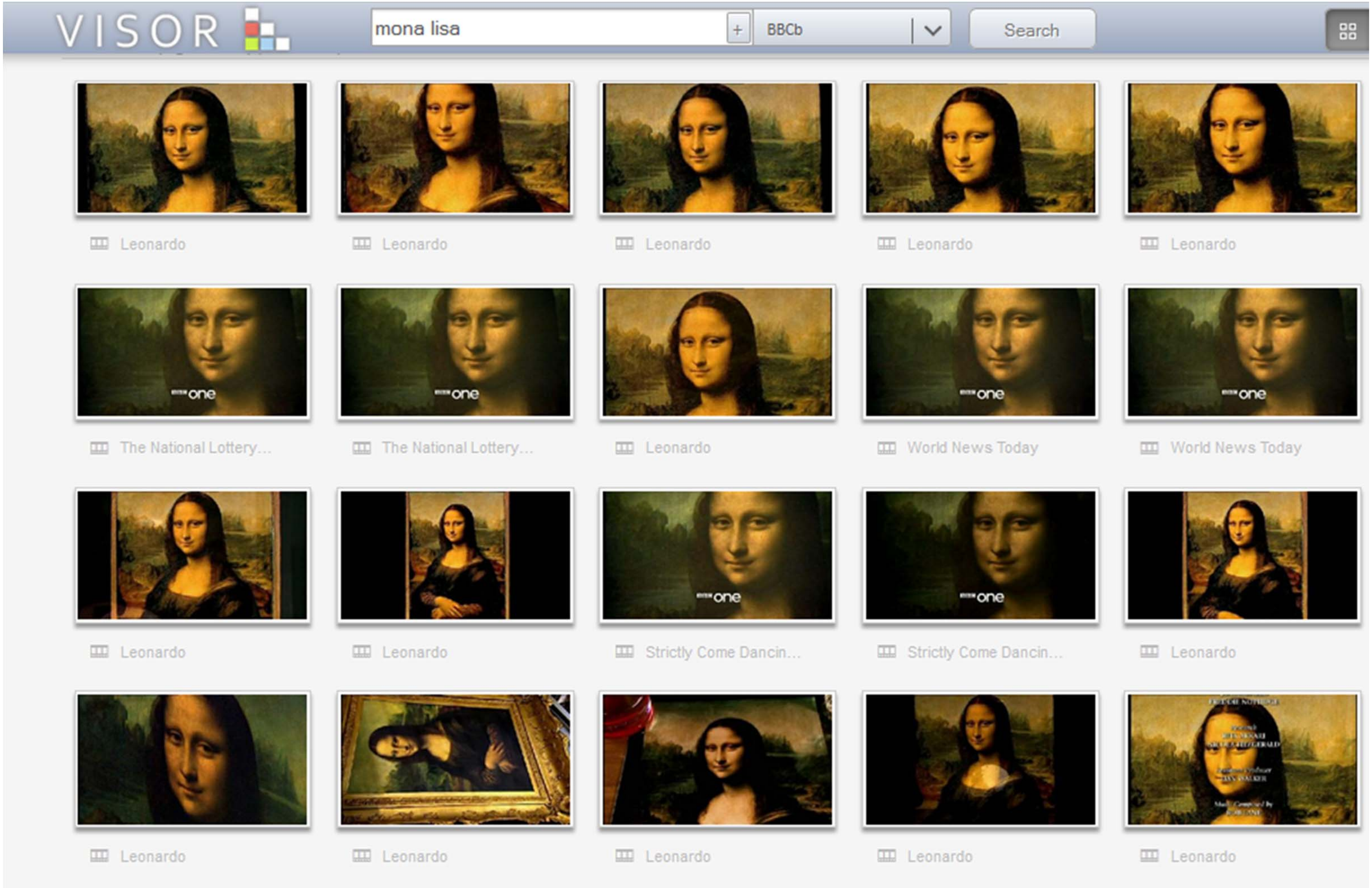
(BBC corpus)

Can search for anyone

Video dataset: BBC TV

- 4372 broadcasts from BBC 1, 2, 3 & 4
- Programmes from late 2011 to early 2012 from prime time slot (7pm-12pm) over five months
- 3007 hours of video represented by 1 frame per second
- 11M seconds of data, 3M keyframes
- Frames are 480 x 270 pixels

Instance Search – Example `Mona Lisa'



Outline

On-the-fly ***instance*** search

- Specific places/scenes/objects e.g. White house, Mona Lisa, HSBC logo

On-the-fly ***category*** search

- Object and scene categories, e.g. cars, crowds, forest

On-the-fly ***face*** search

- Particular people and attributes, e.g. Obama, moustache

Long history of learning from Google images

- Berg & Forsyth, CVPR 06
- Fergus *et al.*, ICCV 05
- Li *et al.*, CVPR 07
- Liu *et al*., ACM MM 09
- Schroff *et al*., ICCV 07
- Sivic & Zisserman, ICCV 03, Proc. IEEE 08
- Torresani *et al*., ECCV 10, NIPS 11

1. On-the-fly Instance Search

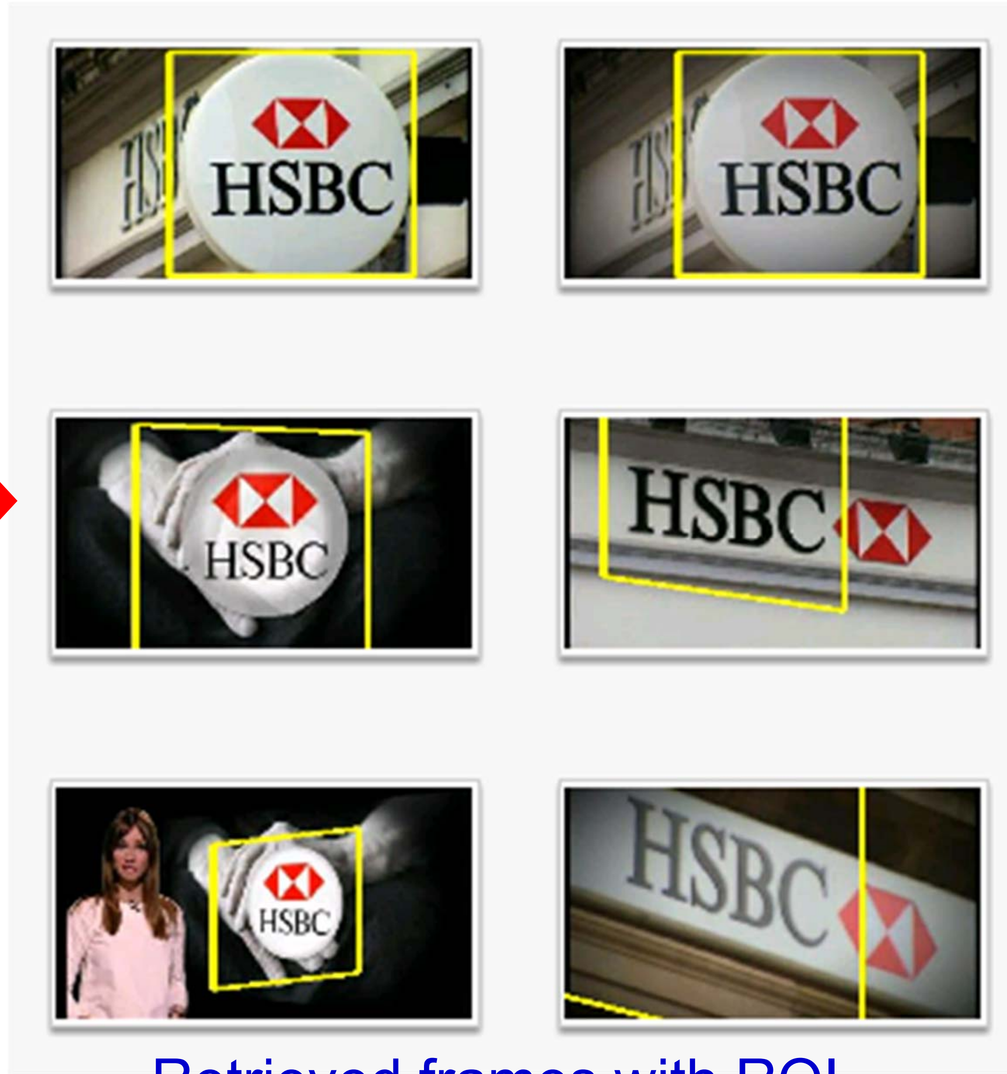
Instance Search

Query by example image: retrieve specific objects, unaffected by: scale, viewpoint, lighting, partial occlusion

Query image

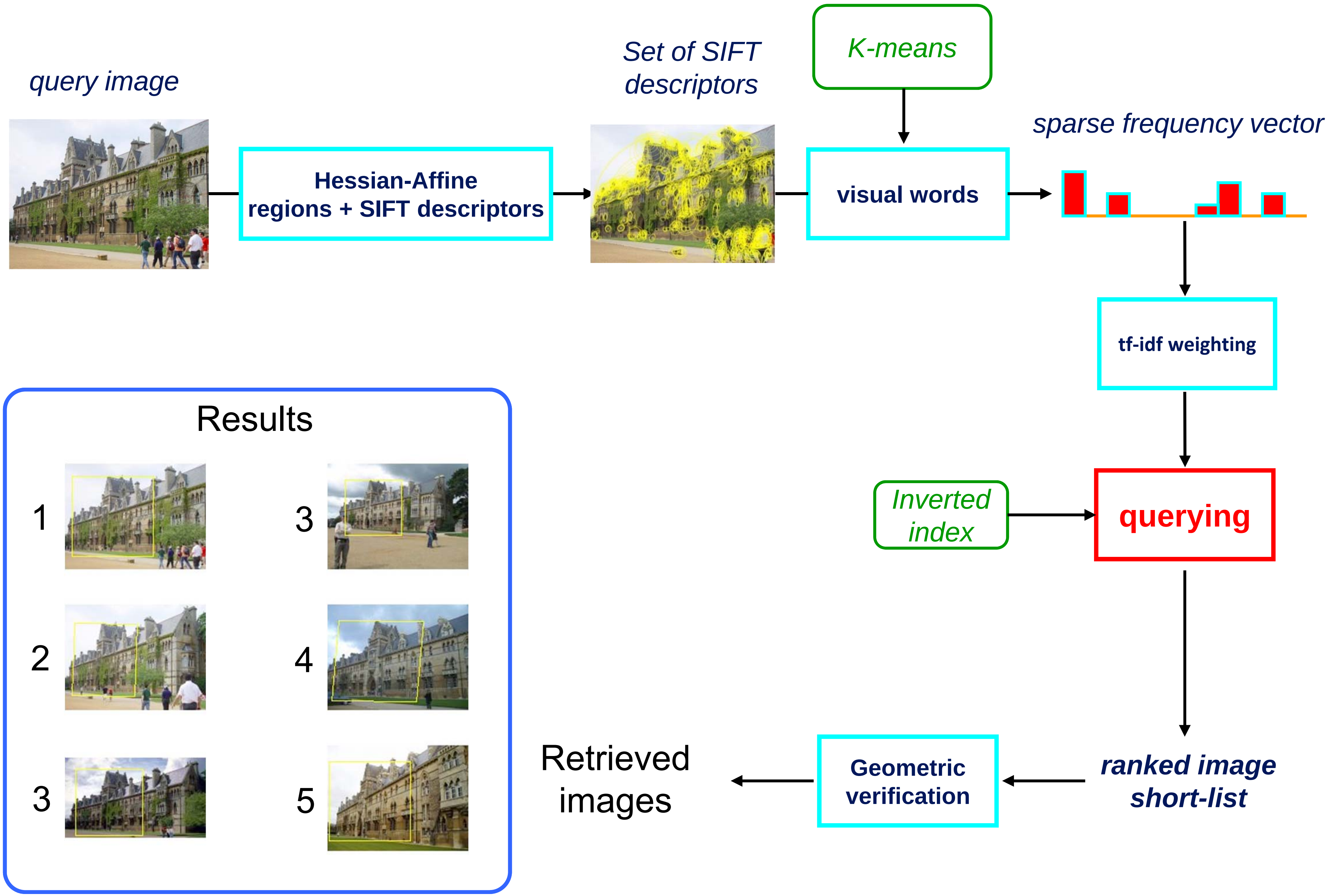
Download from web
(external)

(not category recognition)

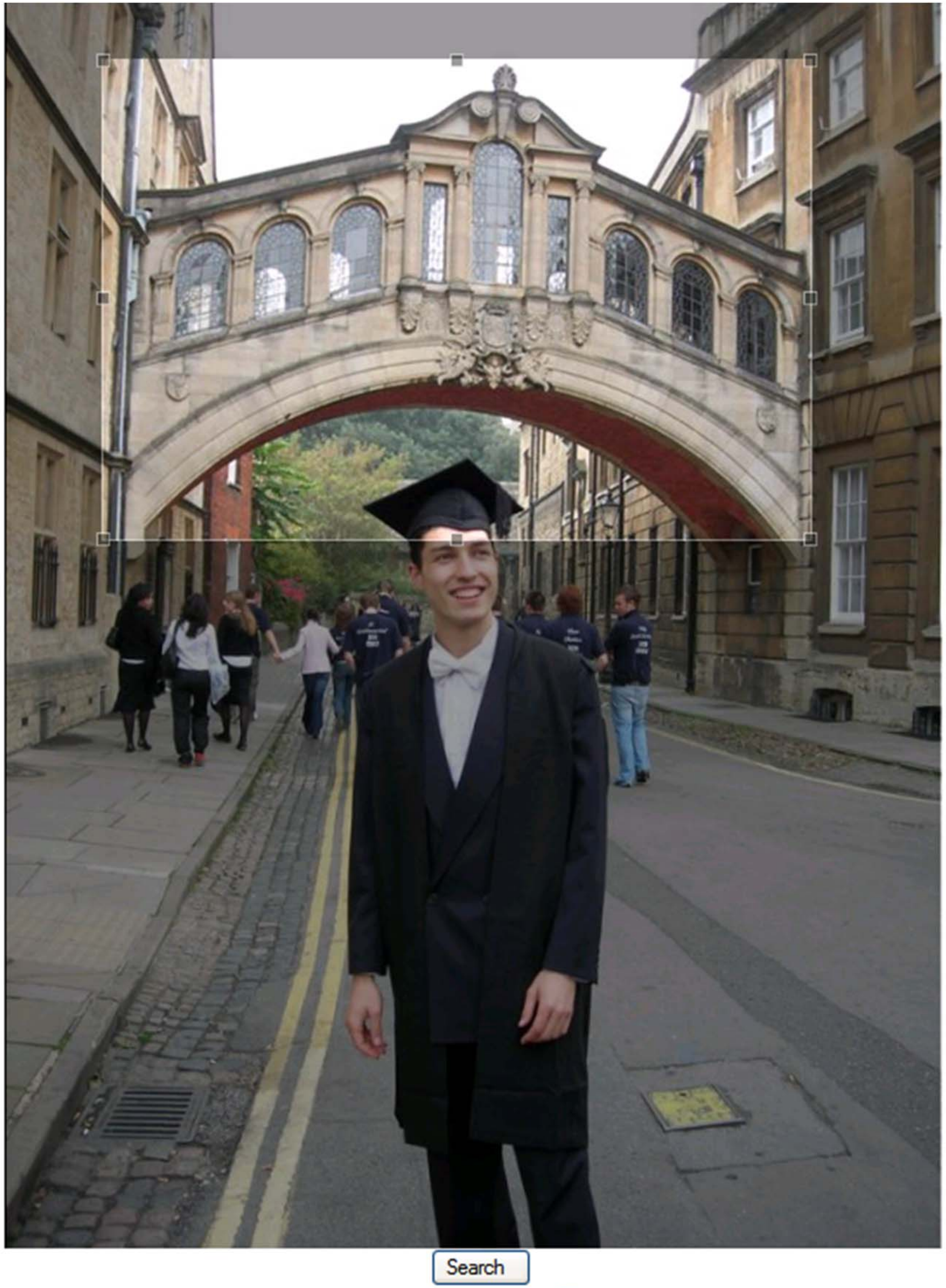


Retrieved frames with ROI

Visual engine: bag of visual words particular object retrieval



Example

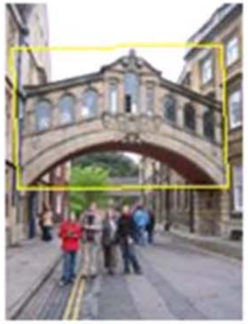


Search results 1 to 20 of 104844

1

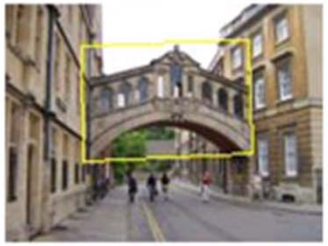
ID: oxc1_hertford_000011
Score: 1816.000000
Putative: 2325
Inliers: 1816
Hypothesis: 1.000000 0.000000 0.000015 0.000000 1.000000 0.000031
Detail

2



ID: oxc1_all_souls_000075
Score: 352.000000
Putative: 645
Inliers: 352
Hypothesis: 1.162245 0.041211 -70.414459 -0.012913 1.146417 91.276093
Detail

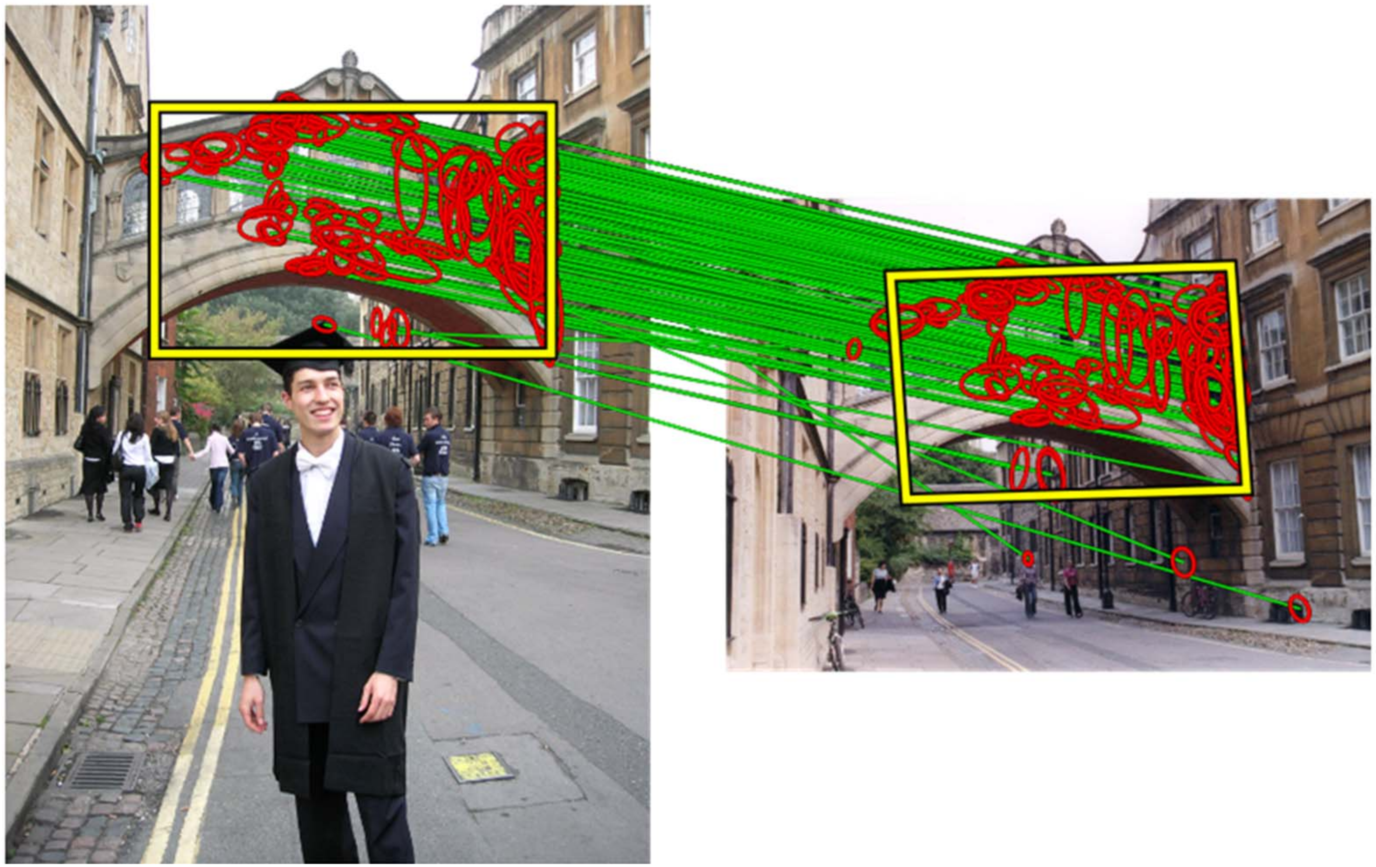
3



ID: oxc1_hertford_000064
Score: 278.000000
Putative: 527
Inliers: 278
Hypothesis: 0.928686 0.026134 169.954620 -0.041703 0.937558 97.962112
Detail

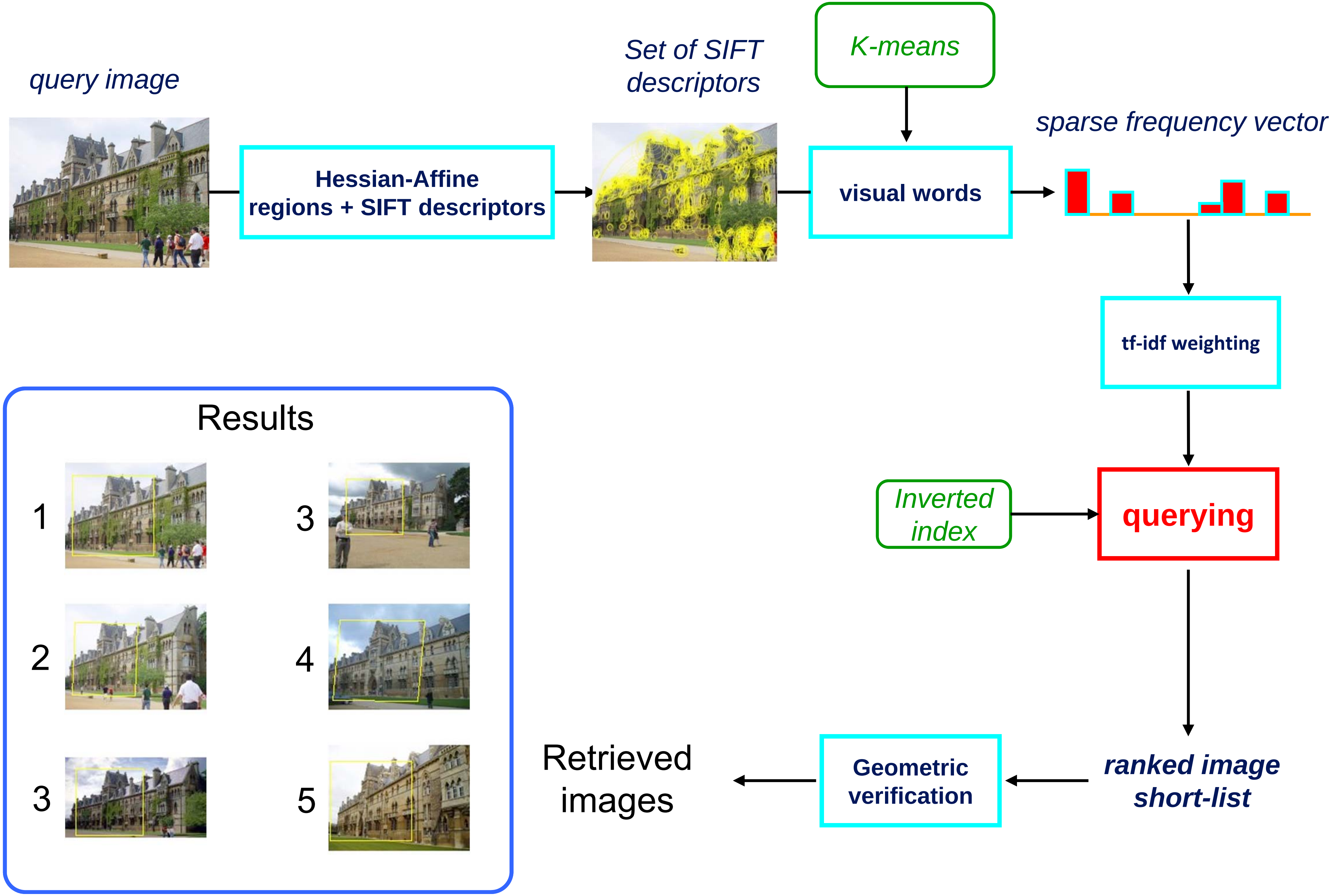
Spatial verification

Use the spatial distribution of the detections in the image to improve retrieval quality – re-rank short list by number of matches



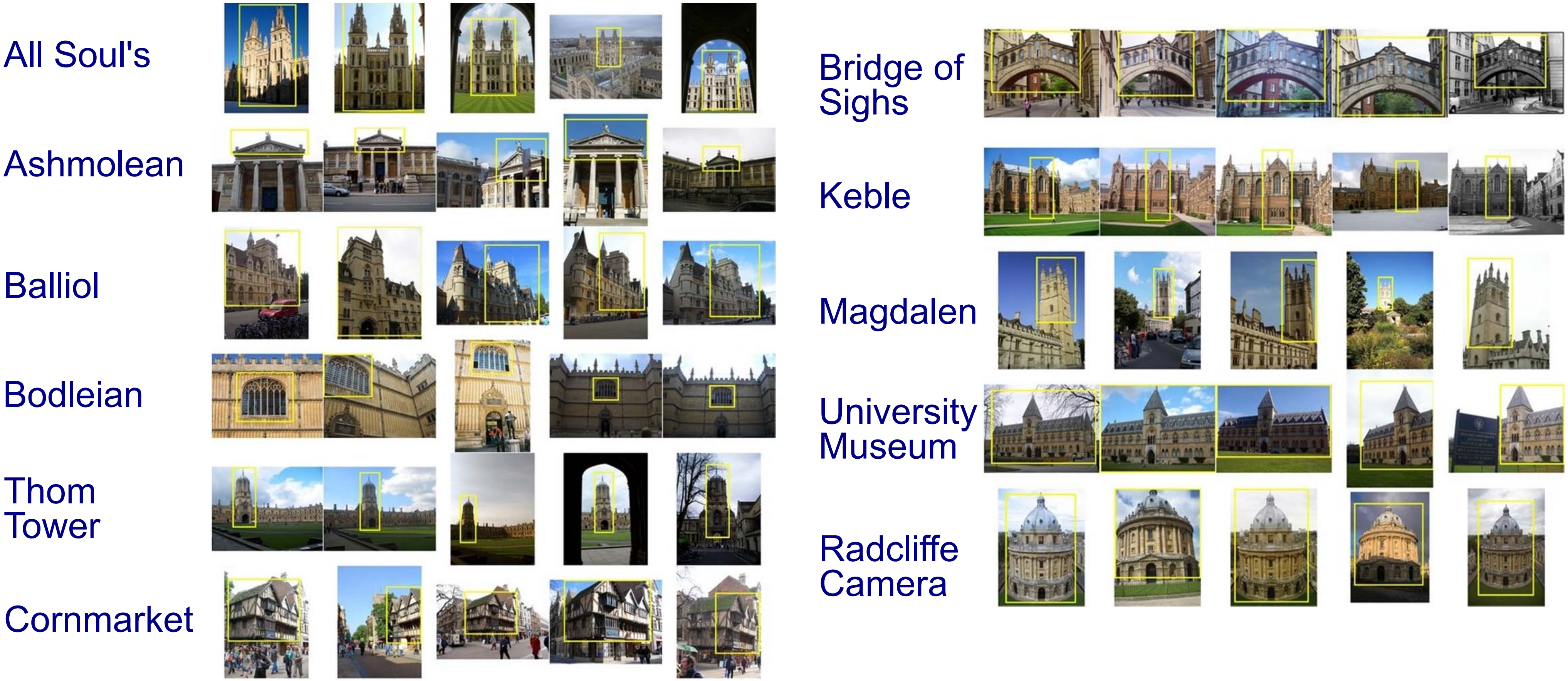
SIFT matches consistent with an affine transformation

Visual engine: bag of visual words particular object retrieval



Oxford buildings dataset

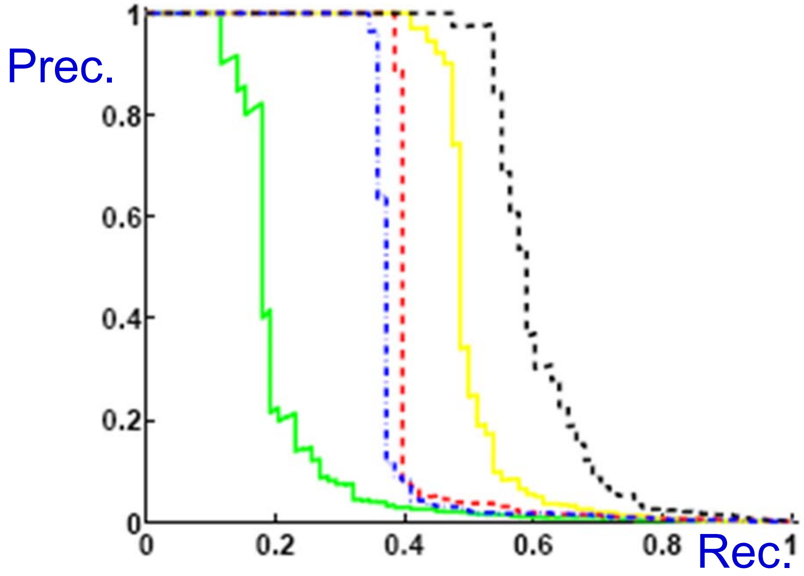
- Landmarks plus queries used for evaluation



- Ground truth obtained for 11 landmarks over 5062 images

- Evaluate performance by Precision - Recall curves

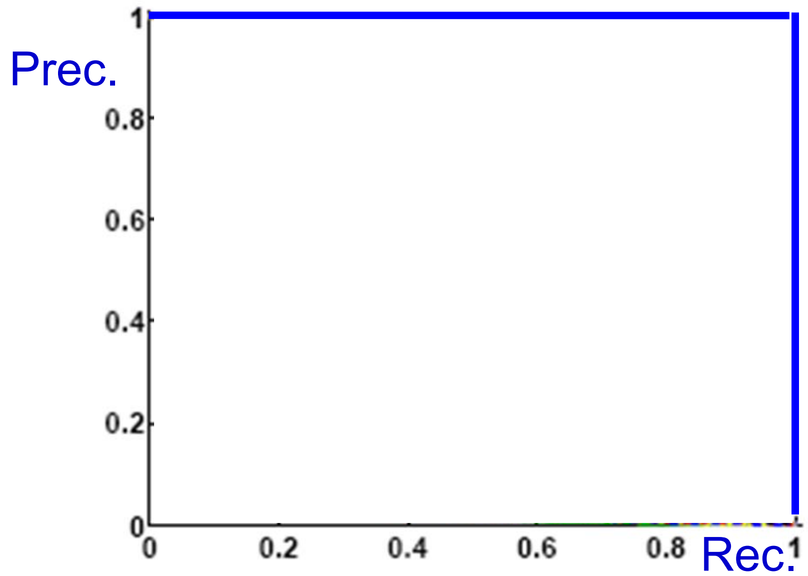
Query images



- high precision at low recall (like Google)
- variation in performance over query
- none retrieve all instances

Total Recall

Query images



Retrieve **all** occurrences of an object in the corpus

Improving SIFT

- Histogram measures such as Hellinger or χ^2, outperform Euclidean distance when comparing histograms (e.g. image classification, object category detection, texture classification etc).

- And these can be implemented efficiently using approximate feature maps in the case of additive kernels

- SIFT is a histogram: can performance be boosted using a better distance measure?

Hellinger distance

Hellinger kernel (Bhattacharyya's coefficient) for L1 normalized histograms x and y:

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

Distances and kernels x and y L2 normalized

$$\begin{aligned} d_{\mathsf{E}}(\mathbf{x},\mathbf{y})^2 &= \|\mathbf{x}-\mathbf{y}\|_2^2 \\ &= \|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2 - 2\mathbf{x}^\top\mathbf{y} \\ &= 2 - 2\underbrace{\sum_{i=1}^{n} x_i\, y_i}_{\text{kernel}} \end{aligned}$$

Hellinger distance

Hellinger kernel (Bhattacharyya's coefficient) for L1 normalized histograms x and y:

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

Distances and kernels x and y L1 normalized

$$\begin{aligned} d_{\mathsf{E}}(\sqrt{\mathbf{x}},\sqrt{\mathbf{y}})^2 &= \|\sqrt{\mathbf{x}}-\sqrt{\mathbf{y}}\|_2^2 \\ &= \|\sqrt{\mathbf{x}}\|_2^2 + \|\sqrt{\mathbf{y}}\|_2^2 - 2\sqrt{\mathbf{x}}^\top\sqrt{\mathbf{y}} \\ &= 2 - 2\underbrace{\sum_{i=1}^{n}\sqrt{x_i\, y_i}}_{\text{kernel}} \end{aligned}$$

Hellinger distance

Hellinger kernel (Bhattacharyya’s coefficient) for L1 normalized histograms x and y:

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

Explicit feature map of x into x’:

- L1 normalize x
- element-wise square root x to give x’

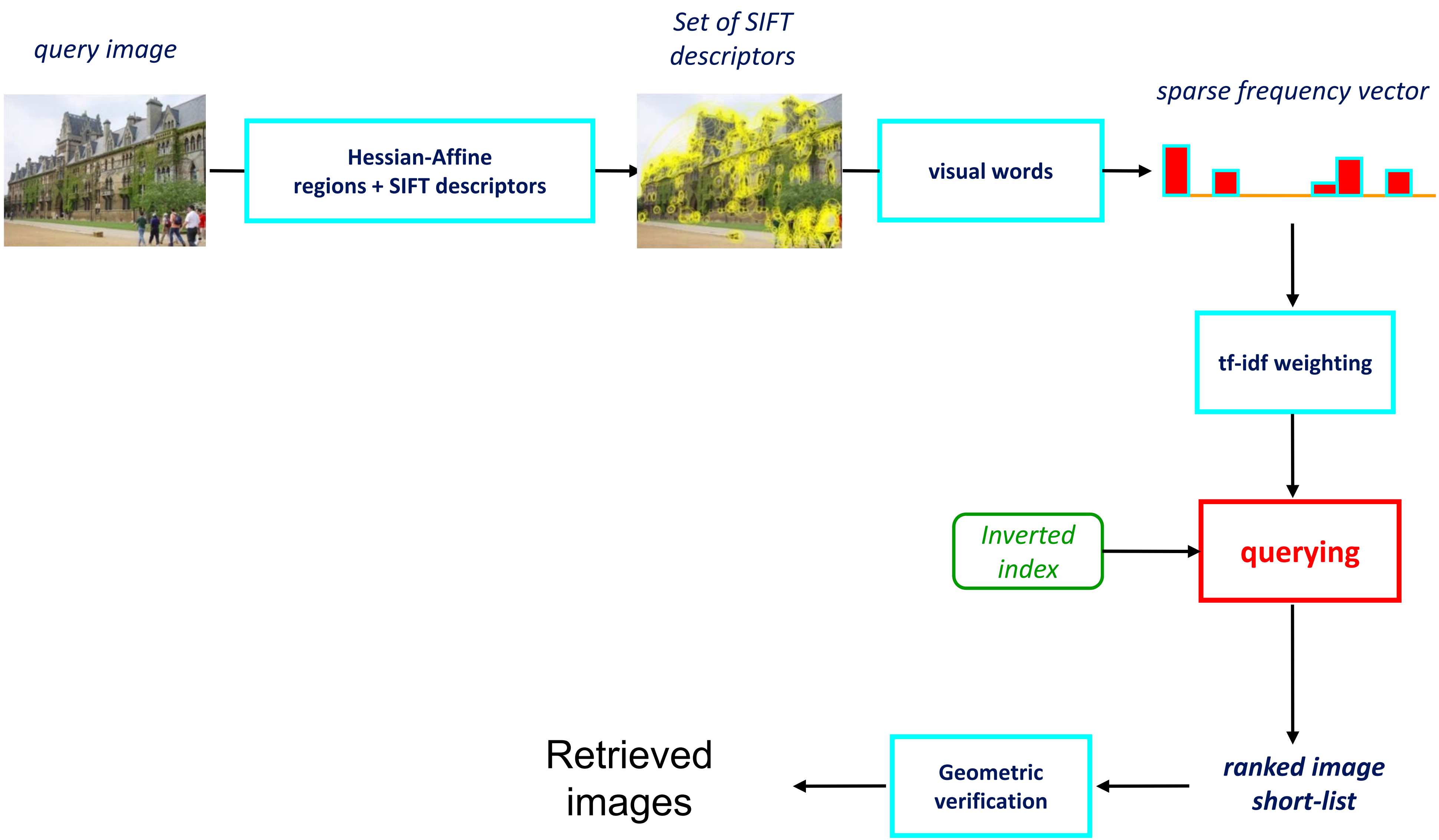
then x’ is L2 normalized

RootSIFT

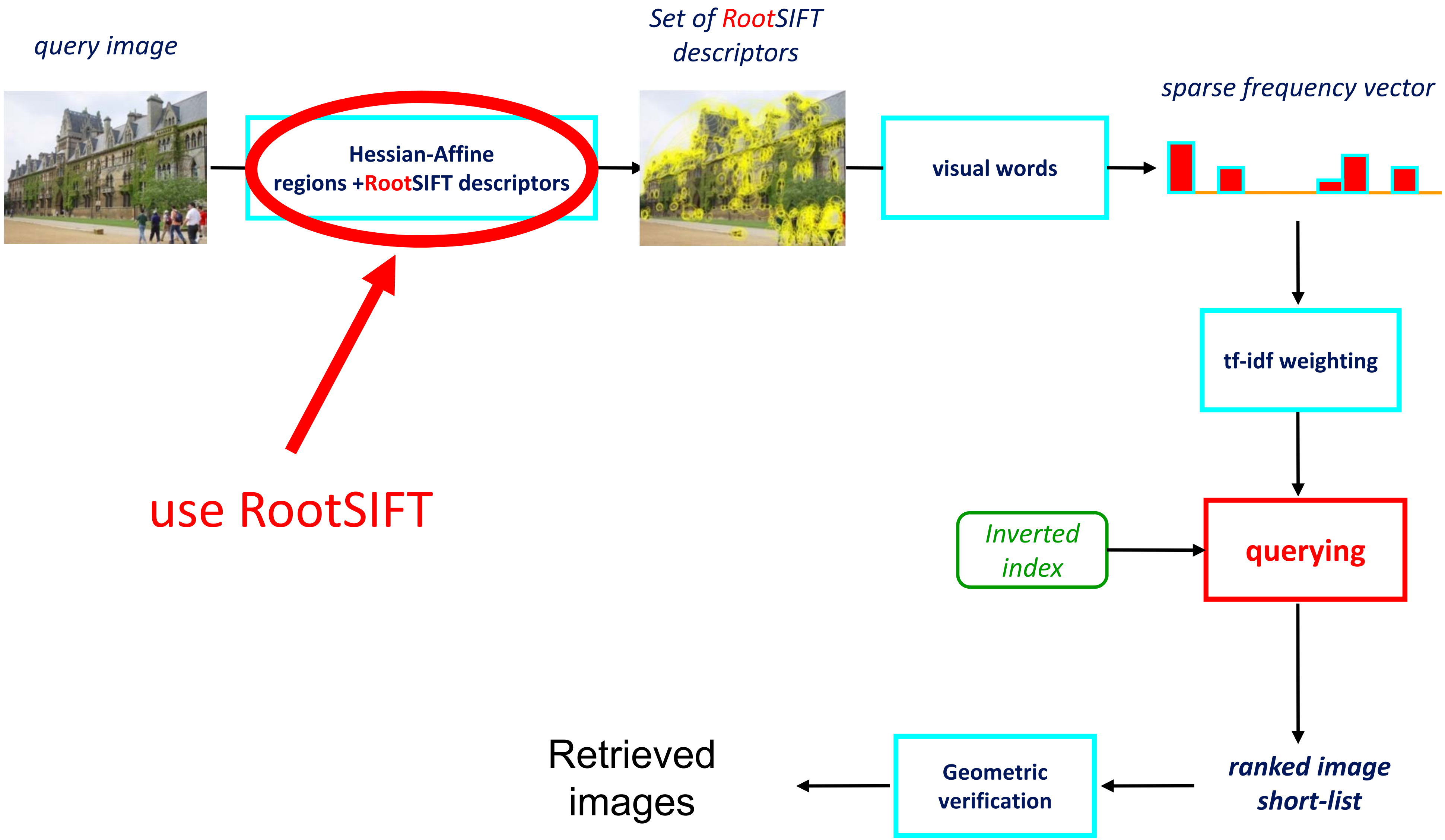
Euclidean distance in the feature map space is equivalent to Hellinger distance in the original space, since:

$$\mathbf{x}'^{\top}\mathbf{y}' = H(\mathbf{x},\mathbf{y})$$

Bag of visual words particular object retrieval



Bag of visual words particular object retrieval



RootSIFT: mAP performance

Philbin *et al*. 2007: bag of visual words either with

- tf-idf ranking,
- or tf-idf ranking and spatial reranking

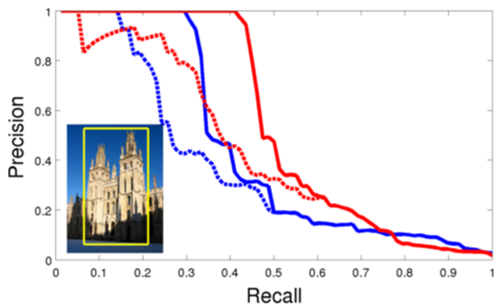
Evaluate on:

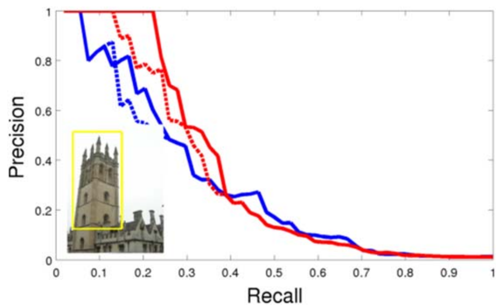
- Oxford 5k buildings,
- and on Oxford105k (5k buildings + 100k distractor images)

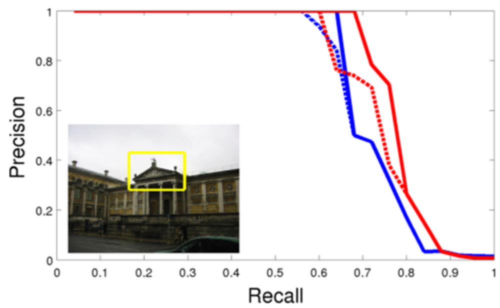
mean Average Precision (mAP)

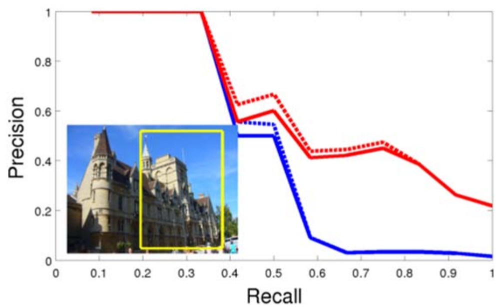
Retrieval method	Oxford 5k	Oxford 105k
SIFT: tf-idf ranking	0.636	0.515
SIFT: tf-idf with spatial reranking	0.672	0.581
RootSIFT: tf-idf ranking	0.683	0.581
RootSIFT: tf-idf with spatial reranking	**0.720**	**0.642**

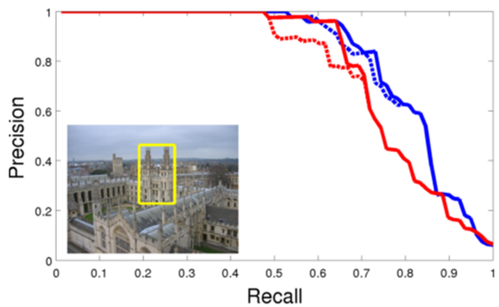
RootSIFT: results, Oxford 5k

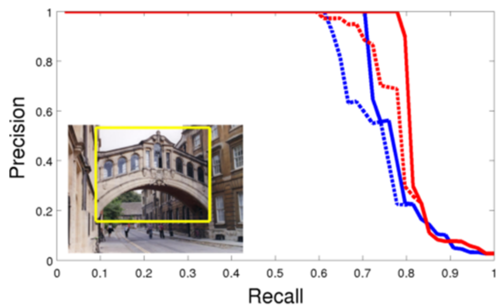


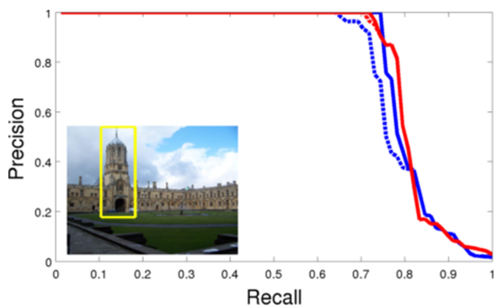


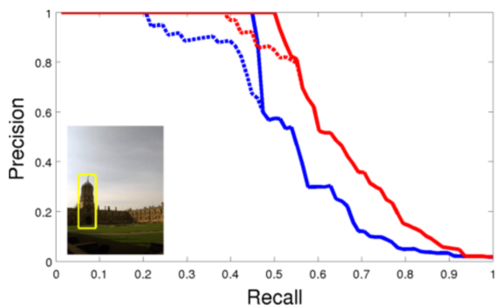


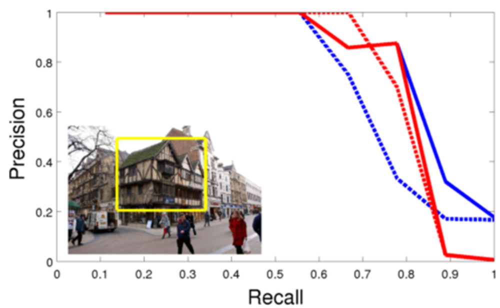


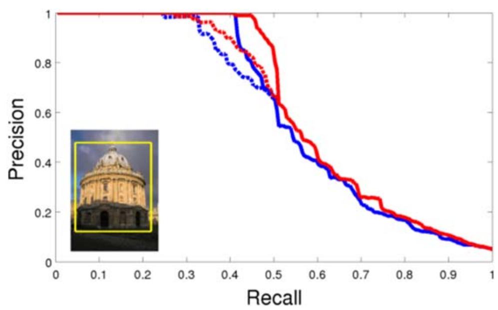


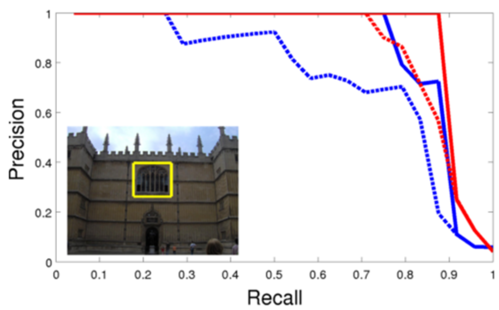










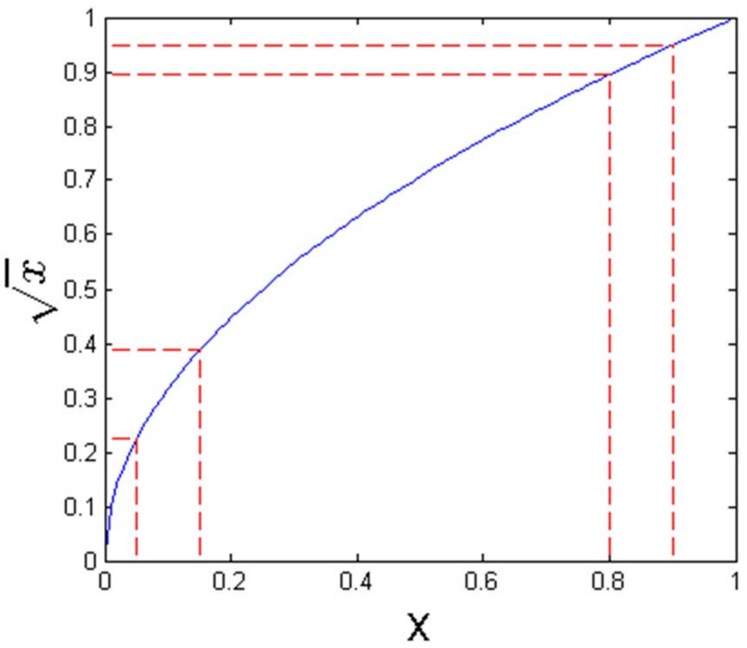


tfidf: dashed - -
spatial rerank: solid –
RootSIFT: red
SIFT: blue

Why does it work better?

Intuition: Euclidean distance can be dominated by large bin values. Hellinger distance is more sensitive to smaller bin values

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



RootSIFT Advantages

- Extremely simple to implement and use
  - one line of Matlab code to convert SIFT to RootSIFT:

```
rootsift= sqrt( sift / sum(sift) );
```

- Conversion from SIFT to RootSIFT can be done on-the-fly

- No need to re-compute stored SIFT descriptors for large image datasets

- Applications throughout computer vision

  k-means, approximate nearest neighbours, soft-assignment to visual words, Fisher vector coding, PCA, descriptor learning, hashing methods, product quantization etc.

There is a magic bullet

Other significant improvements …

Discriminative learning of descriptors, a better SIFT, e.g.

- Winder et al CVPR 09, Brown et al PAMI 2011, Philbin et al ECCV 10
- Convex learning of pooling regions and projection - Simonyan et al ECCV12

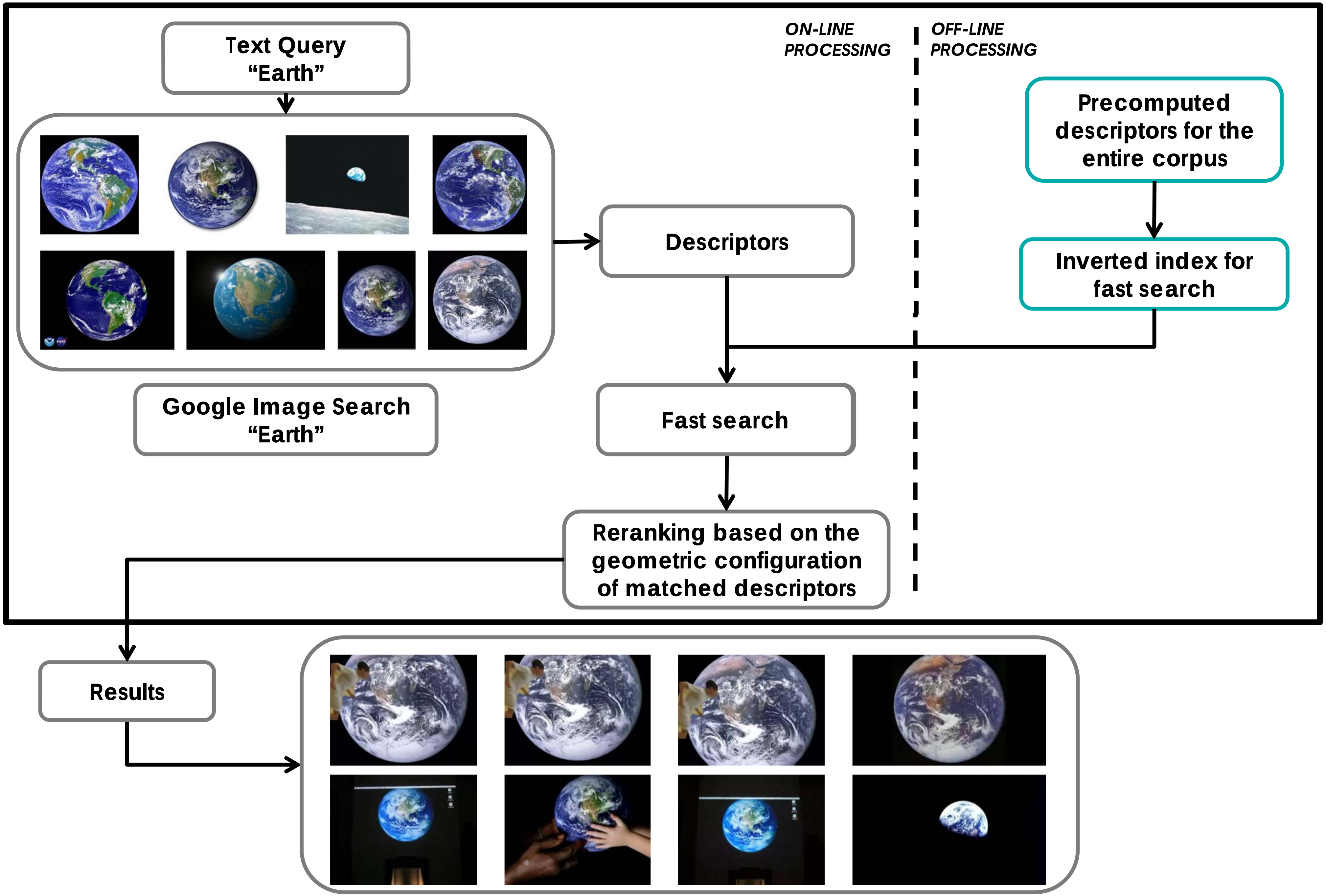
Closer representation of descriptor (reduce quantization errors), e.g.

- Philbin et al CVPR 08, Jegou et al ECCV 08, Mikulik et al ECCV 10
- Product Quantization on residuals – Jegou et al PAMI 2011

Query expansion, e.g.

- Chum et al ICCV 07, Turcot & Lowe ICCV 09 (workshop), Chum et al CVPR11
- Discriminative query expansion – Arandjelovic & Zisserman CVPR 2012

On-the-fly Instance Search



How are positive images used for instance search?

Compute a BOW feature vector x_i for each positive image

Possibilities:

- Average feature vectors x_i into q and query with q

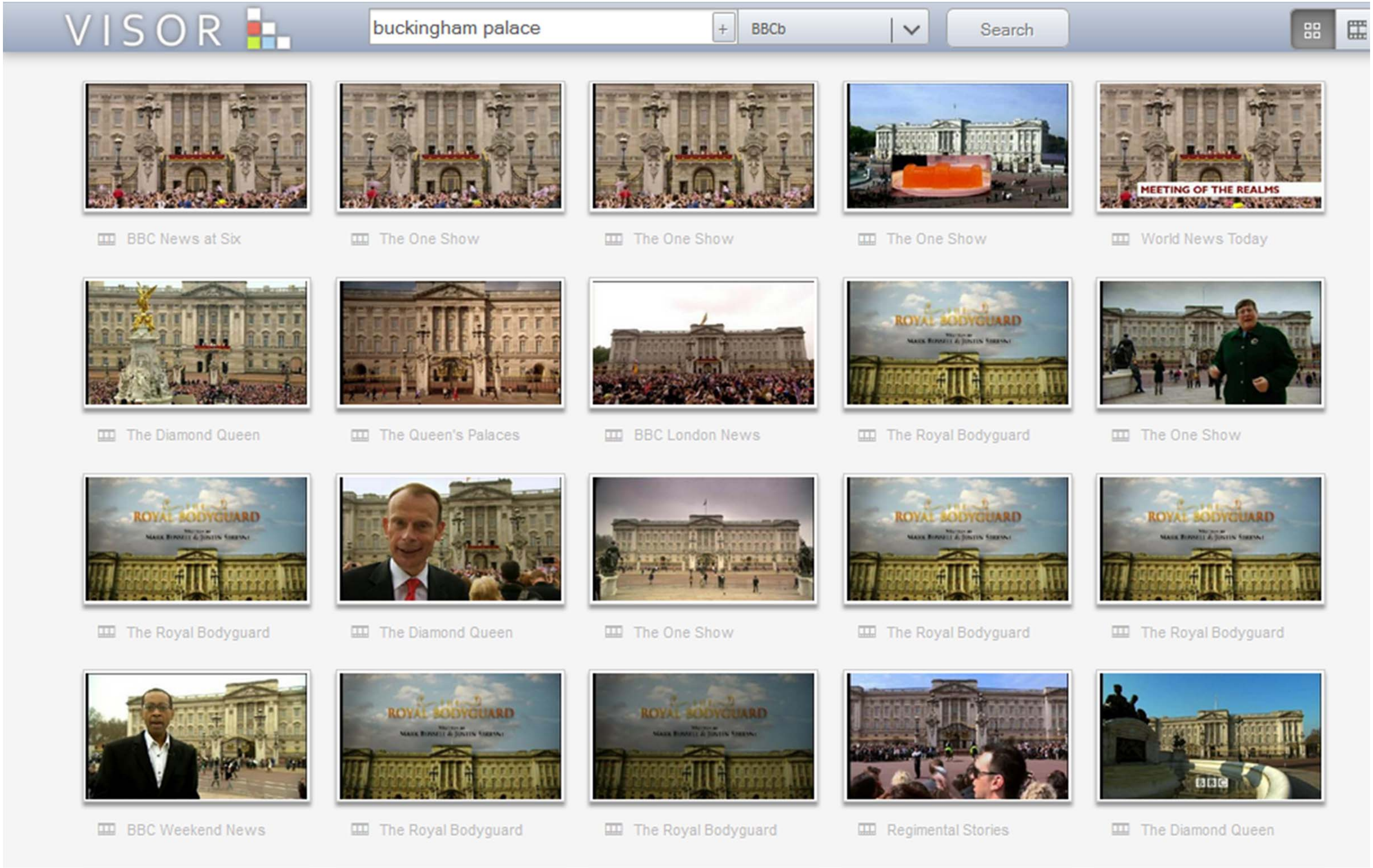
- Query with each feature vector x_i in turn and combine ranked results




Video dataset: BBC TV

- 4372 broadcasts from BBC 1, 2, 3 & 4
- Programmes from late 2011 to early 2012 from prime time slot (7pm-12pm) over five months
- 3007 hours of video represented by 1 frame per second
- 11M seconds of data, 3M keyframes
- Frames are 480 x 270 pixels

Instance Search – Example `Buckingham Palace'



2. On-the-fly Category Search

Image classification

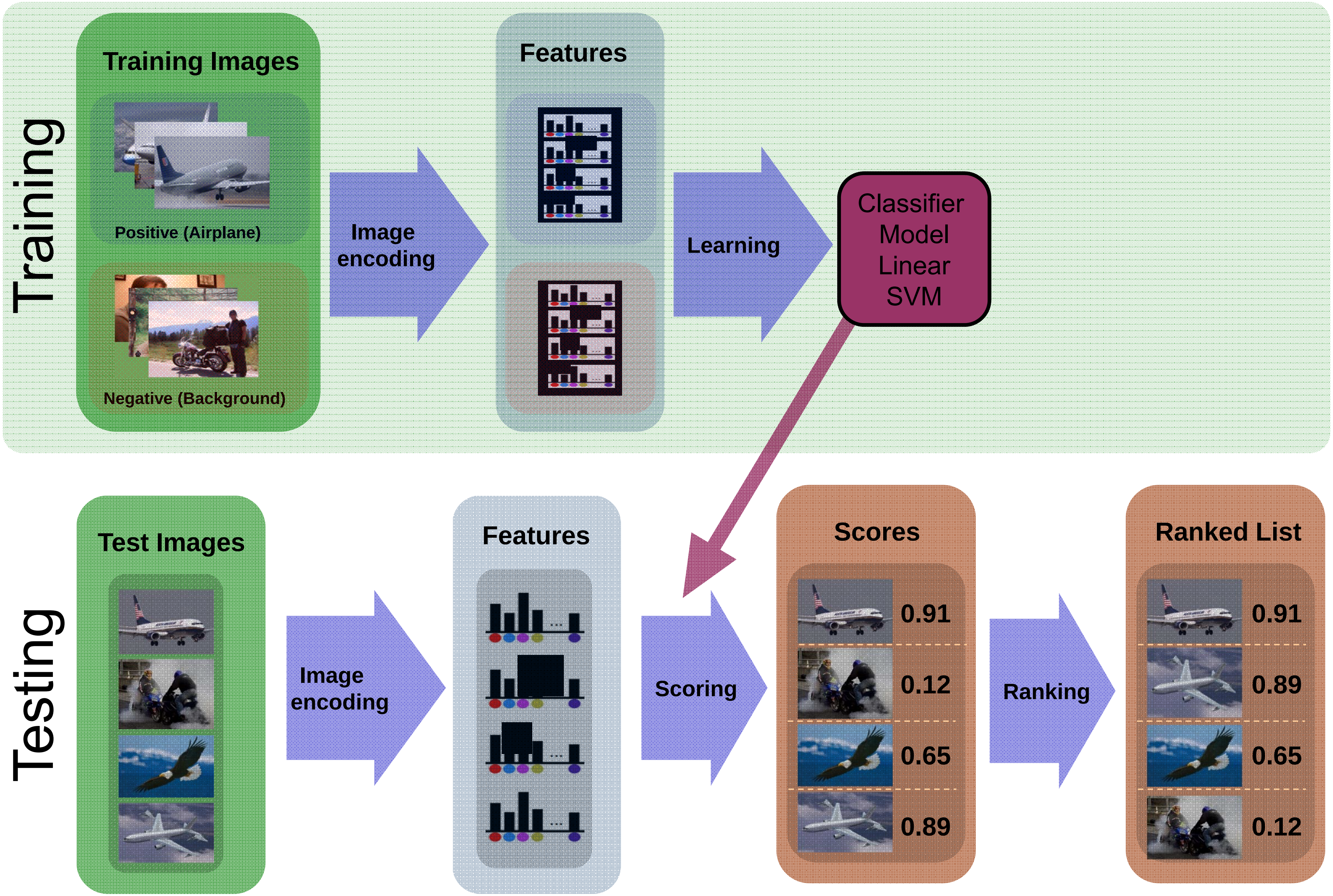


Image classification

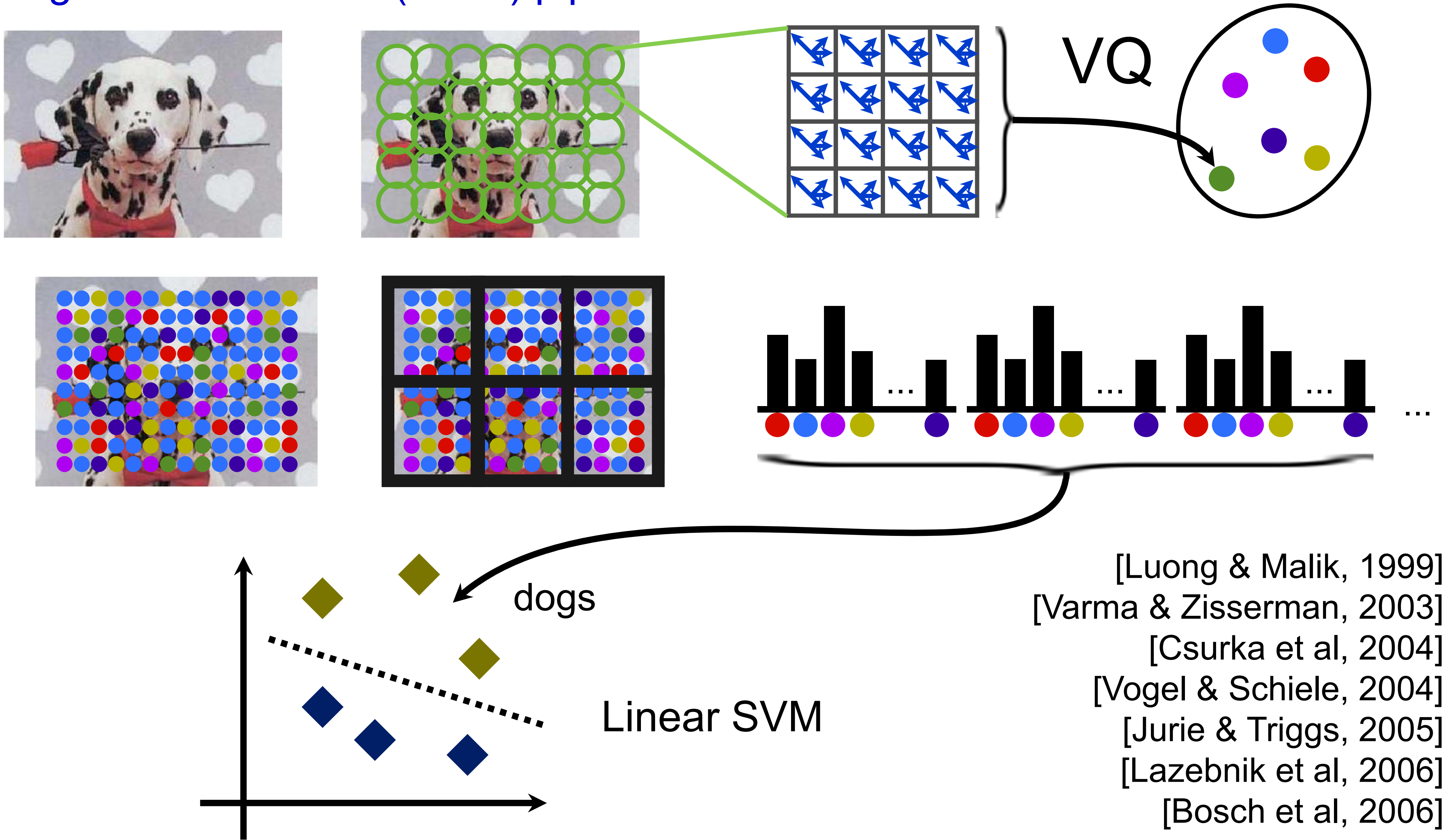
Classify an image by the objects/scenes it contains

- Review recent progress in encoding methods
- Choice of encoding method – trade off:
  - memory footprint
  - speed
  - performance

Image Encoding

Dense SIFT features

- Bag of Visual Words (BOW) pipeline



[Luong & Malik, 1999]
[Varma & Zisserman, 2003]
[Csurka et al, 2004]
[Vogel & Schiele, 2004]
[Jurie & Triggs, 2005]
[Lazebnik et al, 2006]
[Bosch et al, 2006]

Evolution of encodings …

Soft and sparse assignments, e.g.

- Philbin et al CVPR 08, Gemert et al ECCV 08,
- Locality-constrained linear coding (LLC) – Wang et al CVPR 10

Representing SIFT distribution mean in voronoi cell, e.g.

- super-vector coding – Zhou et al ECCV 10
- VLAD – Jegou et al CVPR 10

Representing SIFT distribution mean and covariance in voronoi cell, e.g.

- Fisher vector – Perronnin et al CVPR 07 & 10, ECCV 10

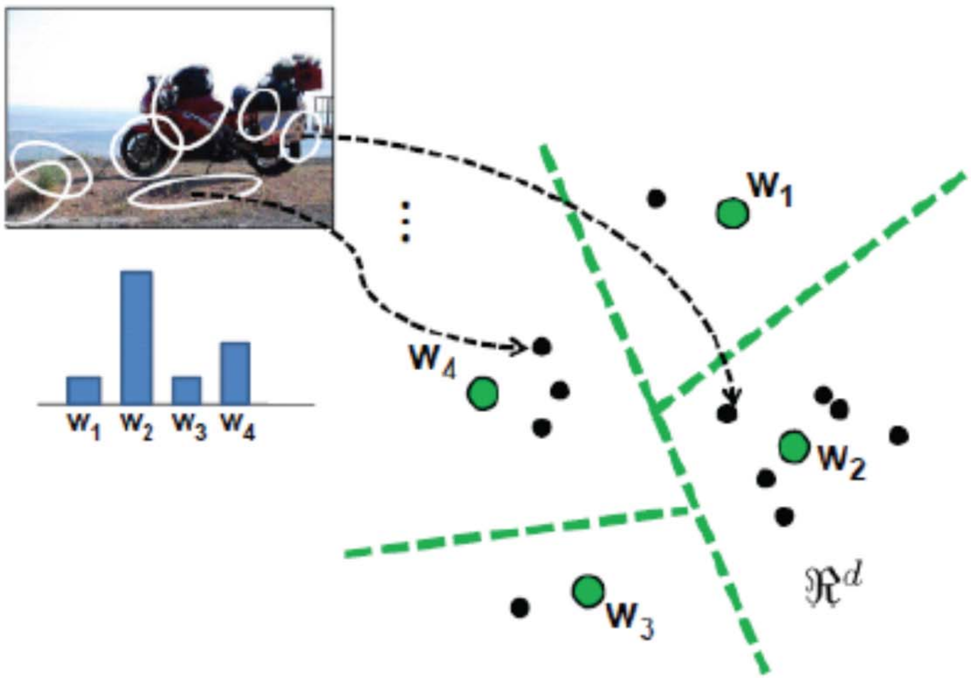
Improvements to normalization, PCA, whitening for VLAD/FV

- Chen et al 2011, Jegou & Chum ECCV 12
- All about VLAD – Arandjelovic & Zisserman CVPR 13

Comparison & code: “The devil is in the details”,Chatfield et al, BMVC11

Encoding the Descriptor Distribution

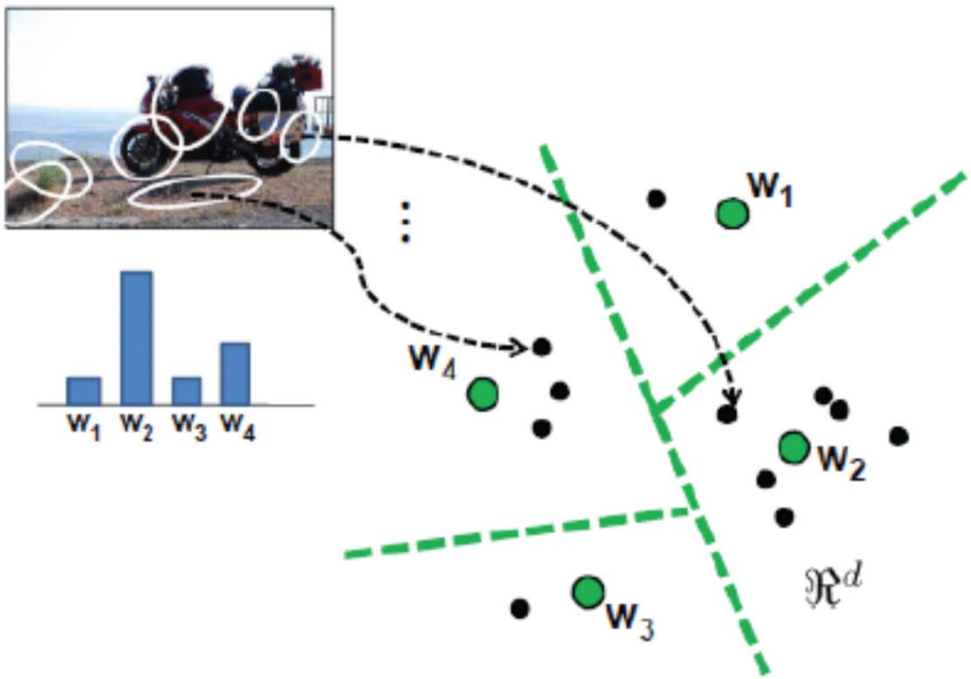
• BOW only **counts** the number of SIFT descriptors assigned to each Voronoi cell



http://www.cs.utexas.edu/~grauman/courses/fall2009/papers/bag_of_visual_words.pdf

Encoding the Descriptor Distribution

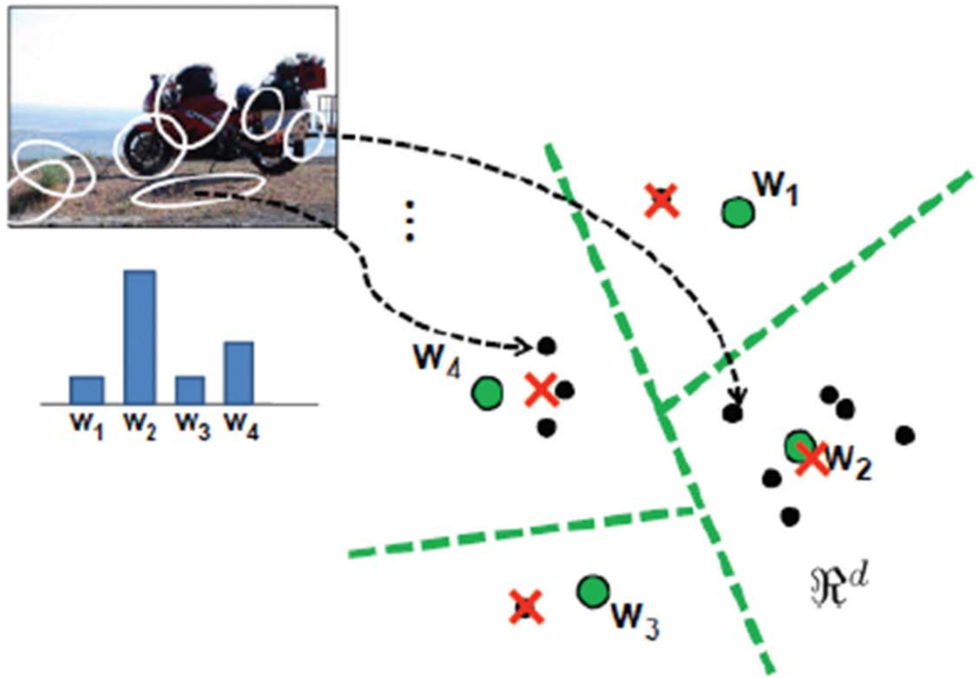
- BOW only **counts** the number of SIFT descriptors assigned to each Voronoi cell
- Why not include **other statistics**? For instance



Slide from Perronnin & Jegou

Encoding the Descriptor Distribution

- BOW only **counts** the number of SIFT descriptors assigned to each Voronoi cell
- Why not include **other statistics**? For instance
  - mean of descriptors ×

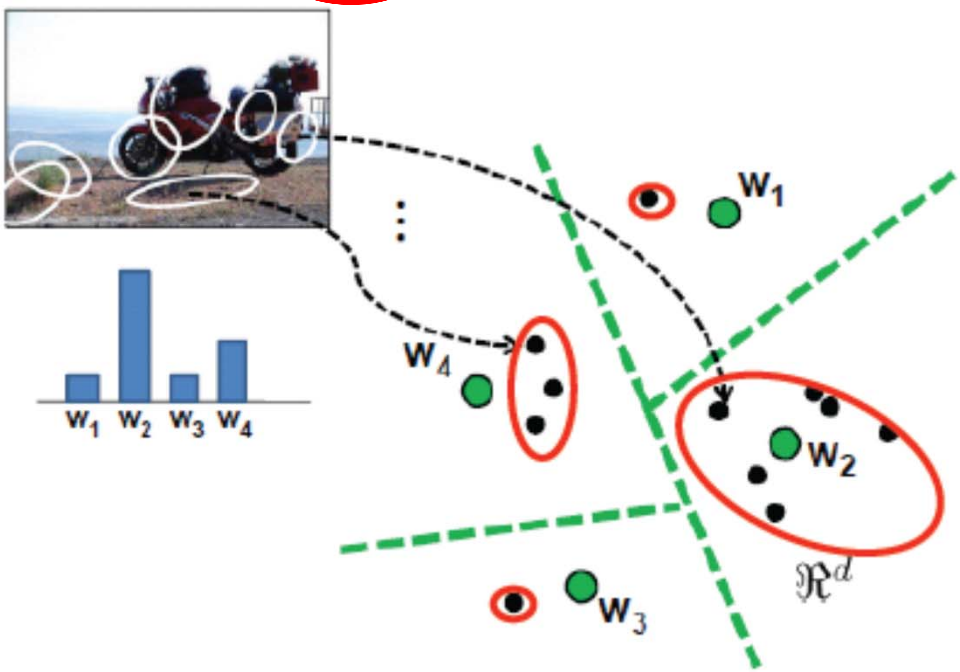


VLAD & Super-vector

Slide from Perronnin & Jegou

Encoding the Descriptor Distribution

- BOW only **counts** the number of SIFT descriptors assigned to each Voronoi cell
- Why not include **other statistics**? For instance
  - mean of descriptors
  - (co)variance of descriptors



Fisher vector

Slide from Perronnin & Jegou

VLAD – Encoding

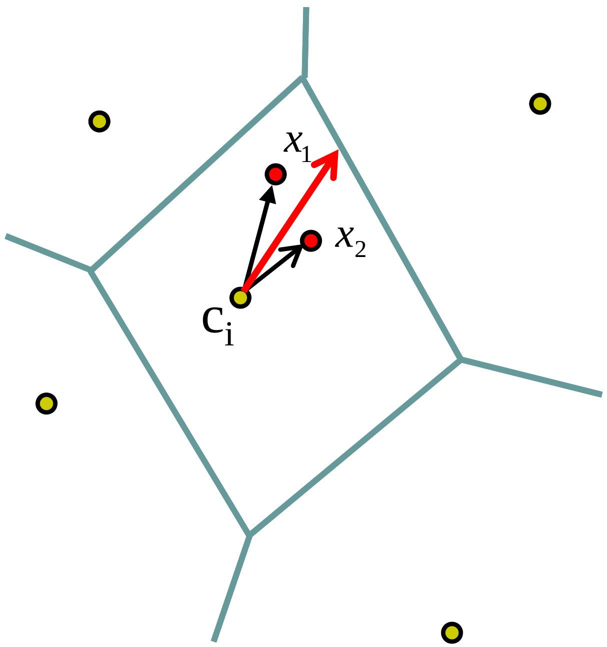
• VLAD : vector of locally aggregated descriptors

- Learn a vector quantizer (*k*-means): $c_1, \ldots, c_i, \ldots c_k$, with c_i centroid of dim. *d*

- For a given image
  - assign each SIFT descriptor to closest center c_i
  - accumulate (sum) descriptors per cell

    $v_i := v_i + (x_j - c_i)$

    measure residual of vectors within a cell



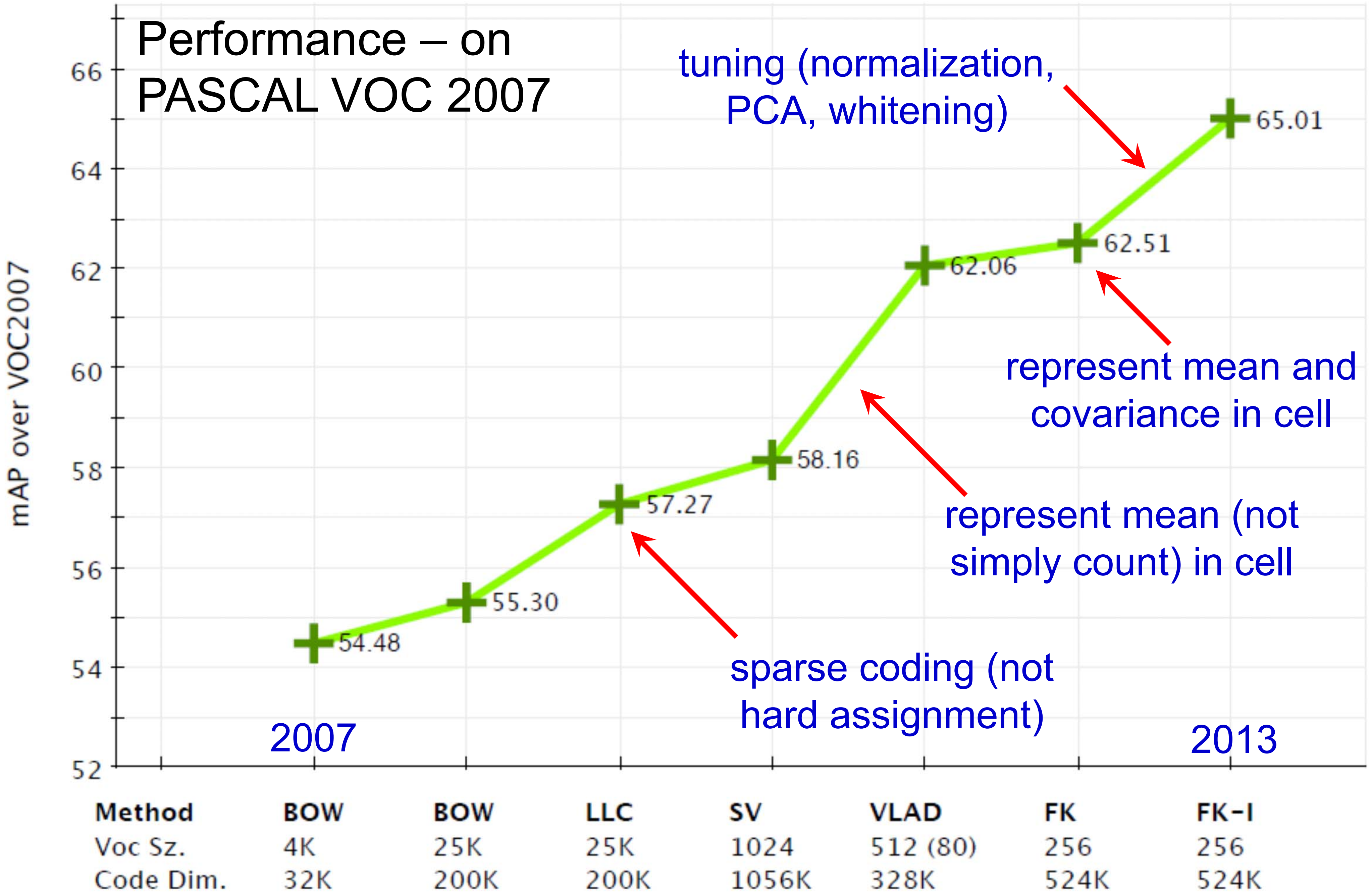
- VLAD of dimension $D = k \times d$

  (k typically between 16 and 512, d = 128 or less)

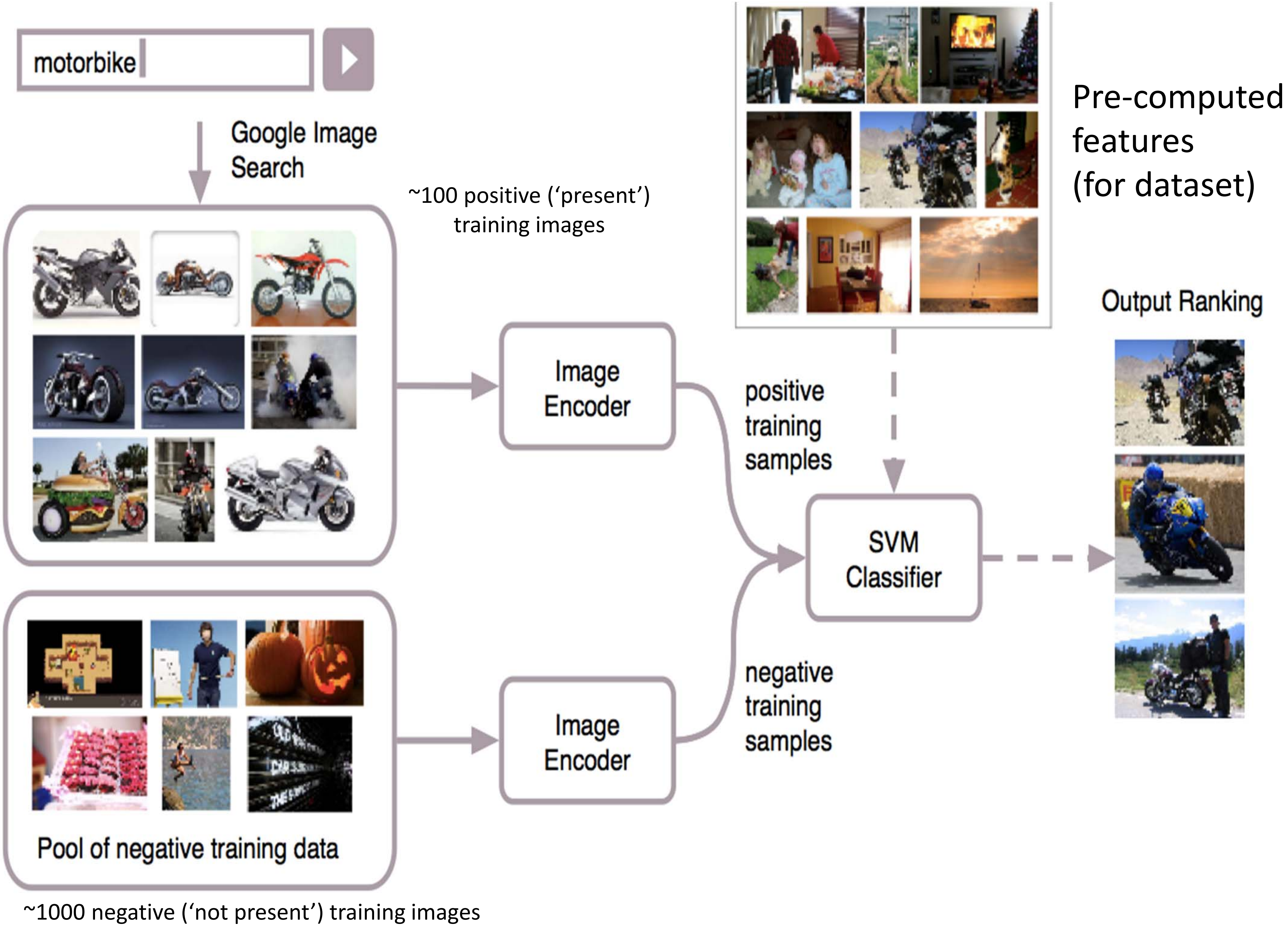
- The vector is square-root + L2-normalized

Jegou, Douze, Schmid, Perez, CVPR'10

Evolution of Encoding Methods



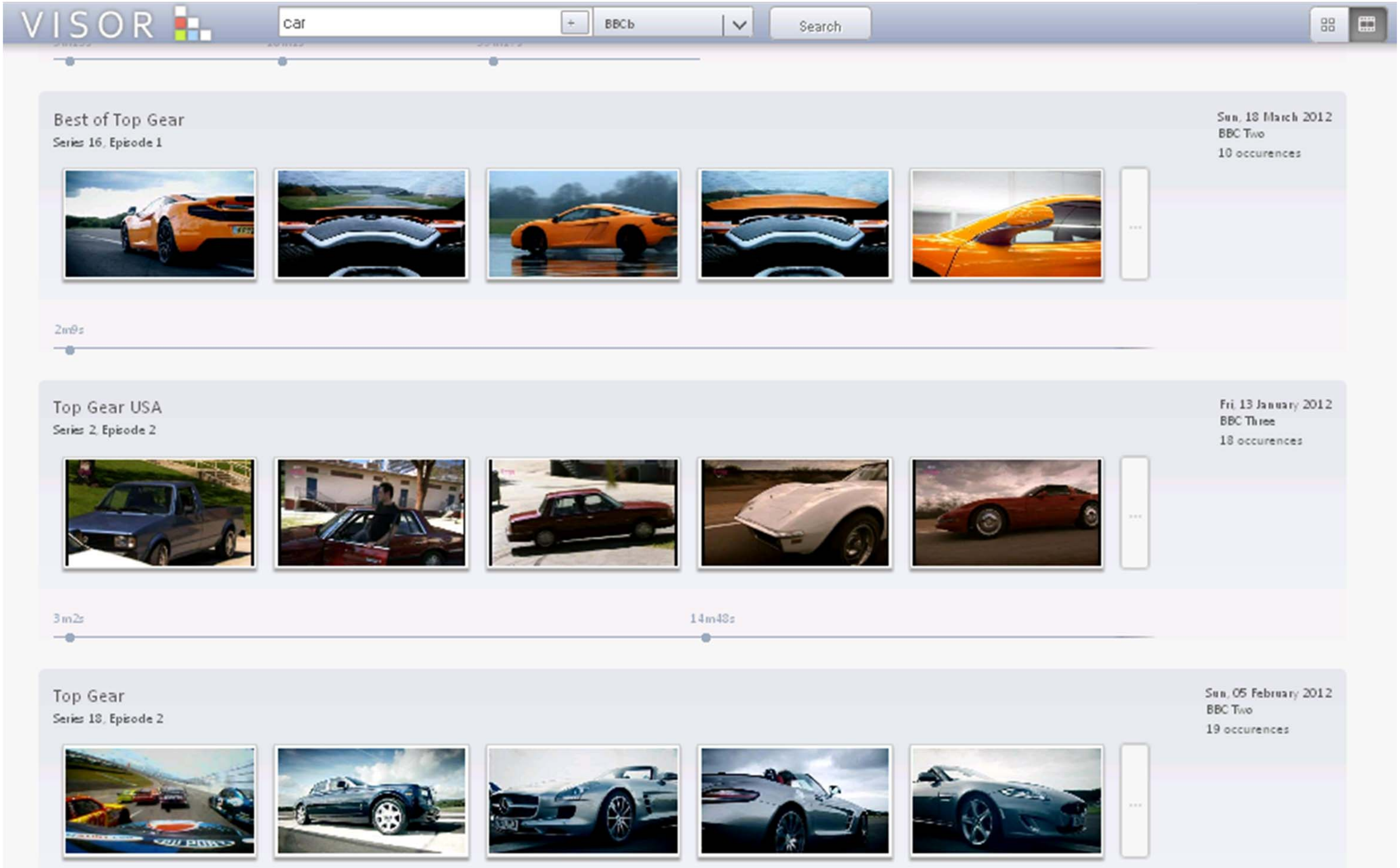
On-the-fly Visual Category Search



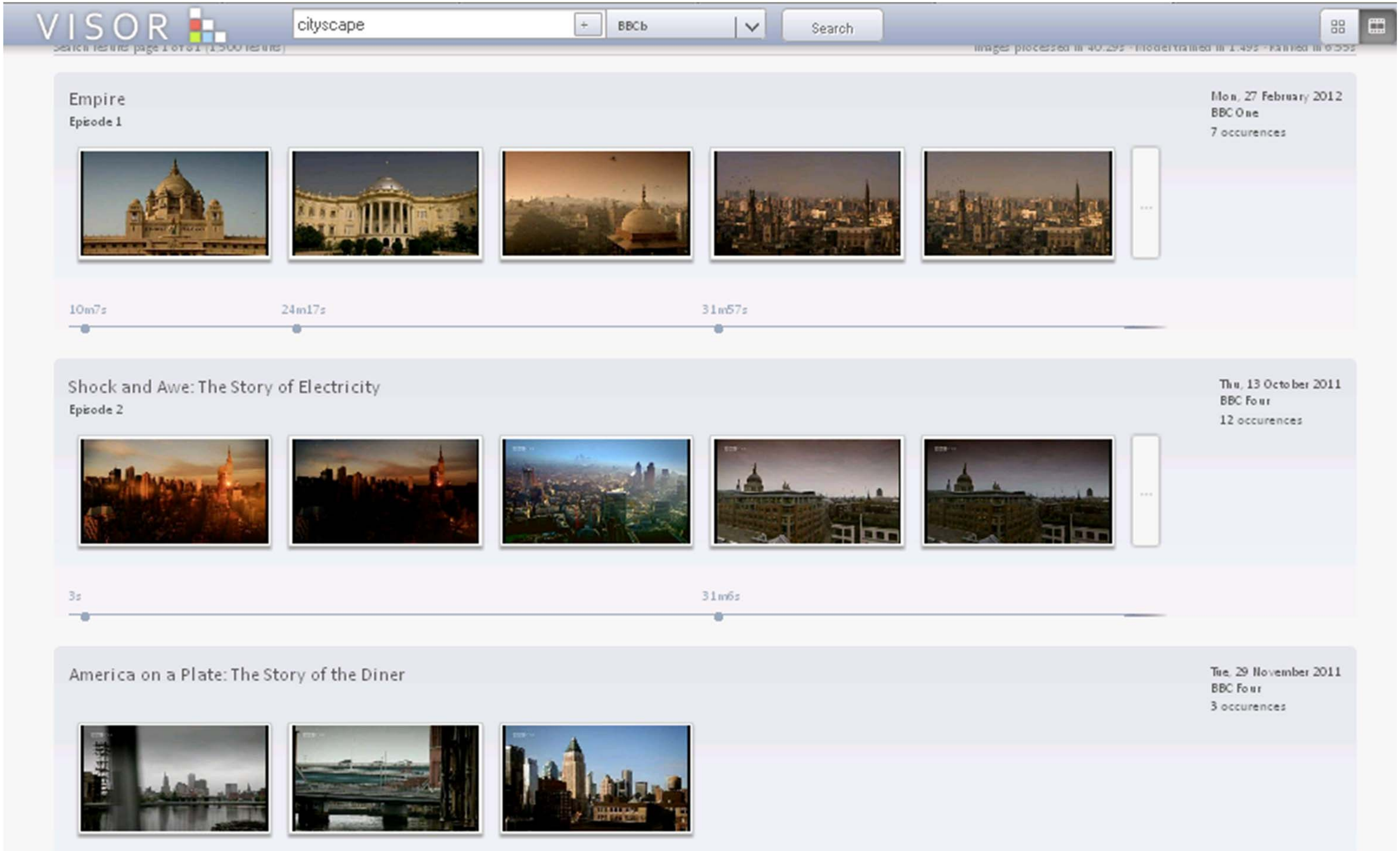
Video dataset: BBC TV

- 4372 broadcasts from BBC 1, 2, 3 & 4
- Programmes from late 2011 to early 2012 from prime time slot (7pm-12pm) over five months
- 3007 hours of video represented by 1 frame per second
- 11M seconds of data, 3M keyframes
- Frames are 480 x 270 pixels


Visual Category Search – Examples `Car'



Visual Category Search – Examples `Cityscape’



VLAD Data Stats

3 Million key frames

Total size of original descriptors: 328k x 4 x 3M = 3936 GB

Dimensionality reduction 328k -> 8k using PCA
(mAP 62.06 -> 60.30)

- Memory footprint: 8k x 4 x 3M = 96 GB

Product Quantization: 8k x 4 -> 2k

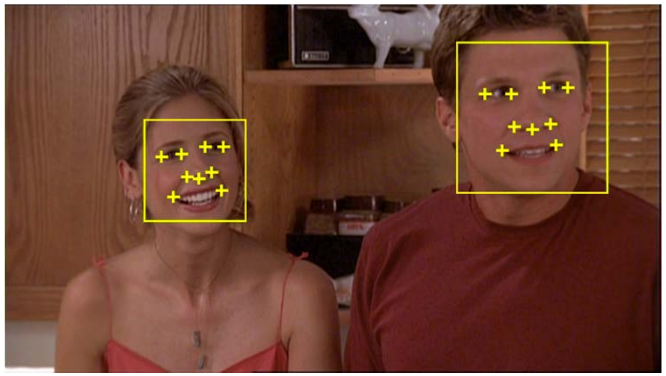
- Memory footprint: 2k x 3M = 6 GB

Product Quantization for vector compression,
Jegou *et al.*, PAMI 2011

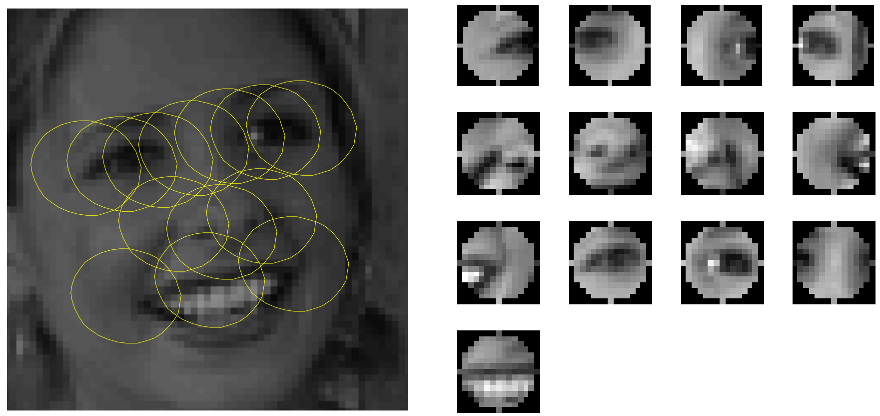
3. On-the-fly Face Search

Feature vectors for face (tracks)

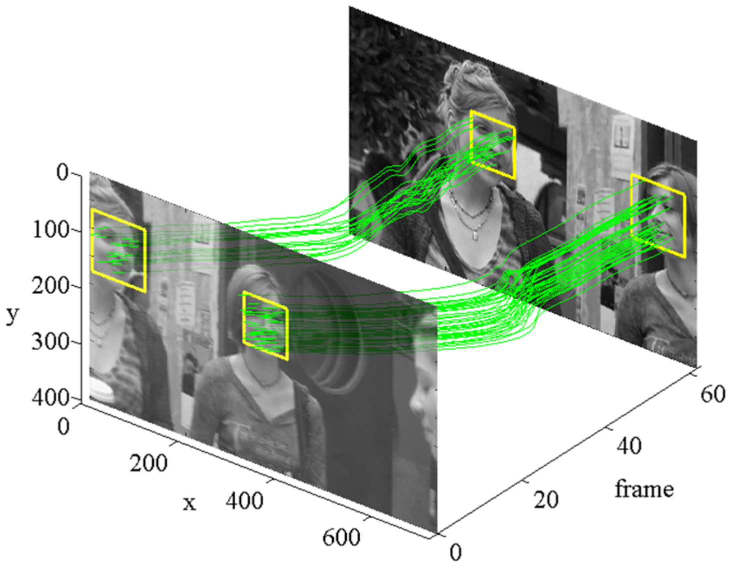
Face detection and facial landmark detection



Feature region descriptors



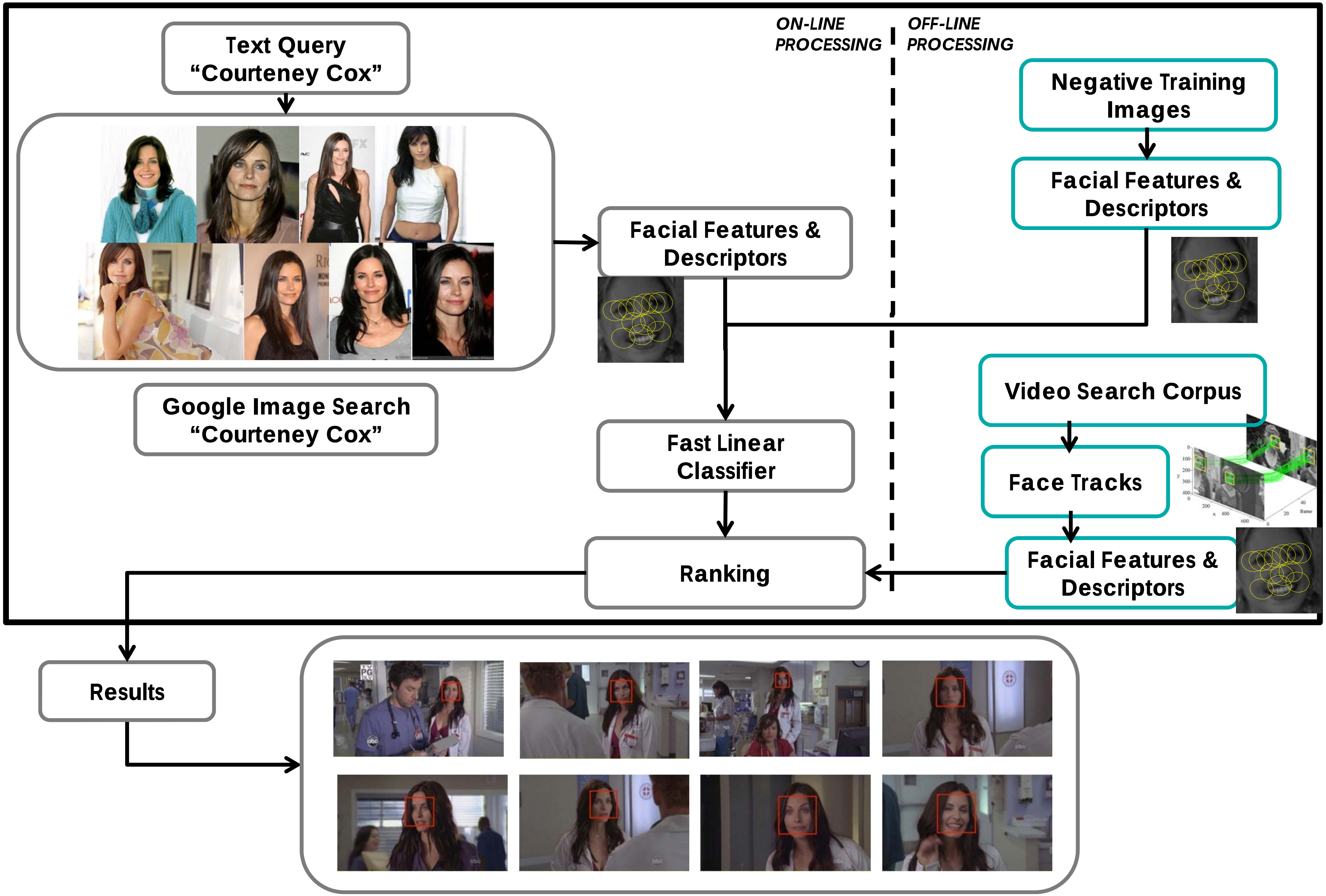
Faces clustered into tracks



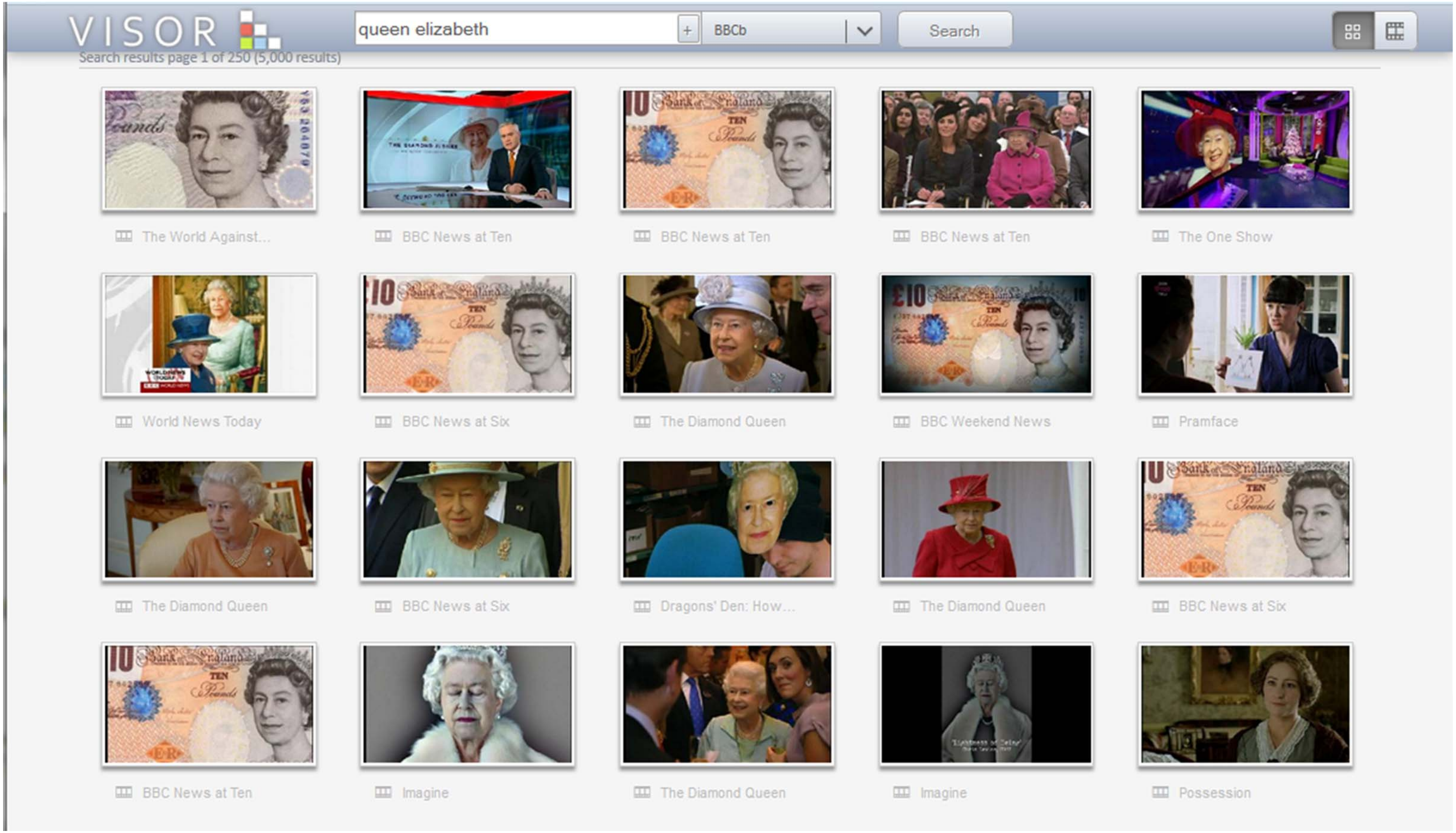
Concatenation

Feature Vector

On-the-fly Person Retrieval



Face Search – Examples `Queen Elizabeth’



Video dataset: BBC TV

- 4372 broadcasts from BBC 1, 2, 3 & 4
- Programmes from late 2011 to early 2012 from prime time slot (7pm-12pm) over five months
- 3007 hours of video represented by 1 frame per second
- 11M seconds of data, 3M keyframes
- Frames are 480 x 270 pixels



Face Data Stats

- 3007 hours of video, 3 M shots
- 0.68 M shots have faces
- 0.8 M face tracks
- Total size of original descriptors: 4k x 4 x 0.8M = 12.8 GB
- Memory footprint (after PQ): 1k x 0.8M = 0.8 GB

- NB no need for PCA dimensionality reduction here

Facial attributes – FaceTracer project

Examples:

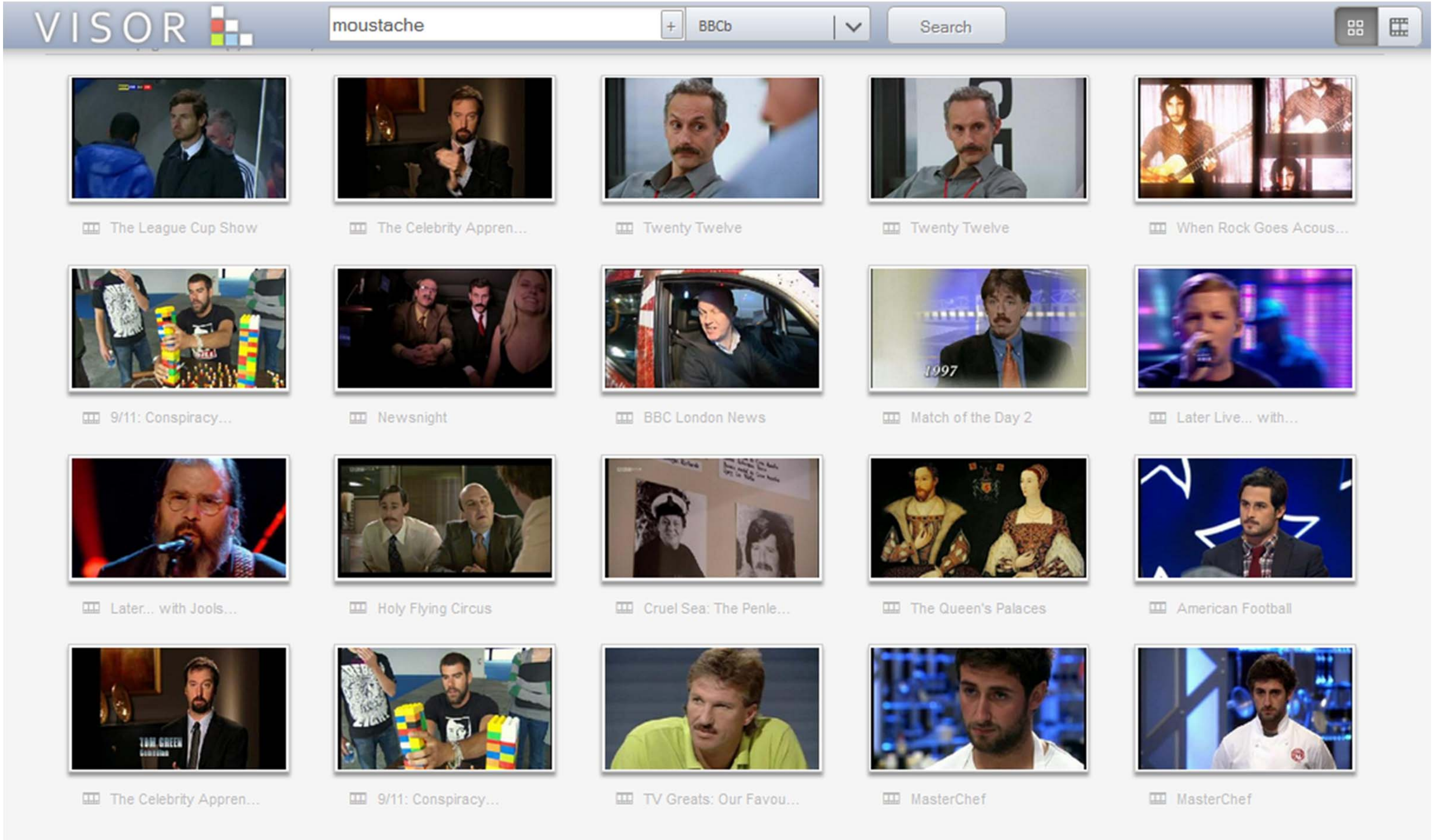
- gender: male, female
- age: baby, child, youth, middle age, senior
- race: white, black, asian
- smiling, mustache, eye-wear, hair colour

Method

- **person independent** training set with attribute
- facial feature representation
- discriminative training of classifier for attribute

N. Kumar, P. N. Belhumeur and S. K. Nayar,
FaceTracer: A Search Engine for Large Collections of Images with Faces, *ECCV 2010*

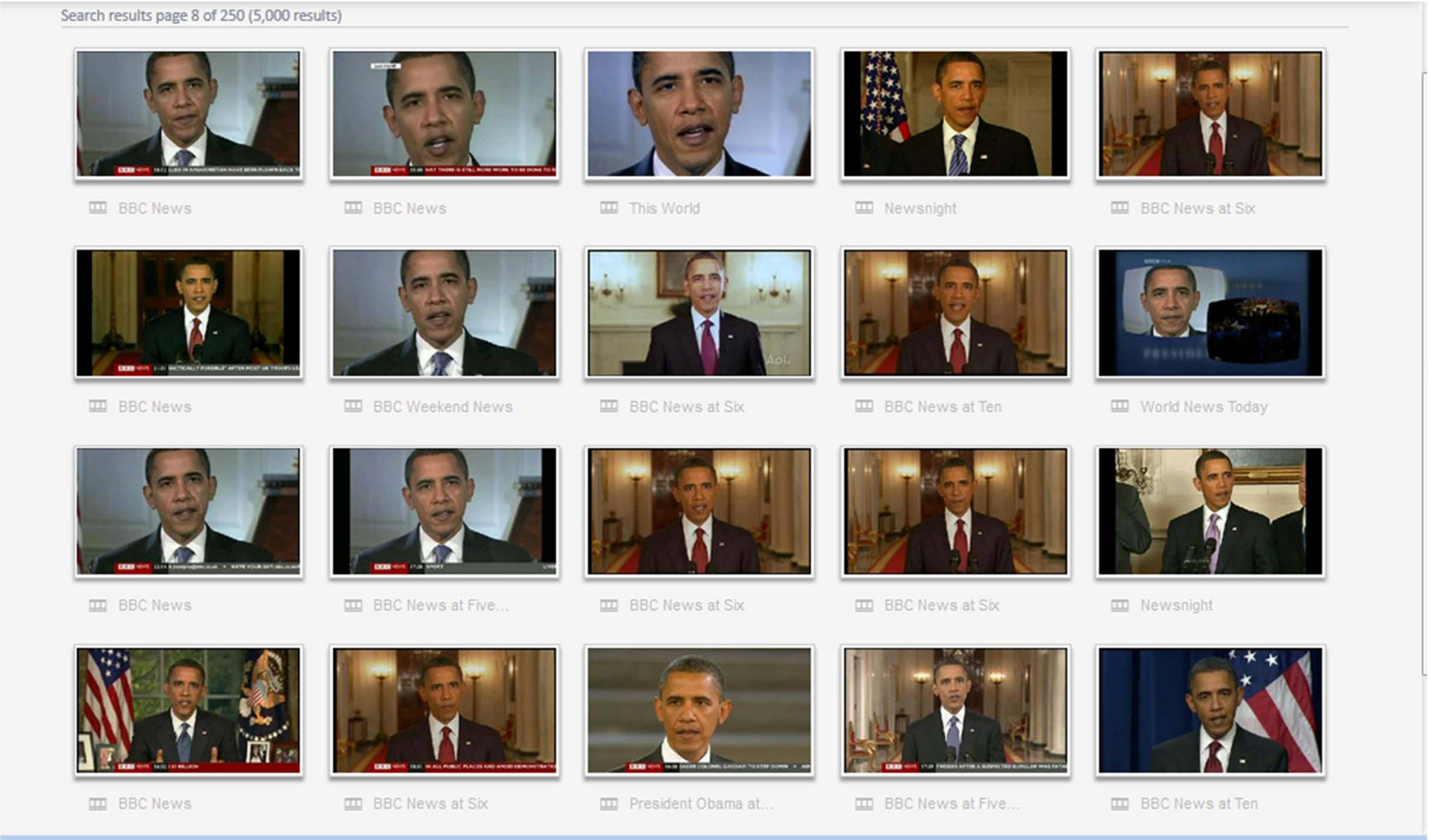
Face Search – Examples `Moustache'



Datasets

Description	BBC 1, 2, 3 & 4 prime time 5 months	BBC 1, 2, 3 & 4, Parliament & News 24 4 years
# broadcasts	4,372	56,078
video / hrs	3,007	39,289
# frames (1 per second)	11 M	141 M
# key frames (1 per shot)	3 M	34.6 M

Face Search on 40 k hrs – Example `Obama'



How can performance be improved?

- Better face descriptor encoding
- See paper by Karen Simonyan *et al.* “Fisher Vector Faces in the Wild”, BMVC 2013

The Vision

All visual material (images, video) should be searchable for anything

- people, object categories, scene categories, particular objects, human actions and interactions, activities …

and retrieved with high precision and high recall








On-the-fly papers

R. Arandjelović, A. Zisserman
Multiple queries for large scale specific object retrieval
British Machine Vision Conference, 2012

K. Chatfield, A. Zisserman
VISOR: Towards On-the-Fly Large-Scale Object Category Retrieval
Asian Conference on Computer Vision, 2012

O. M. Parkhi, A. Vedaldi, A. Zisserman
On-the-fly Specific Person Retrieval
International Workshop on Image Analysis for Multimedia Interactive Services, 2012