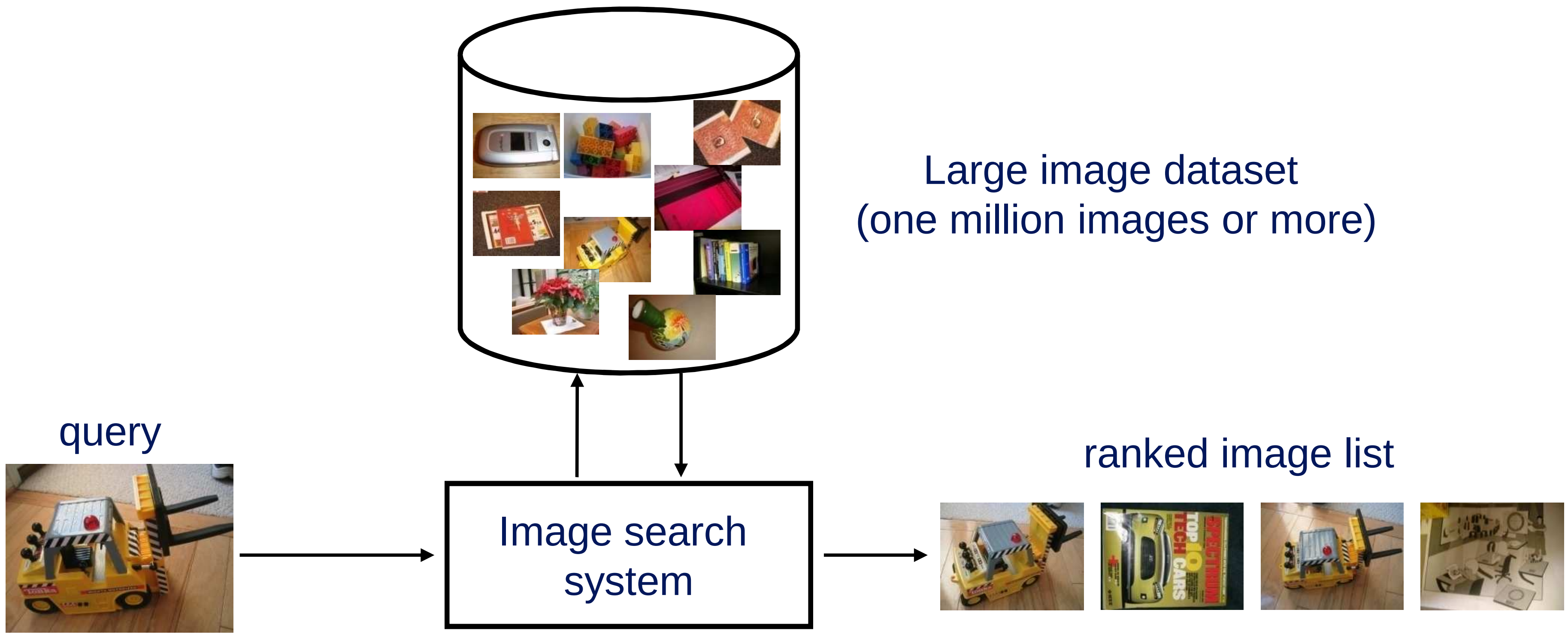
Overview

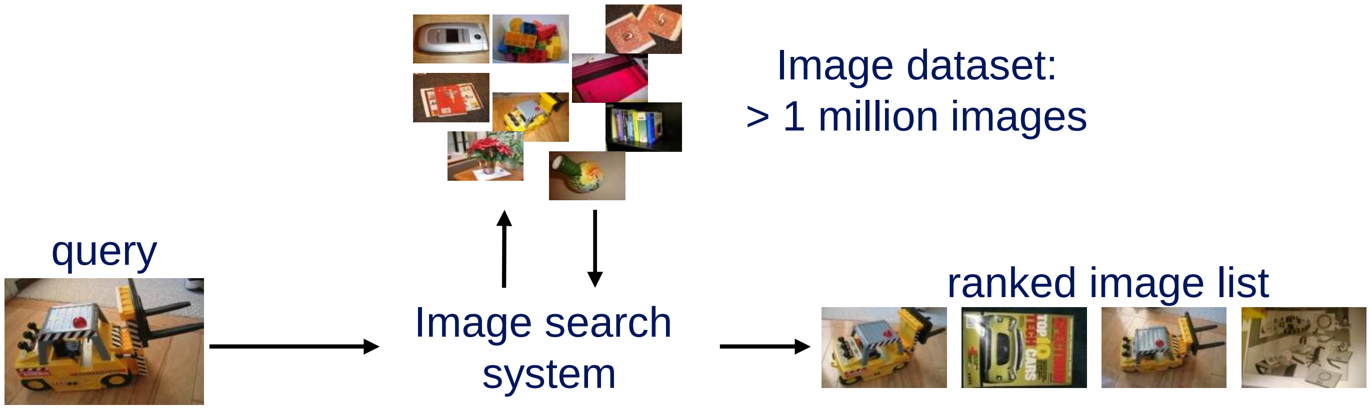
- Local invariant features (C. Schmid)
- Matching and recognition with local features (J. Sivic)
- Efficient visual search (J. Sivic)
- **Very large scale search** (C. Schmid)
- Practical session

Image search system for large datasets



- **Issues** for very large databases
  - to reduce the query time
  - to reduce the storage requirements
  - with minimal loss in retrieval accuracy

Large scale object/scene recognition



- Each image described by approximately 2000 descriptors
  - $2 * 10^9$ descriptors to index for one million images!

- Database representation in RAM:
  - Size of descriptors : 1 TB, search+memory intractable

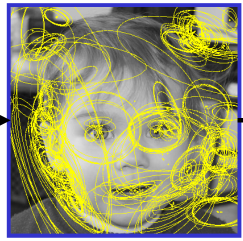
Bag-of-words [Sivic & Zisserman'03]

Query image

Hessian-Affine regions + SIFT descriptors

[Mikolajezyk & Schmid 04]
[Lowe 04]

Set of SIFT descriptors

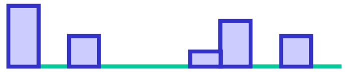


centroids (visual words)

Bag-of-features processing +*tf-idf* weighting

[Nister & al 04, Chum & al 07]

sparse frequency vector



Inverted file

querying

ranked image short-list

Geometric verification

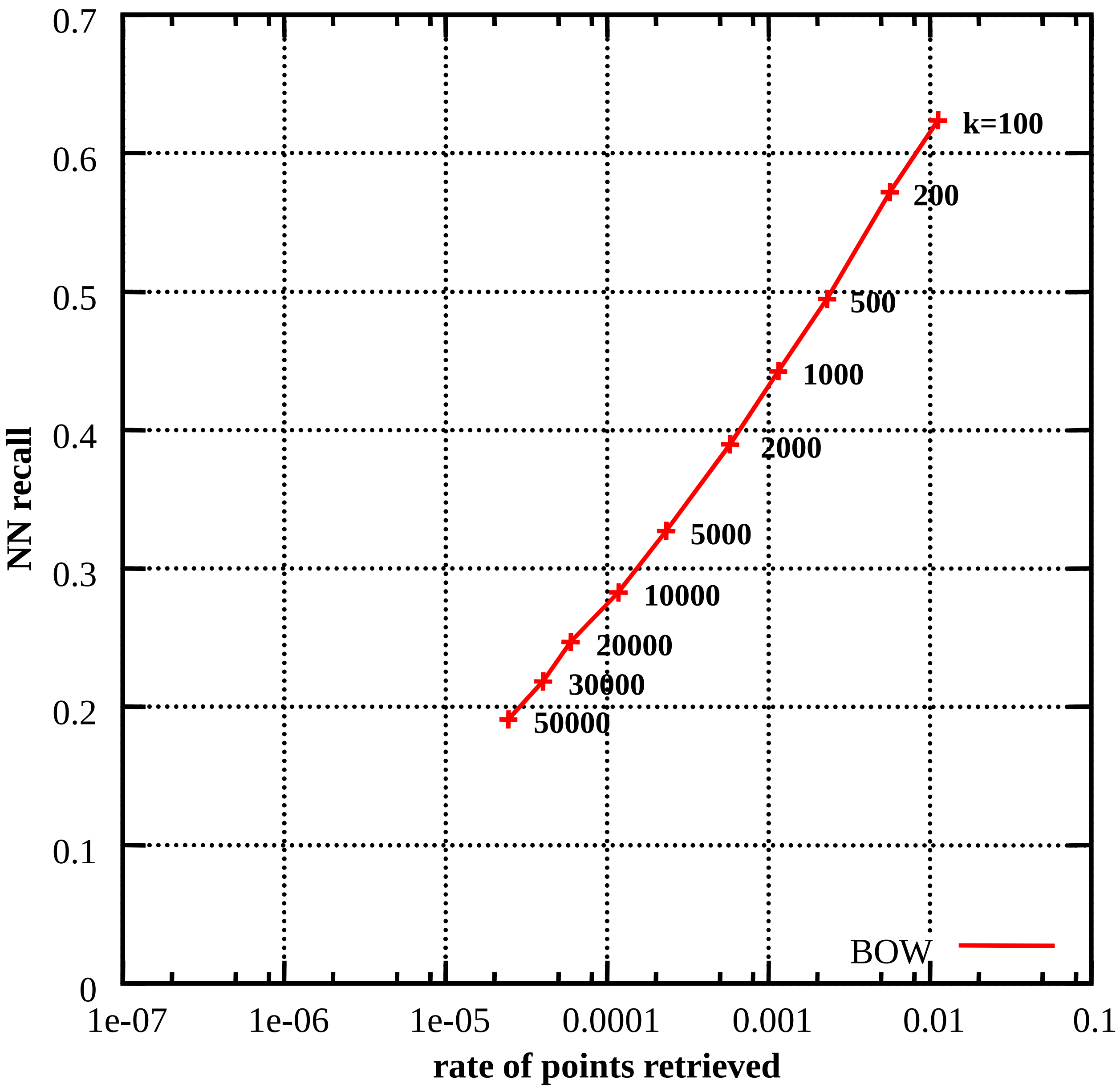
[Lowe 04, Chum & al 2007]

Re-ranked list

- Visual Words
  - 1 word (index) per local descriptor
  - only images ids in inverted file

  ⇒ 8 GB for a million images, fits in RAM
- Problem: Matching approximation

Approximate nearest neighbour (ANN) evaluation of bag-of-features



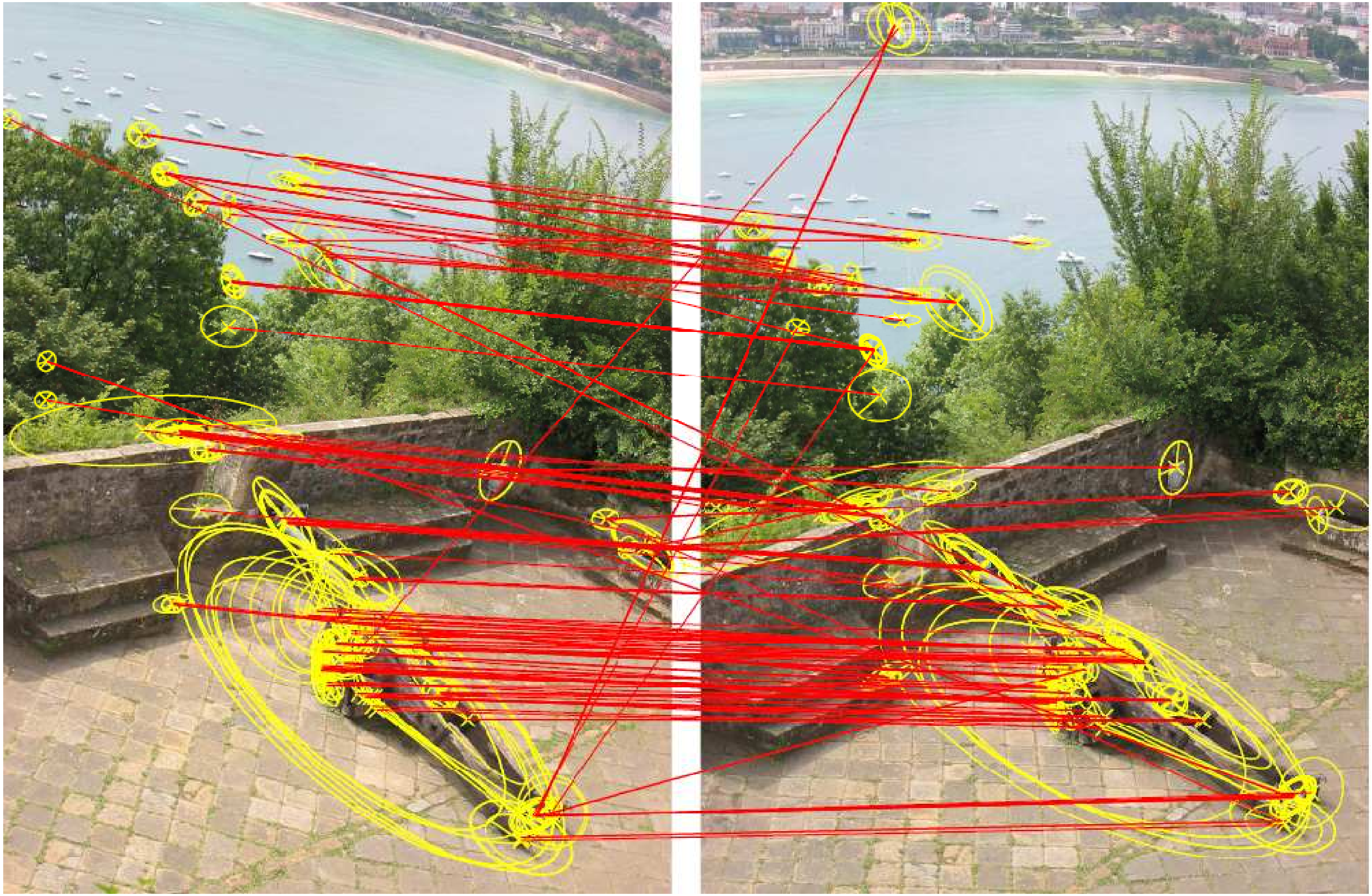
ANN algorithms returns a list of potential neighbors

Accuracy: **NN recall**
= probability that *the* NN is in this list

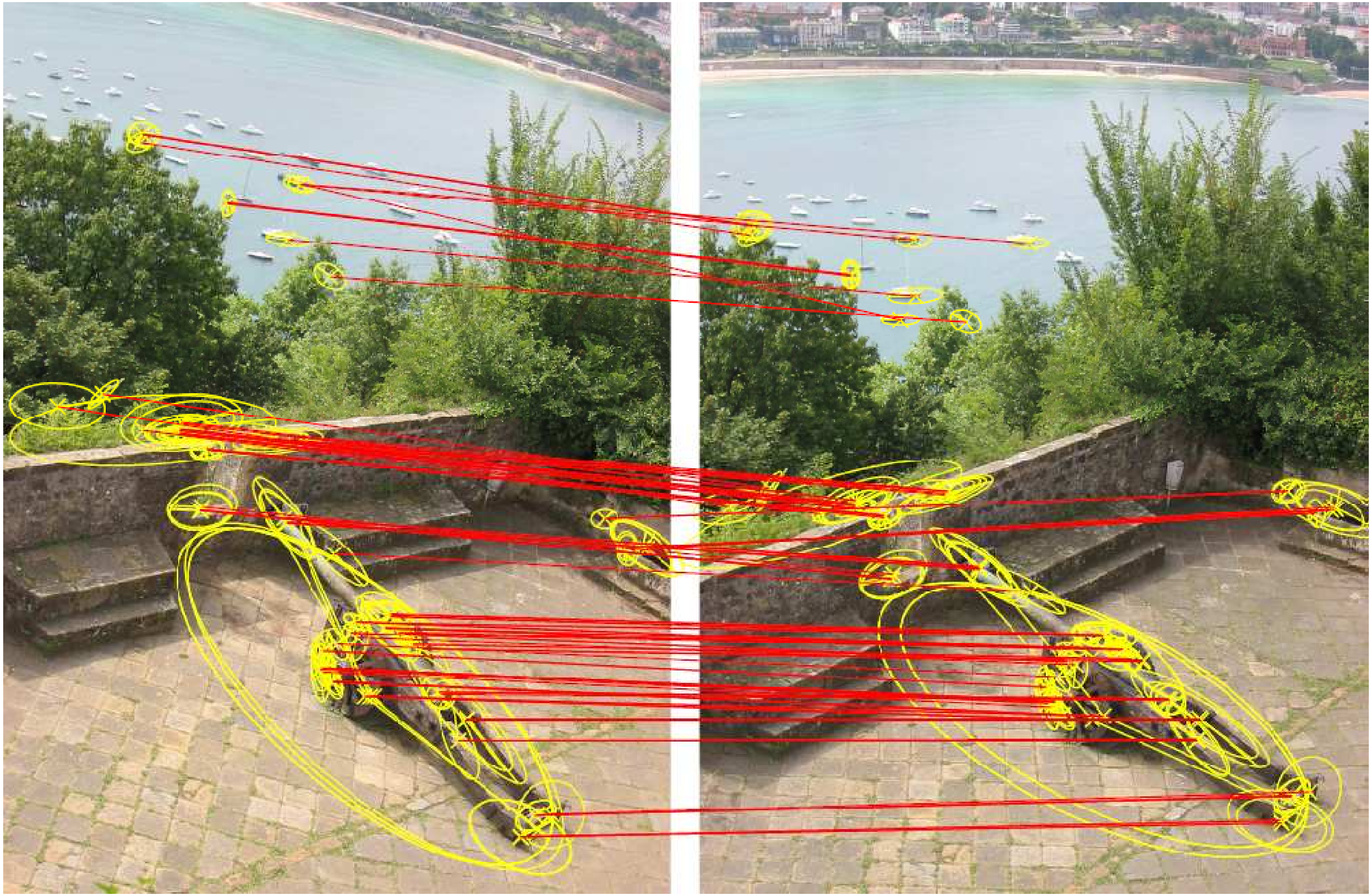
Ambiguity removal:
= proportion of vectors in the short-list

In BOF, this trade-off is managed by the number of clusters *k*

20K visual word: false matchs



200K visual word: good matches missed



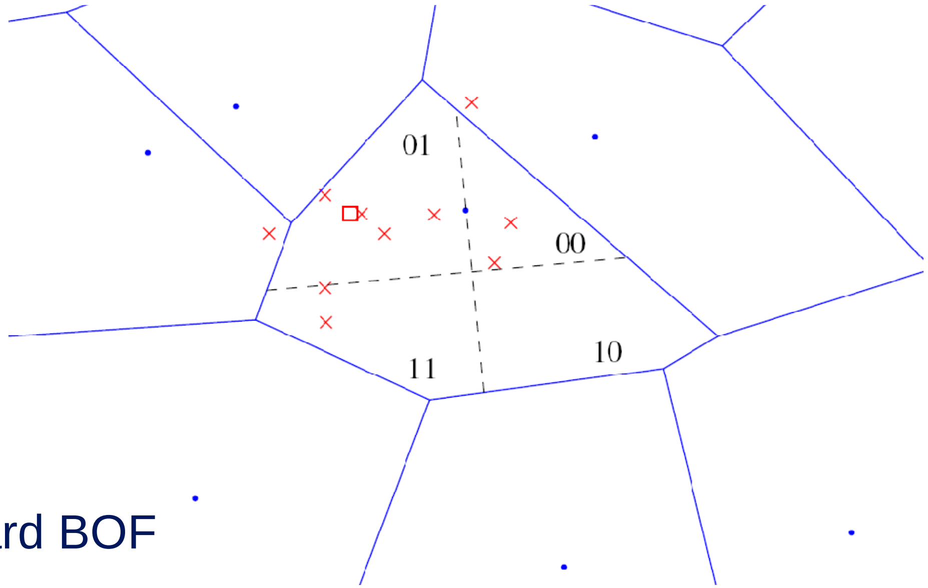
Problem with bag-of-features

- The intrinsic matching scheme performed by BOF is weak
  - for a “small” visual dictionary: too many false matches
  - for a “large” visual dictionary: many true matches are missed

- No good trade-off between “small” and “large” !
  - either the Voronoi cells are too big
  - or these cells can’t absorb the descriptor noise

  $\rightarrow$ intrinsic approximate nearest neighbor search of BOF is not sufficient
  - Possible solutions
    - ➢ Soft assignment [Philbin et al. CVPR’08]
    - ➢ Additional short codes [Jegou et al. ECCV’08]

Hamming Embedding



- Representation of a descriptor *x*
  - Vector-quantized to *q(x)* as in standard BOF
  - \+ short binary vector *b(x)* for an additional localization in the Voronoi cell

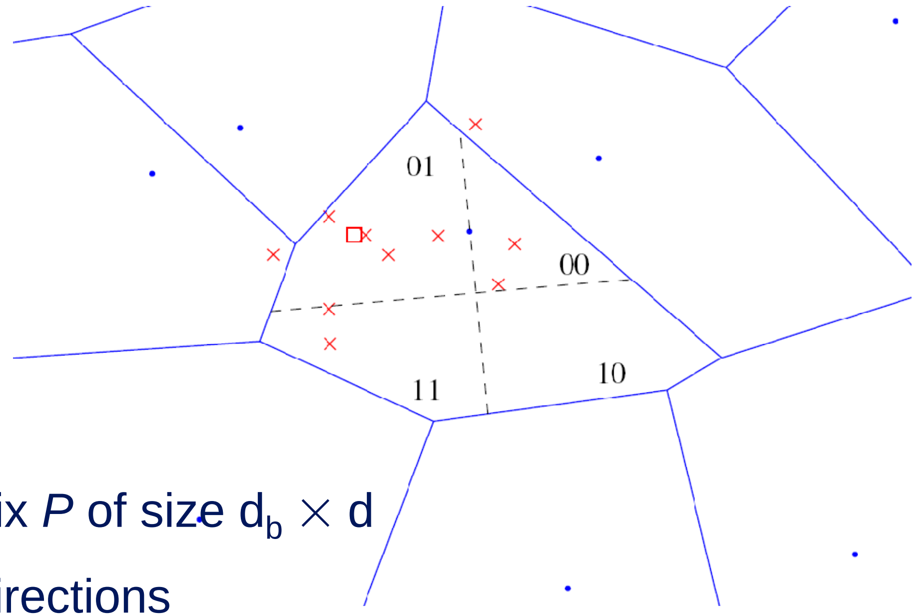
- Two descriptors x and y match iif $q(x) = q(y) \text{ and } h\left(b(x), b(y)\right) \leq h_t$

  where h(*a*,*b*) is the Hamming distance

- Nearest neighbors for Hamming distance ≈ the ones for Euclidean distance

- Efficiency
  - Hamming distance = very few operations
  - Fewer random memory accesses: 3×faster that BOF with same dictionary size!

Hamming Embedding



- **Off-line** (given a quantizer)
  - draw an orthogonal projection matrix P of size $d_b \times d$

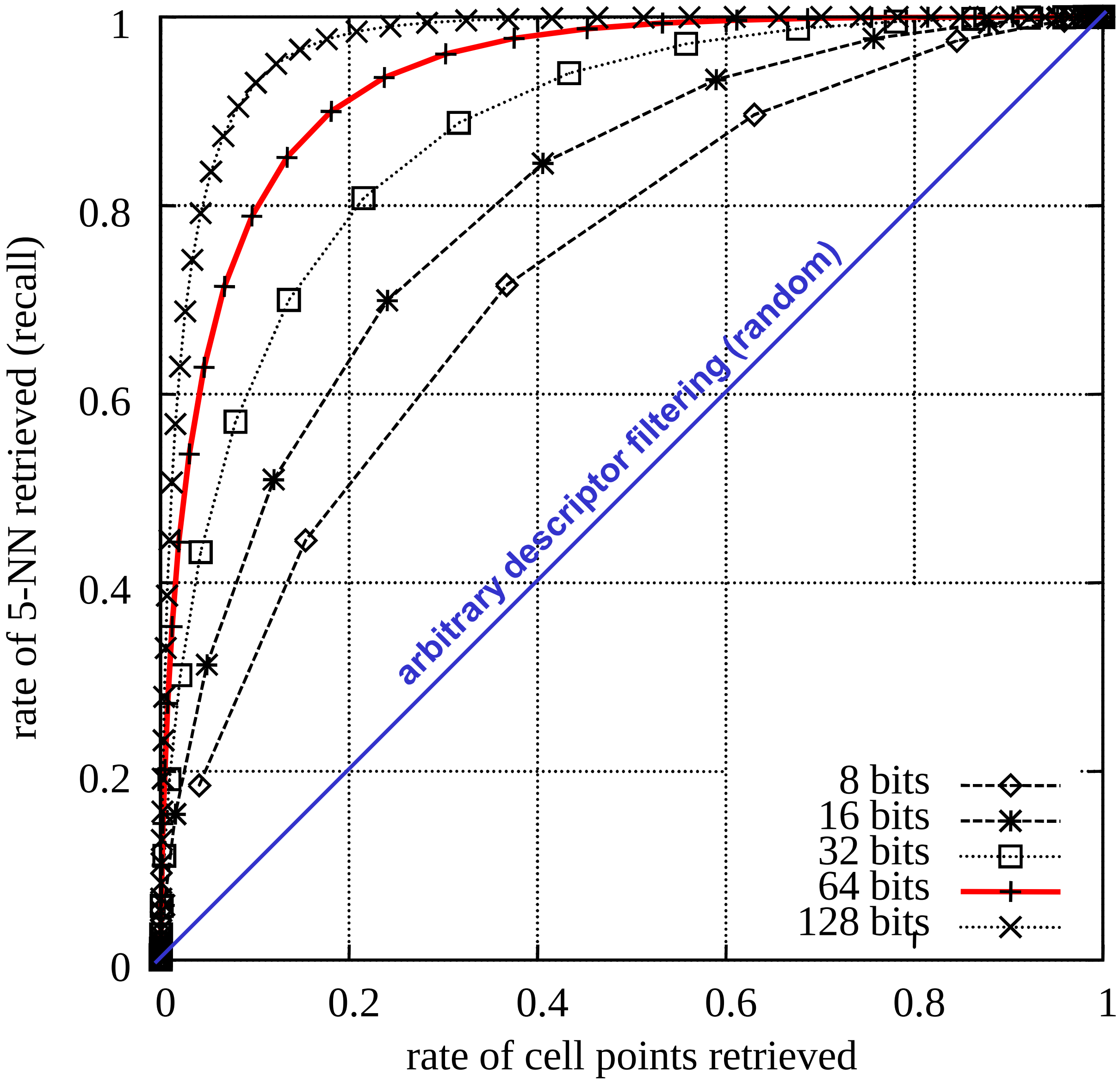
  $\rightarrow$ this defines d_b random projection directions

  - for each Voronoi cell and projection direction, compute the median value from a learning set

- **On-line**: compute the binary signature b(x) of a given descriptor
  - project x onto the projection directions as $z(x) = (z_1, \ldots z_{db})$
  - $b_i(x) = 1$ if $z_i(x)$ is above the learned median value, otherwise 0

[H. Jegou et al., Improving bag of features for large scale image search, ICJV'10]

Hamming and Euclidean neighborhood

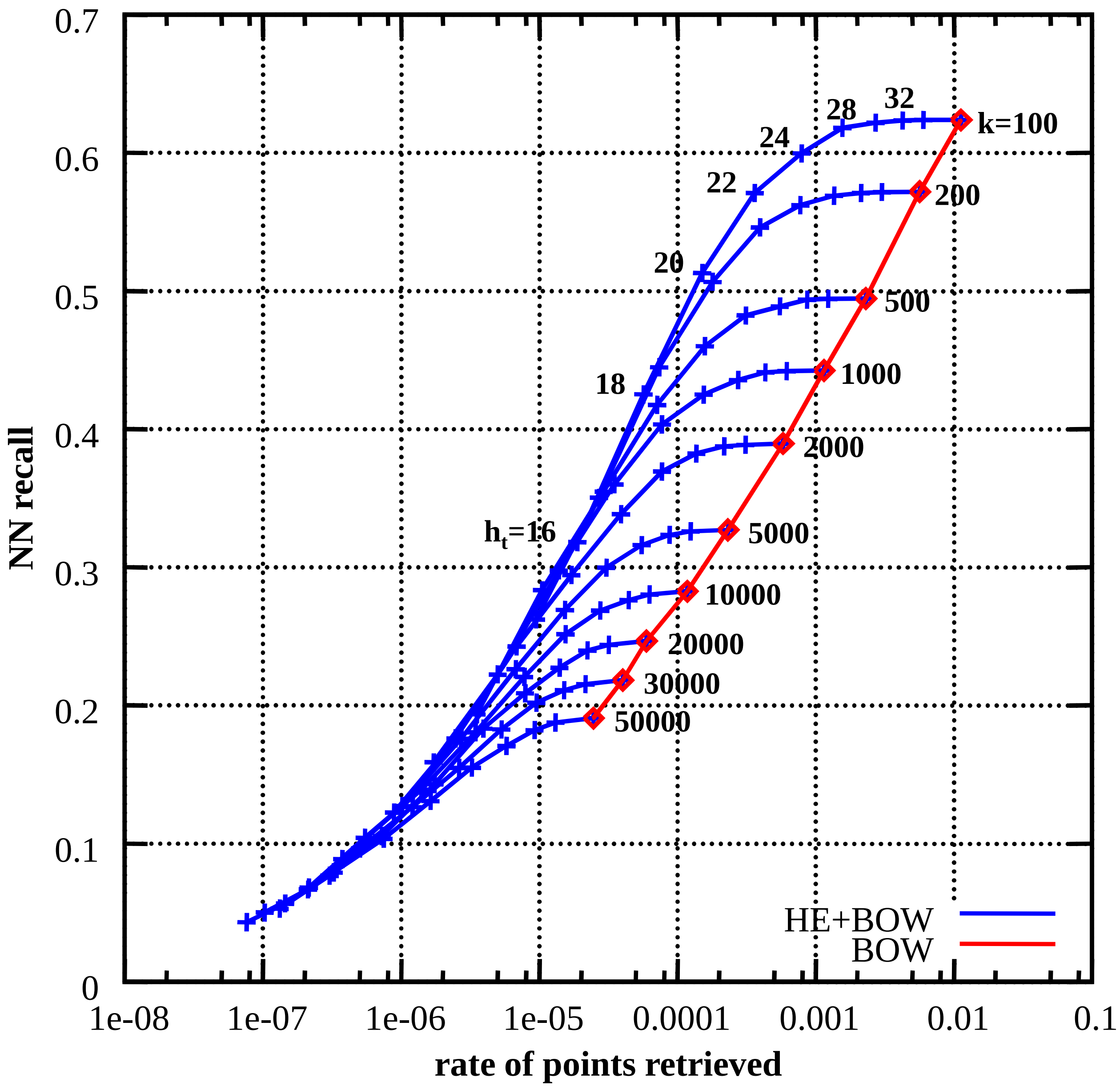


- trade-off between memory usage and accuracy

→ more bits yield higher accuracy

In practice 64 bits (8 bytes)

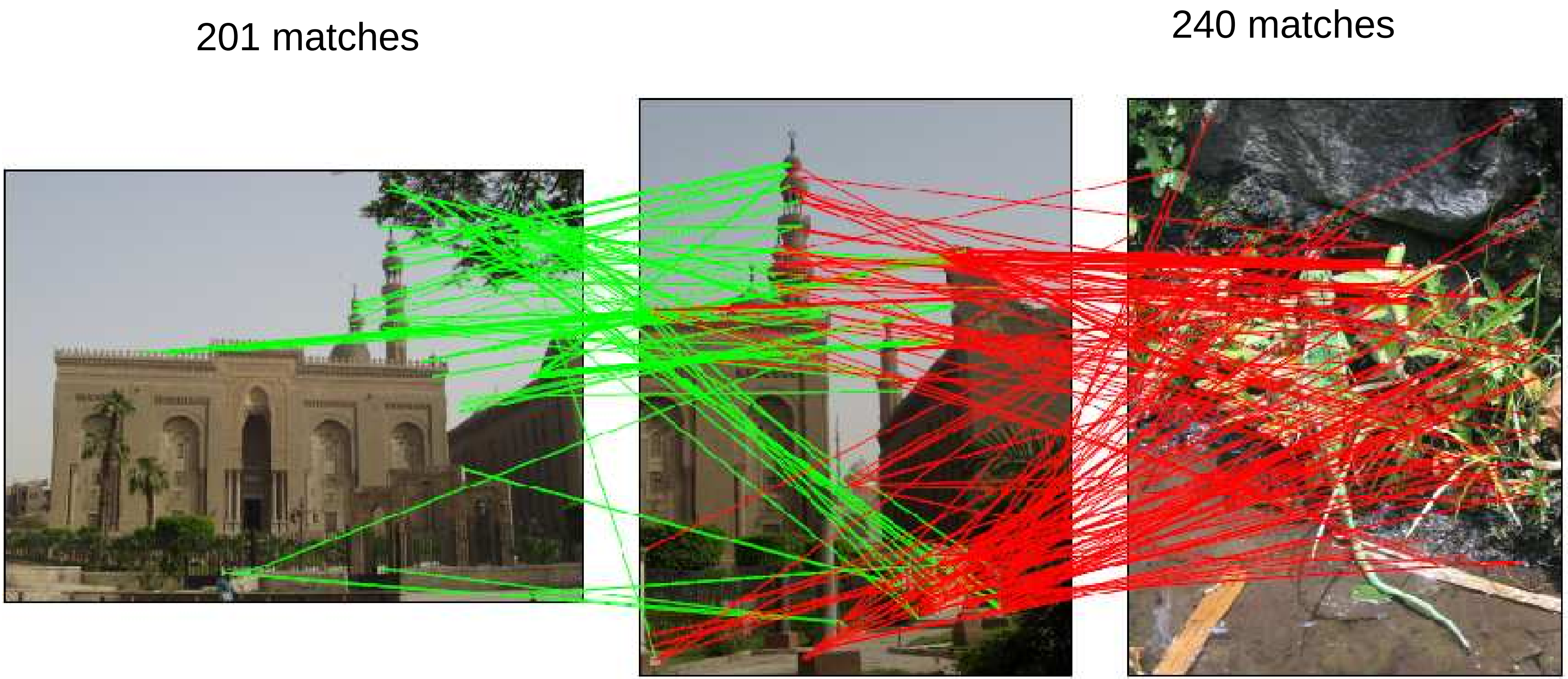
ANN evaluation of Hamming Embedding



compared to BOW: at least 10 times less points in the short-list for the same level of accuracy

Hamming Embedding provides a much better trade-off between recall and ambiguity removal

Matching points - 20k word vocabulary

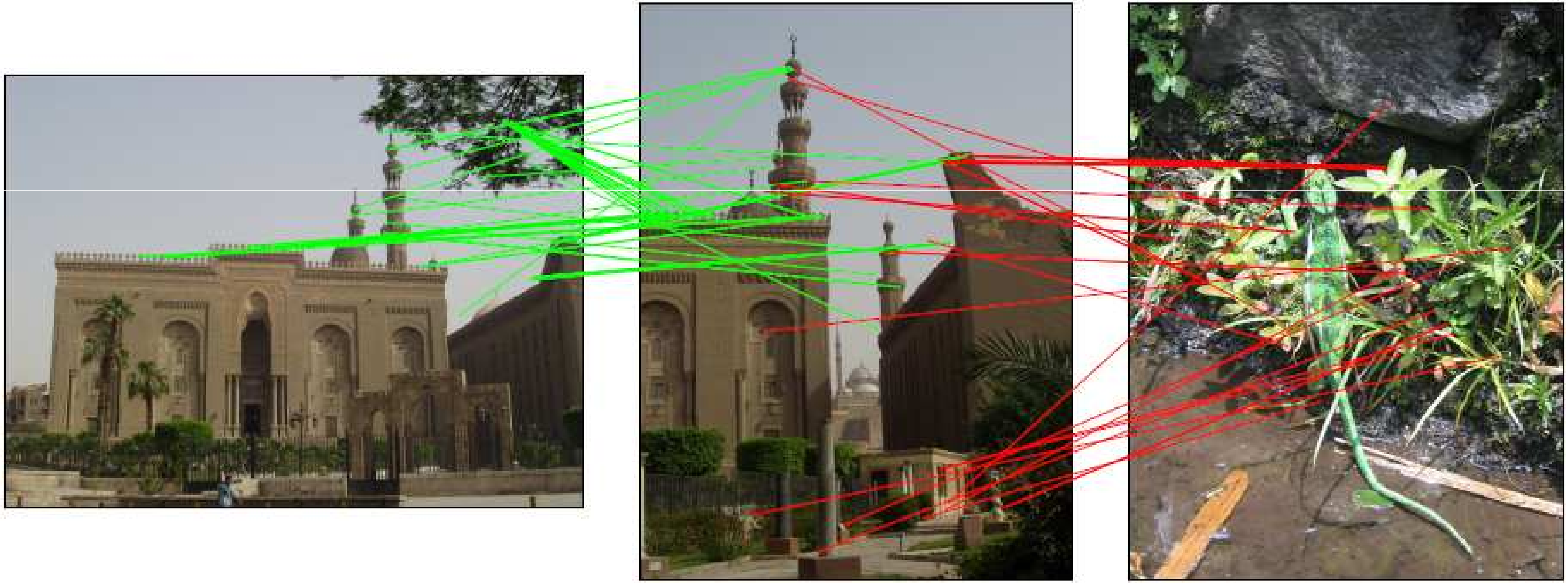


Many matches with the non-corresponding image!

Matching points - 200k word vocabulary

69 matches

35 matches

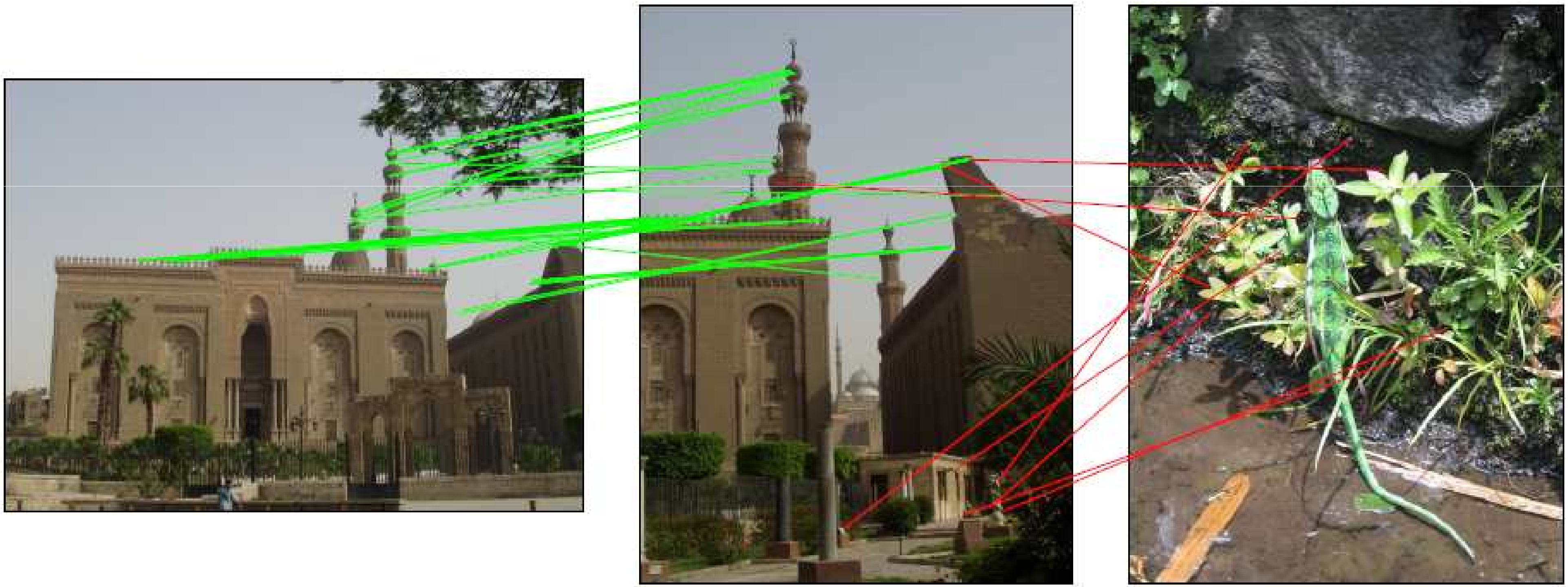


Still many matches with the non-corresponding one

Matching points - 20k word vocabulary + HE

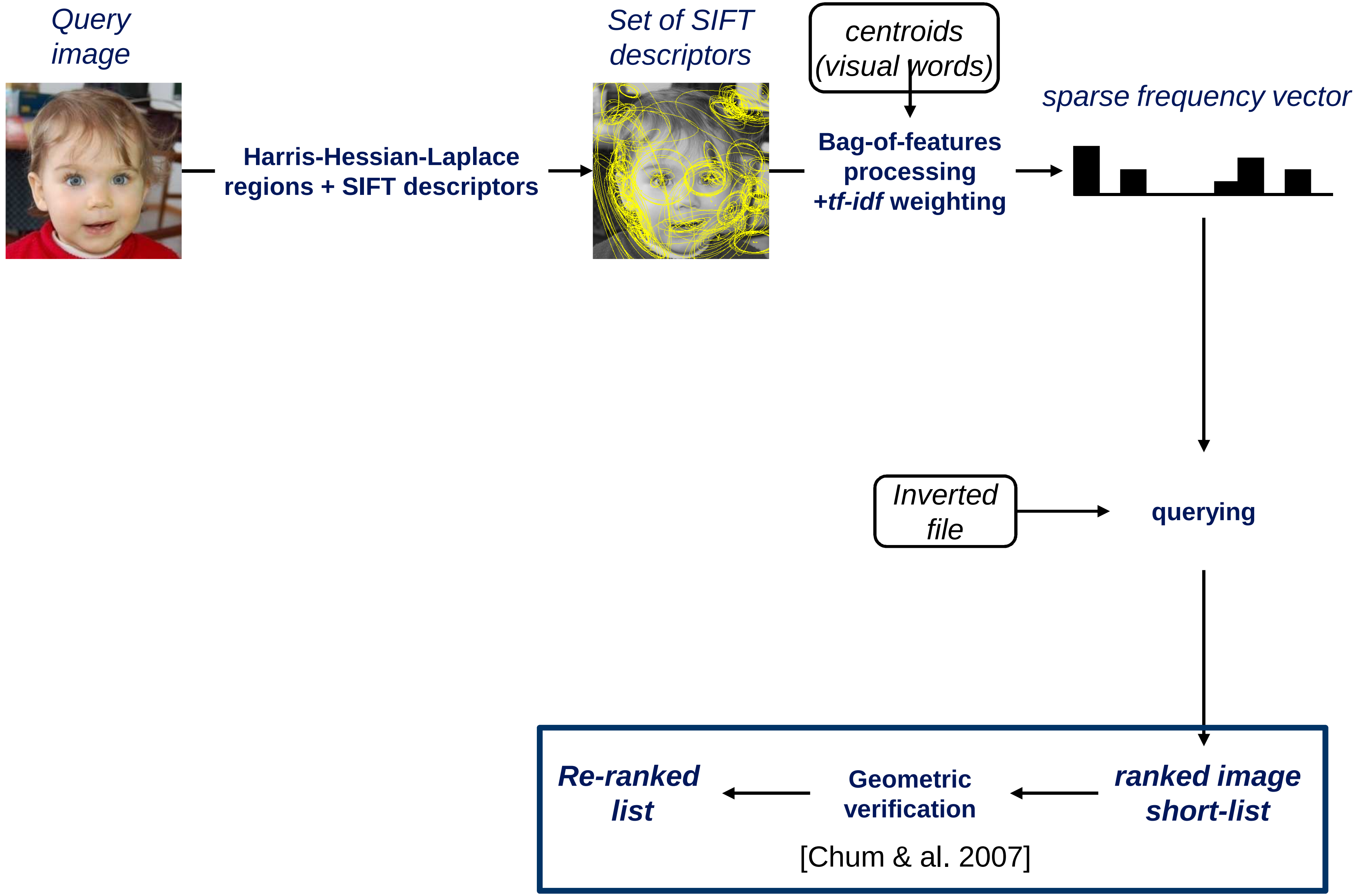
83 matches

8 matches



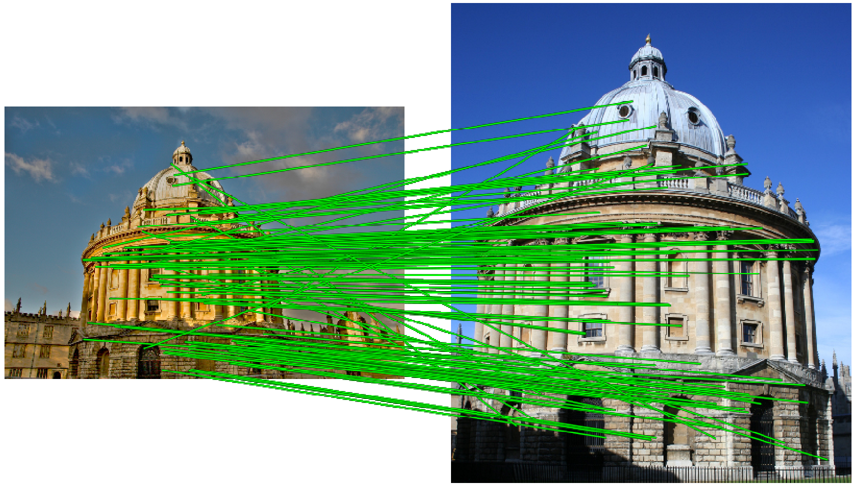
10x more matches with the corresponding image!

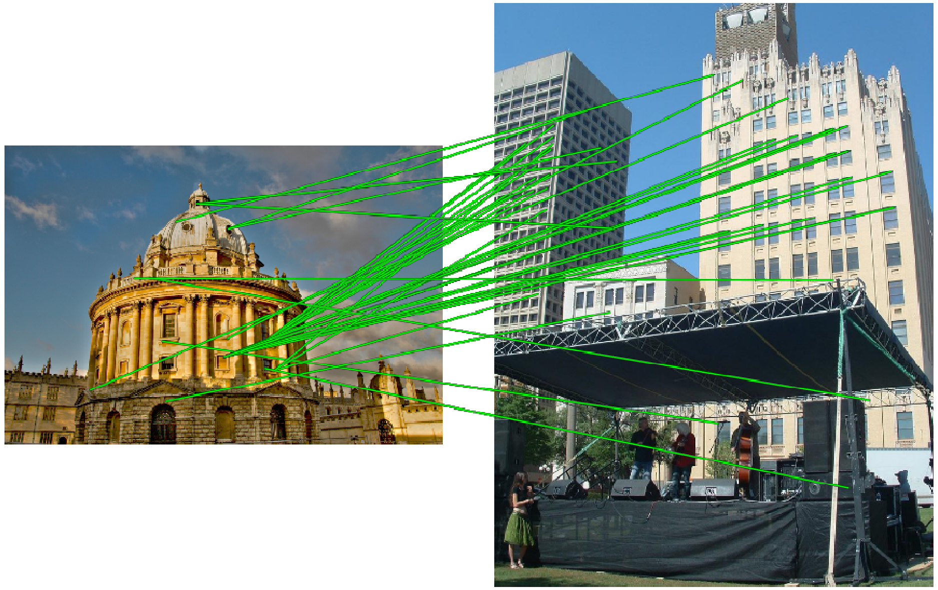
Bag-of-features [Sivic&Zisserman'03]



Geometric verification

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

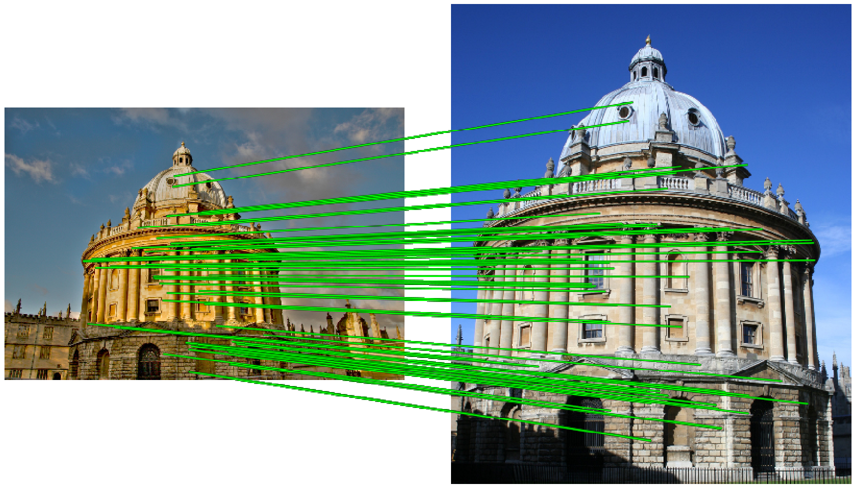




Both images have many matches – which is correct?

Geometric verification

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

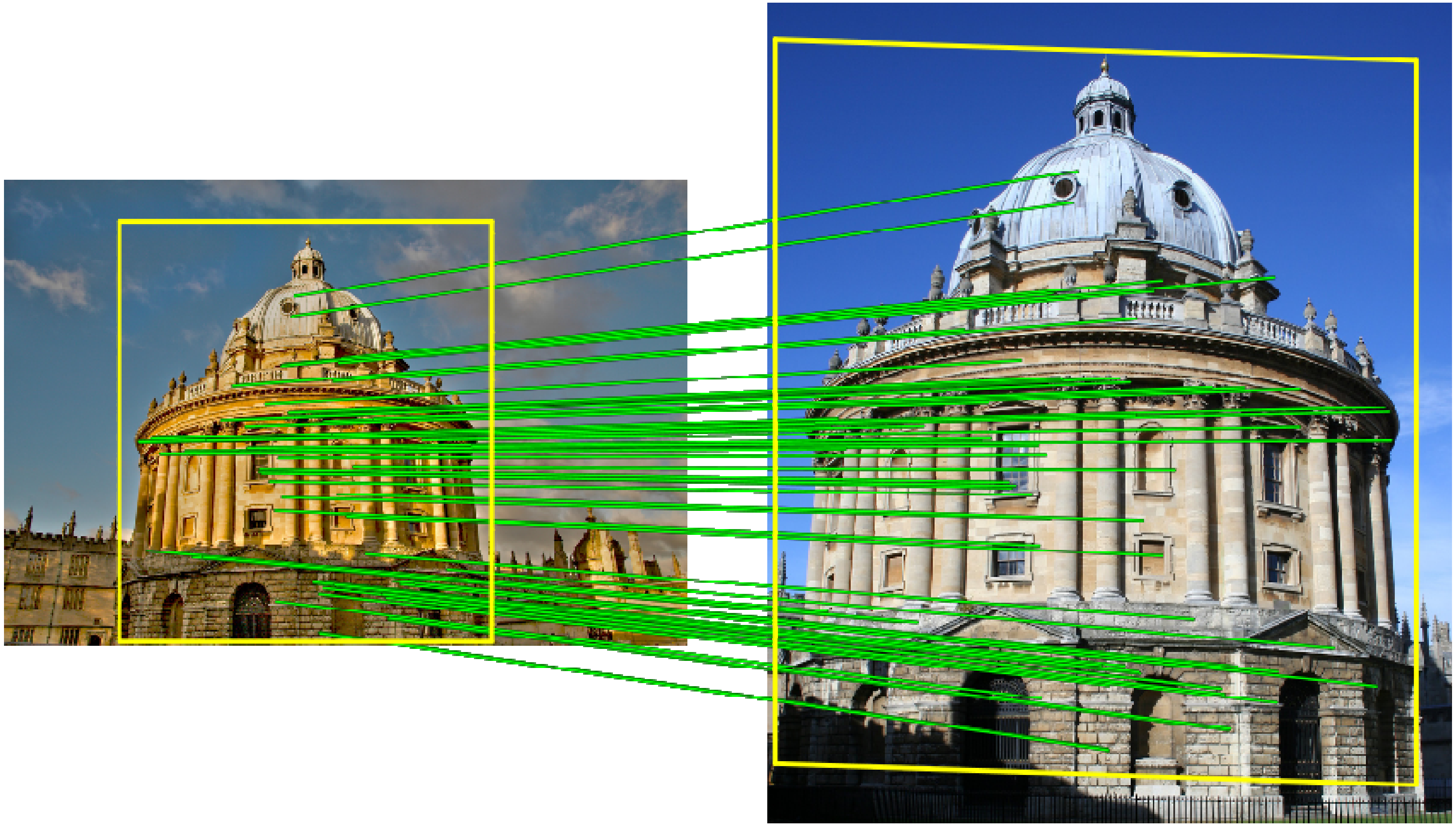


Many spatially consistent matches – **correct result**

Few spatially consistent matches – **incorrect result**

Geometric verification

Gives **localization** of the object

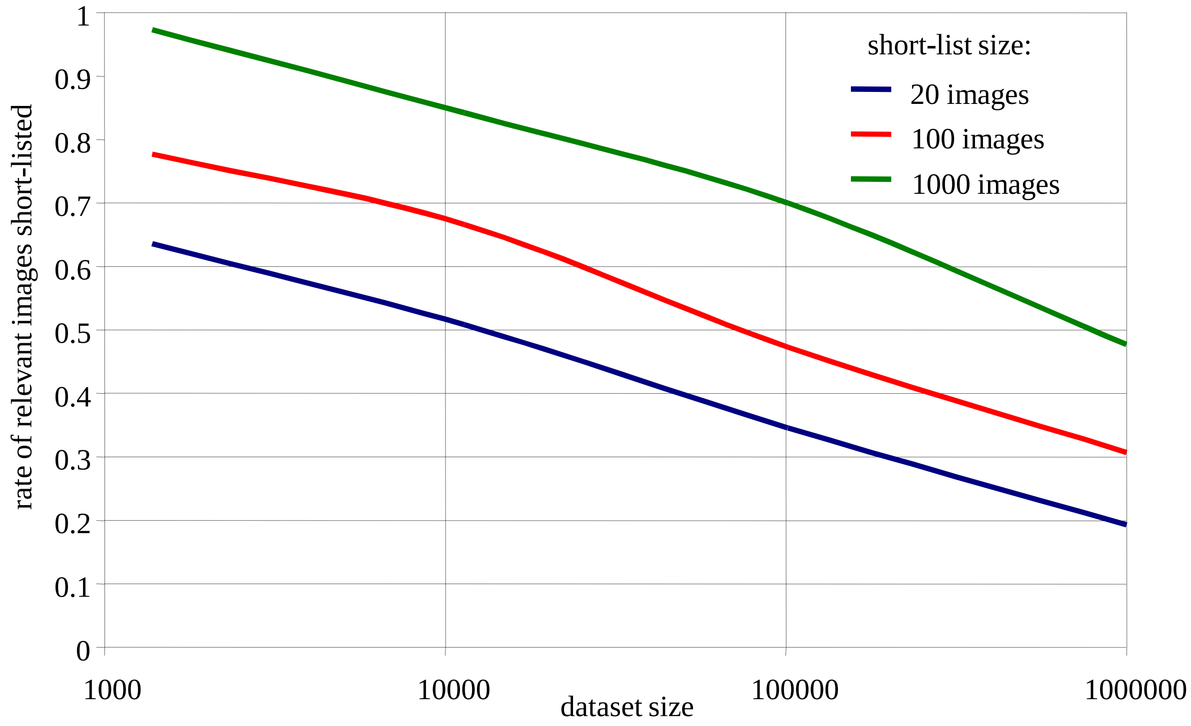


Re-ranking based on geometric verification

- works very well
- but performed on a short-list only (typically, 1000 images)

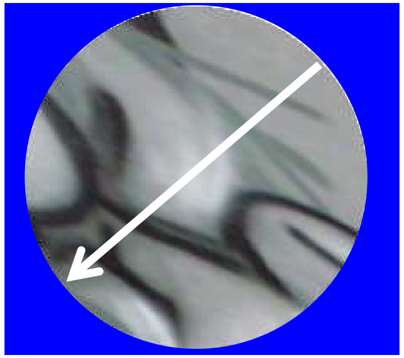
  → for very large datasets, the number of distracting images is so high that relevant images are not even short-listed!

  → Weak geometry



Weak geometry consistency

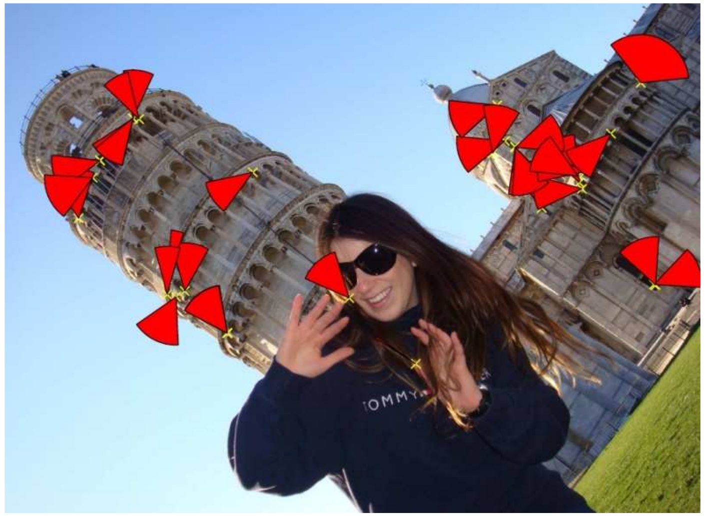
- Weak geometric information used for **all** images (not only the short-list)
- Each invariant interest region detection has a scale and rotation angle associated, here characteristic scale and dominant gradient orientation



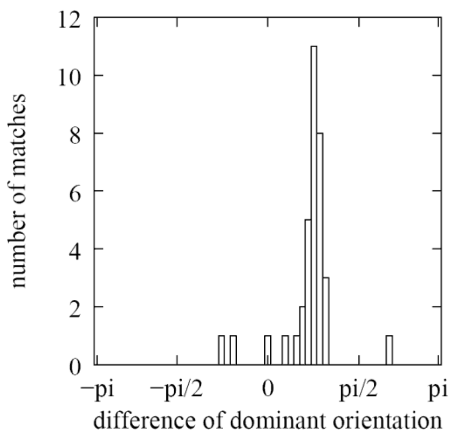
Scale change 2
Rotation angle ca. 20 degrees

- Each matching pair results in a scale and angle difference
- For the global image scale and rotation changes are roughly consistent

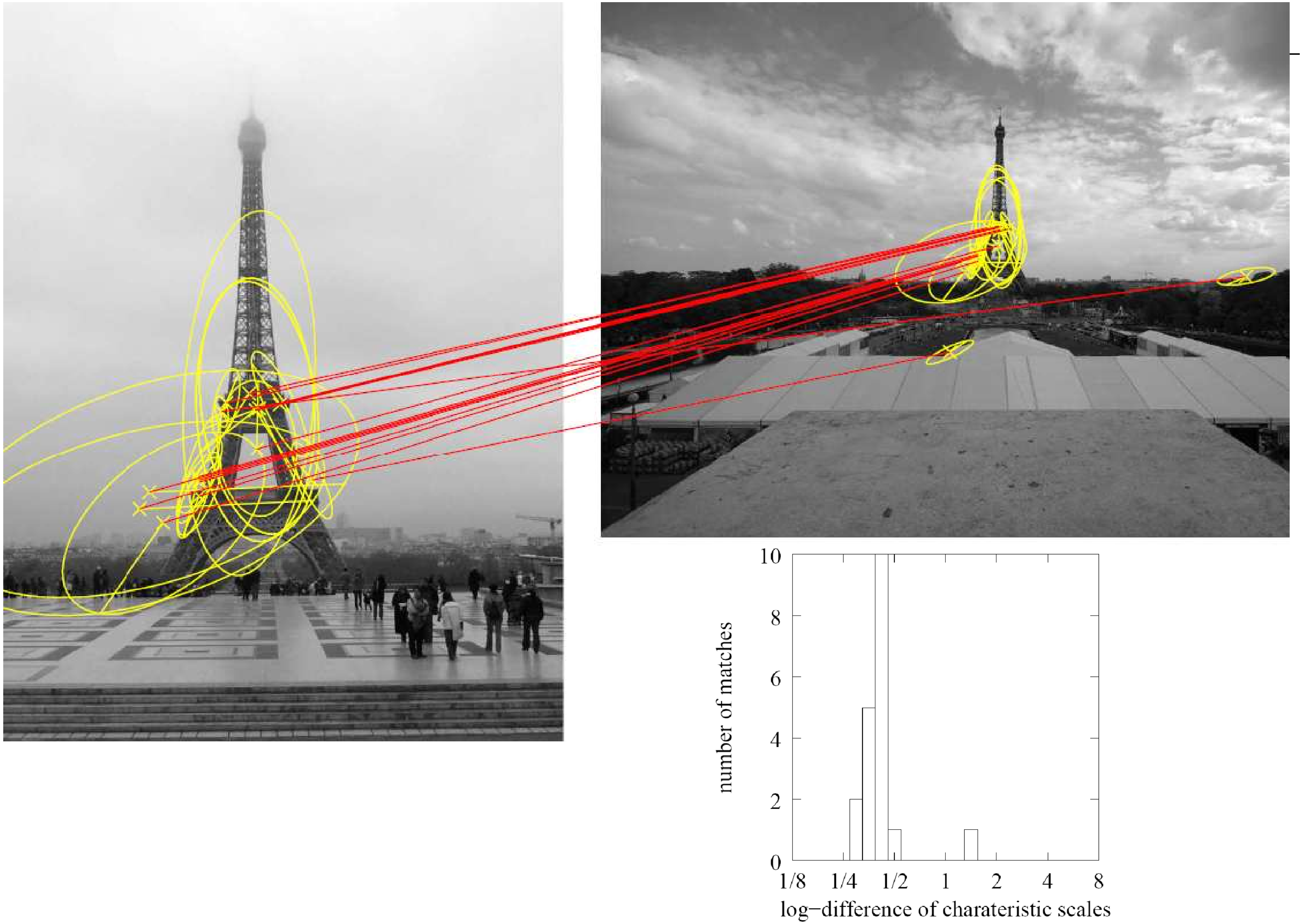
WGC: orientation consistency



Max = rotation angle between images



WGC: scale consistency



Weak geometry consistency

- Integration of the geometric verification into the BOF
  - votes for an image in two quantized subspaces, i.e. for angle & scale
  - these subspace are show to be roughly independent
  - final score: filtering for each parameter (angle and scale)

- Only matches that do agree with the main difference of orientation and scale will be taken into account in the final score

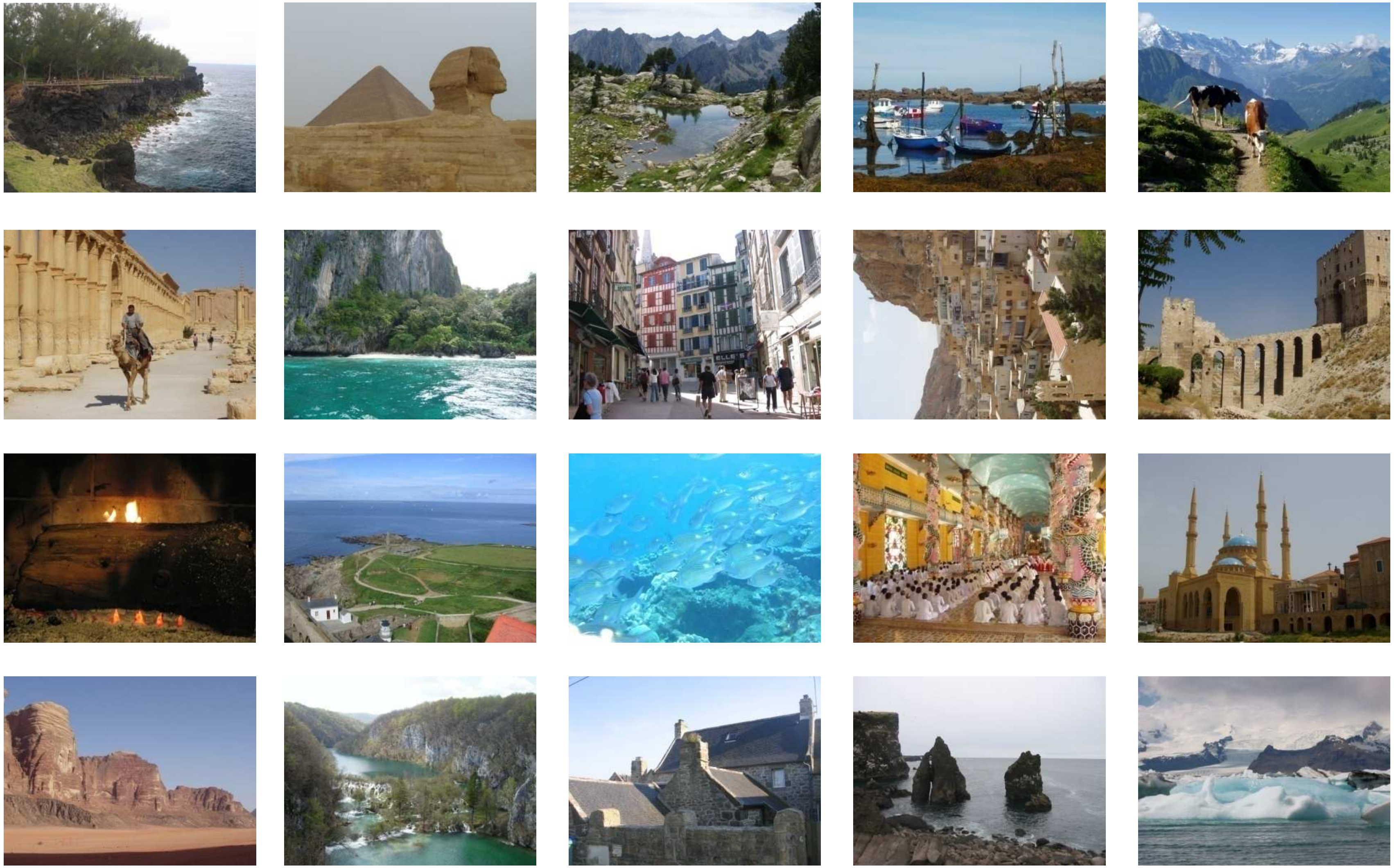
- Re-ranking using full geometric transformation still adds information in a final stage

Experimental results

- Evaluation for the INRIA holidays dataset, 1491 images
  - 500 query images + 991 annotated true positives
  - Most images are holiday photos of friends and family
- 1 million & 10 million distractor images from Flickr
- Vocabulary construction on a different Flickr set
- Almost real-time search speed

- Evaluation metric: mean average precision (in [0,1], bigger = better)
  - Average over precision/recall curve

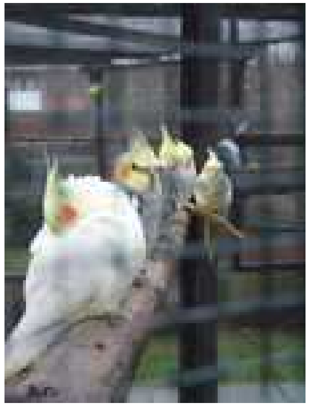
Holiday dataset – example queries



Dataset : Venice Channel

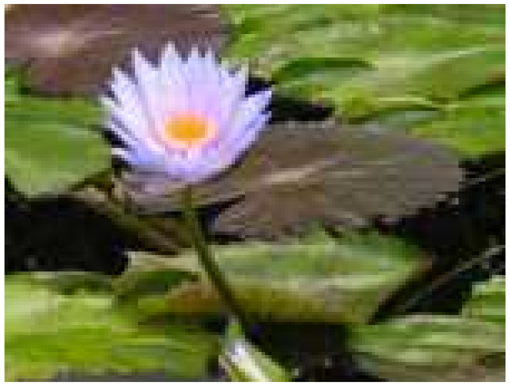
Dataset : San Marco square

Example distractors - Flickr

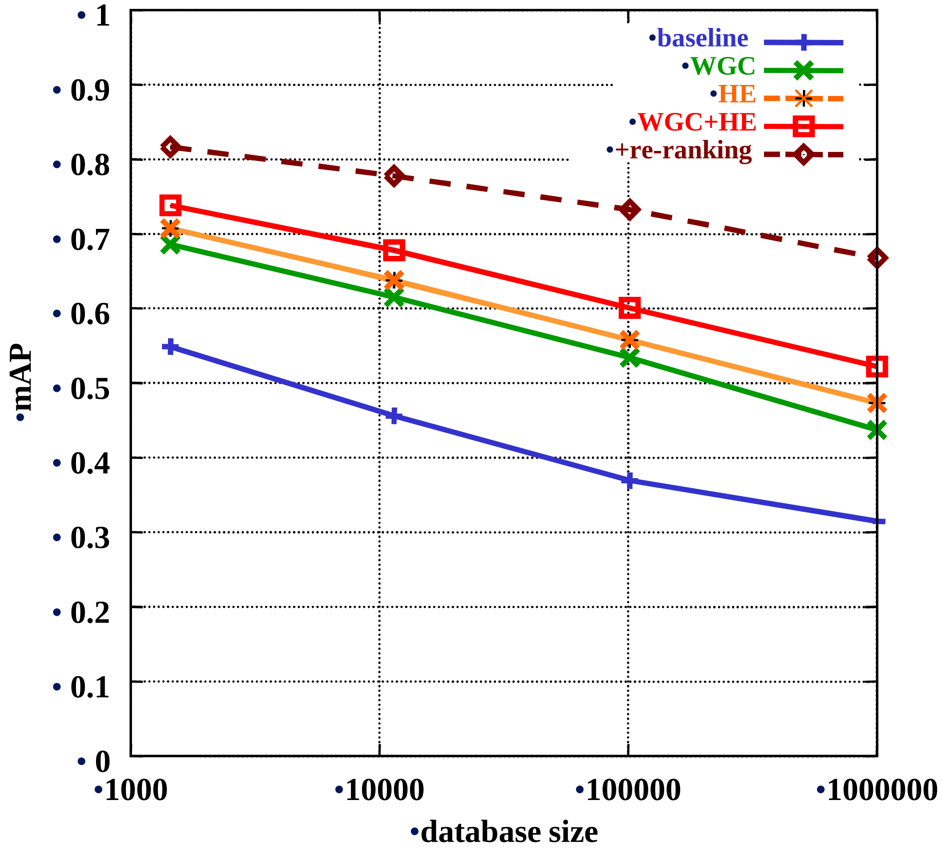





Experimental evaluation

- Evaluation on our holidays dataset, 500 query images, 1 million distracter images
- Metric: mean average precision (in [0,1], bigger = better)



Average query time (4 CPU cores)	
Compute descriptors	880 ms
Quantization	600 ms
Search – baseline	**620** ms
Search – WGC	**2110** ms
Search – HE	**200** ms
Search – HE+WGC	**650** ms

Results – Venice Channel

Comparison with the state of the art: Oxford dataset [Philbin et al. CVPR'07]

Evaluation measure:
Mean average precision (mAP)

Comparison with the state of the art: Kentucky dataset [Nister et al. CVPR'06]

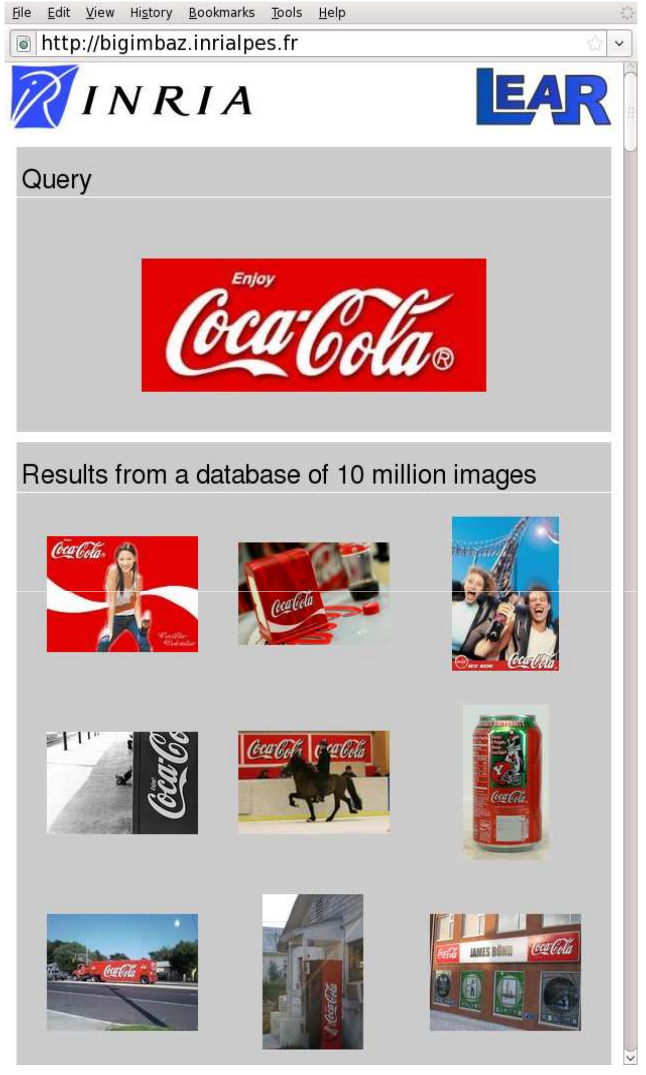
4 images per object

Evaluation measure: among the 4 best retrieval results how many are correct (ranges from 1 to 4)

Comparison with the state of the art

dataset	Oxford		Kentucky	
distractors	0	100K	0	1M
soft assignment [14]	0.493	0.343		
ours	0.615	0.516		
soft + geometrical re-ranking [14]	0.598	0.480		
ours + geometrical re-ranking	0.667	0.591		
soft + query expansion [14]	0.718	0.605		
ours + query expansion	0.747	0.687		
hierarchical vocabulary [6]			3.19	
CDM [11]			3.61	2.93
ours			3.42	3.10
ours + geometrical re-ranking			3.55	3.40

[14] Philbin et al., CVPR'08; [6] Nister et al., CVPR'06; [11] Harzallah et al., CVPR'07



Demo at http://bigimbaz.inrialpes.fr

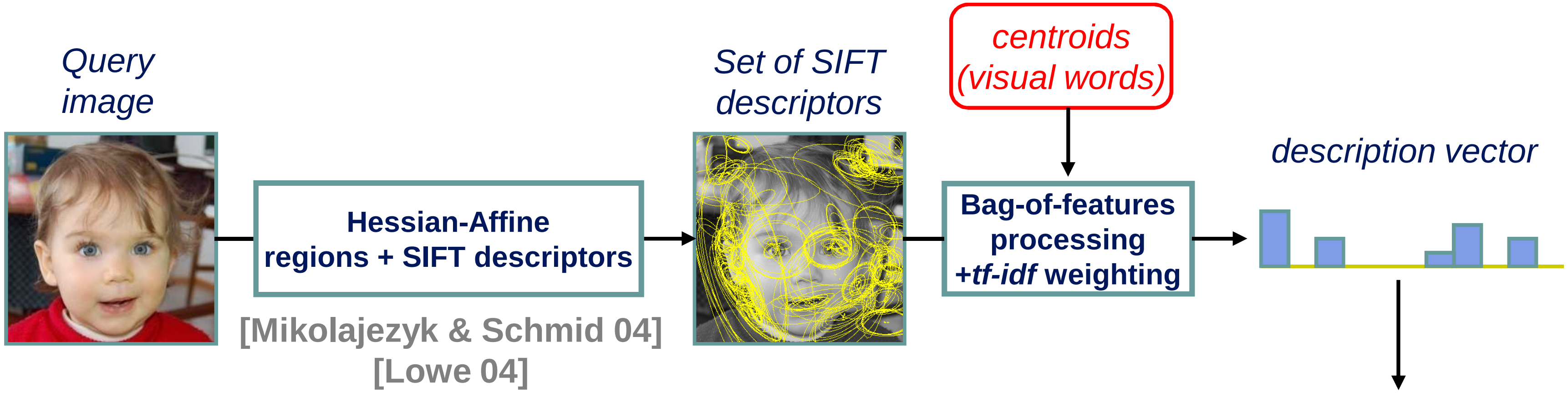
Towards large-scale image search

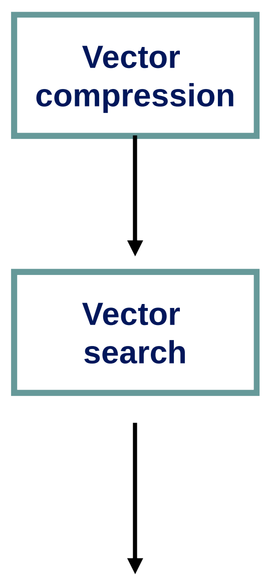
- BOF+inverted file can handle up to ~10 millions images
  - with a limited number of descriptors per image → RAM: 40GB
  - search: 2 seconds

- Web-scale = billions of images
  - with 100 M per machine → search: 20 seconds, RAM: 400 GB
  - not tractable

- Solution: represent each image by one compressed vector

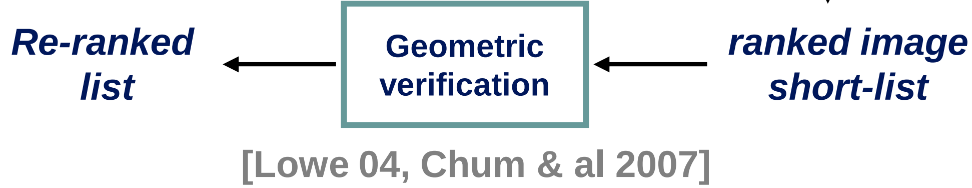
Very large scale image search





- Each image is represented by one vector (Bag-of-features, VLAD, Fisher, GIST)

- Vector compression to reduce storage requirements and search time



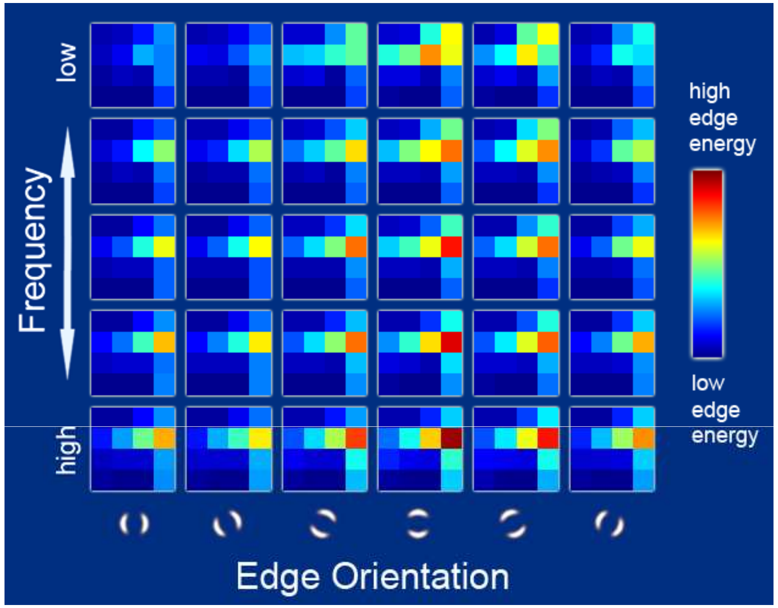
Related work on very large scale image search

- Min-hash and geometrical min-hash [Chum et al. 07-09]
- Compressing the BoF representation (miniBof) [Jegou et al. 09]
  → require hundreds of bytes to obtain a “reasonable quality”

- GIST descriptors with Spectral Hashing [Weiss et al.’08]
  → very limited invariance to scale/rotation/crop

Global scene context – GIST descriptor + spectral hashing

- The "gist" of a scene: Oliva & Torralba (2001)



- 5 frequency bands and 6 orientations for each image location
- Tiling of the image (windowing)
- ~ 900 dimensions

- Spectral hashing produces binary codes similar to spectral clustering

Related work on very large scale image search

- Min-hash and geometrical min-hash [Chum et al. 07-09]
- Compressing the BoF representation (miniBof) [Jegou et al. 09]

  → require hundreds of bytes to obtain a “reasonable quality”

- GIST descriptors with Spectral Hashing [Weiss et al.’08]

  → very limited invariance to scale/rotation/crop

- Efficient object category recognition using classemes [Torresani et al.’10]

- Aggregating local descriptors into a compact image representation [Jegou&al.’10,‘12]