Computer Vision and Object Recognition 2023

Efficient visual search

Josef Sivic

https://people.ciirc.cvut.cz/~sivic/

Czech Institute of Informatics, Robotics and Cybernetics,
Czech Technical University in Prague

External collaborator, Willow Team, Inria, DI ENS, Paris

With slides from: O. Chum, K. Grauman, S. Lazebnik, B. Leibe, D. Lowe, J. Philbin, J. Ponce, D. Nister, C. Schmid, J. Sivic, N. Snavely, A. Zisserman

Announcements

Assignment 2 on neural networks

- Due on Today

Instance-level recognition

Previous lectures:

- Introduction, Basic camera geometry (J. Ponce),
- local invariant features, correspondence and matching (G. Varol)
- Supervised learning (A. Joulin)
- Neural networks for visual recognition (G. Varol)
- Beyond classification: object detection (G. Varol)

This lecture (J. Sivic):

- Efficient visual search

Next week (G. Varol):

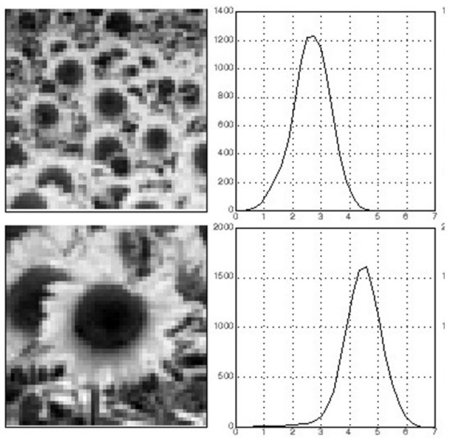
- Generative models, vision and language

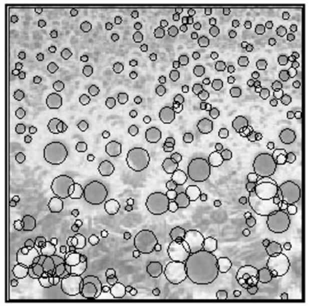
Outline – Efficient visual search

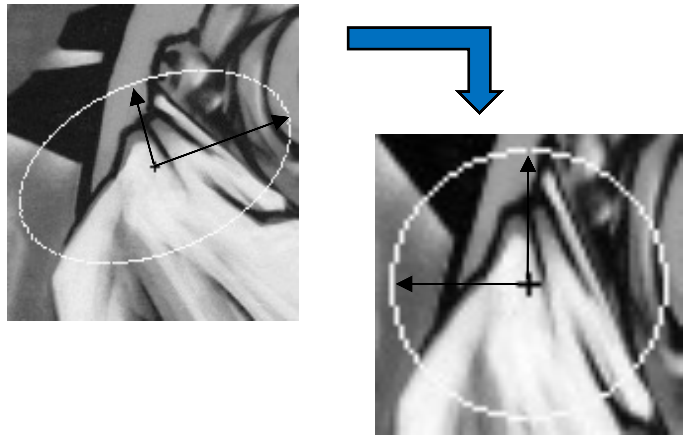
1. Efficient matching of local descriptors
   - Approximate nearest neighbor search
   - k-d trees, locality-sensitive hashing (LSH)
2. Aggregate local descriptors into a single vector
   - Bag-of-visual-words, inverted files, query expansion
3. Compact representations for very large-scale search
   - Product quantization (PQ)
4. Learnable representations
   - Neural representations for large-scale visual search
   - Visual search using natural language query

Recap: Local features

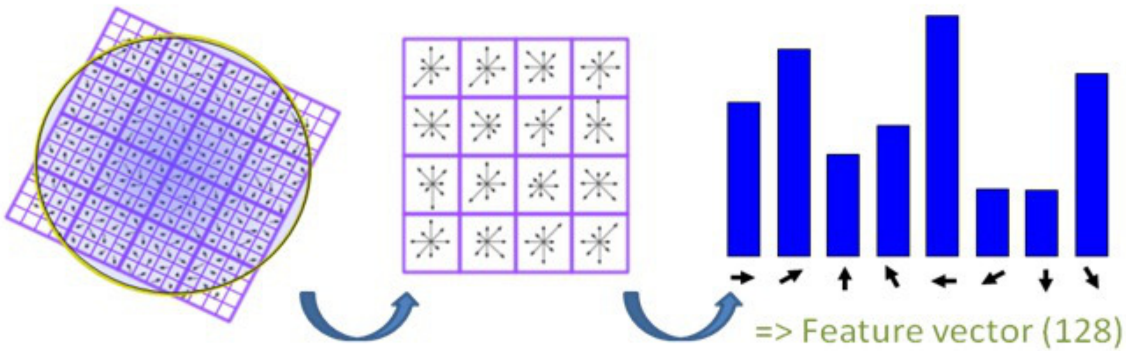
Scale and affine co-variant feature detection





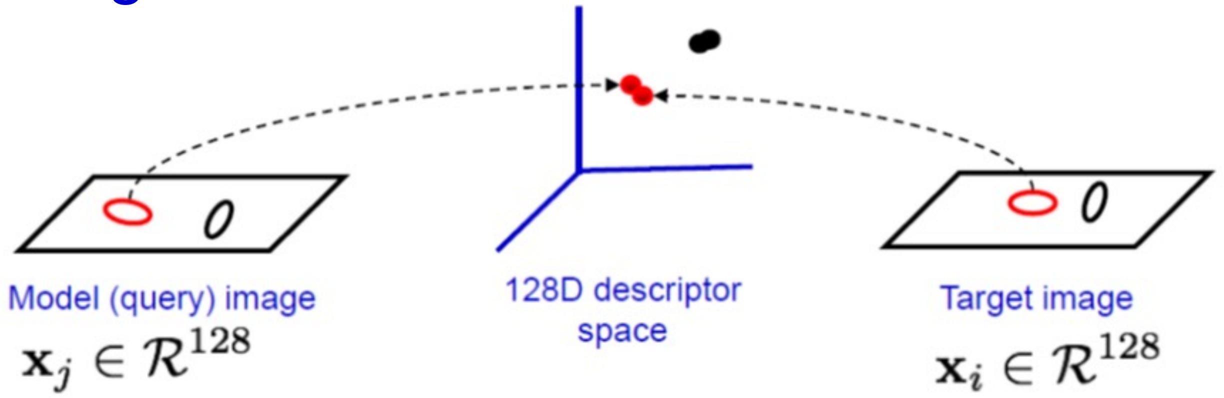


Feature descriptors (SIFT)

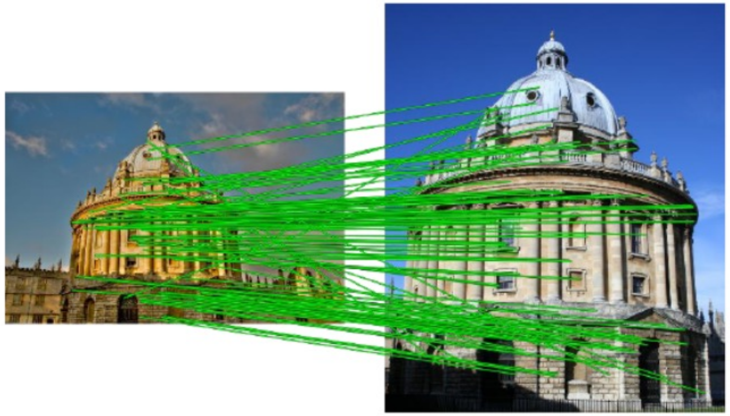


Recap: Matching

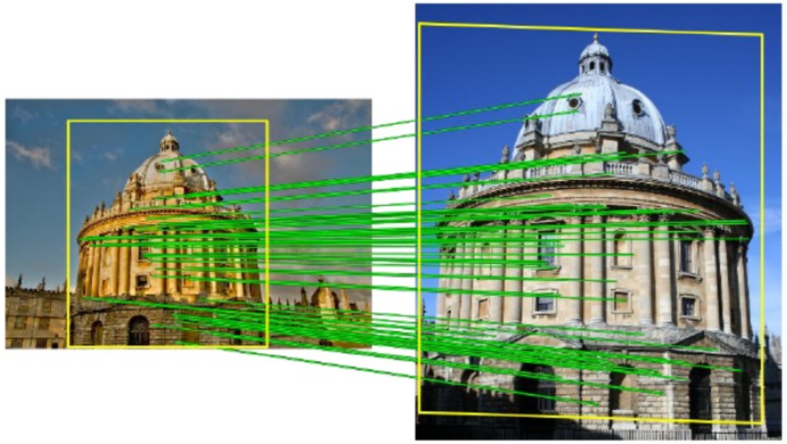
Feature matching



Geometric verification (RANSAC, Hough transform)



Tentative matches



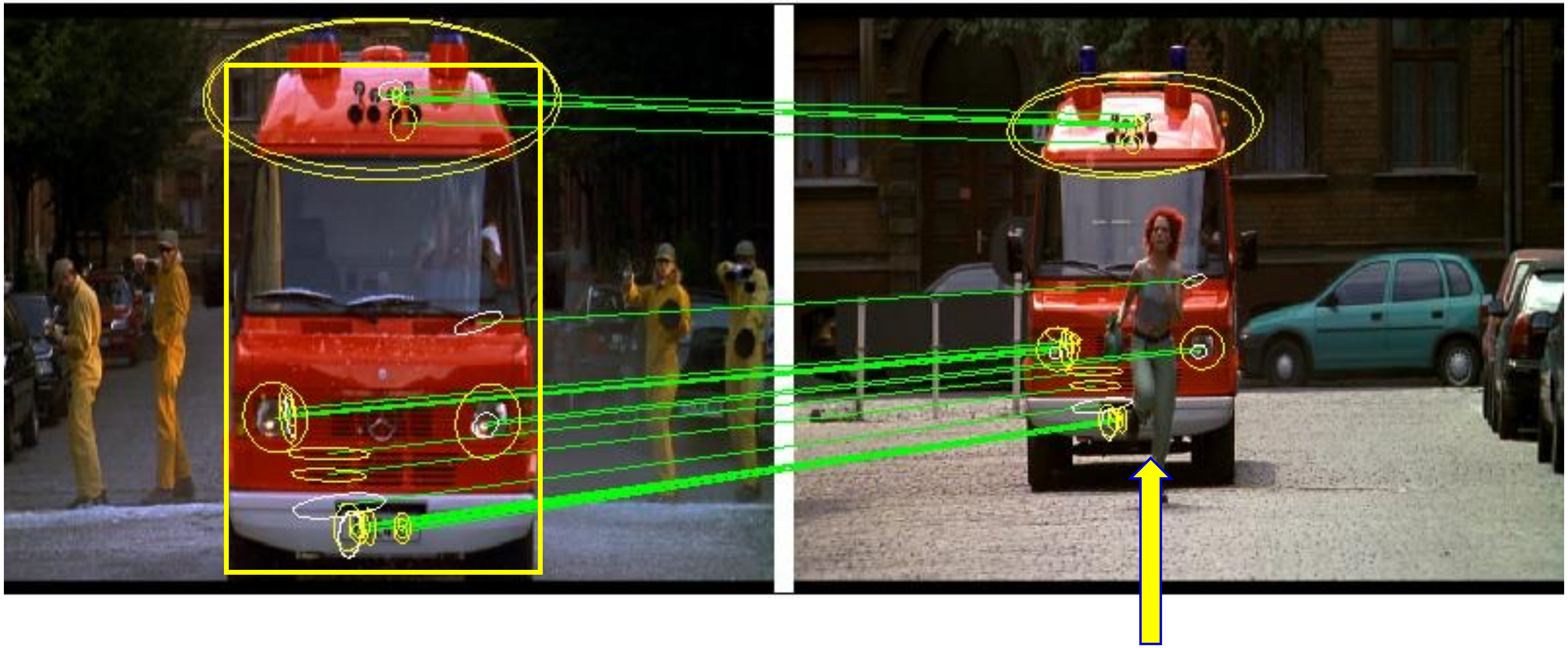
Matches consistent with an affine transformation

Recap: Matching

1000+ descriptors per image

Recap: Matching

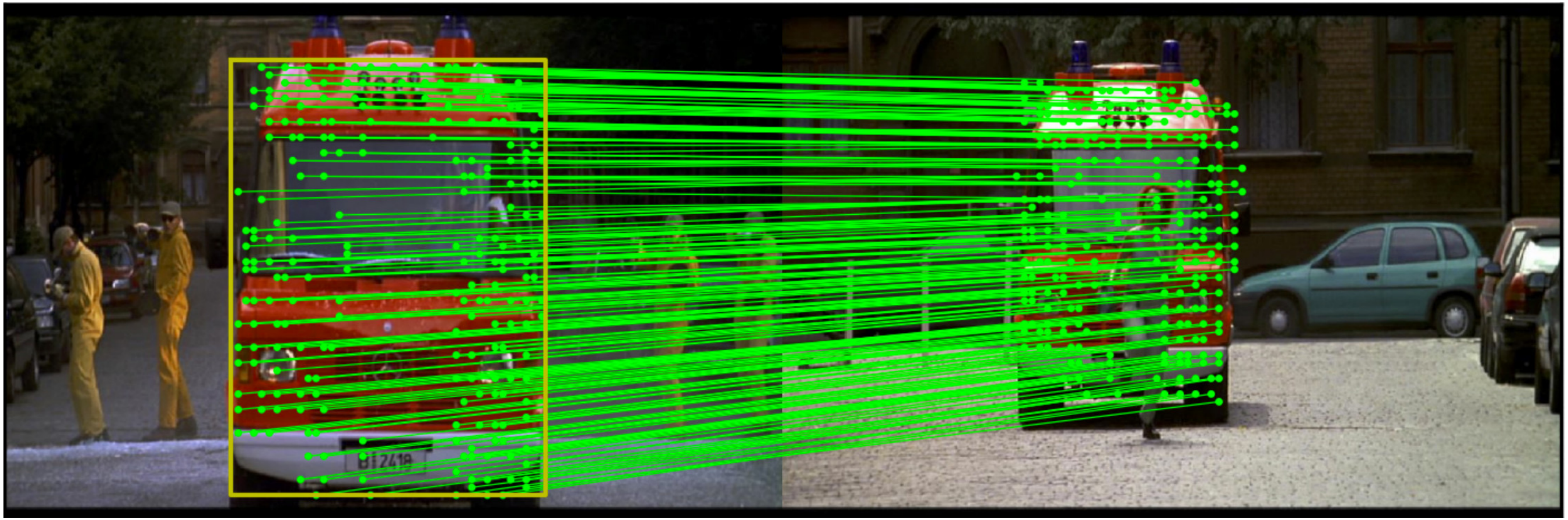
Match regions between frames using SIFT descriptors and spatial consistency



Multiple regions overcome problem of partial occlusion

Matching: Update

Better matches using recent CNN features instead of SIFT



Rocco, Cimpoi, Arandjelovic, Torii, Pajdla, Sivic,
Neighbourhood consensus networks, NIPS 2018

What about multiple images?

So far, we have seen successful matching of a query image to a single target image using local features.

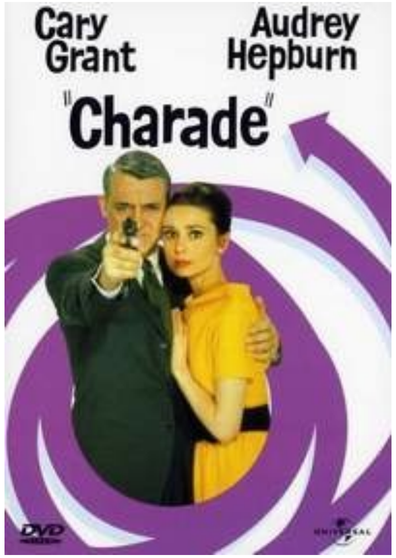
How to generalize this strategy to multiple target images with reasonable complexity?

- 10, 10^2, 10^3, …, 10^7, … 10^{10}, … images?

Example: Visual search in an entire feature length movie

Visually defined query

"Find this bag"



"Charade" [Donen, 1963]

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

Example: Visual search in an entire feature length movie

Visually defined query

"Find this bag"

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

Techniques from efficient search are also used for

- Efficient representation of memory in neural networks

  Lample et al., Large memory layers with product keys, arXiv preprint arXiv:1907.05242.

- Reducing memory footprint of neural networks by quantization

  Fan et al., Training with Quantization Noise for Extreme Model Compression, ICLR 2021 (preprint arXiv:2004.07320)

  P. Stock et al., And the Bit Goes Down: Revisiting the Quantization of Neural Networks, ICLR 2020 (preprint arXiv:1907.05686)

  Gong et al., Compressing deep convolutional networks using vector quantization, arXiv:1412.6115, 2014

- Efficient search of video language embedding spaces.

  Miech et al., HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips, ICCV 2019.

  Demo: https://www.di.ens.fr/willow/research/howto100m/

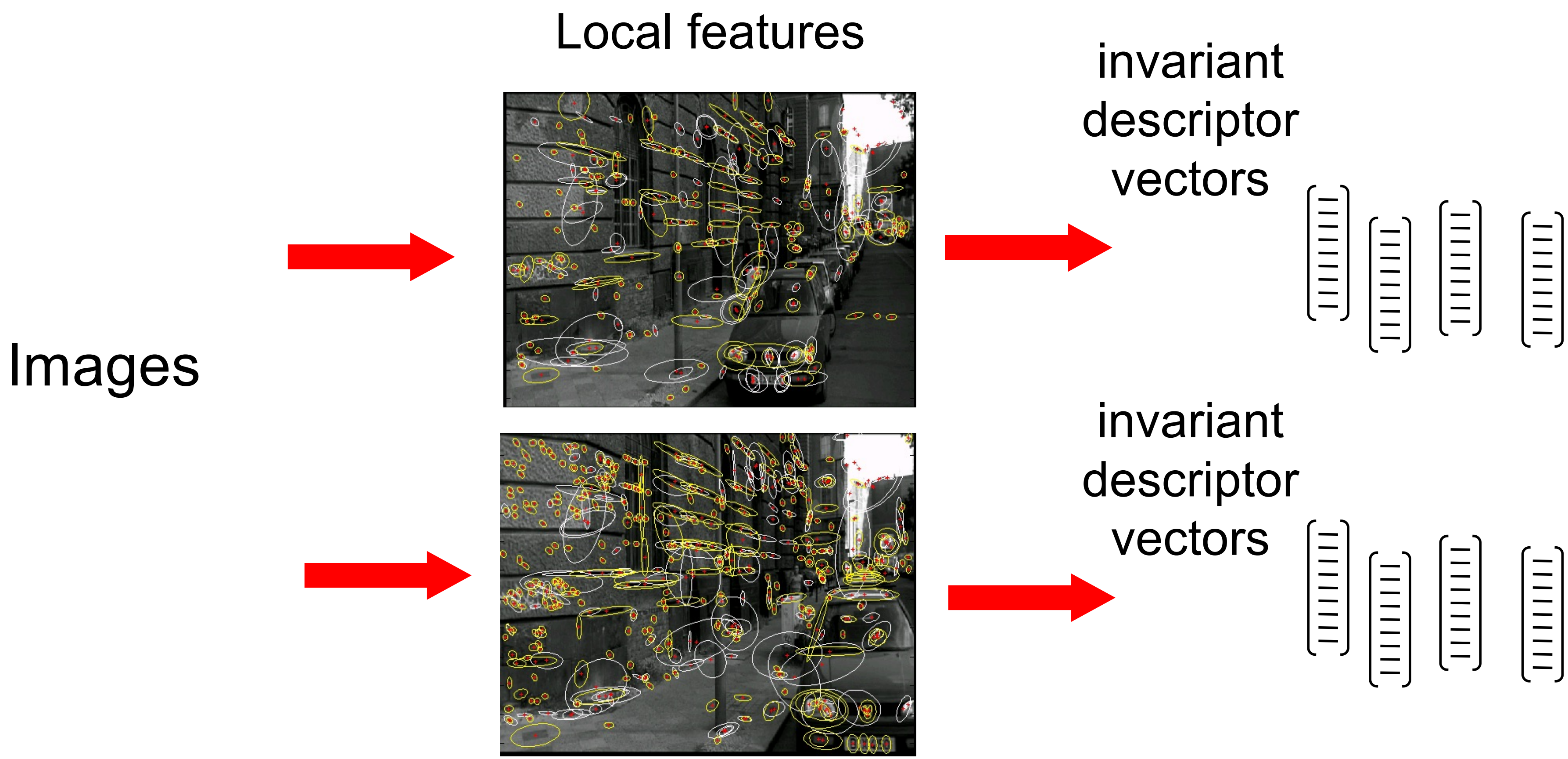
Two strategies

1. Efficient approximate nearest neighbor search on local feature descriptors.

2. Quantize descriptors into a “visual vocabulary” and use efficient techniques from text retrieval.

   (Bag-of-words representation)

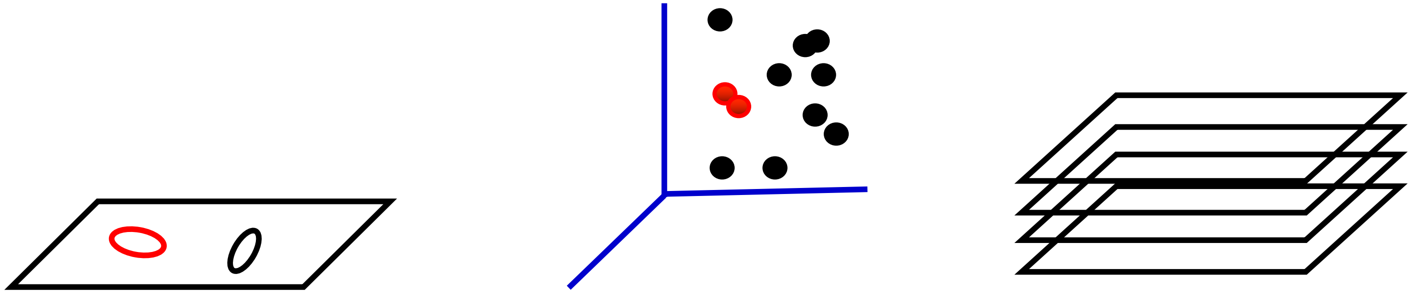
Strategy I: Efficient approximate NN search



1. Compute local features in each image independently
2. “Label” each feature by a descriptor vector based on its intensity
3. Finding corresponding features → finding nearest neighbour vectors
4. Rank matched images by number of (tentatively) corresponding regions
5. Verify top ranked images based on spatial consistency

Finding nearest neighbour vectors

Establish correspondences between a query image and images in the database by **nearest neighbour matching** on SIFT vectors



Model image

128D descriptor space

Image database

Solve following problem for all feature vectors, $\mathbf{x}_j \in \mathcal{R}^{128}$, in the query image:

$$\forall j \; NN(j) = \arg\min_i ||\mathbf{x}_i - \mathbf{x}_j||$$

where, $\mathbf{x}_i \in \mathcal{R}^{128}$, are features from all the database images.

Quick look at the complexity of the NN-search

N … images

M … regions per image (~1000)

D … dimension of the descriptor (~128)

Exhaustive linear search: O(M NMD)

Example:

- Matching two images (N=1), each having 1000 SIFT descriptors
  Nearest neighbors search: 0.4 s (2 GHz CPU, implementation in C)
- Memory footprint: 1000 * 128 = 128kB / image

# of images	CPU time	Memory req.
N = 1,000 …	~7min	(~100MB)
N = 10,000 …	~1h7min	(~ 1GB)
…		
$N = 10^7$	~115 days	(~ 1TB)
…		
(Some) Images on Facebook:		
$N = 10^{10}$ …	~300 years	(~ 1PB)

Nearest-neighbor matching

Solve following problem for all feature vectors, $\mathbf{x_j}$, in the query image:

$$\forall j \; NN(j) = \arg\min_i ||\mathbf{x}_i - \mathbf{x}_j||$$

where $\mathbf{x_i}$ are features in database images.

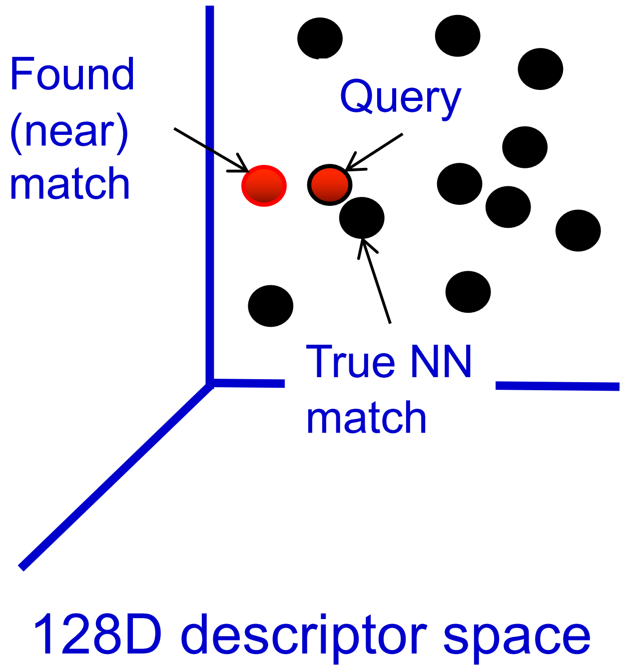
Nearest-neighbour matching is the major computational bottleneck

- Linear search performs *dn* operations for *n* features in the database and *d* dimensions
- No exact methods are faster than linear search for d>10
- Approximate methods can be much faster, but at the cost of missing some correct matches

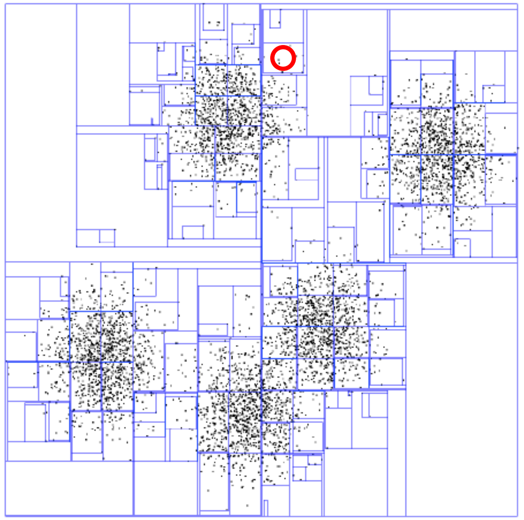
Finding *approximate* nearest neighbour vectors

- Approximate method is not guaranteed to find the nearest neighbour.

- Can be much faster, but at the cost of missing some nearest matches

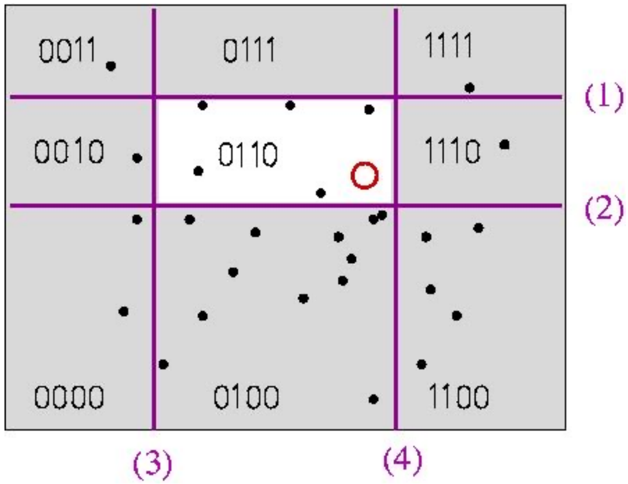


Approximate nearest neighbor search



Best-Bin First (BBF), a variant of k-d trees that uses priority queue to examine most promising branches first
[Beis & Lowe, CVPR 1997]

Extended to multiple randomized trees in :
[Muja & Lowe, 2009]



Locality-Sensitive Hashing (LSH), a randomized hashing technique using hash functions that map similar points to the same bin, with high probability
[Indyk & Motwani, 1998]

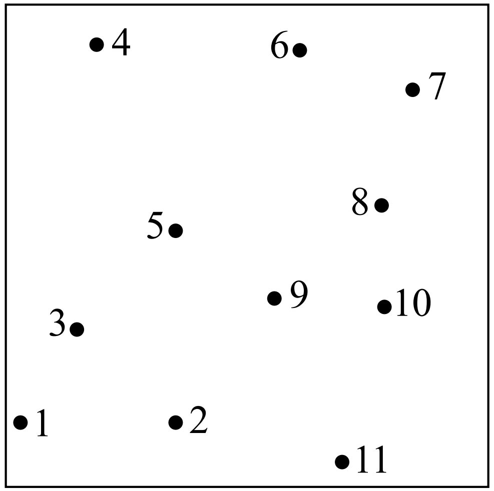
Can reduce the complexity of the search, e.g. O(log N) for k-d tree.

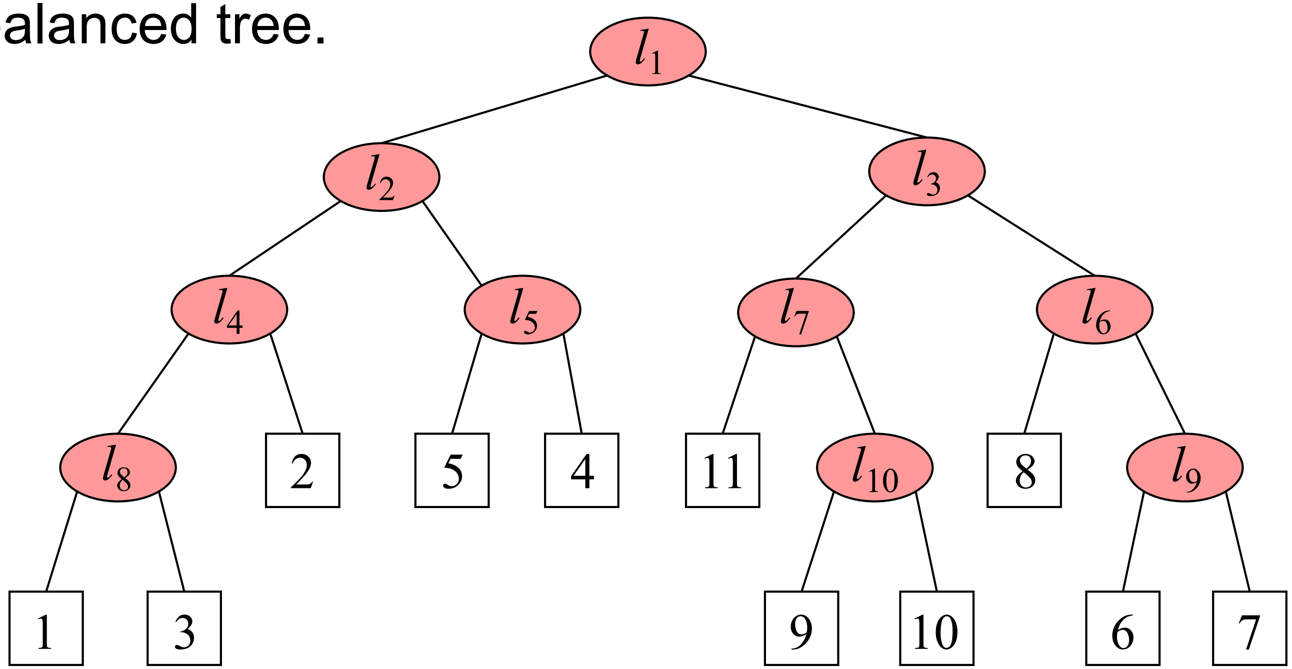
But at the cost of missing some nearest matches.

Adapted from K. Grauman, B. Leibe

K-d tree

- K-d tree is a binary tree data structure for organizing a set of points in a K-dimensional space.
- Each internal node is associated with an axis aligned hyper-plane splitting its associated points into two sub-trees.
- Dimensions with high variance are chosen first.
- Position of the splitting hyper-plane is chosen as the mean/median of the projected points – balanced tree.

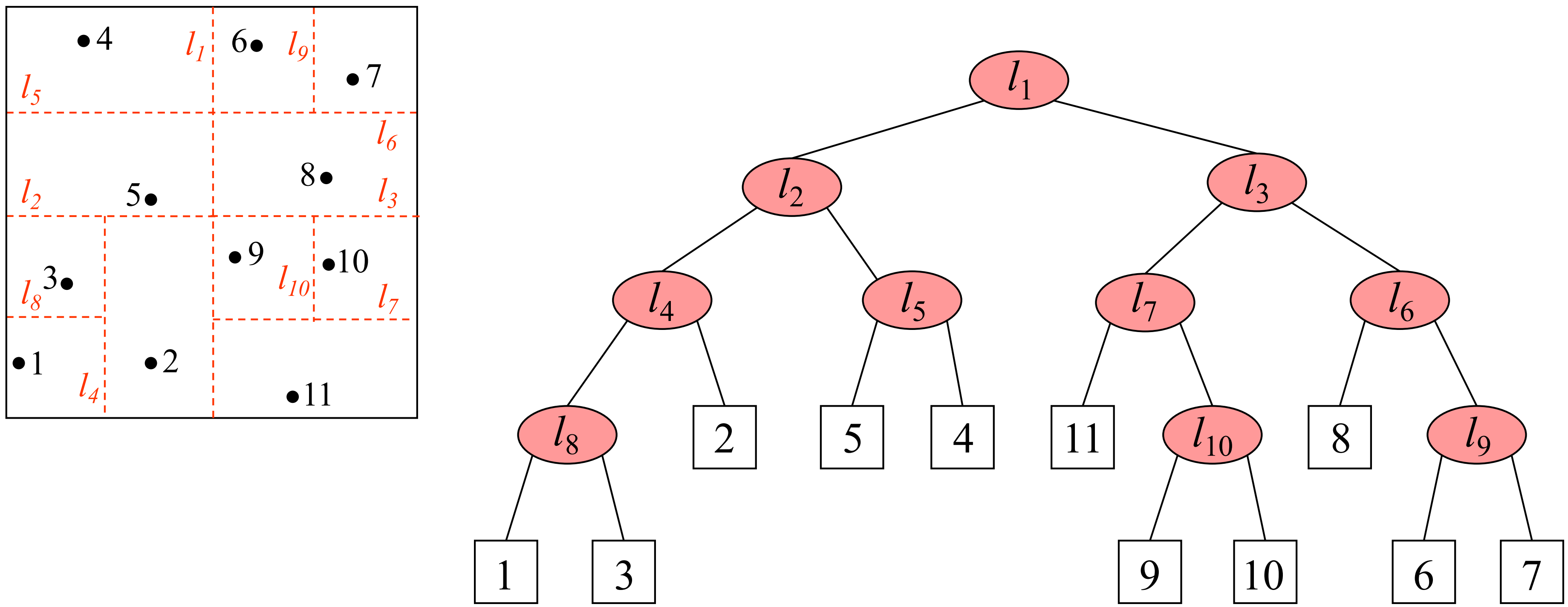




Images: Anna Atramentov

K-d tree construction

Simple 2D example

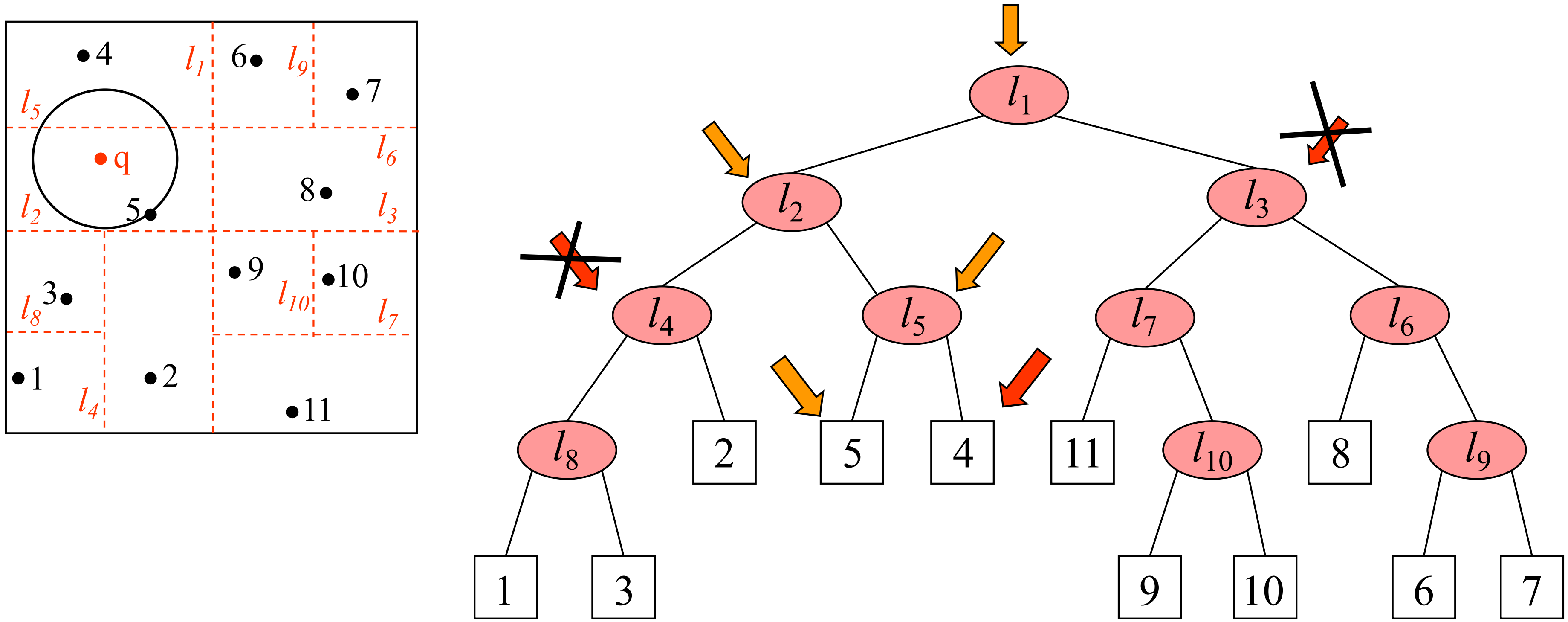


Slide credit: Anna Atramentov

KD-tree: Properties

- Binary tree of depth O(log(n))
- Total nodes: (2n -1) (n-1 internal and n leaves)
- Construction time: O(n d log(n))
- Memory requirements:
  - nonleaf node – (dim, threshold)
  - Leaf node: data id
- Need to also store the original data

K-d tree query



Slide credit: Anna Atramentov

K-d tree: Backtracking

Backtracking is necessary as the true nearest neighbor may not lie in the query cell.

But in some cases, almost all cells need to be inspected.

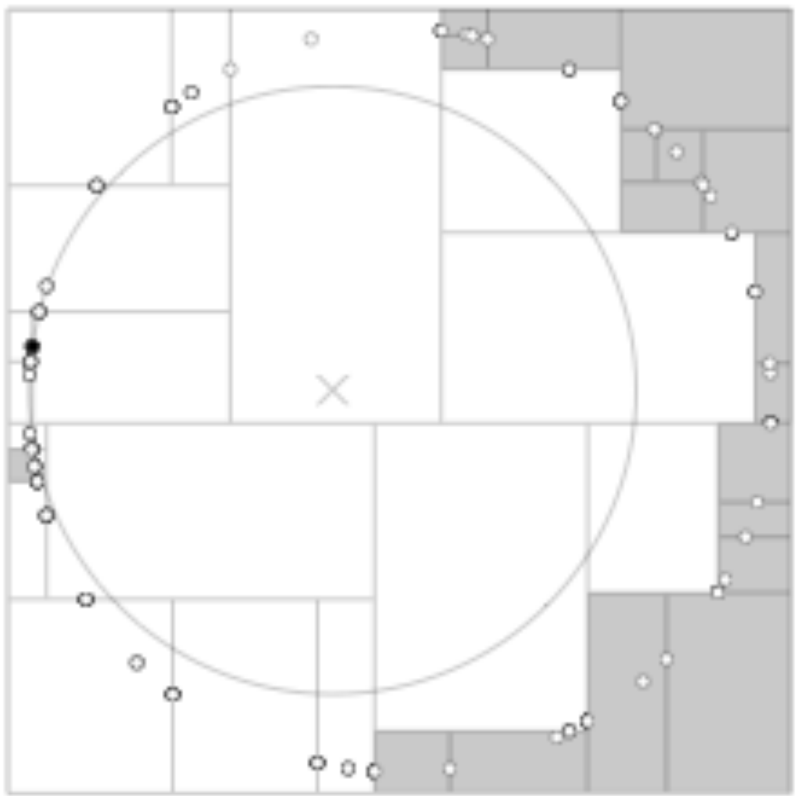


Figure 6.6

A bad distribution which forces almost all nodes to be inspected.

Figure: A. Moore

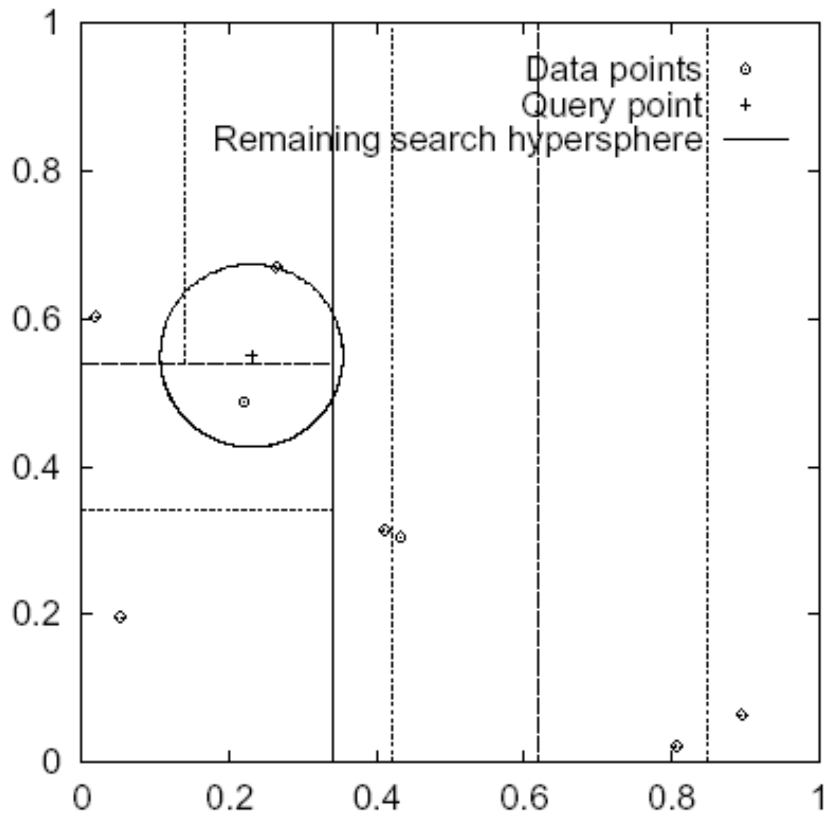
Solution: Approximate nearest neighbor K-d tree

Key ideas:

• Search k-d tree bins in order of distance from query

• Requires use of a priority queue

• Limit the number of neighbouring k-d tree bins to explore: only approximate NN is found

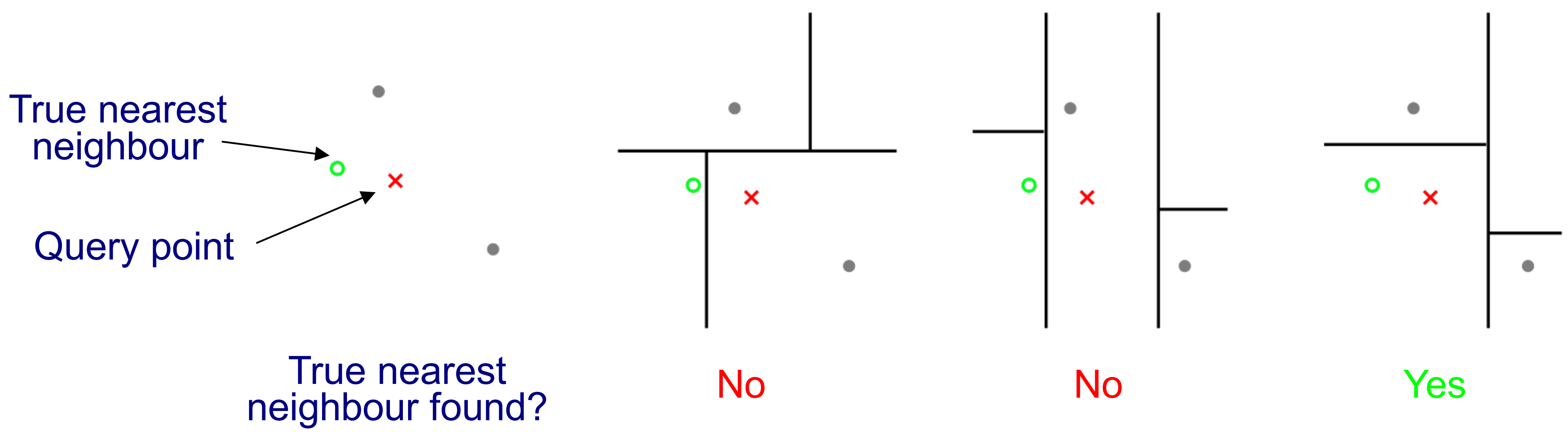


• Reduce the boundary effects by randomization

Randomized K-d trees

- How to choose the dimension to split and the splitting point?
  - Pick dimension with the highest variance
  - Split at the mean/median

- Multiple randomized trees increase the chances of finding nearby points



[Silpa-Anan and Hartley 2008, Philbin et al. 2007]

Approximate NN search using a randomized forest of K-d trees: Algorithm summary

1. Descent all (typically 8) trees to the leaf node

2. Search k-d tree bins in order of distance from query
   - Distance between the query and the bin is defined as the minimum distance between the query and any point on the bin boundary

   - Requires the use of a priority queue:
     - During lookup an entry is added to the priority queue about the option not taken
     - For multiple trees, the queue is shared among the trees

   - Limit the number of neighbouring K-d tree bins to explore (parameter of the algorithm, typically set to 512)

Randomized K-d trees: discussion

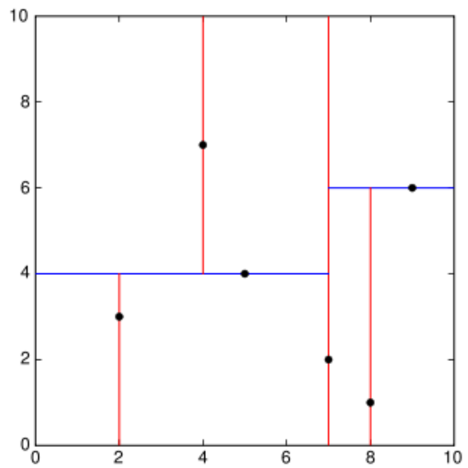
- Find approximate nearest neighbor in O(logN) time, where N is the number of data points.

- Increased memory requirements: needs to store multiple (~8) trees

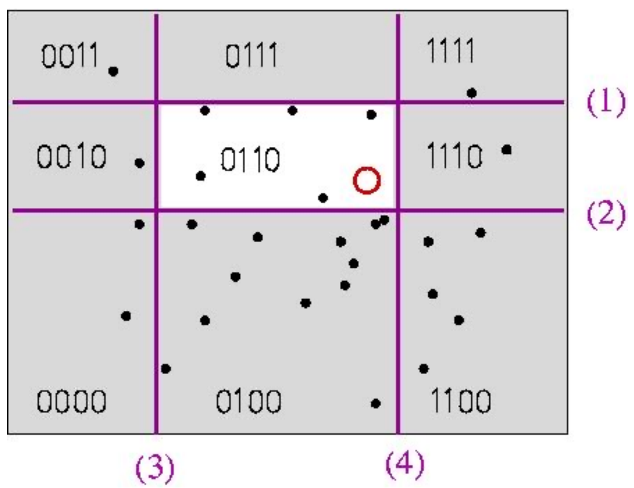
- Good performance in practice for recognition problems (NN-search for SIFT descriptors and image patches).

- Code available online:
  http://people.cs.ubc.ca/~mariusm/index.php/FLANN/FLANN

Indexing local features: approximate nearest neighbor search



Best-Bin First (BBF), a variant of k-d trees that uses priority queue to examine most promising branches first [Beis & Lowe, CVPR 1997]



Locality-Sensitive Hashing (LSH), a randomized hashing technique using hash functions that map similar points to the same bin, with high probability [Indyk & Motwani, 1998]

Locality Sensitive Hashing (LSH)

Idea: construct hash functions $g: R^d \rightarrow Z^k$ such that

for any points p,q:

If $||p-q|| \leq r$, then $Pr[g(p)=g(q)]$ is "high" or "not-so-small"
If $||p-q|| > cr$, then $Pr[g(p)=g(q)]$ is "small"

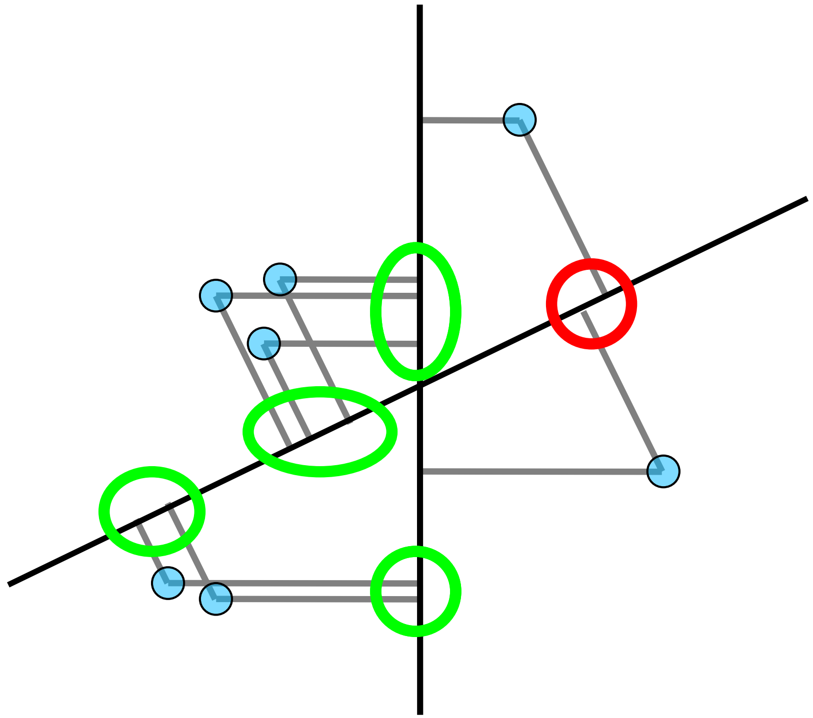
Example of g: linear projections

$g(p)=<h_1(p),h_2(p),\ldots,h_k(p)>$, where $h_{X,b}(p)=\lfloor(p*X+b)/w\rfloor$

$\lfloor . \rfloor$ is the "floor" operator.
X_i are sampled from a Gaussian.
w is the width of each quantization bin.
b is sampled from uniform distr. $[0,w]$. [Datar-Immorlica-Indyk-Mirrokni'04]

Locality Sensitive Hashing (LSH)

- Choose a random projection
- Project points
- Points close in the original space remain close under the projection
- Unfortunately, converse not true



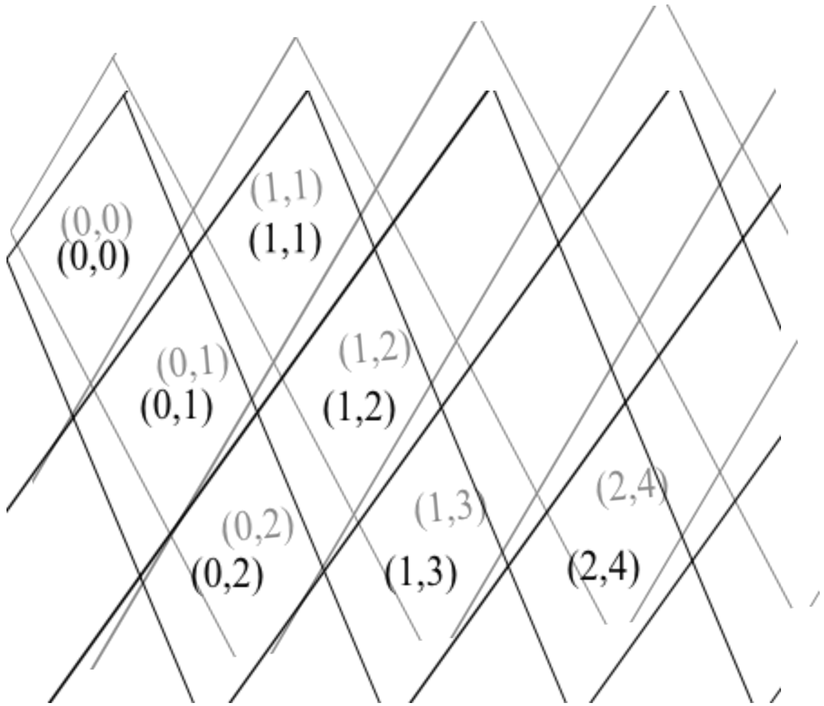
- Answer: use multiple quantized projections which define a high-dimensional “grid”

Slide: Philbin, Chum, Isard, Zissrman

Locality Sensitive Hashing (LSH)

- Cell contents can be efficiently indexed using a hash table

- Repeat to avoid quantization errors near the cell boundaries



- Point that shares at least one cell = potential candidate

- Compute distance to all candidates

Slide: Philbin, Chum, Isard, Zissrman

LSH: discussion

In theory, query time is O(kL), where k is the number of projections and L is the number of hash tables, i.e. independent of the number of points, N.

In practice, LSH has high memory requirements as large number of projections/hash tables are needed.

Code and more materials available online:
http://www.mit.edu/~andoni/LSH/

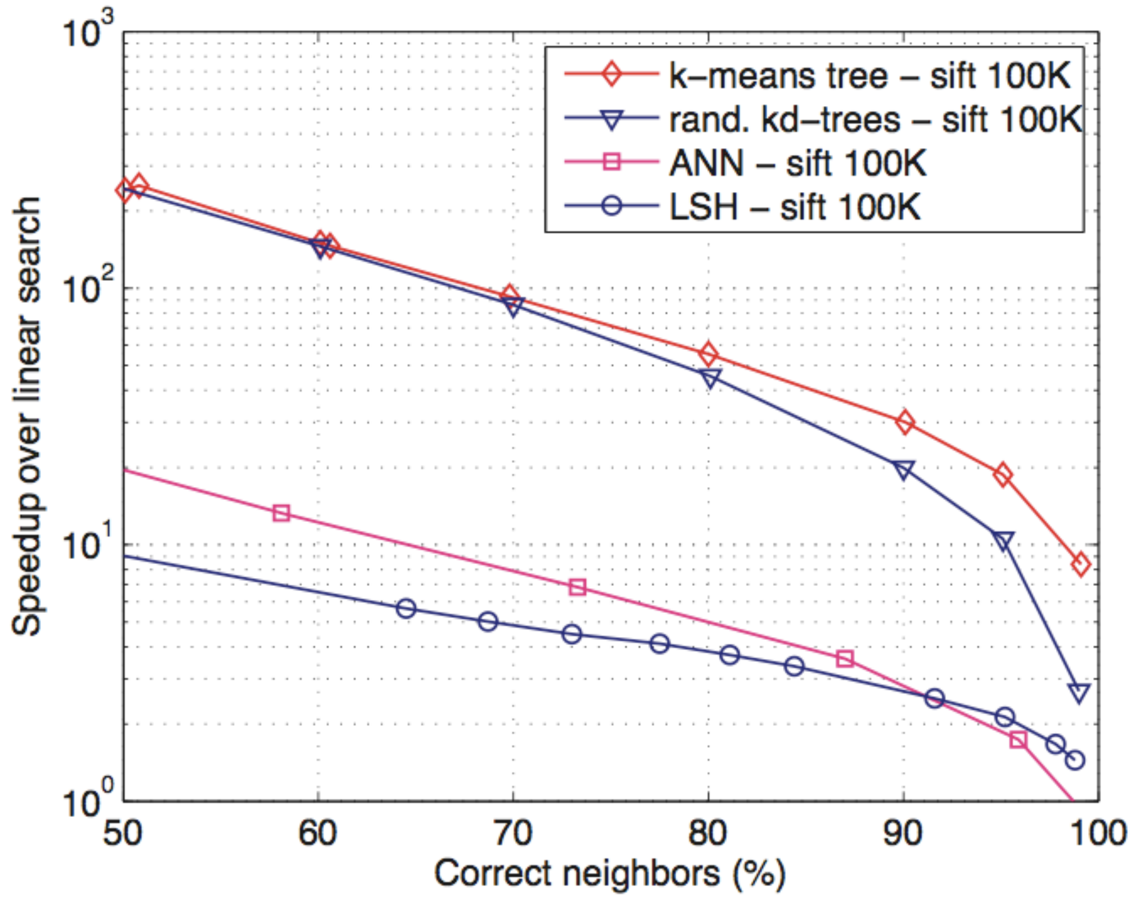
Hashing functions could be also data-dependent (PCA) or learnt from labeled point pairs (close/far).

Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *NIPS, 2008.*
R. Salakhutdinov and G. Hinton, "Semantic Hashing," ACM SIGIR, 2007.

See also:
http://cobweb.ecn.purdue.edu/~malcolm/yahoo Slaney2008(LSHTutorialDraft).pdf
http://www.sanjivk.com/EECS6898/ApproxNearestNeighbors_2.pdf

Comparison of approximate NN-search methods

Dataset: 100K SIFT descriptors



Code for all methods available online, see Muja&Lowe'09

Figure: Muja&Lowe'09

Approximate nearest neighbour search (references)

J. L. Bentley. Multidimensional binary search trees used for associative searching. Comm. ACM, 18(9), 1975.

Freidman, J. H., Bentley, J. L., and Finkel, R. A. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw., 3:209–226, 1977.*

Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R., and Wu, A. Y. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *Journal of the ACM, 45:891–923, 1998.*

C. Silpa-Anan and R. Hartley. Optimised KD-trees for fast image descriptor matching. In CVPR, 2008.

M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In VISAPP, 2009.

P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proc. of 30th ACM Symposium on Theory of Computing, 1998*

G. Shakhnarovich, P. Viola, and T. Darrell, "Fast pose estimation with parameter-sensitive hashing," in *Proc. of the IEEE International Conference on Computer Vision, 2003.*

R. Salakhutdinov and G. Hinton, "Semantic Hashing," ACM SIGIR, 2007.

Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *NIPS, 2008.*

ANN - search (references continued)

O. Chum, J. Philbin, and A. Zisserman. Near duplicate image detection: min-hash and tf-idf weighting. BMVC., 2008.

B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing for scalable image search," *Proc. of the IEEE International Conference on Computer Vision, 2009.*

J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for scalable image retrieval," *in CVPR, 2010.*

H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *PAMI*, 2011.

A.Gordo and F.Perronnin. Asymmetric distances for binary embeddings. CVPR, 2011.

Y. Gong and S. Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. CVPR, *2011.*

A. Babenko and V. Lempitsky. The inverted multi-index. CVPR, 2012.

T. Ge, K. He, Q. Ke, and J. Sun. Optimized product quantization for approximate nearest neighbor search. CVPR, 2013.

T. Norouzi and D. Fleet, Cartesian k-means., CVPR, 2013

See tutorial at CVPR'13 by H. Jegou: https://sites.google.com/site/lsvr13

Code: https://github.com/facebookresearch/faiss

Outline – Efficient visual search

1. Efficient matching of local descriptors
   - Approximate nearest neighbor search
   - k-d trees, locality-sensitive hashing (LSH)
2. Aggregate local descriptors into a single vector
   - Bag-of-visual-words, inverted files, query expansion
3. Compact representations for very large-scale search
   - Product quantization (PQ)
4. Learnable representations
   - Neural representations for large-scale visual search
   - Visual search using natural language query

So far …

- Linear exhaustive search can be prohibitively expensive for large image collections

- Answer (so far): approximate NN search methods
  - Randomized KD-trees
  - Locality sensitive hashing

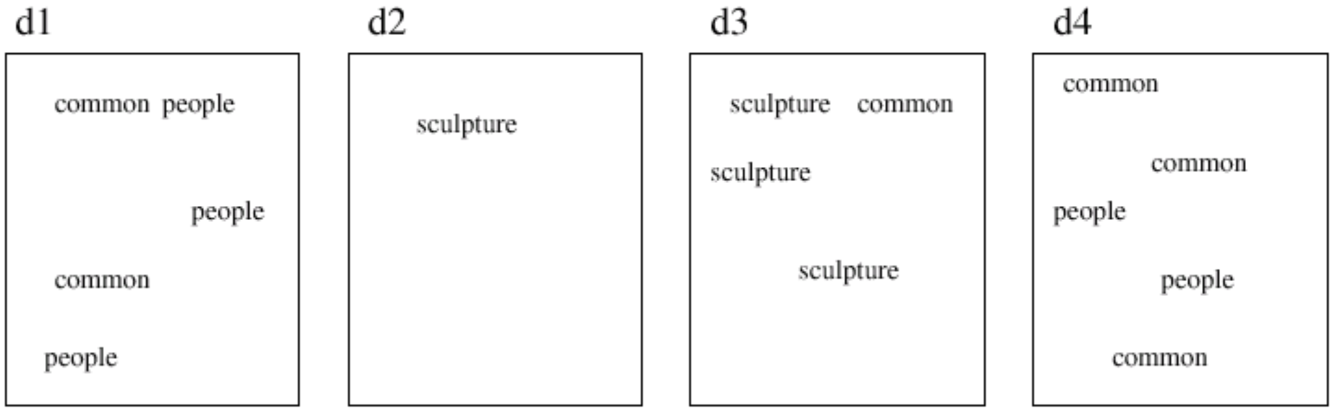
- However, memory footprint can be still high.

  Example: N = 10^7 images, 10^{10} SIFT features with 128B per feature $\Longrightarrow$ 1TB of memory

  Look how text-based search engines (Google) index documents – **inverted files**.

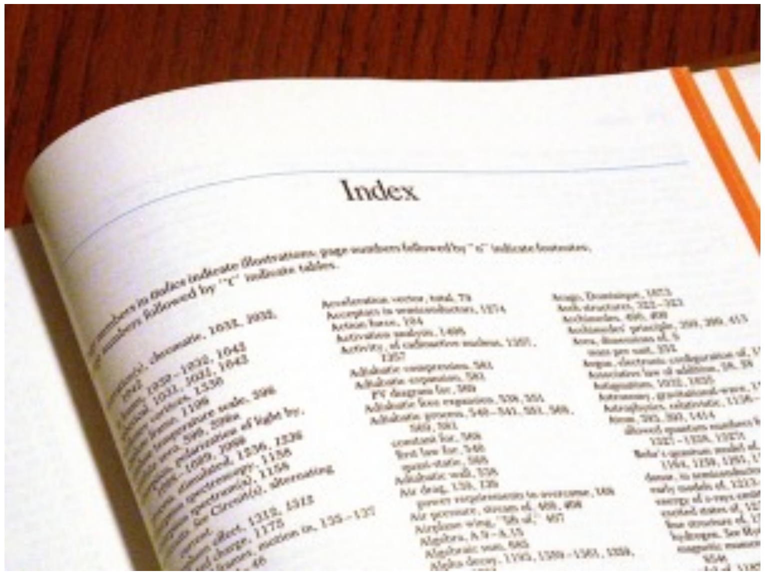
Indexing text with inverted files

Document collection:



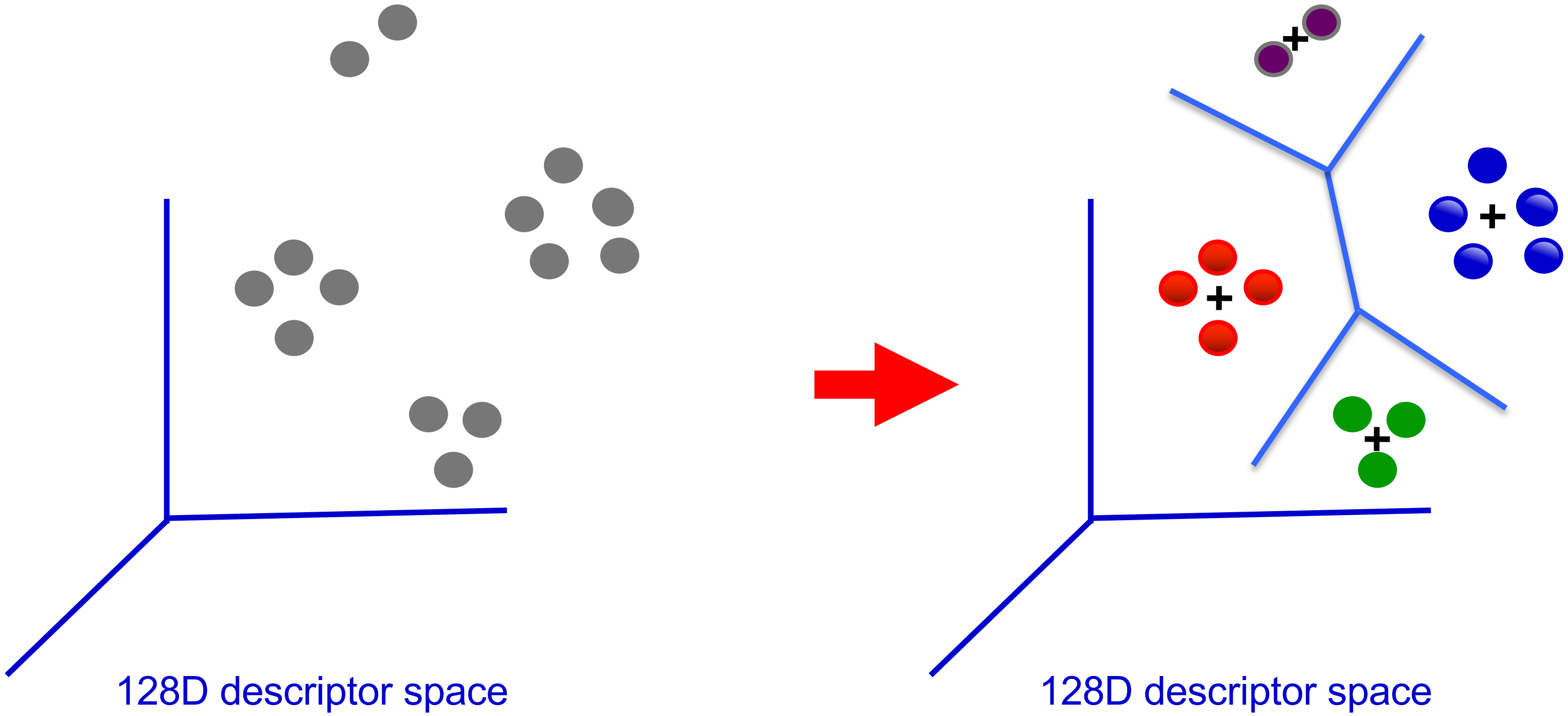
Inverted file:

Term	**List of hits** (occurrences in documents)
People	[d1:hit hit hit], [d4:hit hit] …
Common	[d1:hit hit], [d3: hit], [d4: hit hit hit] …
Sculpture	[d2:hit], [d3: hit hit hit] …



Need to map feature descriptors to “visual words”.

Build a visual vocabulary



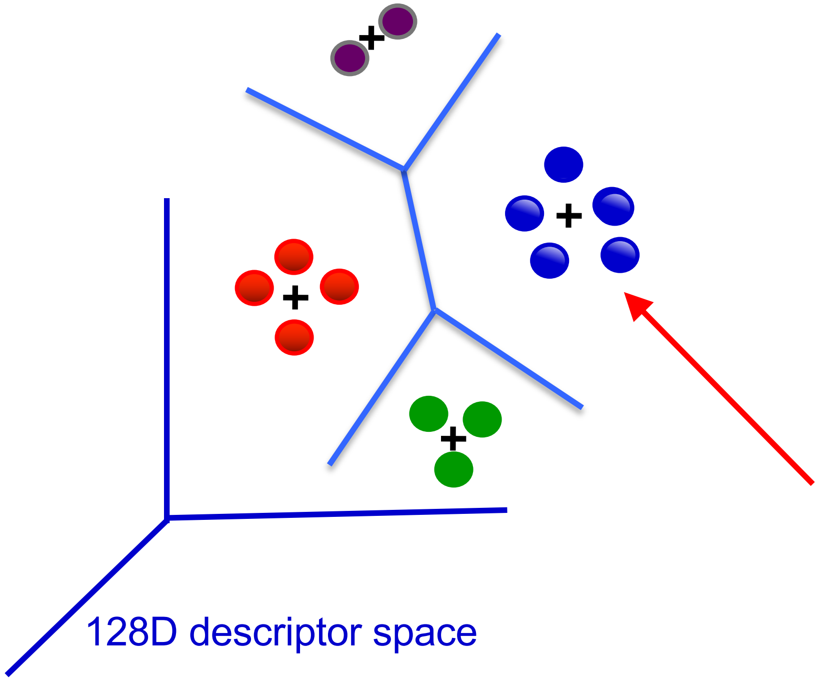
Vector quantize descriptors

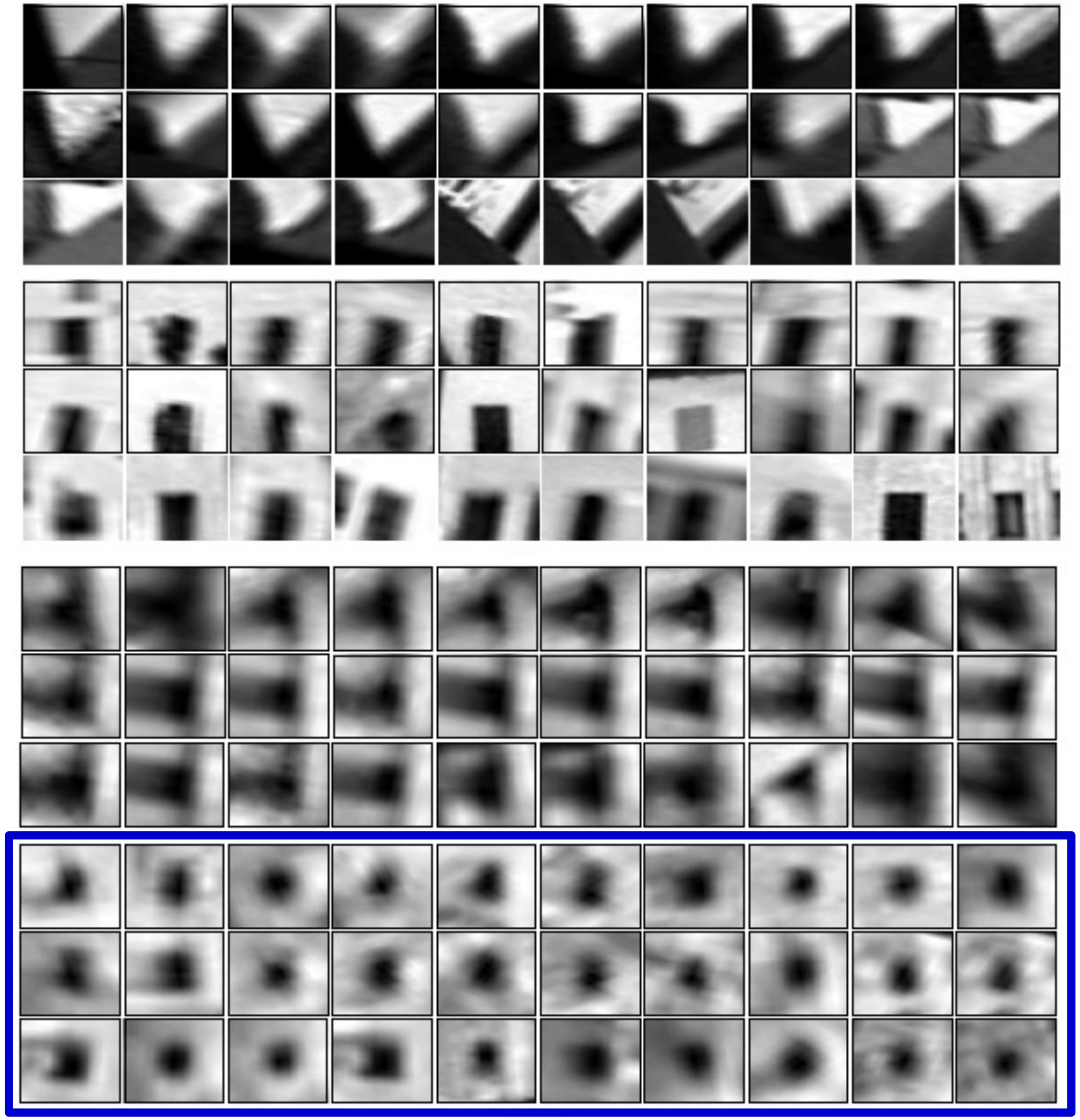
- Compute SIFT features from a subset of images
- K-means clustering (need to choose K)

[Sivic and Zisserman, ICCV 2003]

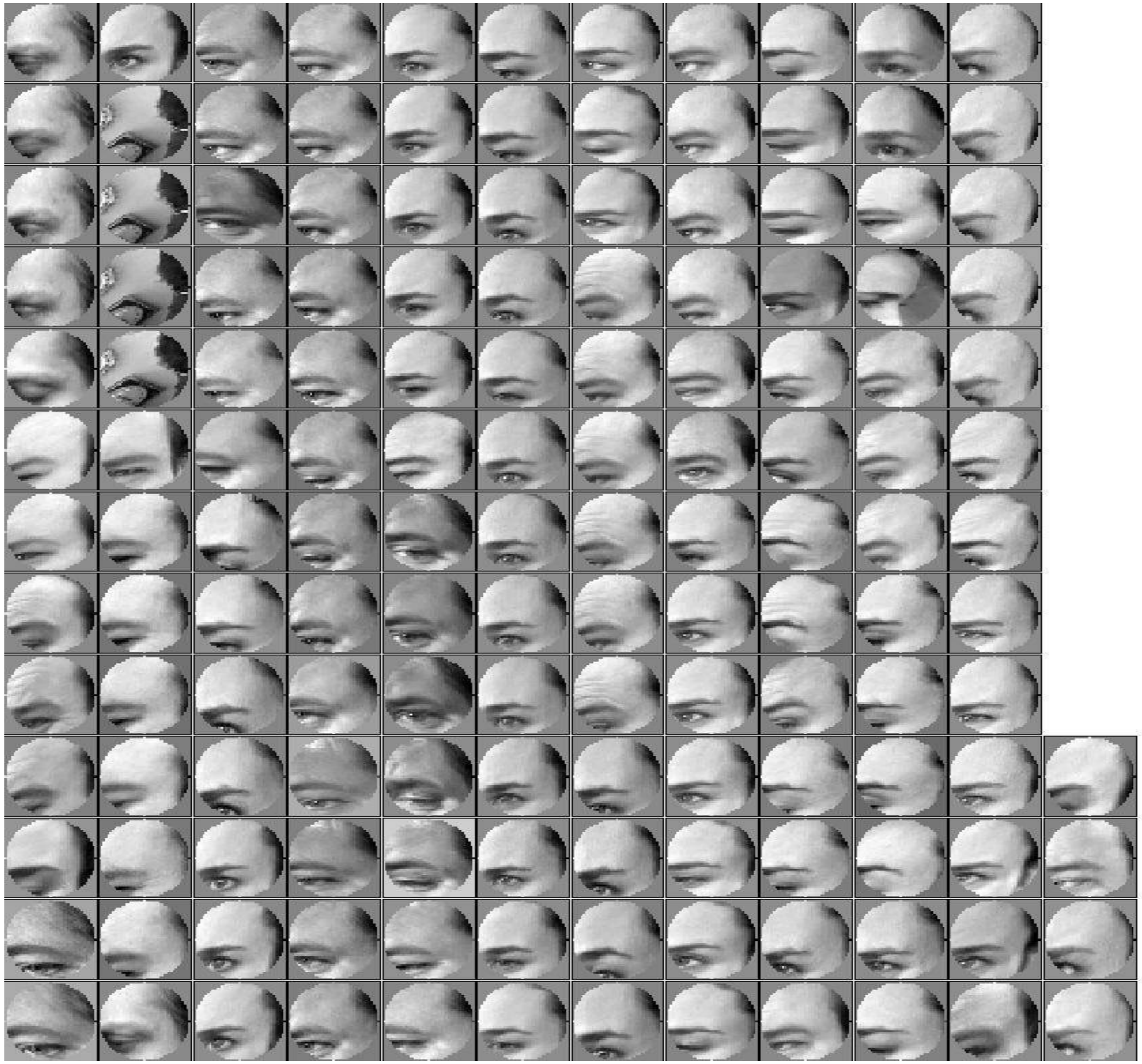
Visual words

Example: each group of patches belongs to the same visual word



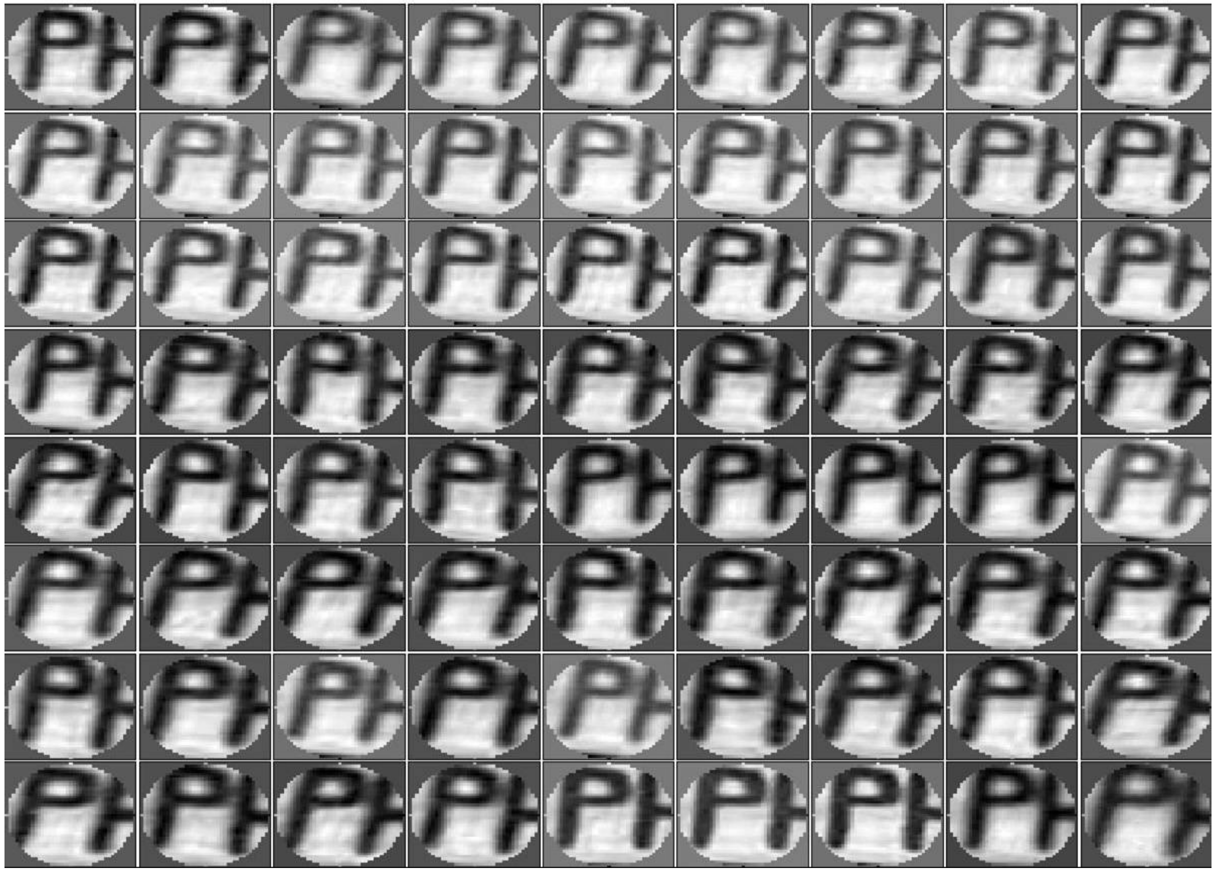


Samples of visual words (clusters on SIFT descriptors):



More specific example

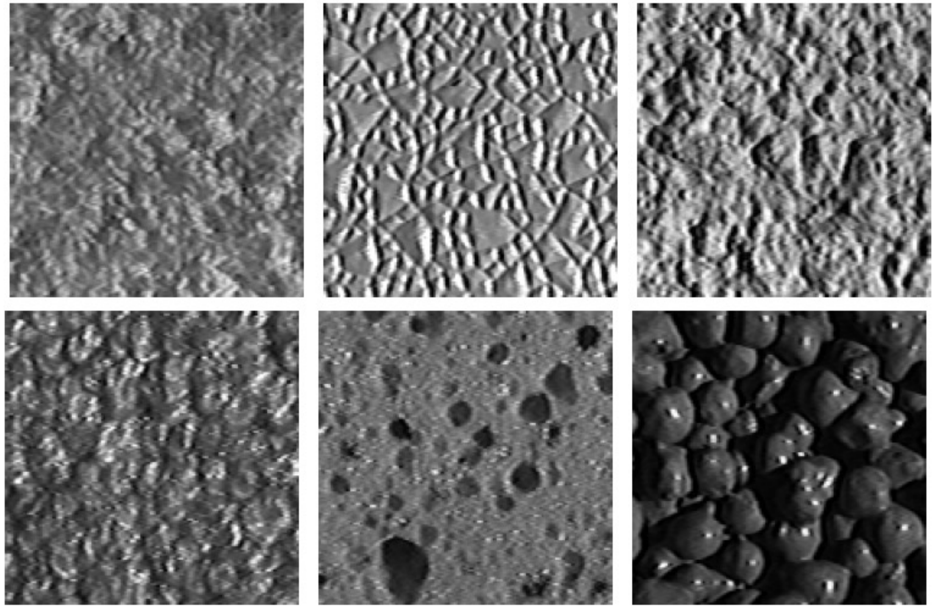
Samples of visual words (clusters on SIFT descriptors):



More specific example

Visual words

- First explored for texture and material representations
- *Texton* = cluster center of filter responses over collection of images
- Describe textures and materials based on distribution of prototypical texture elements.



Leung & Malik 1999; Varma & Zisserman, 2002; Lazebnik, Schmid & Ponce, 2003;

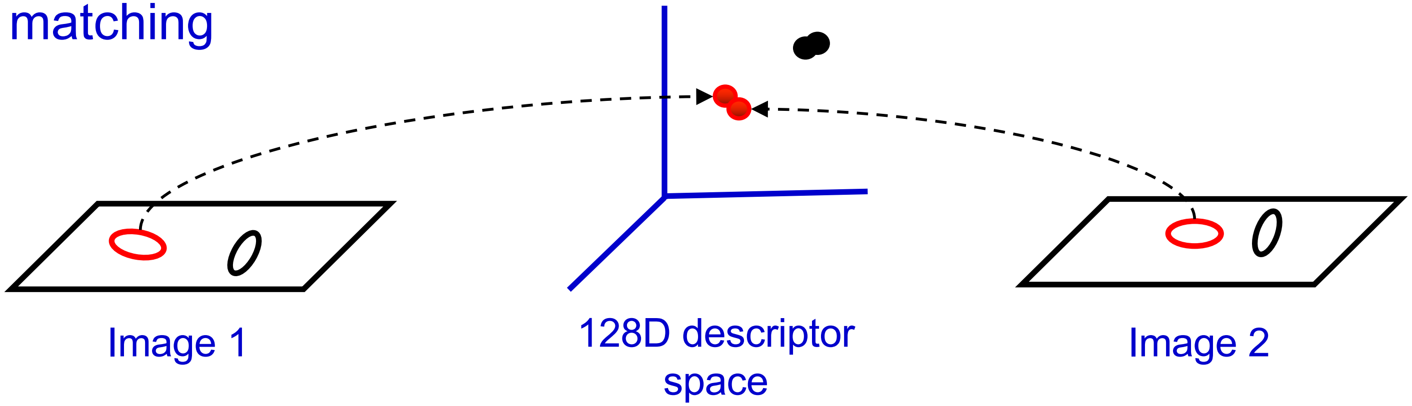
Slide: Grauman&Leibe

Visual words: quantize descriptor space

Sivic and Zisserman, ICCV 2003

Nearest neighbour matching

- expensive to do for all frames

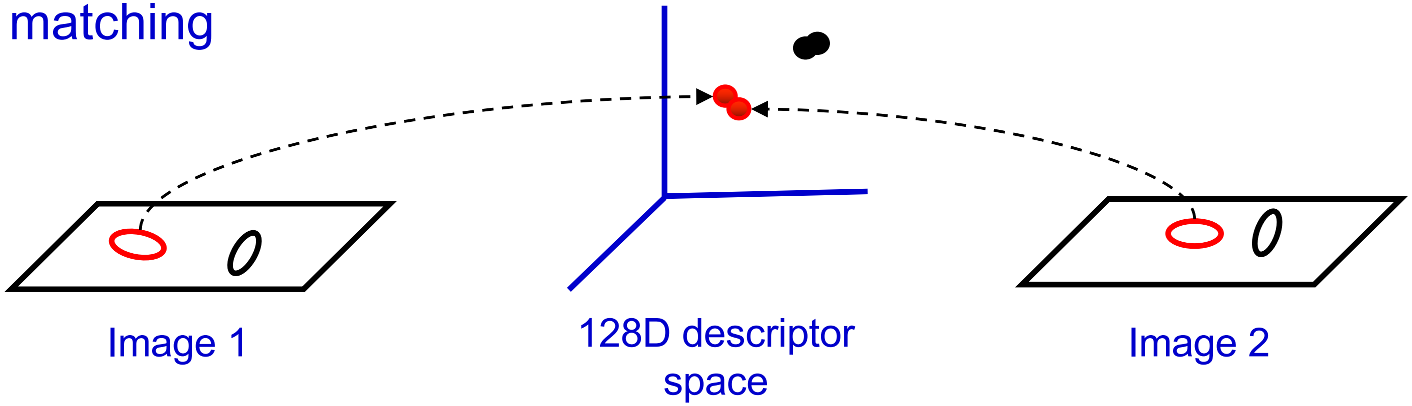


Visual words: quantize descriptor space

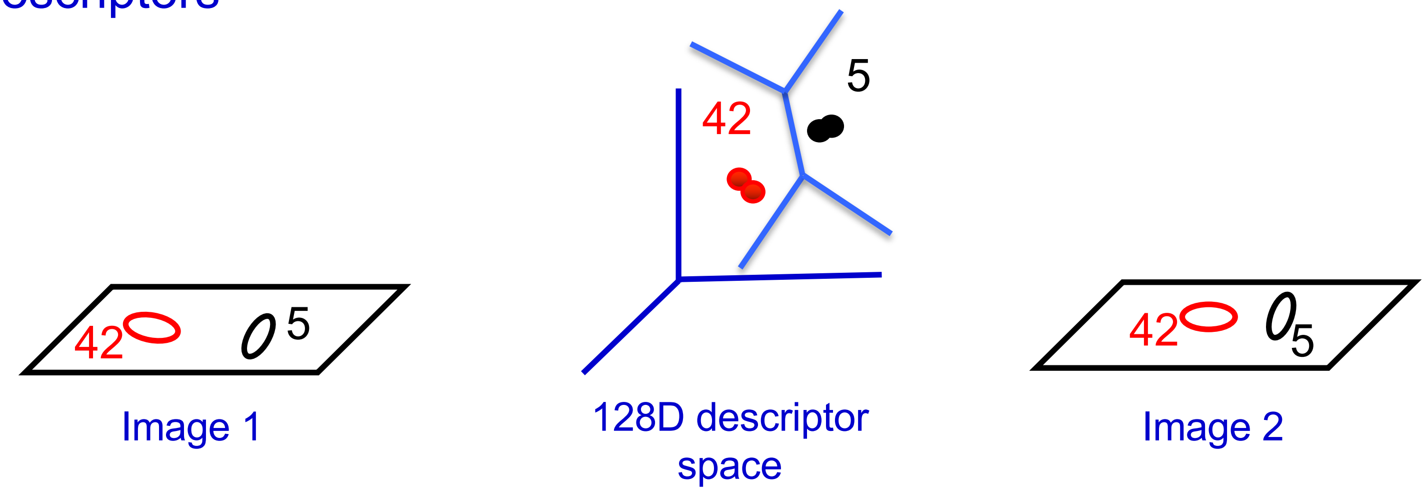
Sivic and Zisserman, ICCV 2003

Nearest neighbour matching

- expensive to do for all frames



Vector quantize descriptors

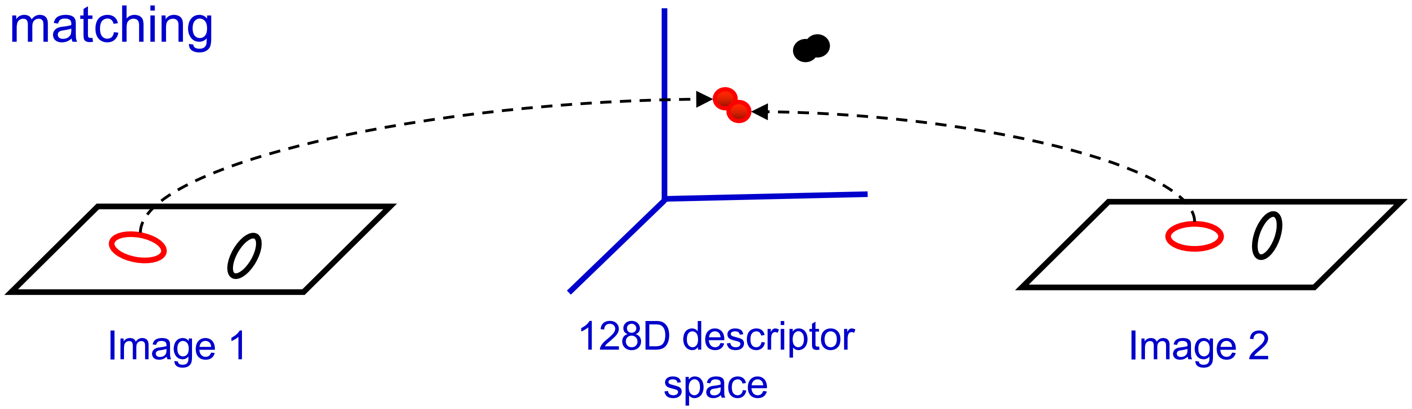


Visual words: quantize descriptor space

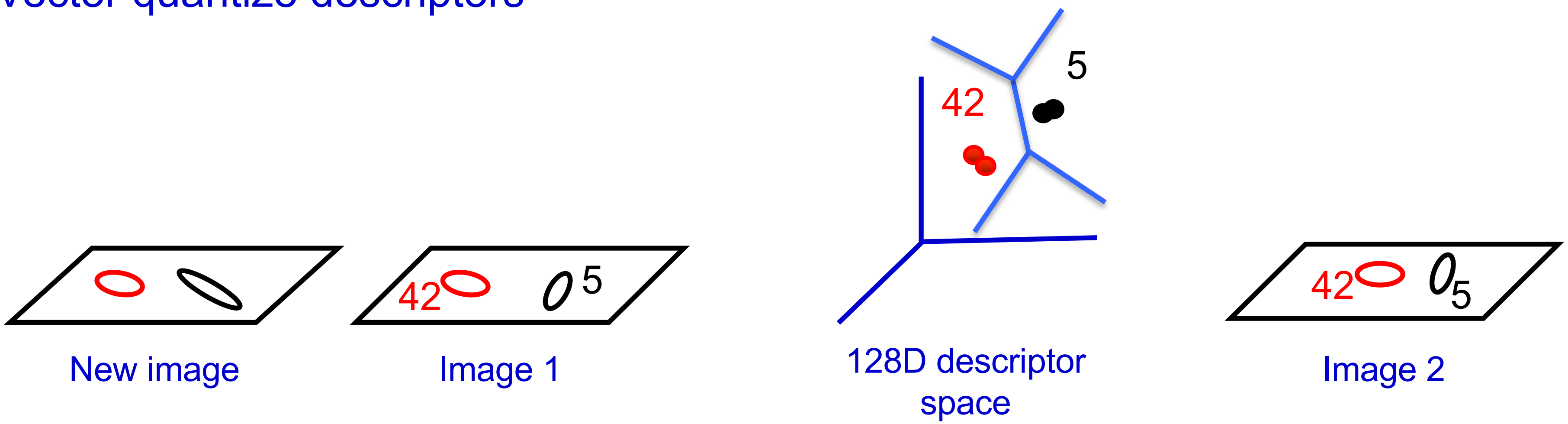
Sivic and Zisserman, ICCV 2003

Nearest neighbour matching

- expensive to do for all frames



Vector quantize descriptors

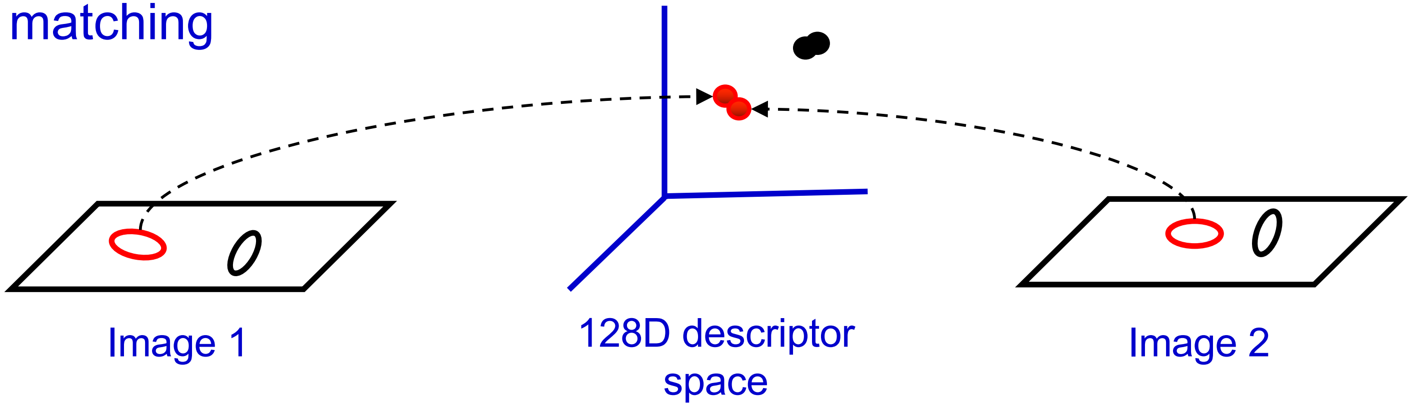


Visual words: quantize descriptor space

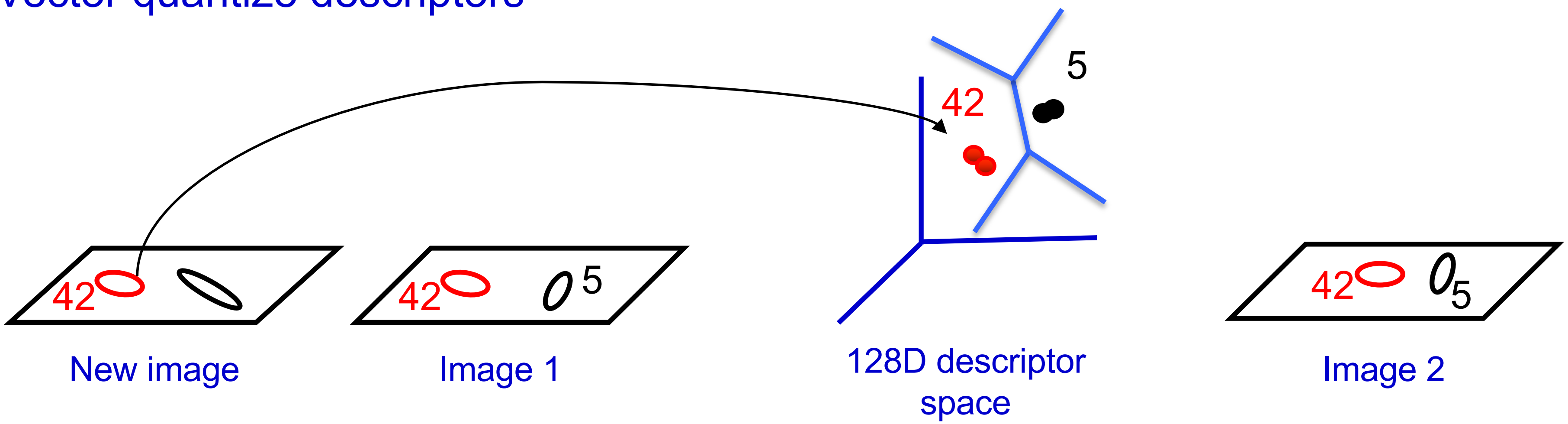
Sivic and Zisserman, ICCV 2003

Nearest neighbour matching

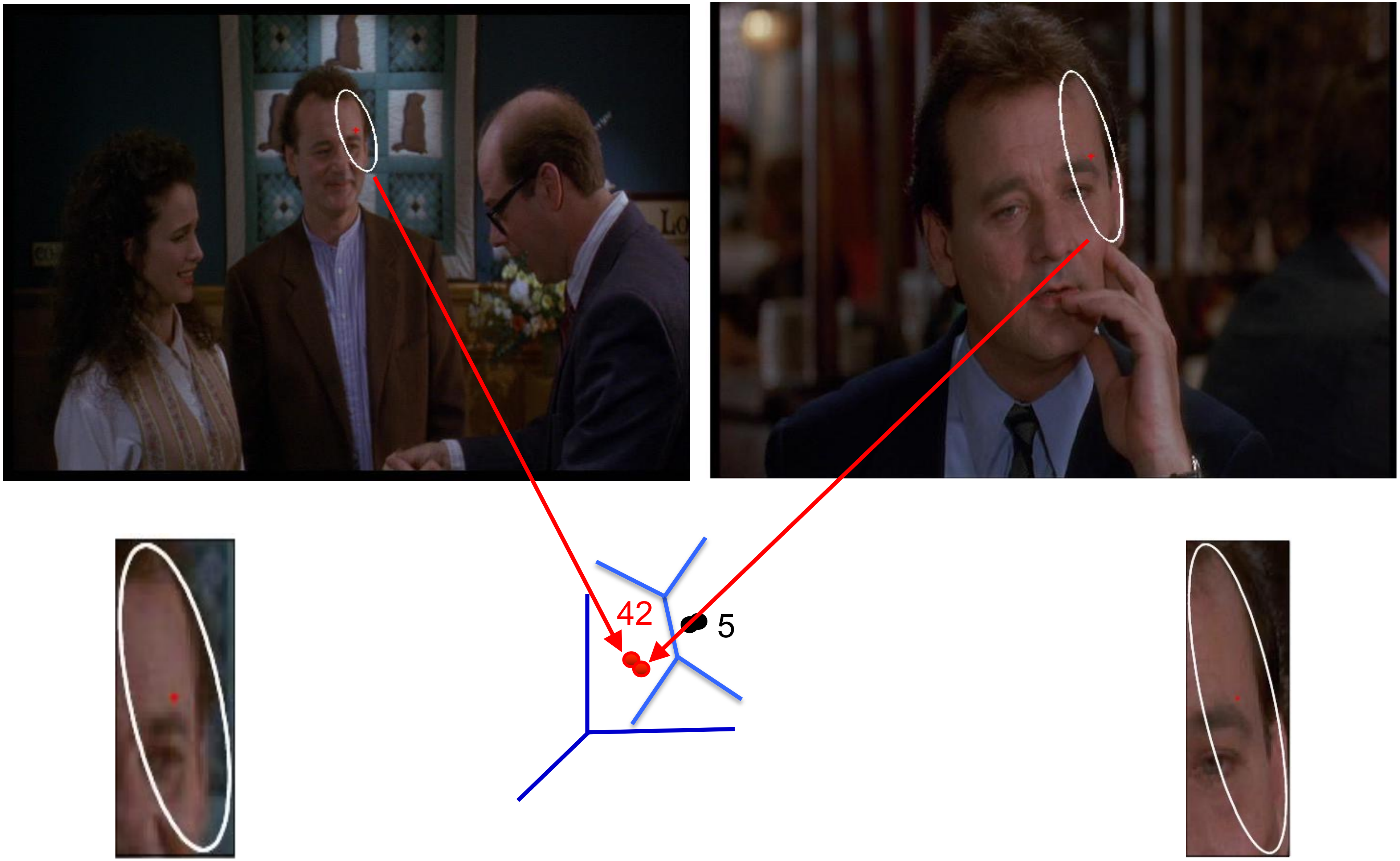
- expensive to do for all frames



Vector quantize descriptors



Vector quantize the descriptor space (SIFT)



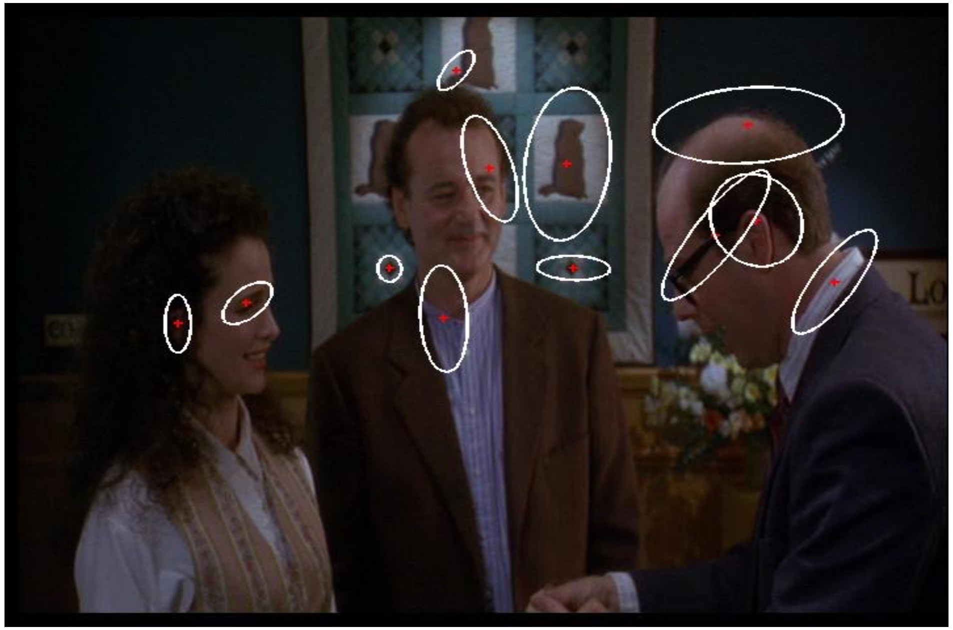
The same visual word

Representation: bag of (visual) words

Visual words are 'iconic' image patches or fragments

• represent their frequency of occurrence

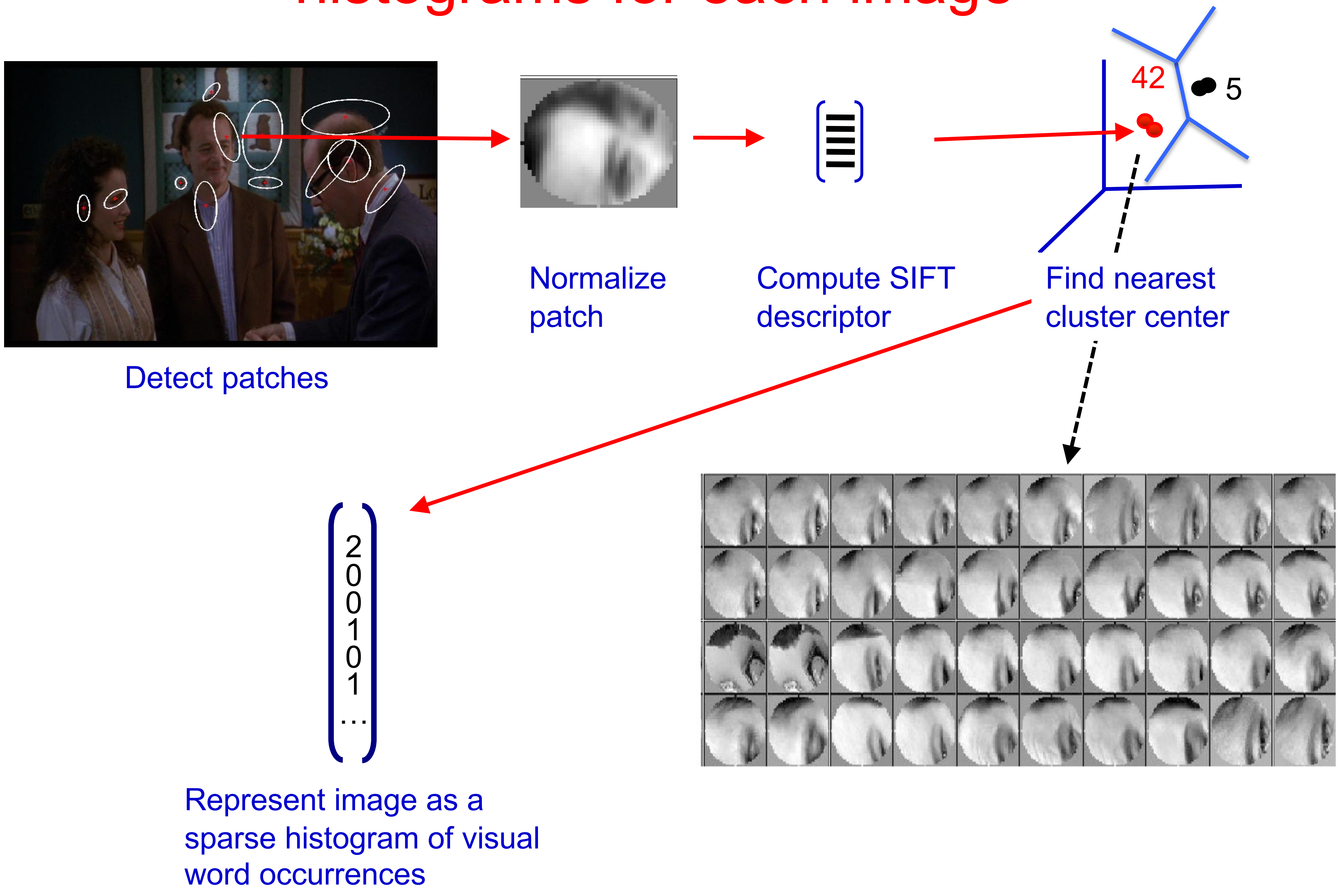
• but not their position



Image

Colelction of visual words

Offline: Assign visual words and compute histograms for each image



Offline: create an index

frame #5

frame #10

Word number	Posting list
1	5,10, ...
2	10,...
...	...

- For fast search, store a "posting list" for the dataset
- This maps visual word occurrences to the images they occur in

(i.e. like the "book index")

At run time

frame #5

frame #10

Word number		Posting list
1	→	5,10, ...
2	→	10,...
...		...

- User specifies a query region
- Generate a short-list of images using visual words in the region
  1. Accumulate all visual words within the query region
  2. Use “book index” to find other frames with these words
  3. Compute similarity for images which share at least on word

At run time

frame #5

frame #10

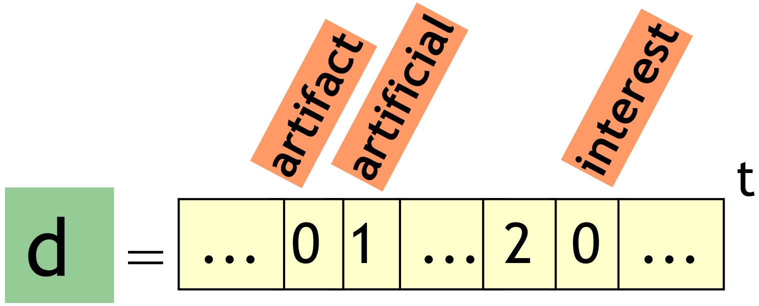
Word number	Posting list
1	5,10, ...
2	10,...
...	...

- Score each image by the (weighted) number of common visual words (tentative correspondences)
- Worst case complexity is linear in the number of images N
- In practice, it is linear in the length of the lists (<< N)

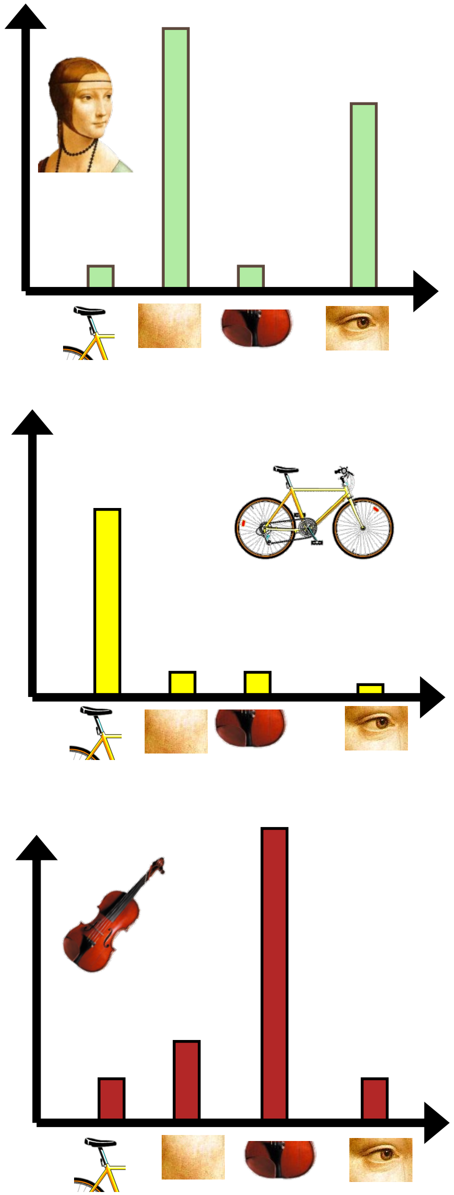
Bags of visual words

Summarize entire image based on its distribution (histogram) of visual word occurrences.

Analogous to bag of words representation commonly used for text documents.



Hofmann 2001



Slide: Grauman&Leibe, Image: L. Fei-Fei

Another interpretation: the bag-of-visual-words model

For a vocabulary of size K, each image is represented by a K-vector

$$\mathbf{v}_d = (t_1, \ldots, t_i, \ldots, t_K)^\top$$

where t_i is the number of occurrences of visual word i.

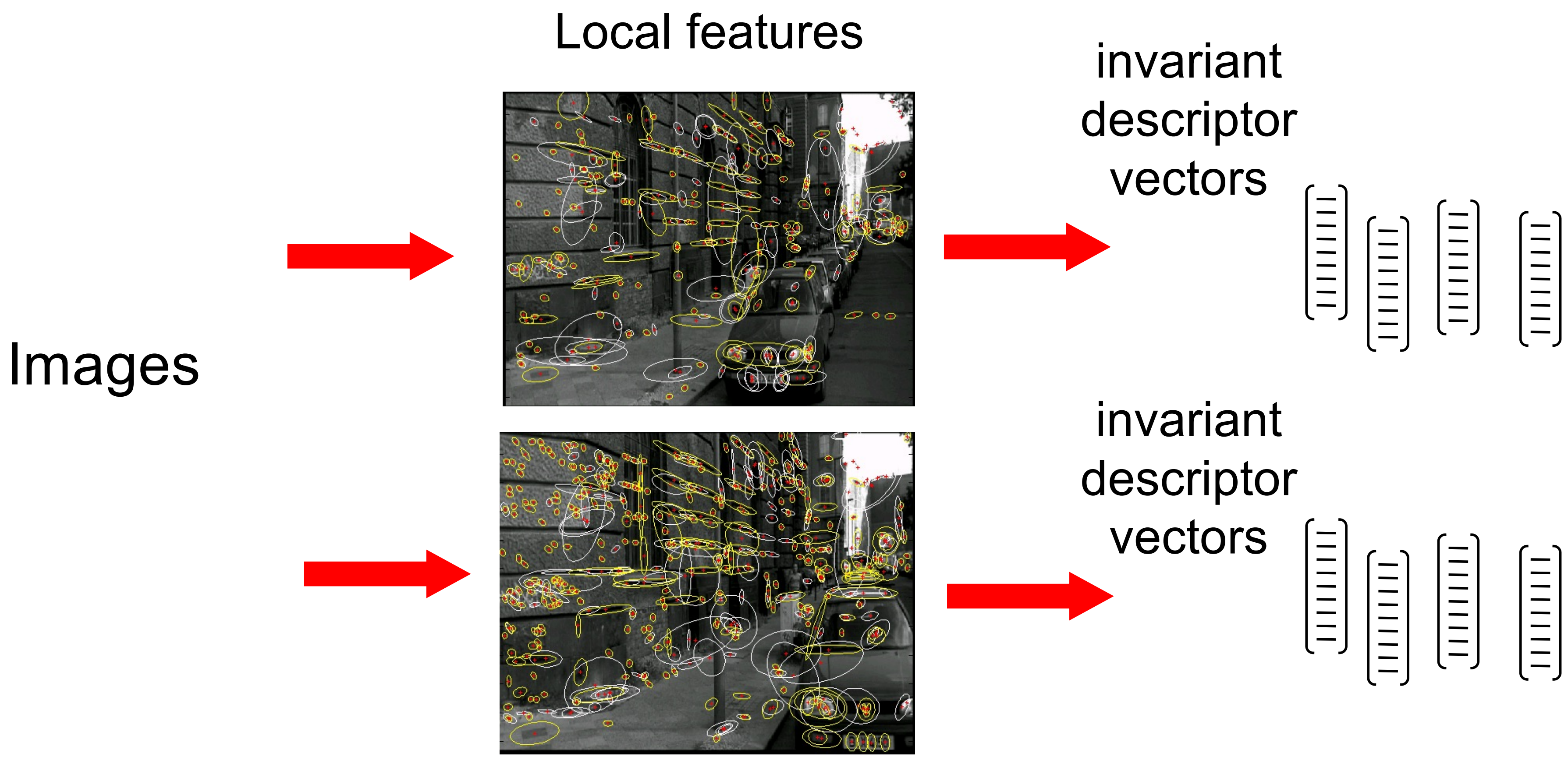
Images are ranked by the normalized scalar product between the query vector v_q and all vectors in the database v_d:

$$f_d = \frac{\mathbf{v}_q{}^\top \mathbf{v}_d}{\|\mathbf{v}_q\|_2 \; \|\mathbf{v}_d\|_2}$$

Scalar product can be computed efficiently using inverted file.

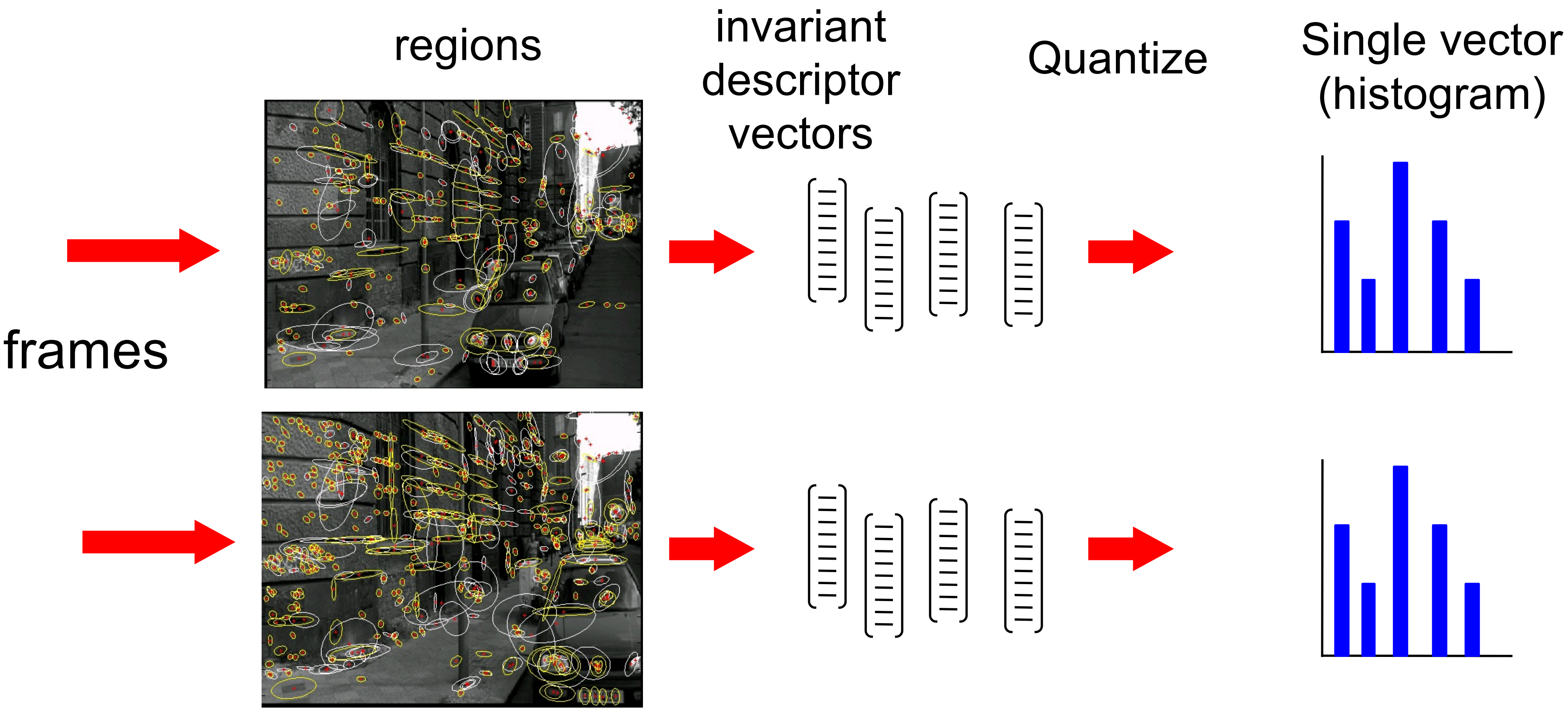
What if vectors are binary? What is the meaning of $\mathbf{v}_q{}^\top \mathbf{v}_d$?

Strategy I: Efficient approximate NN search



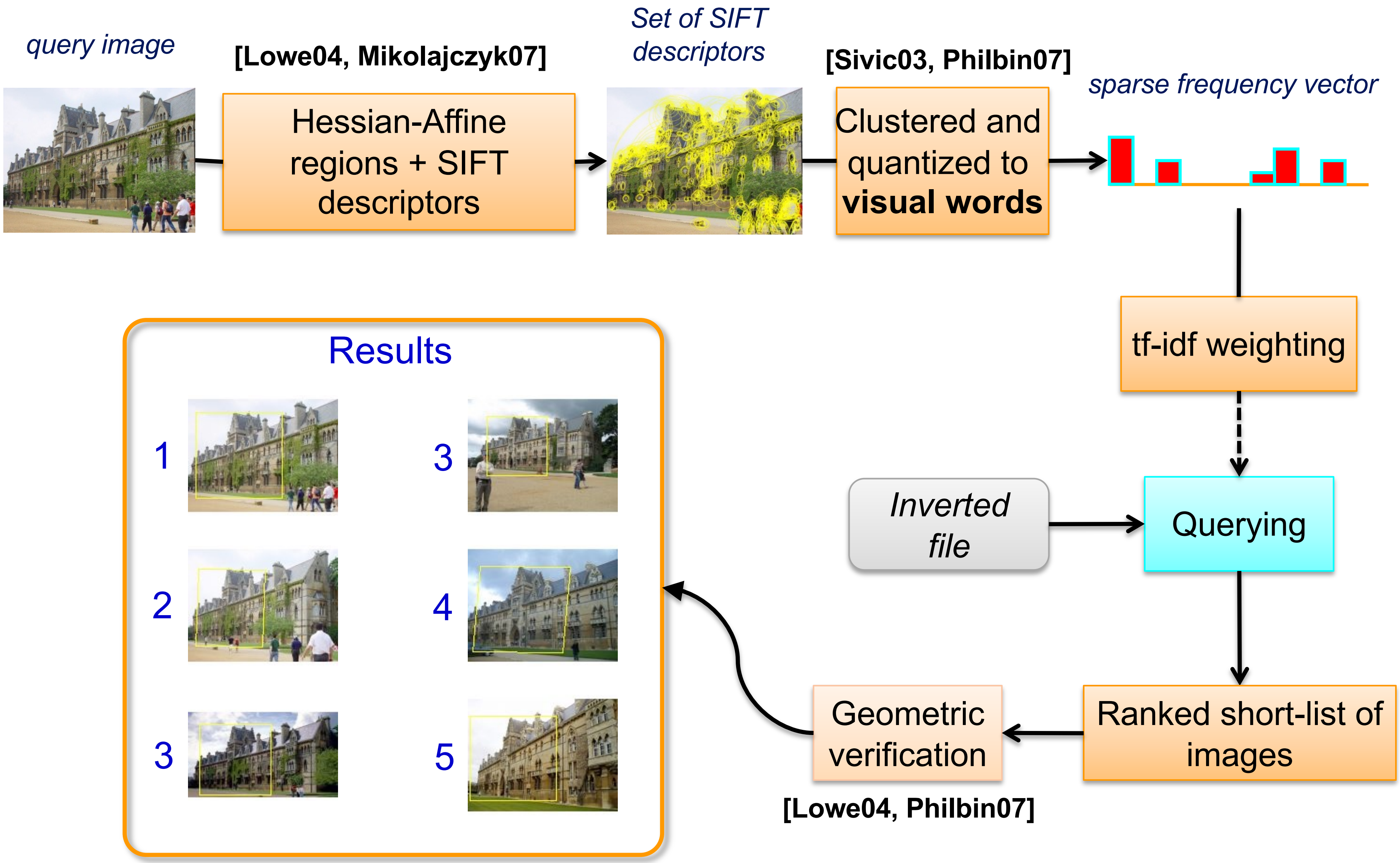
1. Compute local features in each image independently (offline)
2. “Label” each feature by a descriptor vector based on its intensity (offline)
3. Finding corresponding features is transformed to finding nearest neighbour vectors
4. Rank matched images by number of (tentatively) corresponding regions
5. Verify top ranked images based on spatial consistency.

Strategy II: Match histograms of visual words



1. Compute affine covariant regions in each frame independently (offline)
2. “Label” each region by a vector of descriptors based on its intensity (offline)
3. **Build histograms of visual words by descriptor quantization (offline)**
4. **Rank retrieved frames by matching vis. word histograms using inverted files.**
5. Verify retrieved frame based on spatial consistency.

Overview of the retrieval system



Visual search using local regions (references)

C. Schmid, R. Mohr, Local Greyvalue Invariants for Image Retrieval, PAMI, 1997

J. Sivic, A. Zisserman, Text retrieval approach to object matching in videos, ICCV, 2003

D. Nister, H. Stewenius, Scalable Recognition with a Vocabulary Tree, CVPR, 2006.

J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, CVPR, 2007

O. Chum, J. Philbin, M. Isard, J. Sivic, A. Zisserman, Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval, ICCV, 2007

H. Jegou, M. Douze, C. Schmid, Hamming embedding and weak geometric consistency for large scale image search, ECCV’2008

O. Chum, M. Perdoch, J. Matas: Geometric min-Hashing: Finding a (Thick) Needle in a Haystack, CVPR 2009

H. Jégou, M. Douze and C. Schmid, On the burstiness of visual elements, CVPR, 2009

Visual search using local regions (references)

T. Turcot and D. G. Lowe. Better matching with fewer features: The selection of useful features in large database recognition problems. In ICCV Workshop on Emergent Issues in Large Amounts of Visual Data (WS-LAVD), 2009.

H. Jégou, M. Douze, C. Schmid and P. Pérez, Aggregating local descriptors into a compact image representation, CVPR 2010

A. Mikulík, M. Perdoch, O. Chum, J. Matas, Learning a fine vocabulary, ECCV 2010.

O. Chum, A. Mikulik, M. Perdoch, J. Matas, Total recall II: Query expansion revisited, CVPR 2011

D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Van Gool. Hello neighbor: accurate object retrieval with k-reciprocal nearest neighbors. CVPR, 2011.

R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR,* 2012.

R. Arandjelović, A. Zisserman. DisLocation: Scalable descriptor distinctiveness for location recognition, In Asian Conference on Computer Vision, 2014

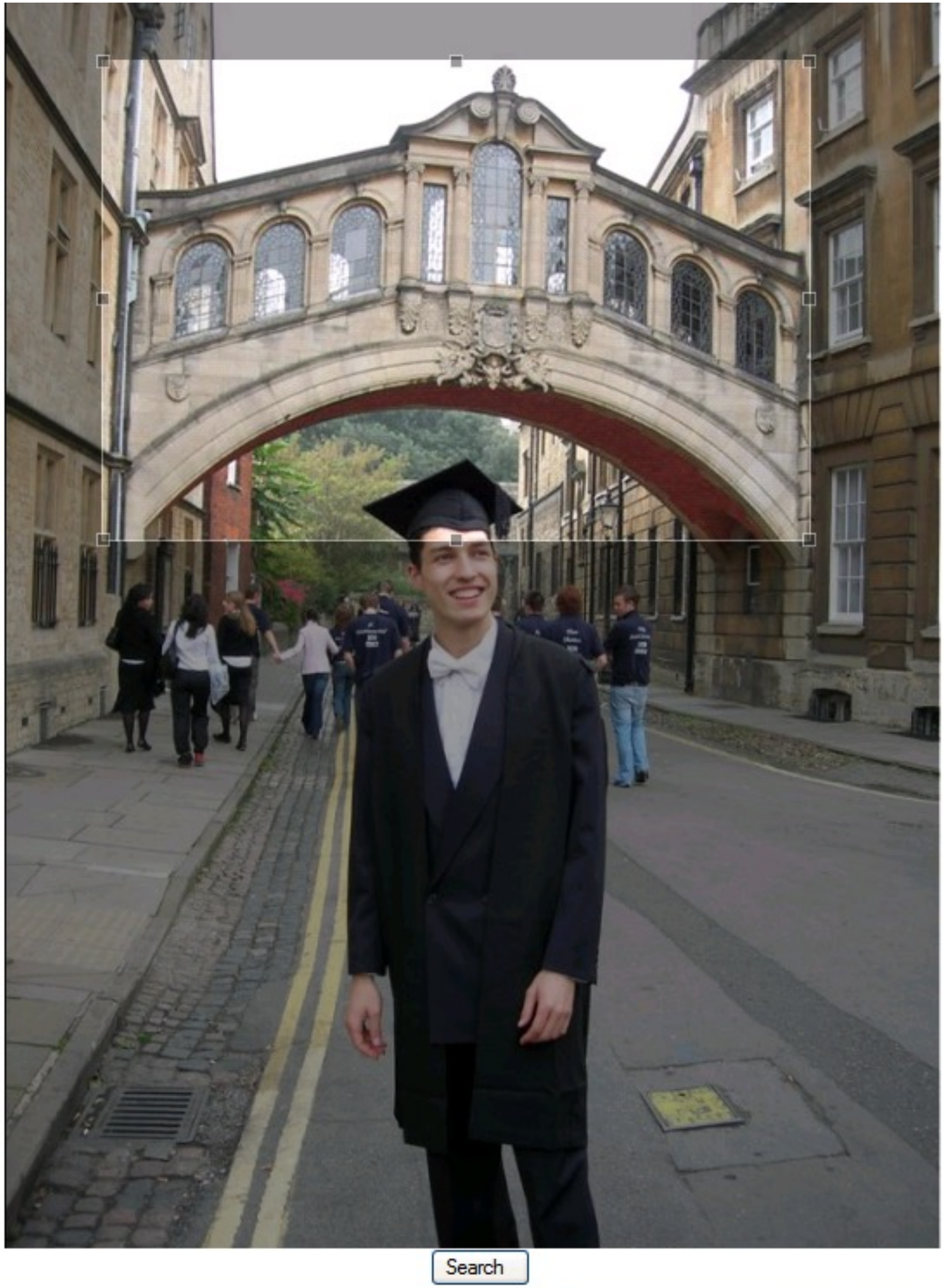
G. Tolias, Y. Avrithis, H. Jégou. Image search with selective match kernels: aggregation across single and multiple. International Journal of Computer Vision, 2016

Efficient visual search for objects and places

Oxford Buildings Search - demo

http://www.robots.ox.ac.uk/~vgg/research/oxbuildings/index.html

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

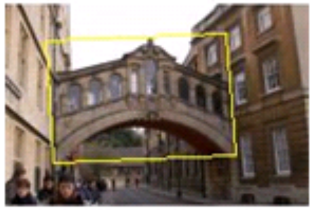
2

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

4

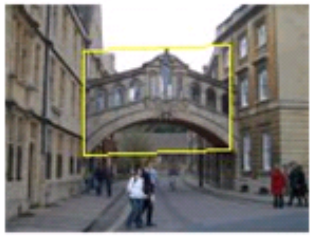


ID: oxc1_oxford_001612
Score: 252.000000
Putative: 451
Inliers: 252
Hypothesis: 1.046026 0.069416 51.576881 -0.044949 1.046938 76.264442
Detail

5

ID: oxc1_hertford_000123
Score: 225.000000
Putative: 446
Inliers: 225
Hypothesis: 1.361741 0.090413 -34.673317 -0.084659 1.301689 -32.281090
Detail

6



ID: oxc1_oxford_001085
Score: 224.000000
Putative: 389
Inliers: 224
Hypothesis: 0.848997 0.000000 195.707611 -0.031077 0.895546 114.583961
Detail

7

ID: oxc1_hertford_000077
Score: 195.000000
Putative: 386
Inliers: 195
Hypothesis: 1.465144 0.069286 -108.473091 -0.097598 1.461877 -30.205191
Detail

Oxford buildings dataset

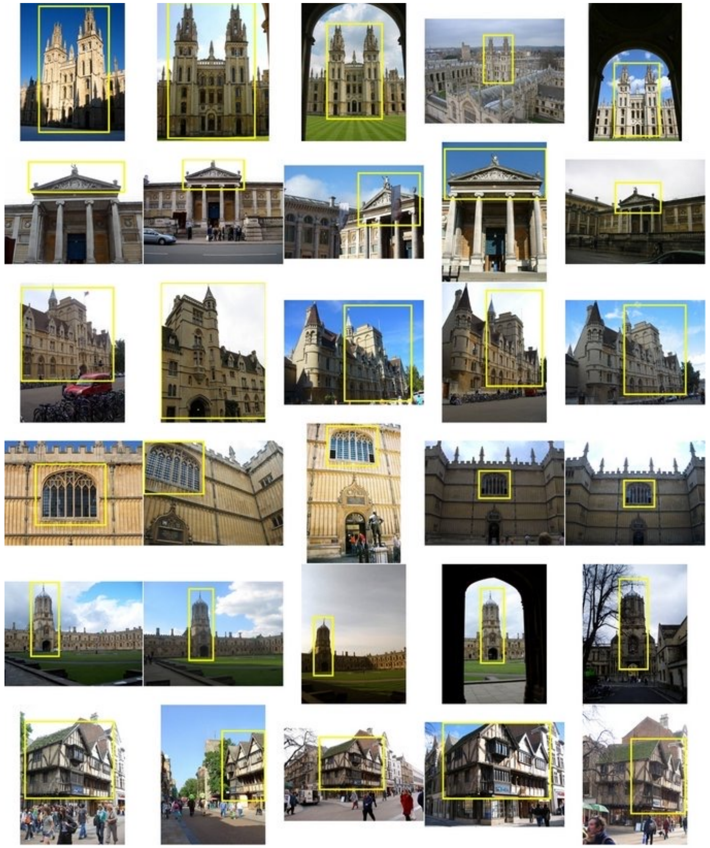
- Automatically crawled from flickr
- Consists of:

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

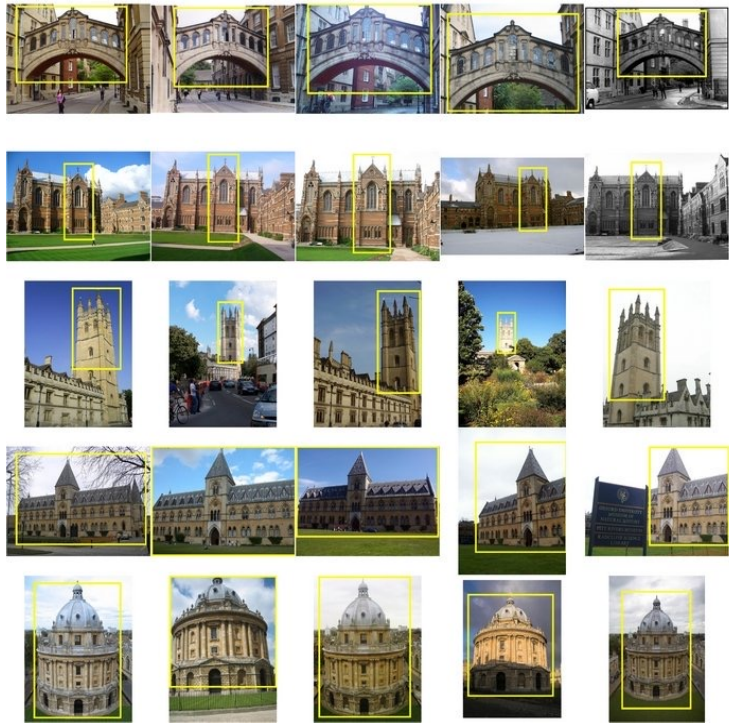
Oxford buildings dataset

- Landmarks plus queries used for evaluation

All Soul's
Ashmolean
Balliol
Bodleian
Thom Tower
Cornmarket



Bridge of Sighs
Keble
Magdalen
University Museum
Radcliffe Camera

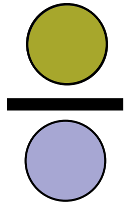


- Ground truth obtained for 11 landmarks

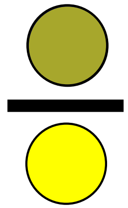
- Evaluate performance by mean Average Precision

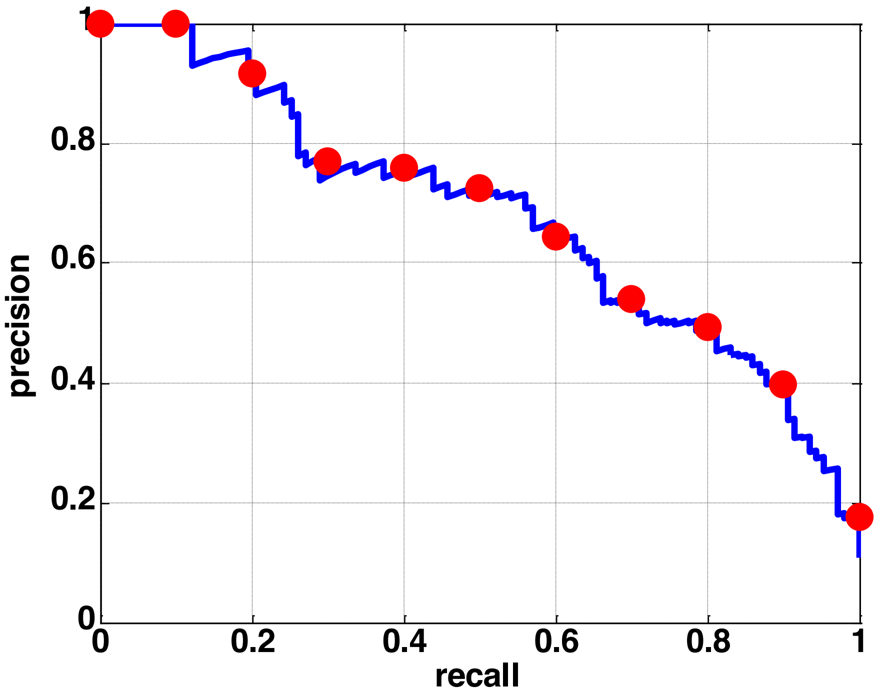
Measuring retrieval performance: Precision - Recall

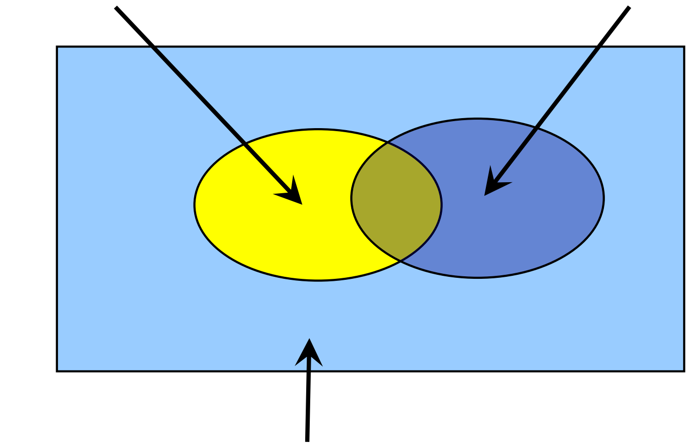
- Precision: % of returned images that are relevant



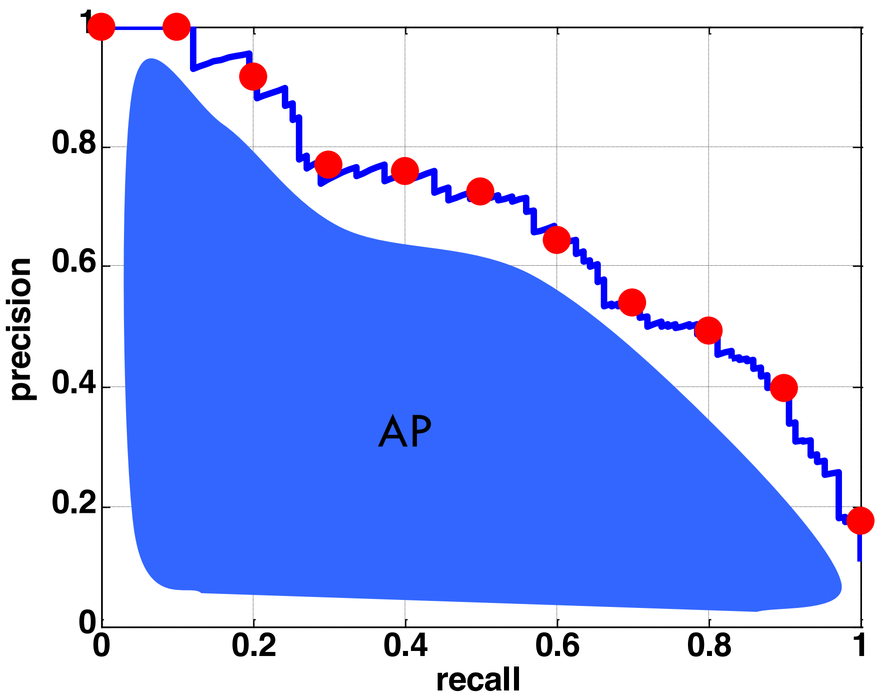
- Recall: % of relevant images that are returned





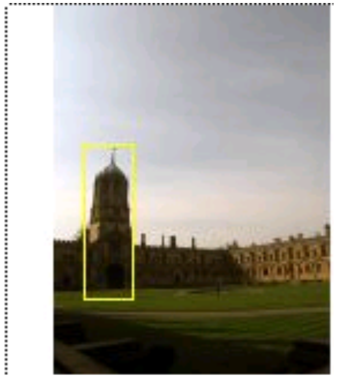


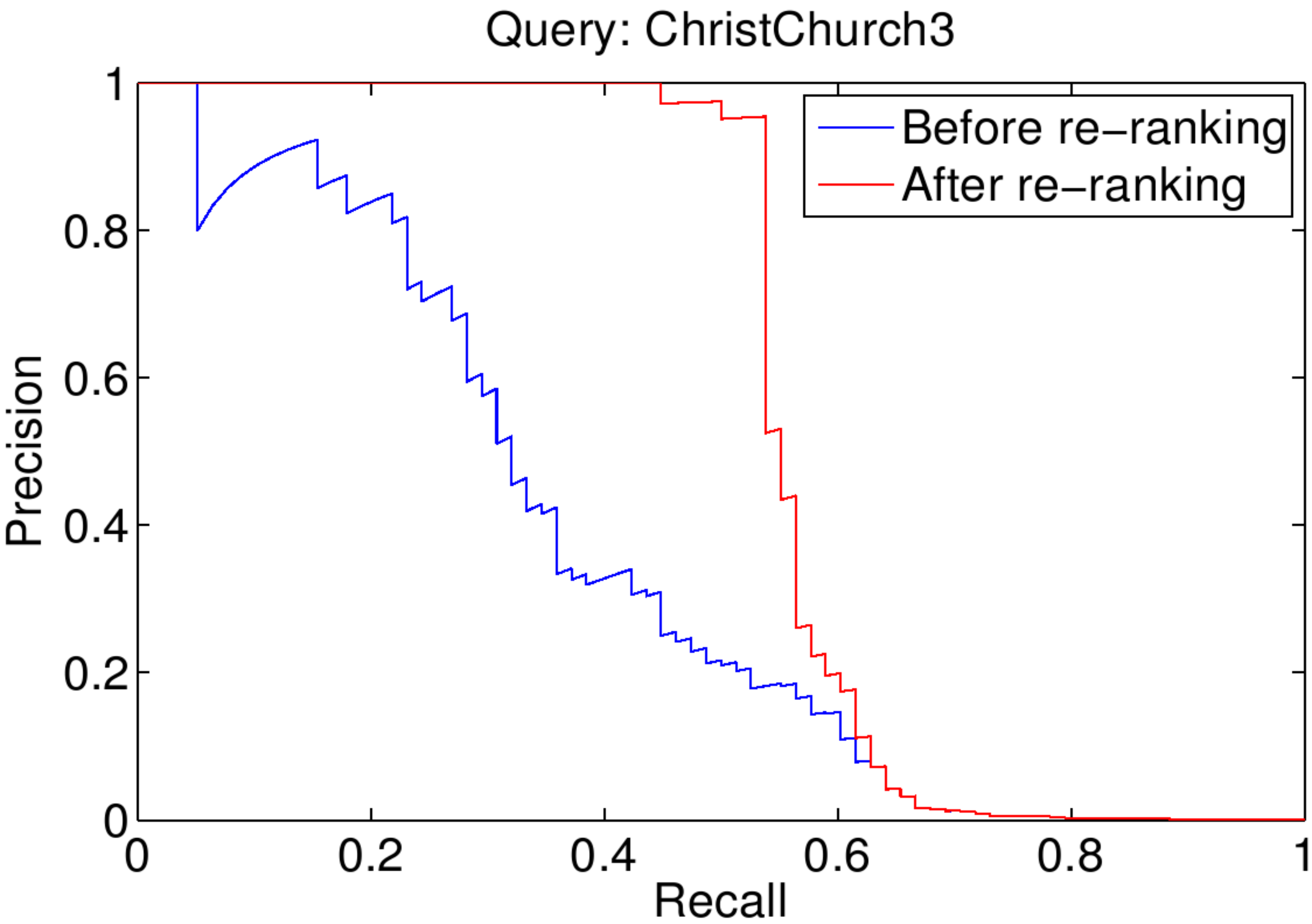
Average Precision



- A good AP score requires both high recall and high precision
- Application-independent

Performance measured by mean Average Precision (mAP) over 55 queries on 100K or 1.1M image datasets

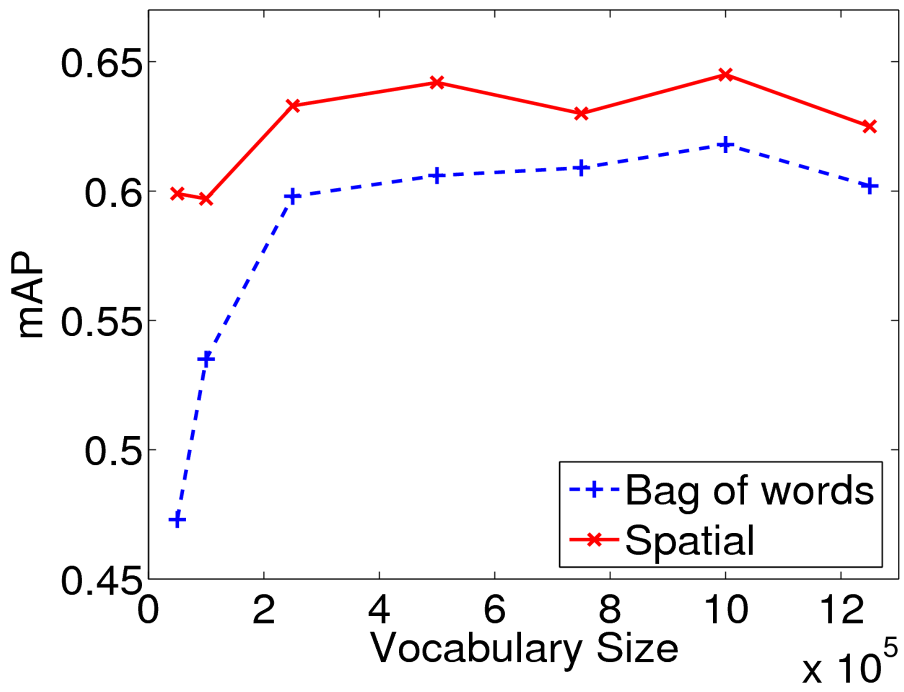




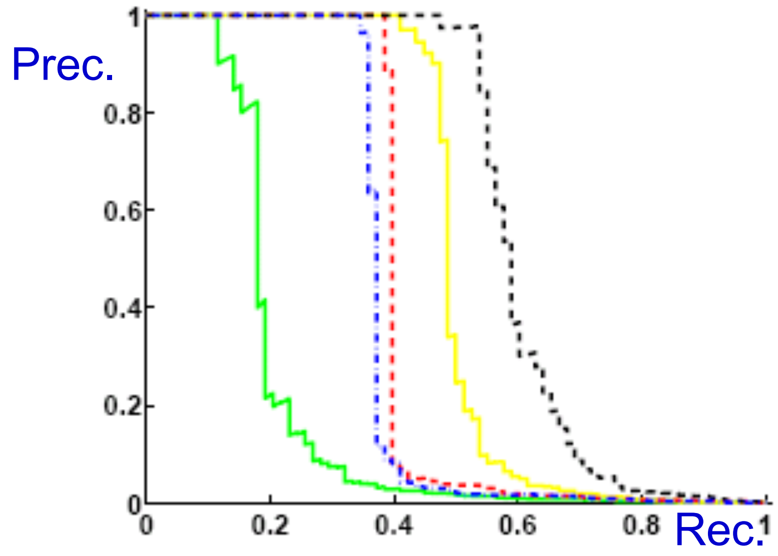
Query: ChristChurch3
Before re−ranking
After re−ranking
Precision
Recall
0
0.2
0.4
0.6
0.8
1

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 images



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

Visual search (references)

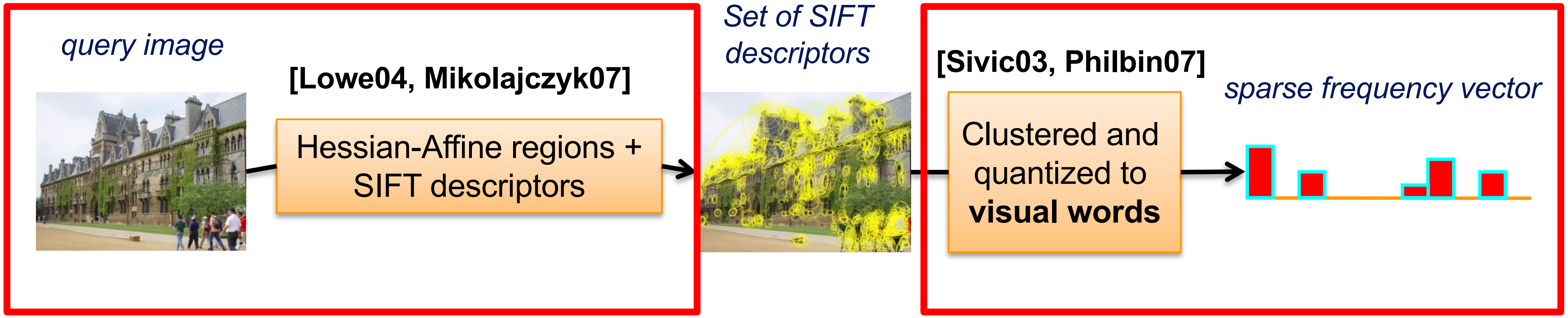
G. Tolias, Y. Avrithis, H. Jégou. Image search with selective match kernels: aggregation across single and multiple. International Journal of Computer Vision, 2016

G Tolias, R Sicre, H Jégou, Particular object retrieval with integral max-pooling of CNN activations, International Conference on Learning Representations (ICLR) 2016

F Radenović, G Tolias, O Chum, CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples, European Conference on Computer Vision (ECCV) 2016.

F Radenović, A Iscen, G Tolias, Y Avrithis, O Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

Why aren't all objects retrieved?



Obtaining visual words is like a sensor measuring the image

"noise" in the measurement process means that some visual words are missing or incorrect, e.g. due to

- Missed detections
- Changes beyond built in invariance
- Quantization effects

1. Query expansion
2. Better quantization

Consequence: Visual word in query is missing in target image

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: Text

Original query: Hubble Telescope Achievements

Query expansion: Select top 20 terms from top 20 documents according to tf-idf

Added terms: Telescope, hubble, space, nasa, ultraviolet, shuttle, mirror, telescopes, earth, discovery, orbit, flaw, scientists, launch, stars, universe, mirrors, light, optical, species

Example from: Jimmy Lin, University of Maryland

Automatic query expansion

Visual word representations of two images of the same object may differ (due to e.g. detection/quantization noise) resulting in missed returns

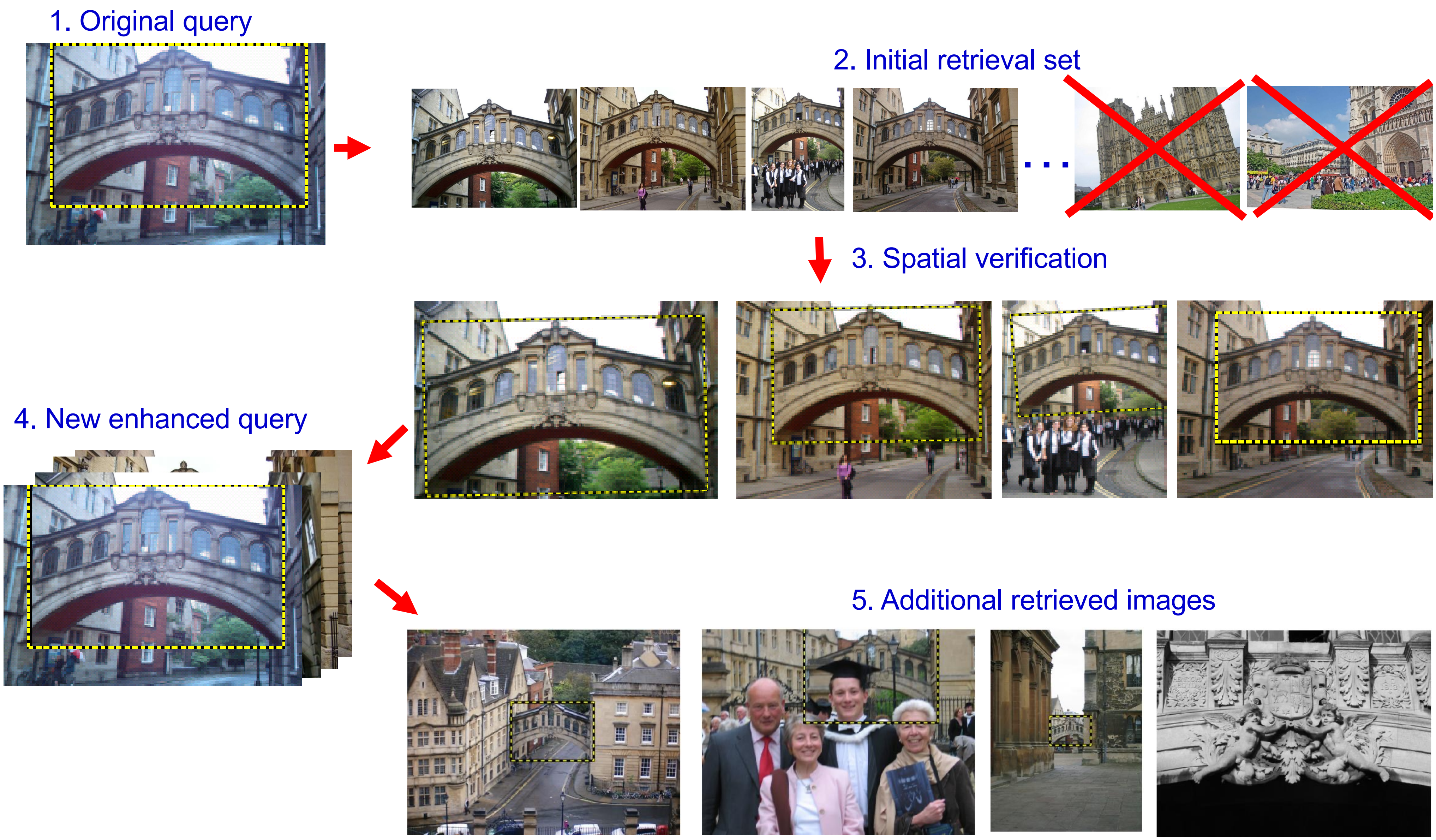
Initial returns may be used to add new relevant visual words to the query

Strong spatial model prevents ‘drift’ by discarding false positives

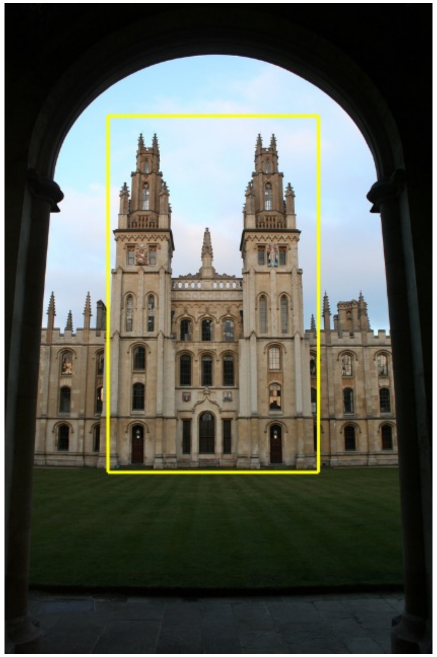
[Chum, Philbin, Sivic, Isard, Zisserman, ICCV’07;

Chum, Mikulik, Perdoch, Matas, CVPR’11]

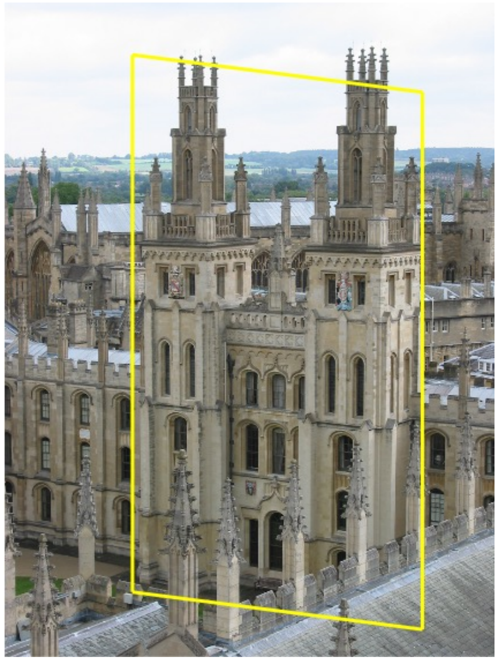
Visual query expansion - overview



Query Expansion



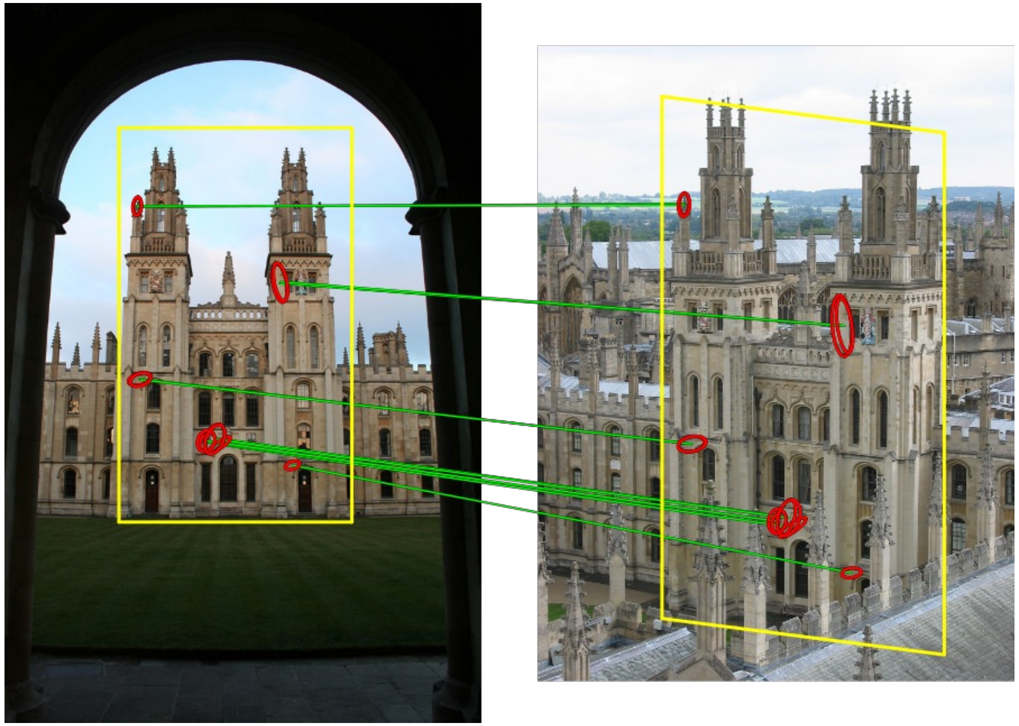
Query Image



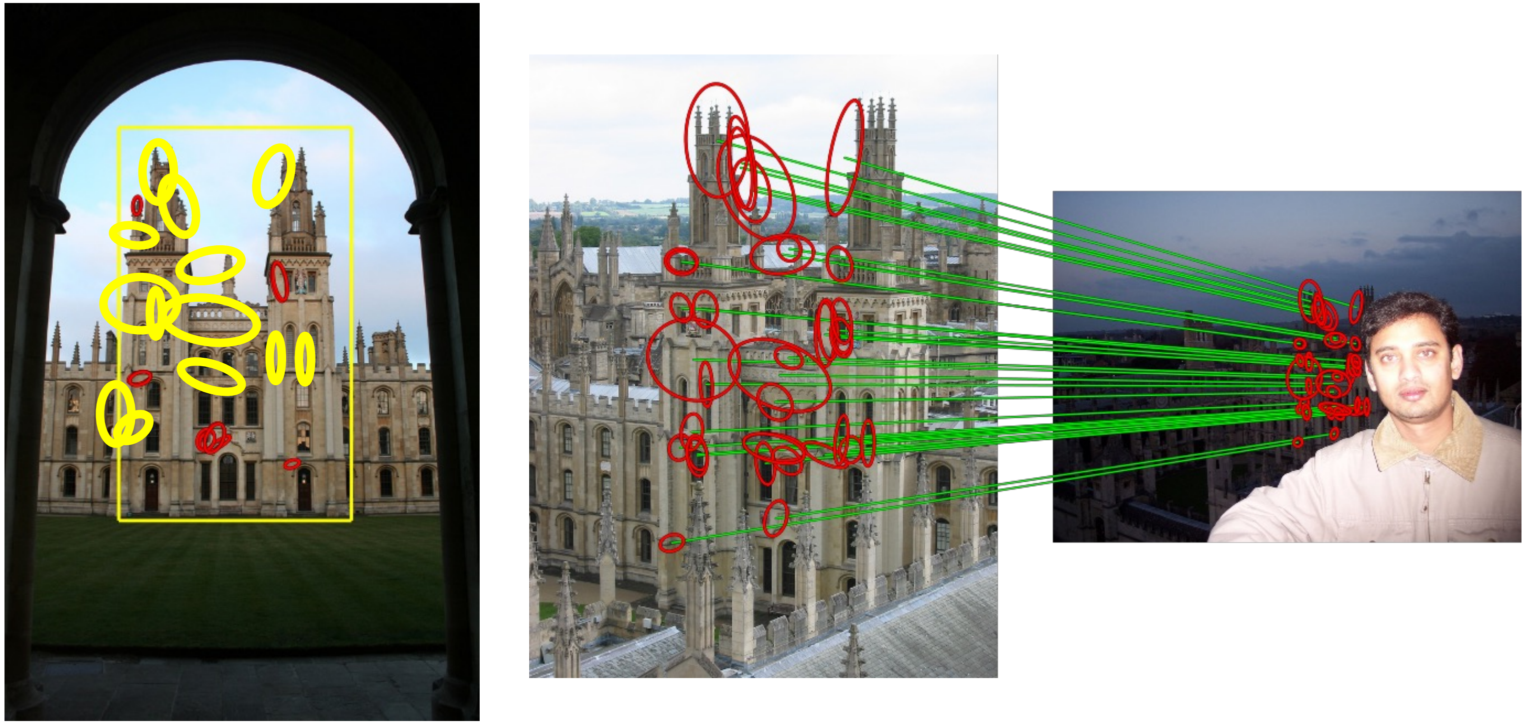
Originally retrieved image

Originally not retrieved

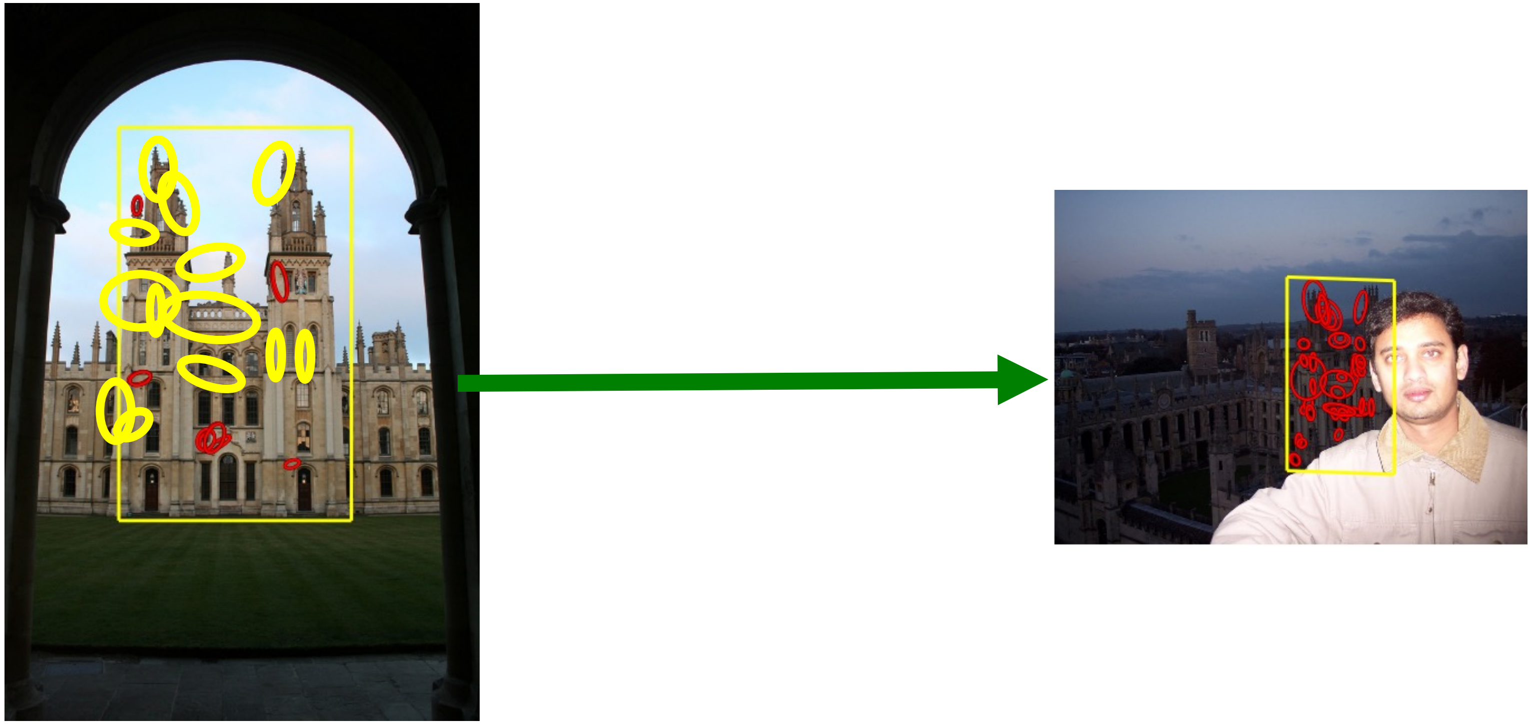
Query Expansion



Query Expansion

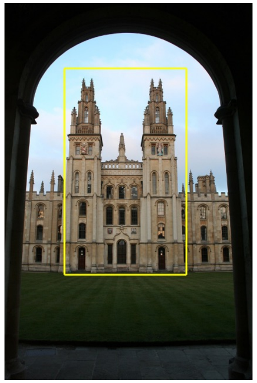


Query Expansion

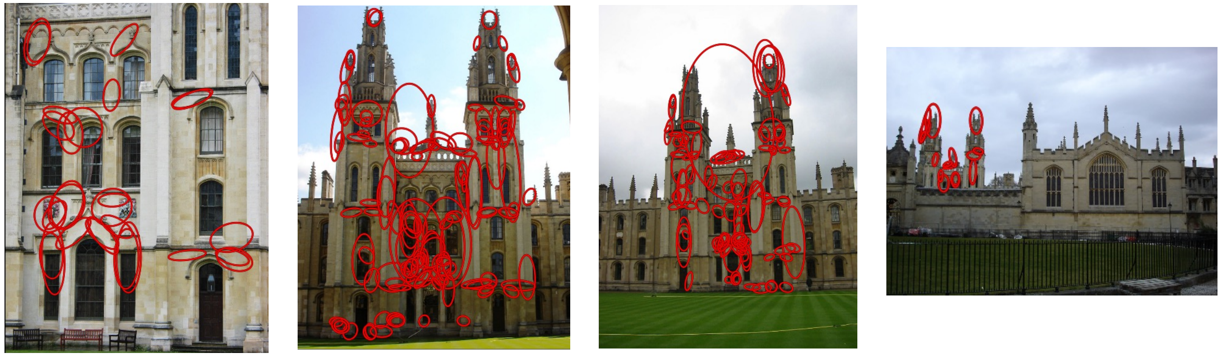


Query Expansion

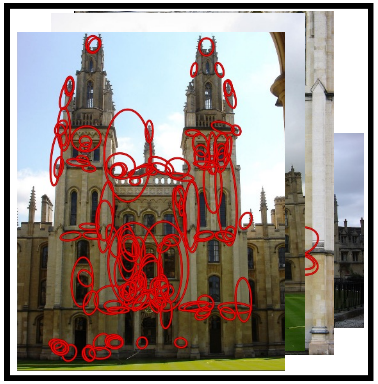
Query Image



Spatially verified retrievals with matching regions overlaid



…



New expanded query

New expanded query is formed as

- the average of visual word vectors of spatially verified returns
- only inliers are considered
- regions are back-projected to the original query image

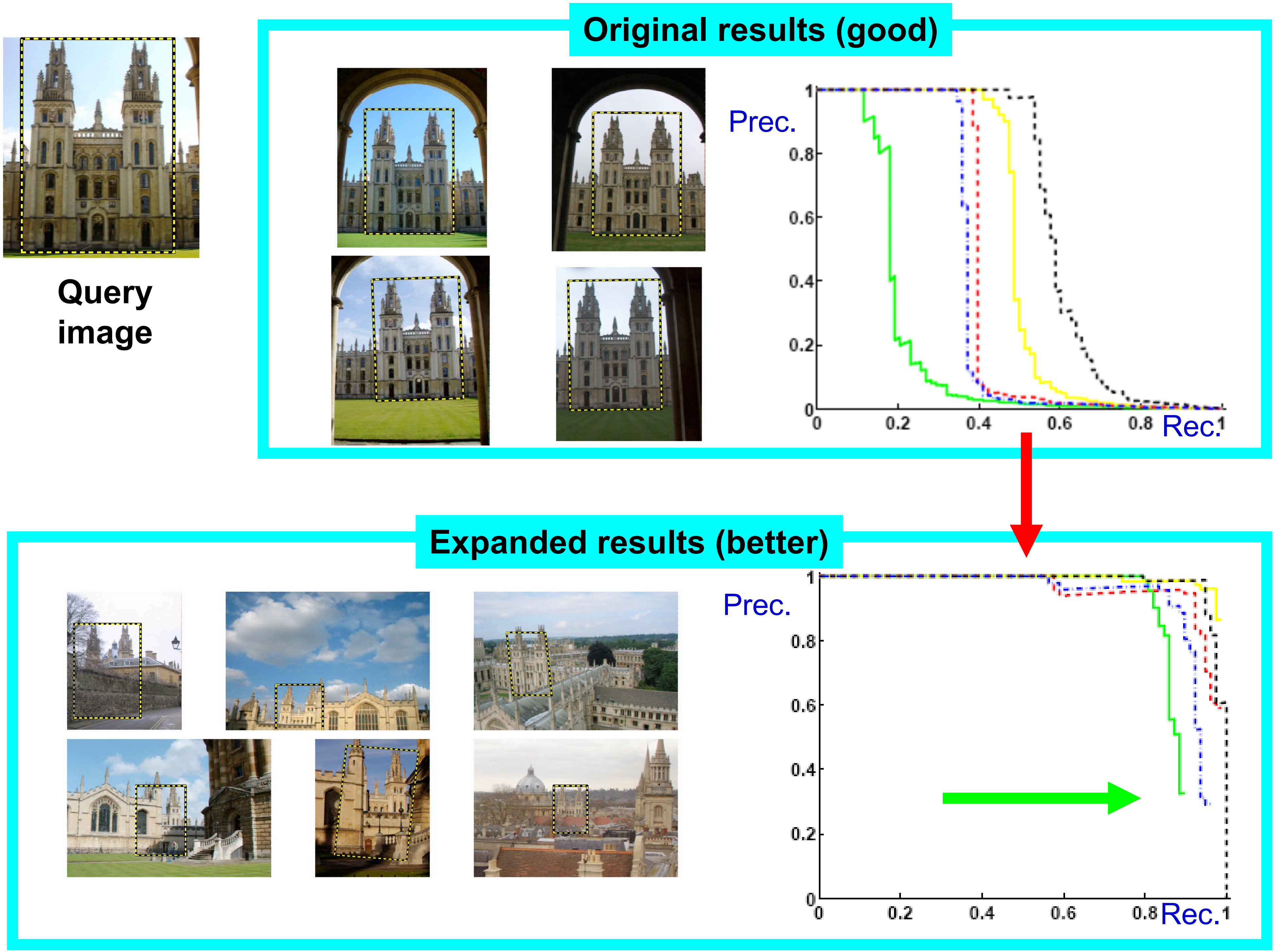
Demo

Query Expansion

Query image

Originally retrieved

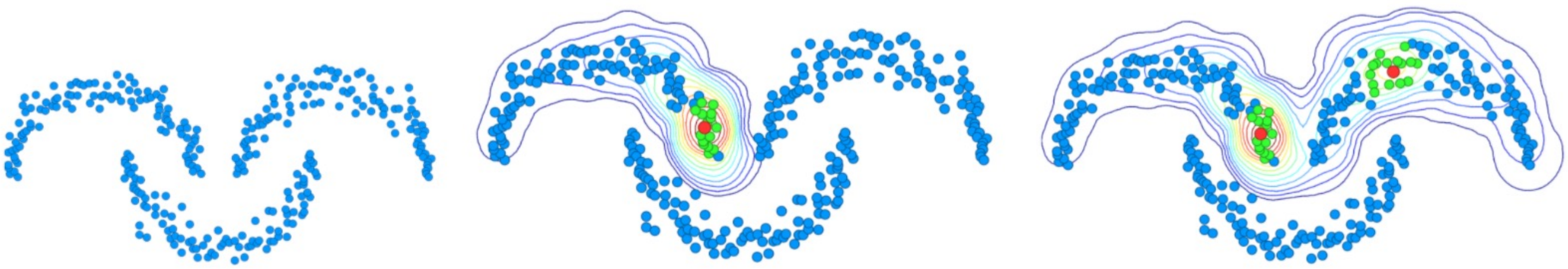
Retrieved only after expansion


Query image
Original results (good)
Prec.
Rec.
Expanded results (better)
Prec.
Rec.

Beyond query expansion – image region graphs

Efficient Diffusion on Region Manifolds: Recovering Small Objects with Compact CNN Representations

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Teddy Furon, Ondřej Chum



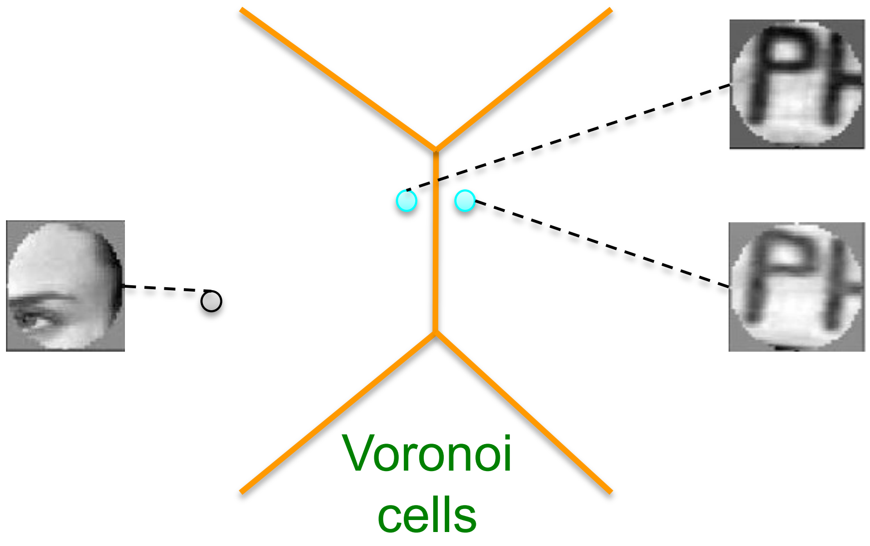
(a) dataset (b) single query (c) multiple queries

[Iscen et al., CVPR 2017]
https://cmp.felk.cvut.cz/~iscenahm/_pages/diffusion.html

Quantization errors

Typically, quantization has a significant impact on the final performance of the system **[Sivic03,Nister06,Philbin07]**

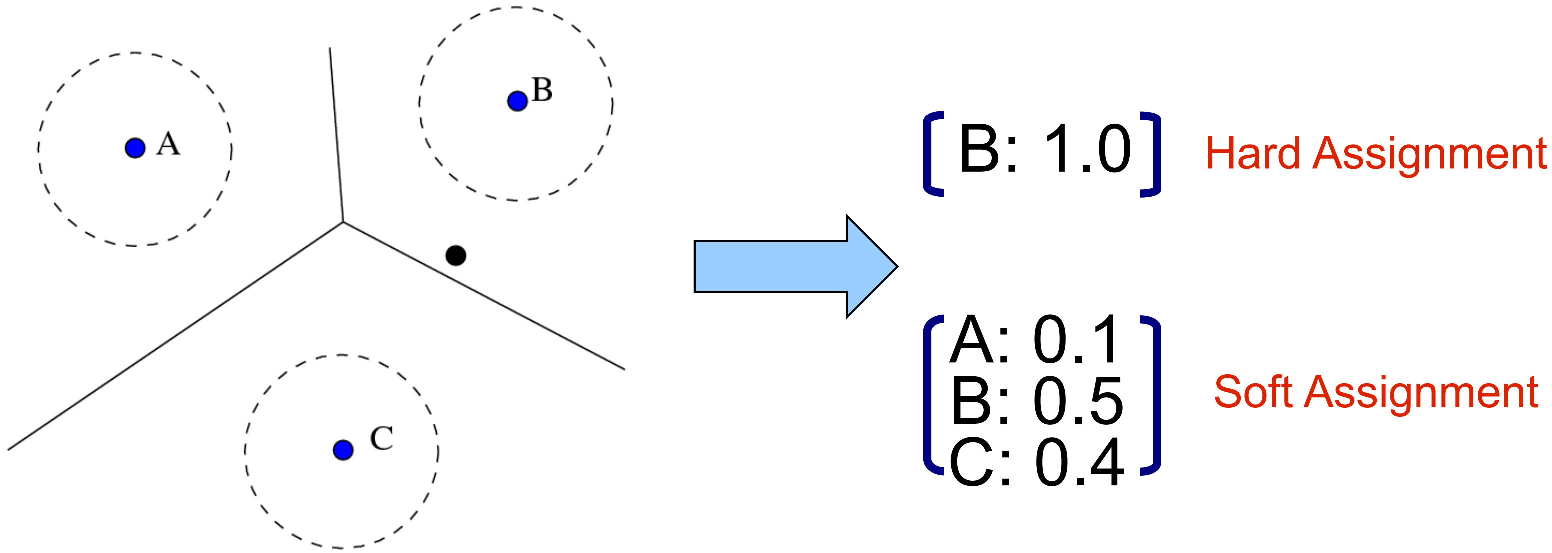
Quantization errors split features that should be grouped together and confuse features that should be separated



Overcoming quantization errors

- Soft-assign each descriptor to multiple cluster centers

[Philbin et al. 2008, Van Gemert et al. 2008]



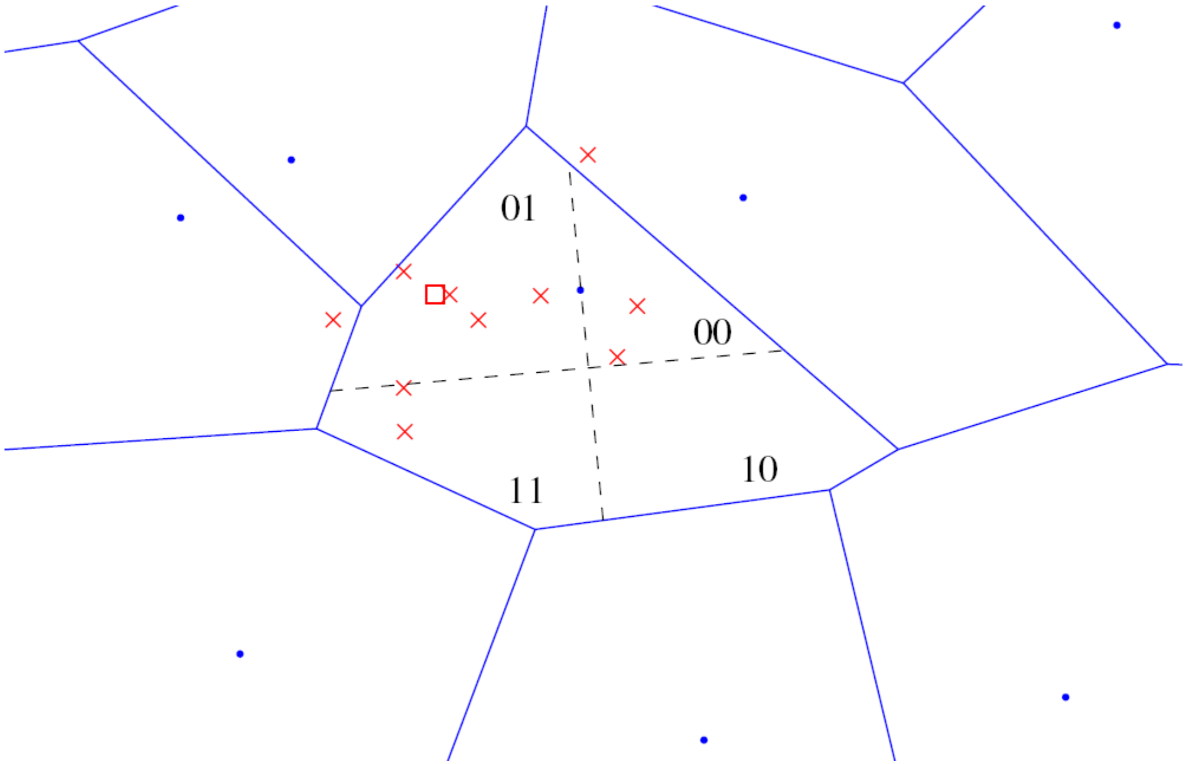
Learning a vocabulary to overcome quantization errors

[Mikulik et al. ECCV 2010, Philbin et al. ECCV 2010]

Beyond bag-of-visual-words I.

Hamming embedding [Jegou&Schmid 2008]

- Standard quantization using bag-of-visual-words
- Additional localization in the Voronoi cell by a binary signature

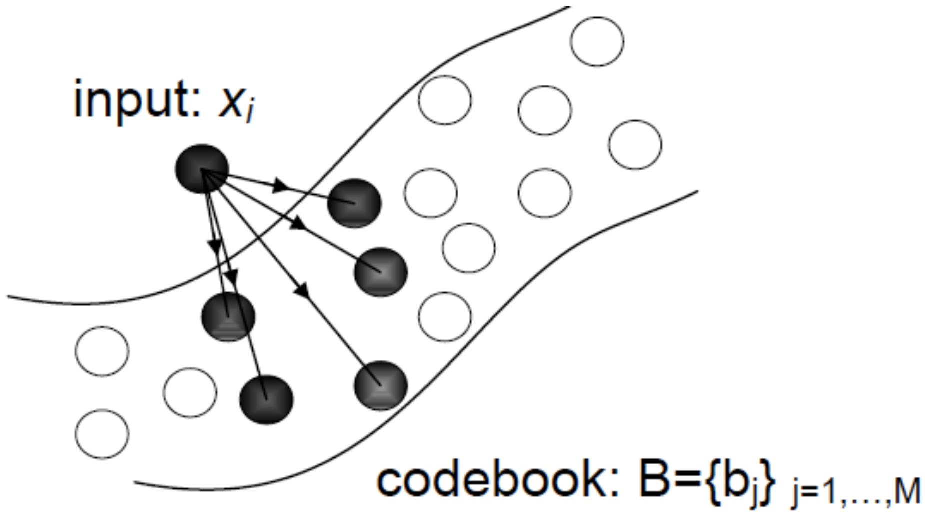


Beyond bag-of-visual-words II.

Locality-constrained linear coding.
[Wang et al. CVPR 2010]

- Represent data point as a linear combination of nearby cluster centers.
- Store the coefficients of linear combination.

Used for category-level classification.



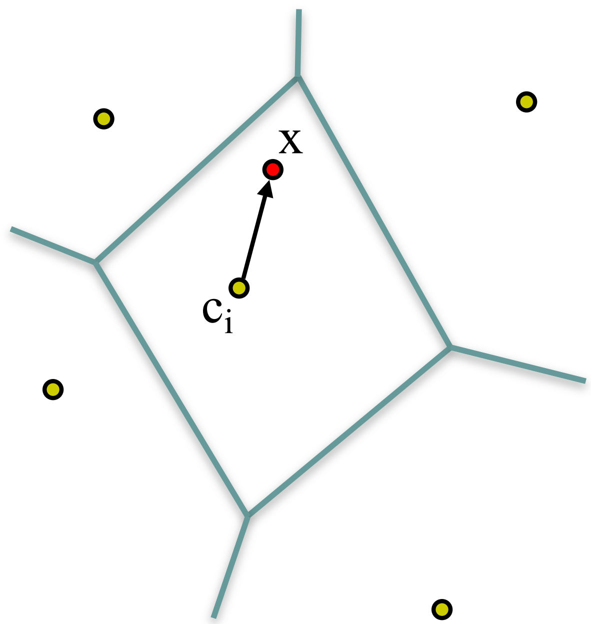
Connection to sparse coding - more at lecture by J. Ponce

Beyond bag-of-visual-words III.

VLAD – Vector of locally aggregated descriptors

[Jegou et al. 2010] but see also [Perronin et al. 2010]

Measure (and quantize) the difference vectors from the cluster center.



Outline – Efficient visual search

1. Efficient matching of local descriptors
   - Approximate nearest neighbor search
   - k-d trees, locality-sensitive hashing (LSH)

2. Aggregate local descriptors into a single vector
   - Bag-of-visual-words, inverted files, query expansion

3. Compact representations for very large-scale search
   - Product quantization (PQ)

4. Learnable representations
   - Neural representations for large-scale visual search
   - Visual search using natural language query

Towards very large-scale image search

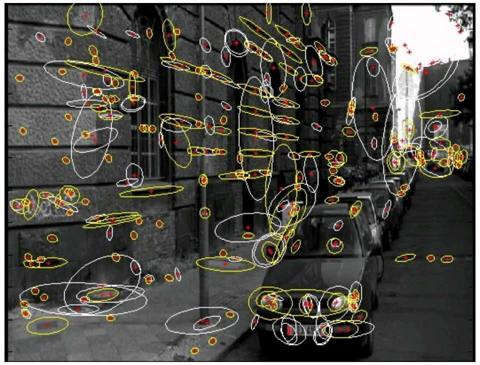
- BOF+inverted file can handle up to ~10 millions images
  - with a limited number of descriptors per image → RAM: 40GB
  - search: 2 seconds
- Web-scale = billions of images
  - with 100 M per machine → search: 20 seconds, RAM: 400 GB
  - not tractable
- Solution: represent each image by one **compressed** vector

Slide credit: C. Schmid

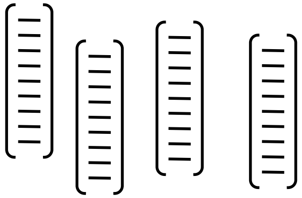
Strategy I: Efficient approximate NN search

Images

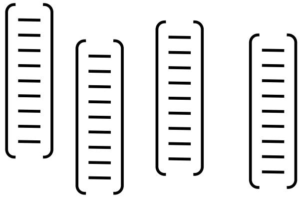
Local features



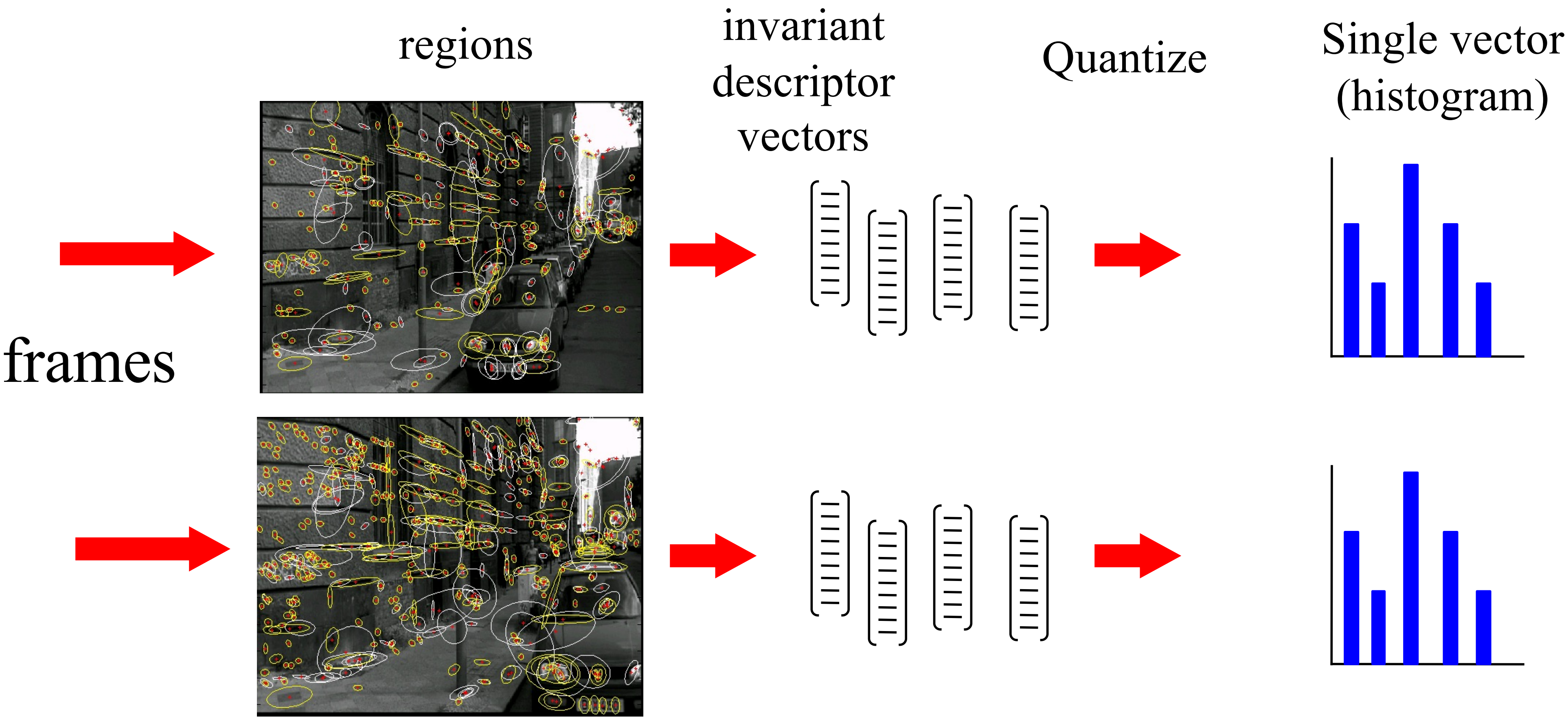
invariant descriptor vectors



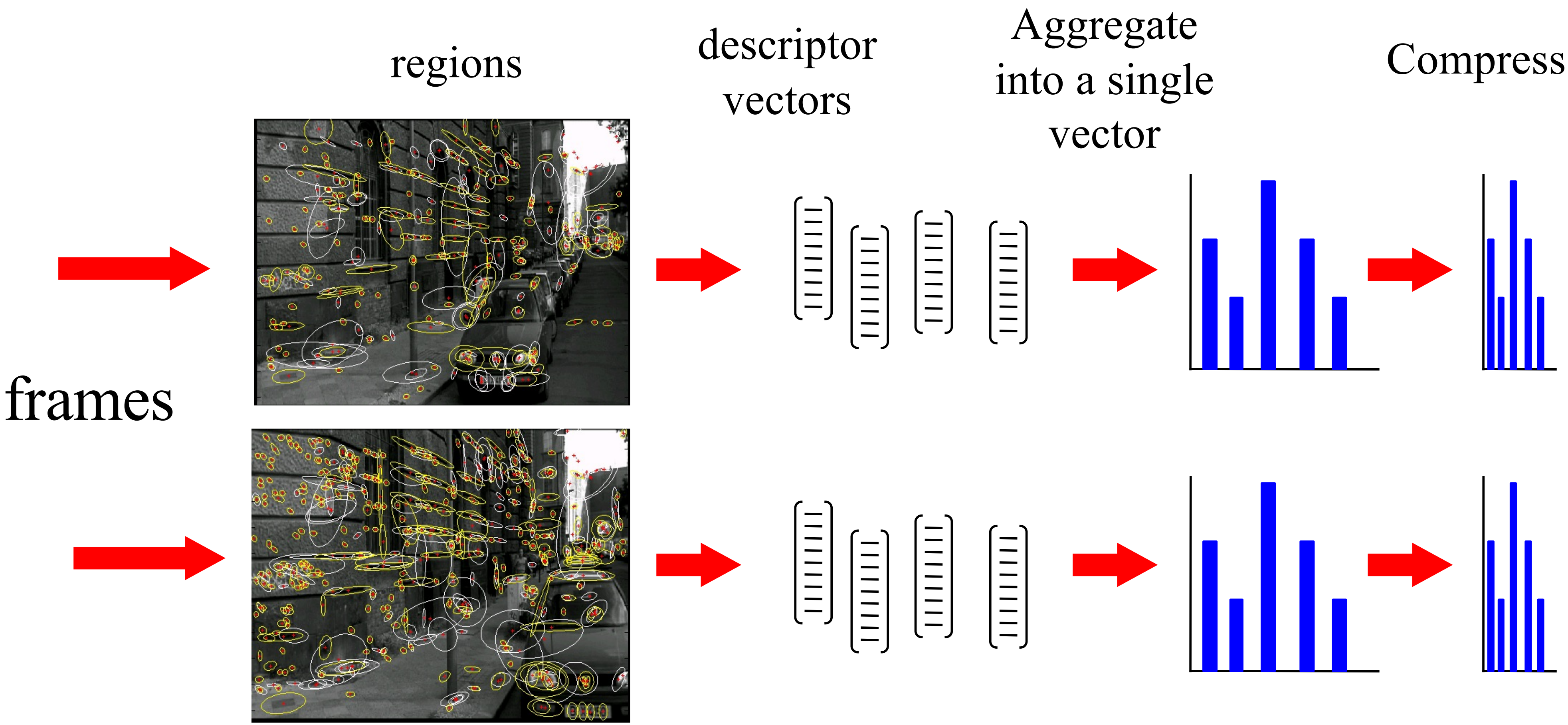
invariant descriptor vectors



Strategy II: Match histograms of visual words



Strategy III: Match compressed vectors



Slide credit: C. Schmid

Compact image representation

- Aim: improving the tradeoff between
  - search speed
  - memory usage
  - search quality

- Approach: joint optimization of three stages
  - descriptor aggregation
  - dimension reduction
  - indexing algorithm

Image representation	→	Descriptor aggregation	→	Descriptor Compression	→	(Non) – exhaustive search
Local features CNN descriptors		BOW VLAD		PCA + PQ codes		Approx. nearest neighbor Inverted files

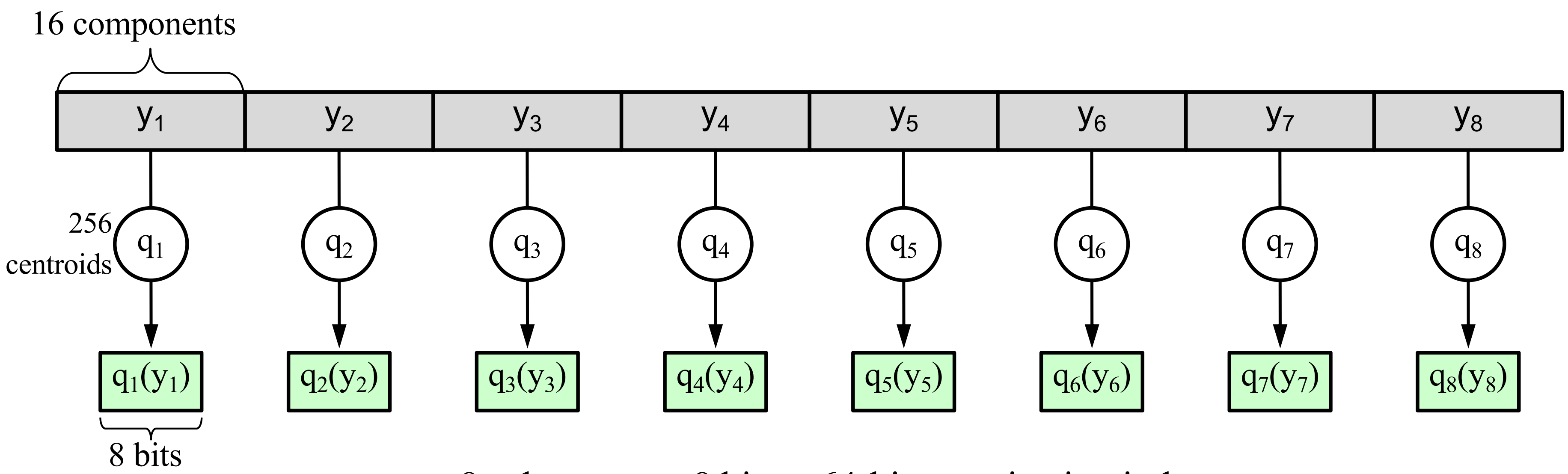
Slide credit: C. Schmid

Product quantization for nearest neighbor search

- Vector split into *m* subvectors: $y \rightarrow [y_1 | \ldots | y_m]$

- Subvectors are quantized separately by quantizers $q(y) = [q_1(y_1) | \ldots | q_m(y_m)]$
  where each q_i is learned by *k*-means with a limited number of centroids

- Example: y = 128-dim vector split in 8 subvectors of dimension 16
  - each subvector is quantized with 256 centroids -> 8 bit
  - very large codebook 256^8 ~ 1.8x10^19

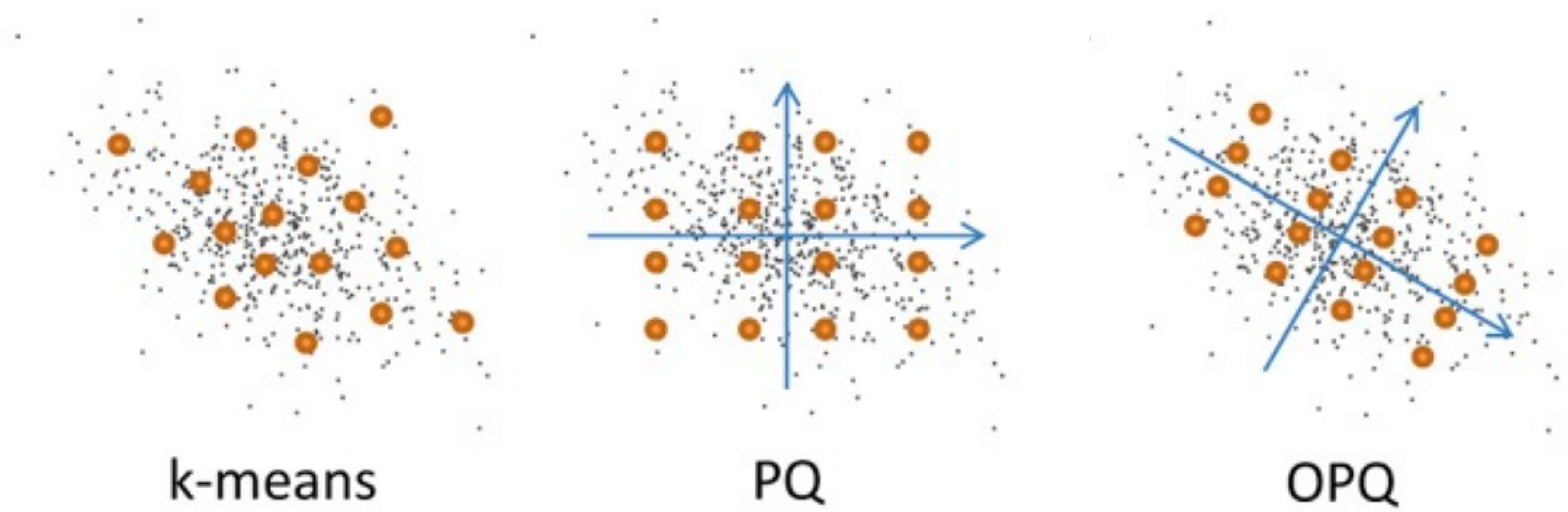


$\Rightarrow$ 8 subvectors x 8 bits = 64-bit quantization index

Slide credit: C. Schmid

[Jegou, Douze, Schmid, PAMI'11]

Product quantization



PQ: Product quantization, H. Jegou, M. Douze and C. Schmid, TPAMI 2011
OPQ: Optimized Product Quantization, by Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun, TPAMI, 2013.

FAISS library for efficient indexing
https://github.com/facebookresearch/faiss

Image credit: K. He
http://kaiminghe.com/cvpr13/index.html

Outline – Efficient visual search

1. Efficient matching of local descriptors
   - Approximate nearest neighbor search
   - k-d trees, locality-sensitive hashing (LSH)

2. Aggregate local descriptors into a single vector
   - Bag-of-visual-words, inverted files, query expansion

3. Compact representations for very large-scale search
   - Product quantization (PQ)

4. Learnable representations
   - Neural representations for large-scale visual search
   - Visual search using natural language query

Beyond local invariant features:
What objects/scenes local regions do not work on?

What objects/scenes local regions do not work on?

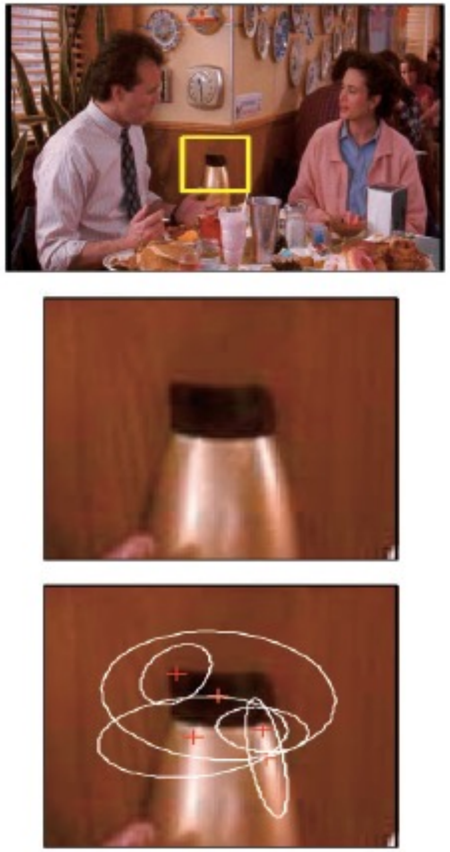
E.g. texture-less objects, objects defined by shape, deformable objects, wiry objects.


(e)


(f)

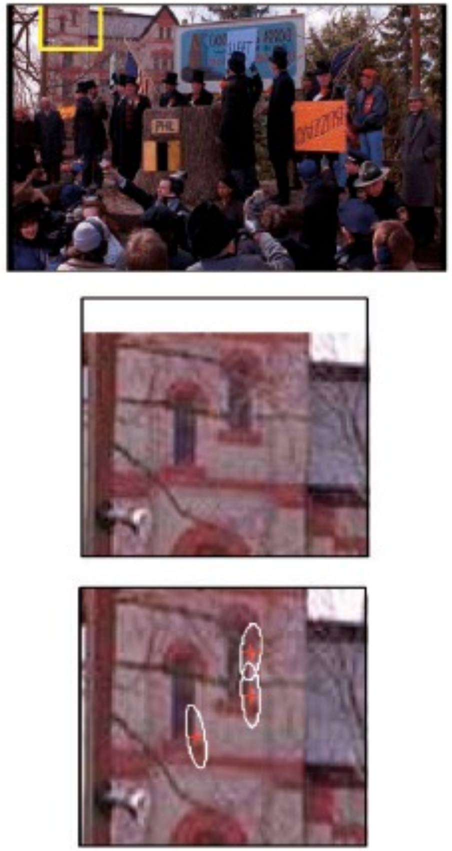

(g)


(h)


(i)


(j)


(k)


(l)

Other types of objects

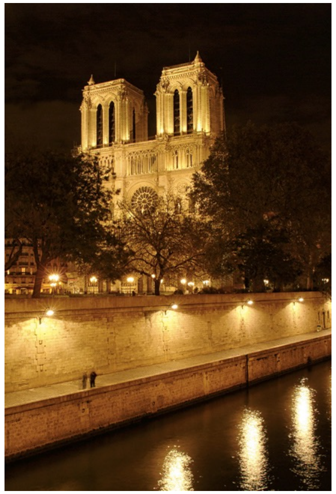
Visual search for texture-less, wiry, deformable and 3D objects..

See e.g.
Where to buy it: Matching street clothing photos in online shops, M Hadi Kiapour, X Han, S Lazebnik, AC Berg, TL Berg, ICCV 2015.

Other types of appearance variations

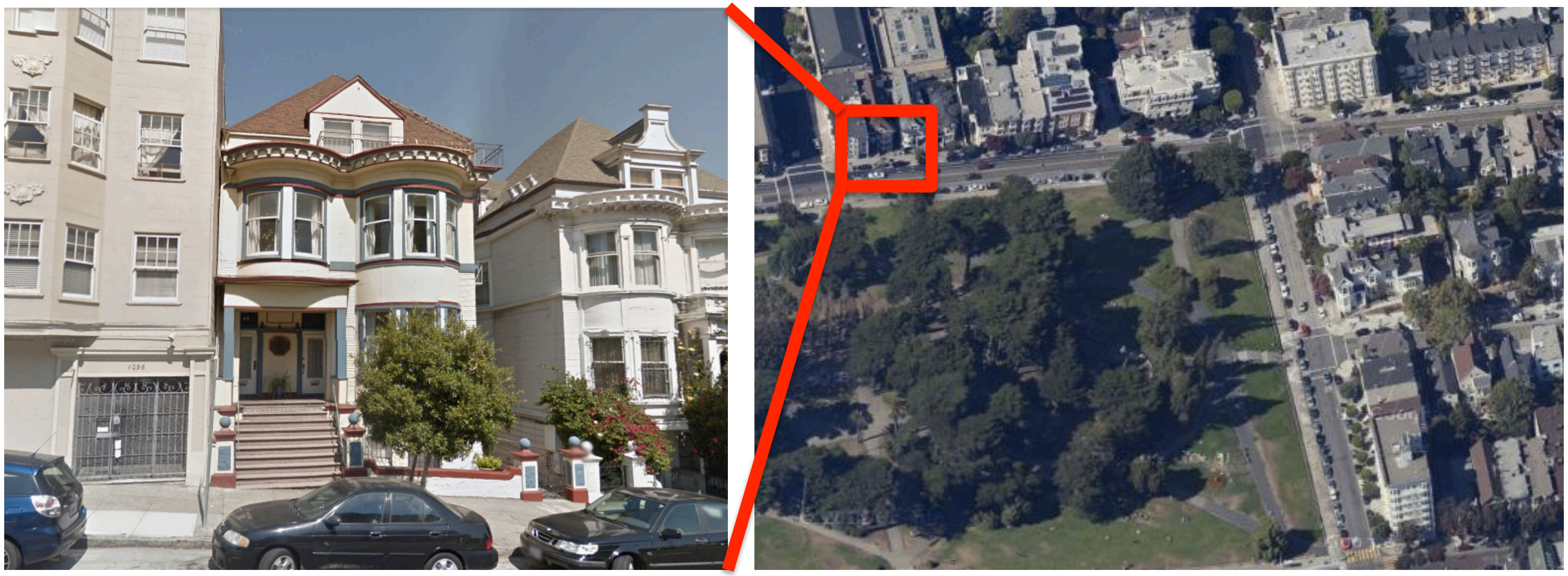
Match objects across large changes of appearance

Examples: non-photographic depictions, degradation over time, change of season, change of illumination, …


Extreme viewpoint changes

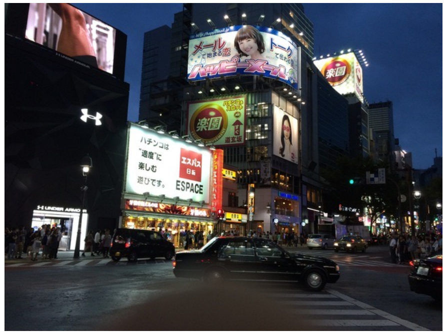
Query image　　　　Matching dataset



[Lin et al., CVPR’15]

See also: [Bansal et al.’11, Shan et al.’14]

Changes over time


See also: e.g. Perdoch et al.'15, Fernando et al.'14, Schindler et al.'06, Martin-Buralla'15, Matzen&Snavely'14

Example I.: Localize non-photographic depictions

Inputs: paintings, drawings, historical photographs, reference 3D model

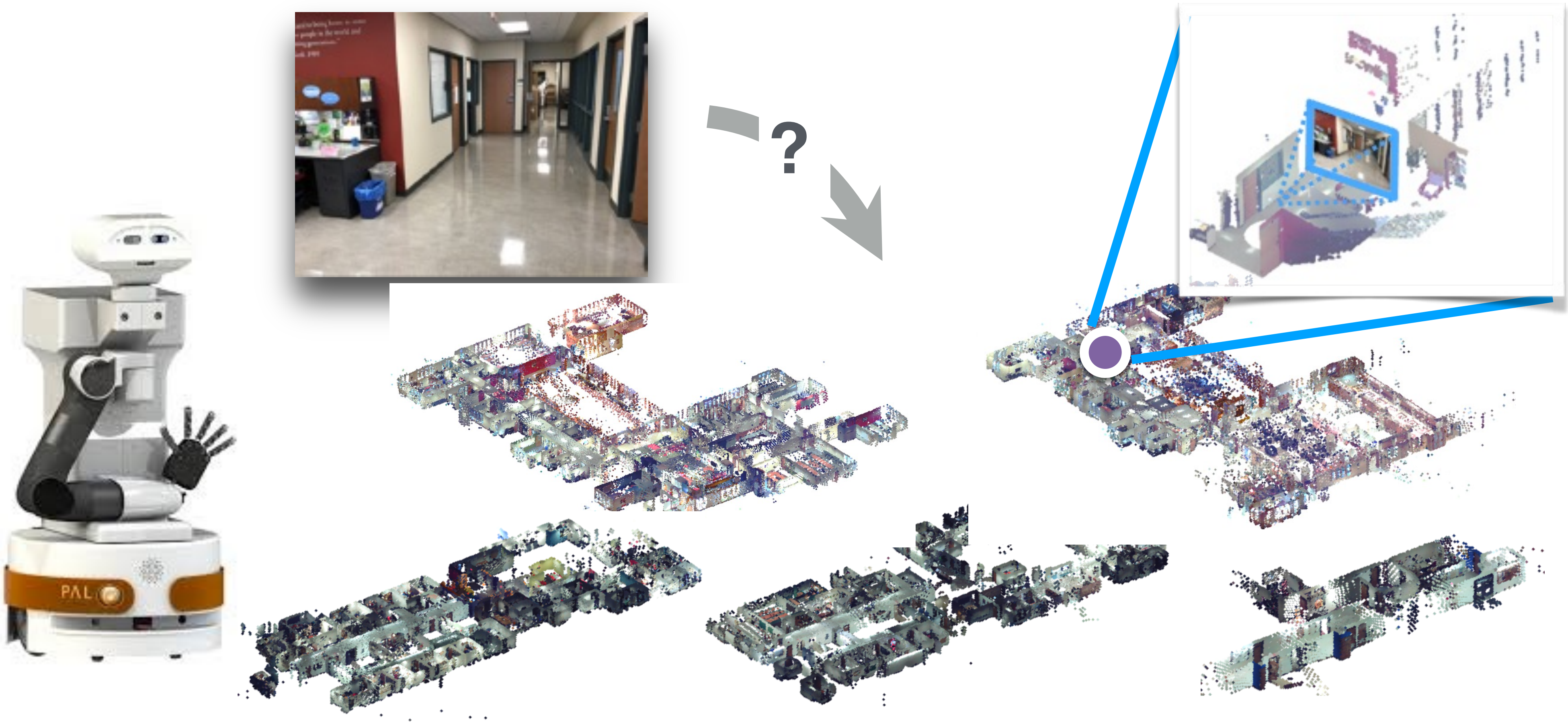
Output: recovered artist/camera viewpoints

[Aubry, Russell, Sivic, 2013]
http://www.di.ens.fr/willow/research/painting_to_3d/

Geo-localization of historical and non-photographic depictions

Example: Visual localization in indoor environments

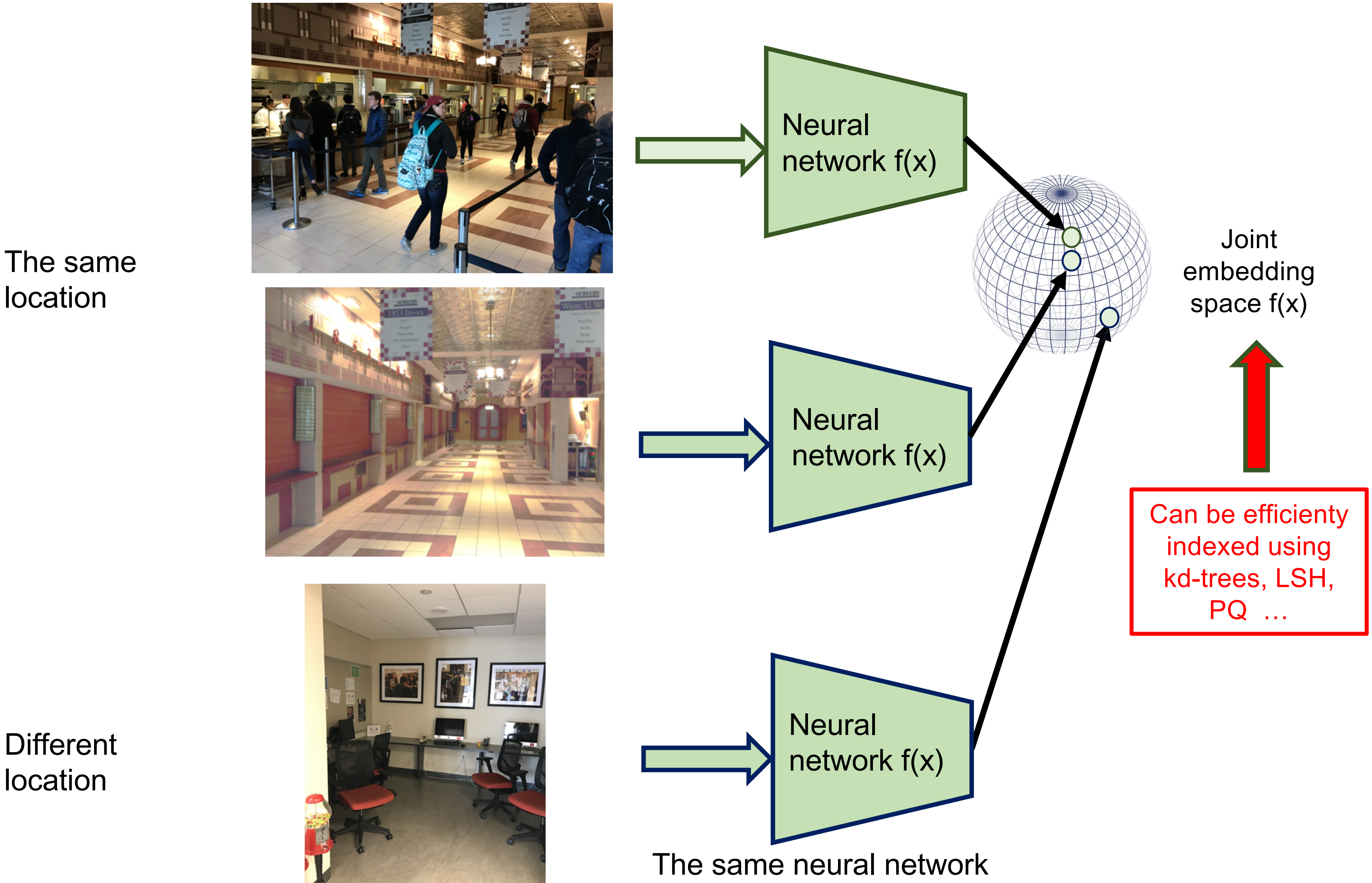
[Taira et al., CVPR 2018]



Visual localization indoors

[Taira et al., CVPR 2018]

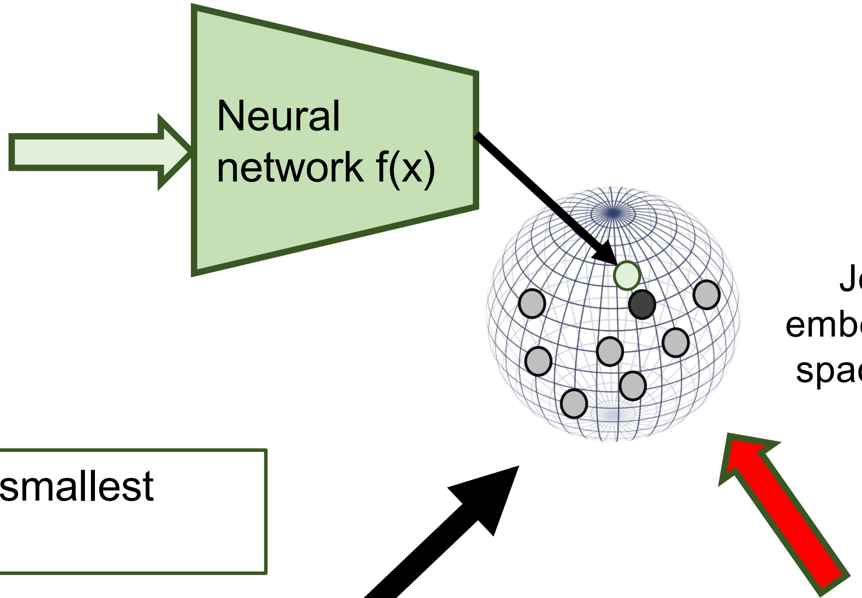
Solution: Learn neural distance functions



See also [Miech et al., 1706.06905, 1804.02516], and e.g. [Frome et al., NIPS 2013] [Gong et al., IJCV 2014]

Use learnt f(x) for efficient retrieval

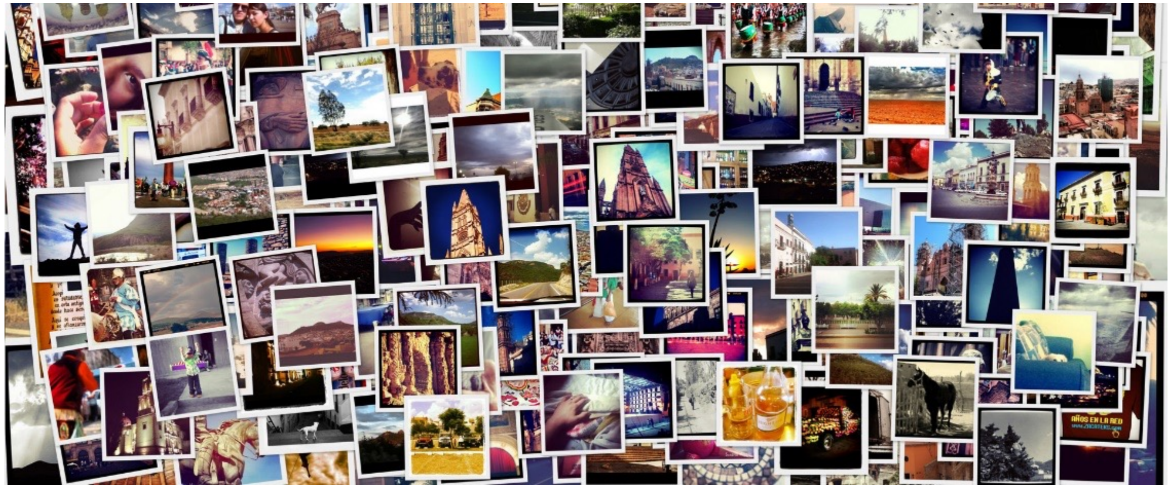
Query



Joint
embedding
space f(x)

Find an image from the database that has the smallest
distance in the learnt embedding space f(x)

Can be efficienty
indexed using
kd-trees, LSH,
PQ …



Database of images from different locations

Example: Visual localization in changing conditions

- [Sattler et al., arXiv:1707.09092]

Why is it difficult?

- Lighting changes: Different time of day / year
- Changes in camera viewpoint
- Occluders and ambiguous objects: People, cars, trees, pavement...
- Big data: World-scale localization

Why is it difficult?

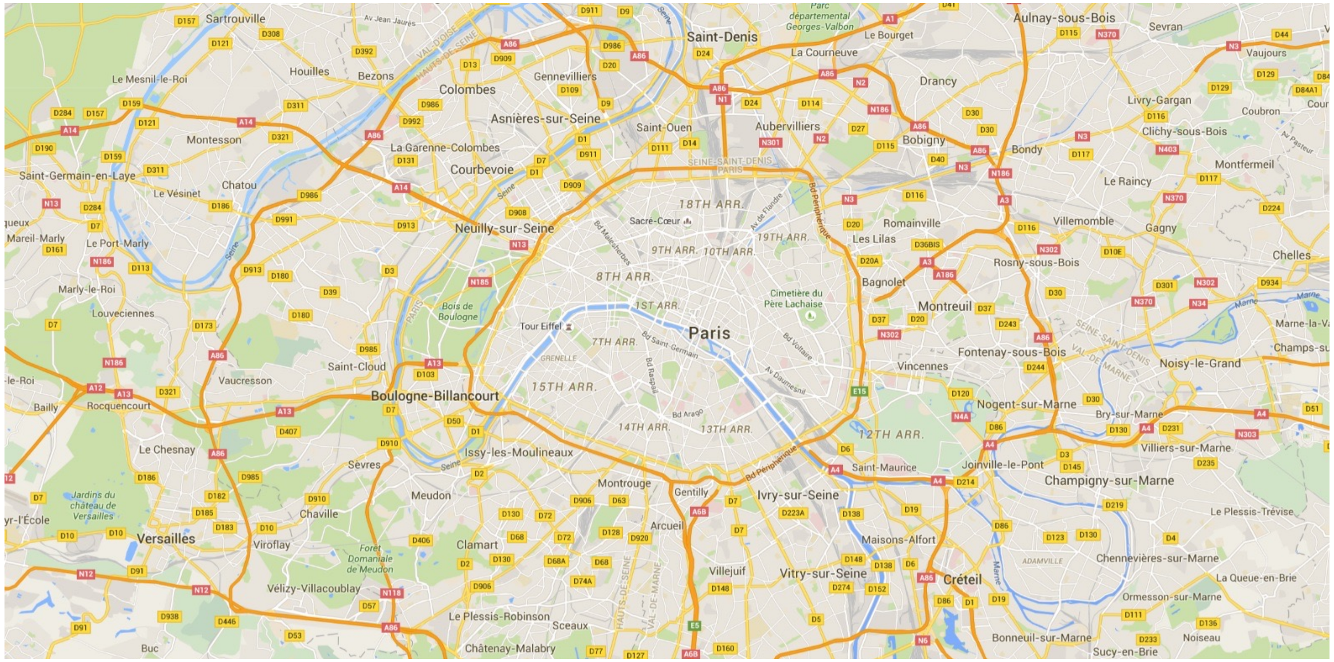
- Lighting changes: Different time of day / year
- Changes in camera viewpoint
- Occluders and ambiguous objects: People, cars, trees, pavement...
- Big data: World-scale localization

Why is it difficult?

- Lighting changes: Different time of day / year
- Changes in camera viewpoint
- Occluders and ambiguous objects: People, cars, trees, pavement...
- Big data: World-scale localization

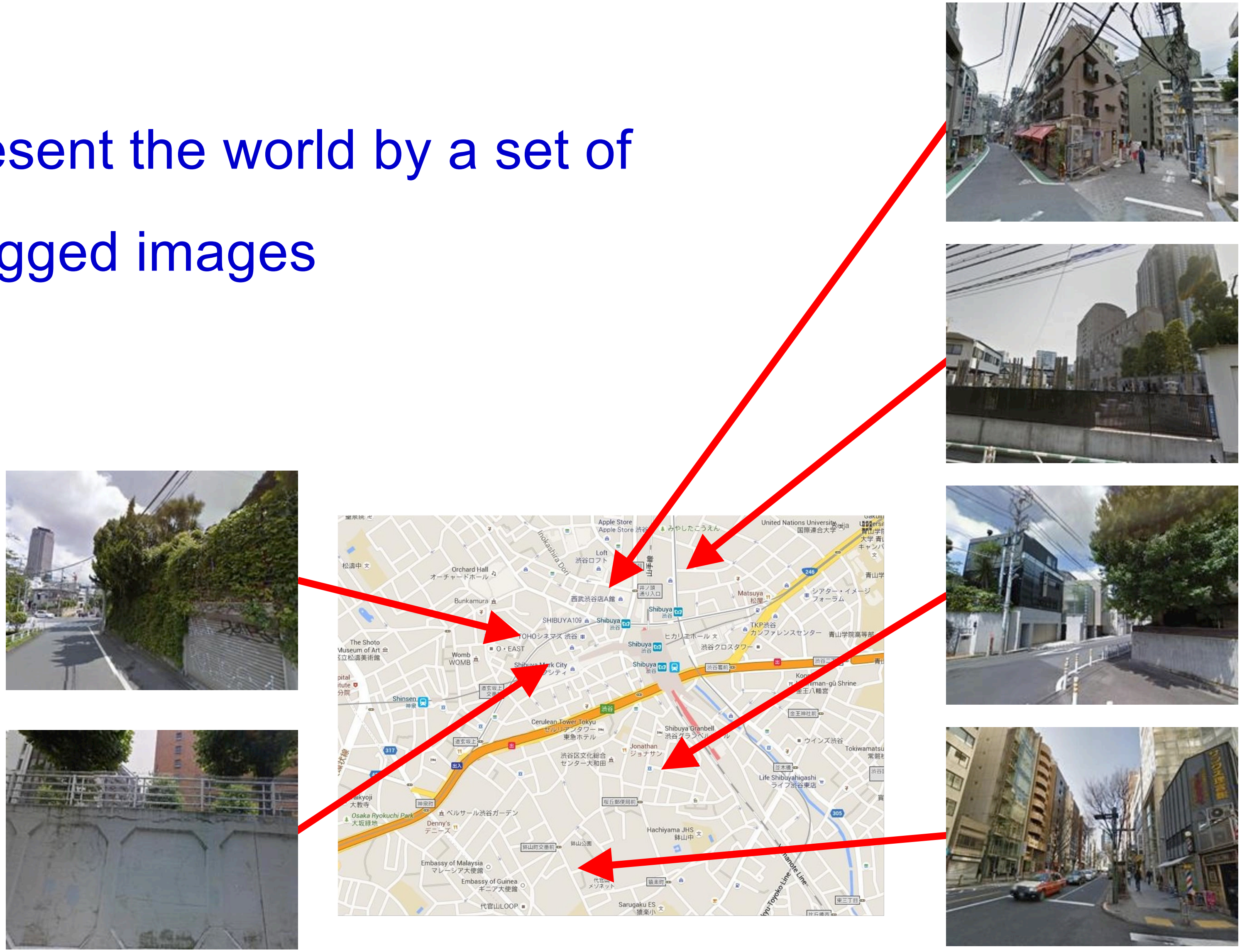
Why is it difficult?

- Lighting changes: Different time of day / year
- Changes in camera viewpoint
- Occluders and ambiguous objects: Trees, cars, pavement...
- Big data: World-scale localization

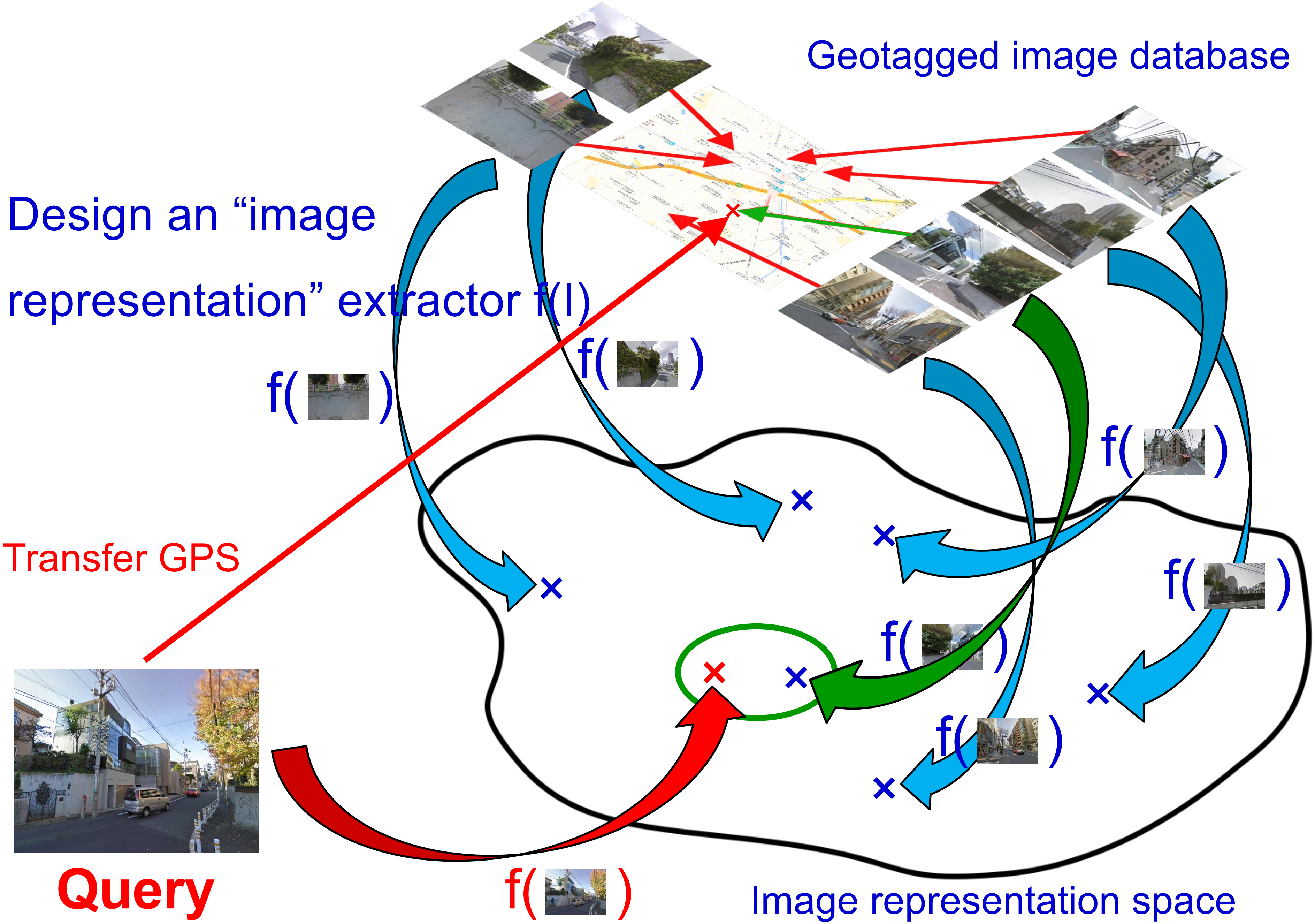


The approach: visual instance recognition

Represent the world by a set of geotagged images

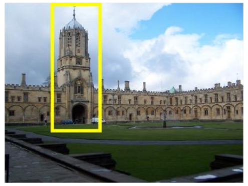


The approach: visual instance recognition



Results on standard retrieval benchmarks

- Test our network on a related task: specific image/object retrieval
- Sets the new state-of-the-art for compact image representations (256-D) on all 3 datasets



Method	Oxford 5k (full)	Oxford 5k (crop)	Paris 6k (full)	Paris 6k (crop)	Holidays (original)	Holidays (rotated)
***Jégou and Zisserman* CVPR14**		47.2			65.7	65.7
***Gordo et al.* CVPR12**					78.3	
***Razavian et al.* ICLR15**	53.3		67.0		74.2	
***Babenko and Lempitsky* ICCV15**	58.9	53.1				80.2
NetVLAD off-the-shelf	53.4	55.5	64.3	67.7	**82.1**	**86.0**
NetVLAD trained	**62.5**	**63.5**	**72.0**	**73.5**	79.9	84.3

Ours

[Radenović et al. arXiv 16, Gordo et al. arXiv 16]

Example result

Query image

Top retrieved image

References learnable representations for large-scale matching

Example: Visual place recognition

R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla and J. Sivic

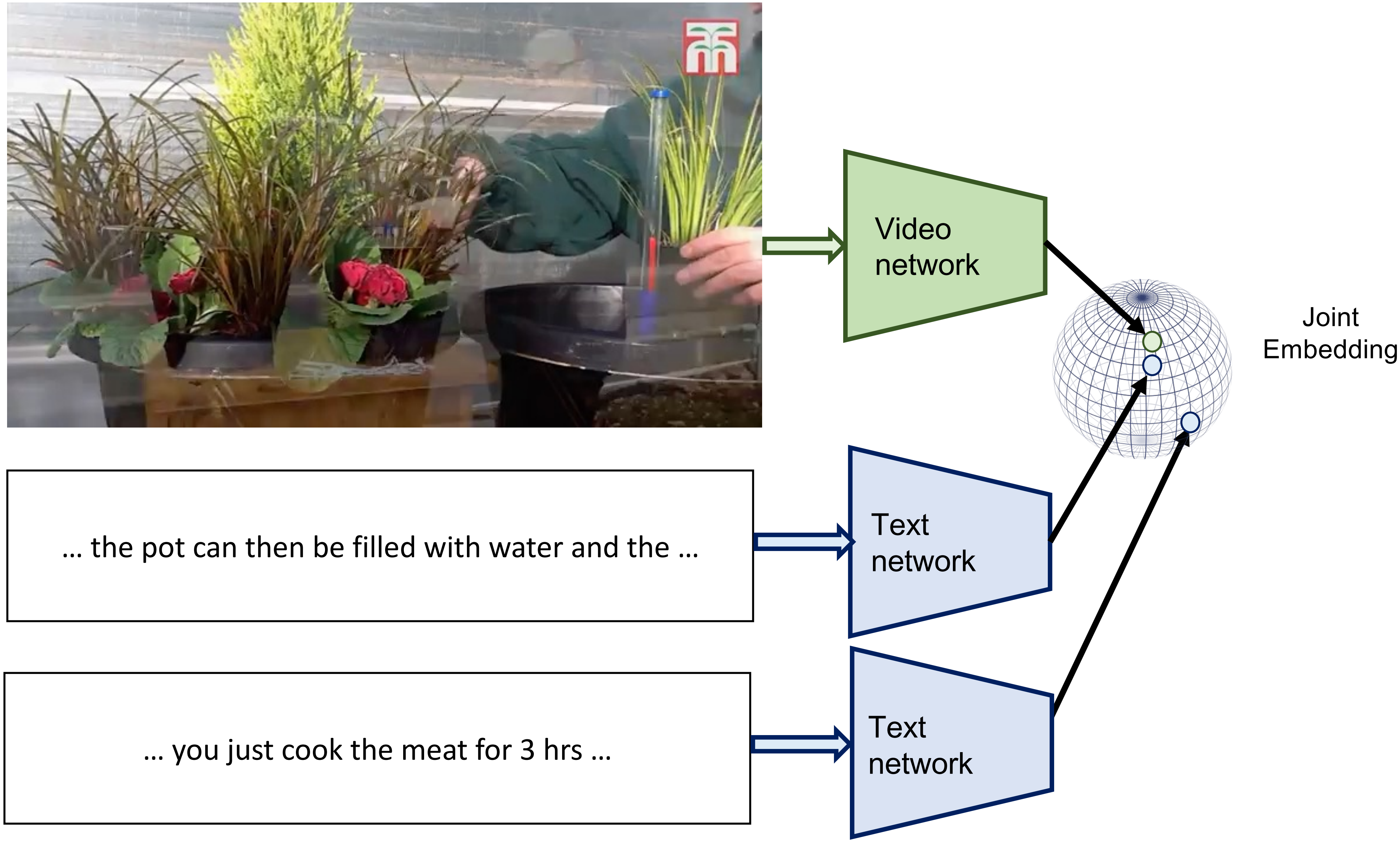
NetVLAD: CNN architecture for weakly-supervised place recognition, CVPR 2016.

See also:

A. Gordo, J. Almazan, J. Revaud, D. Larlus. Deep Image Retrieval: Learning global representations for image search, ECCV 2016

F. Radenovic, G. Tolias and O. Chum, CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples, ECCV 2016

But also: learn joint video and text embedding



See also [Miech et al., 1706.06905, 1804.02516], and e.g. [Frome et al., NIPS 2013] [Gong et al., IJCV 2014]

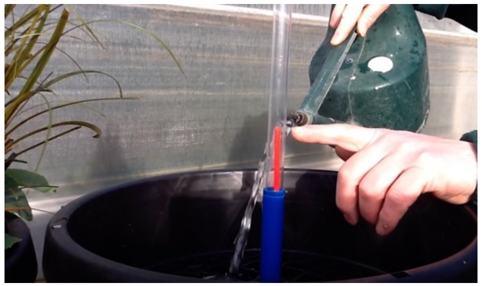
Example loss function: Max-margin triplet loss

$$S_{i,j} = S(X_i, Y_j)$$ (dot product)

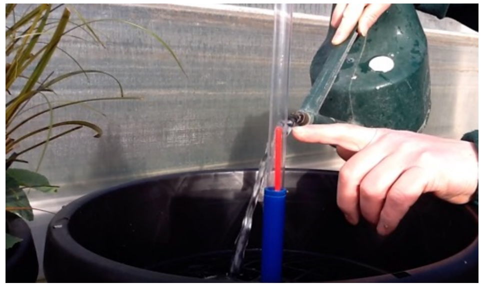
We want: $$\forall (i,j),\ j \neq i,\ S_{i,i} > S_{i,j}$$
$$,\ S_{i,i} > S_{j,i}$$

$$L = \frac{1}{B} \sum_{i=1}^{B} \sum_{j \neq i} \left[\max(0, m + S_{i,j} - S_{i,i}) + \max(0, m + S_{j,i} - S_{i,i}) \right]$$

... filled with water ...



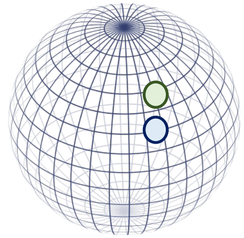
... filled with water ...



... you just cook the meat for 3 hrs ...

Example loss function: Max-margin triplet loss

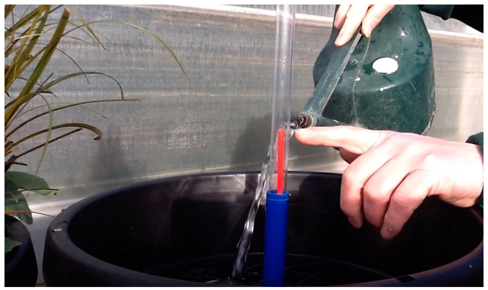
$$S_{i,j} = S(X_i, Y_j)$$ (dot product)



We want: $$\forall (i,j),\ \ j \neq i,\ S_{i,i} > S_{i,j}$$
$$,\ S_{i,i} > S_{j,i}$$

$$L = \frac{1}{B}\sum_{i=1}^{B}\sum_{j\neq i}\left[\max(0, m + S_{i,j} - S_{i,i}) + \max(0, m + S_{j,i} - S_{i,i})\right]$$

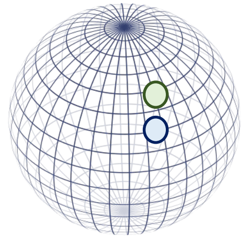
... filled with water ...



... filled with water ...

Example loss function: Max-margin triplet loss

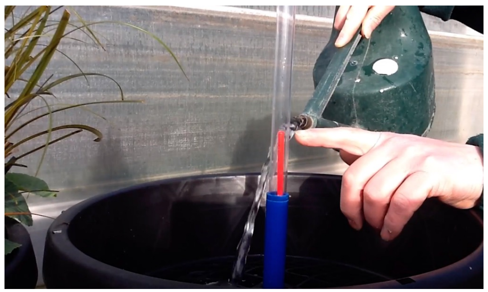
$$S_{i,j} = S(X_i, Y_j)$$ (dot product)



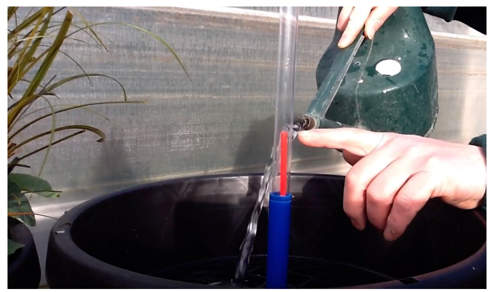
We want: $\forall (i,j),\ j \neq i,\ S_{i,i} > S_{i,j}$

$, S_{i,i} > S_{j,i}$

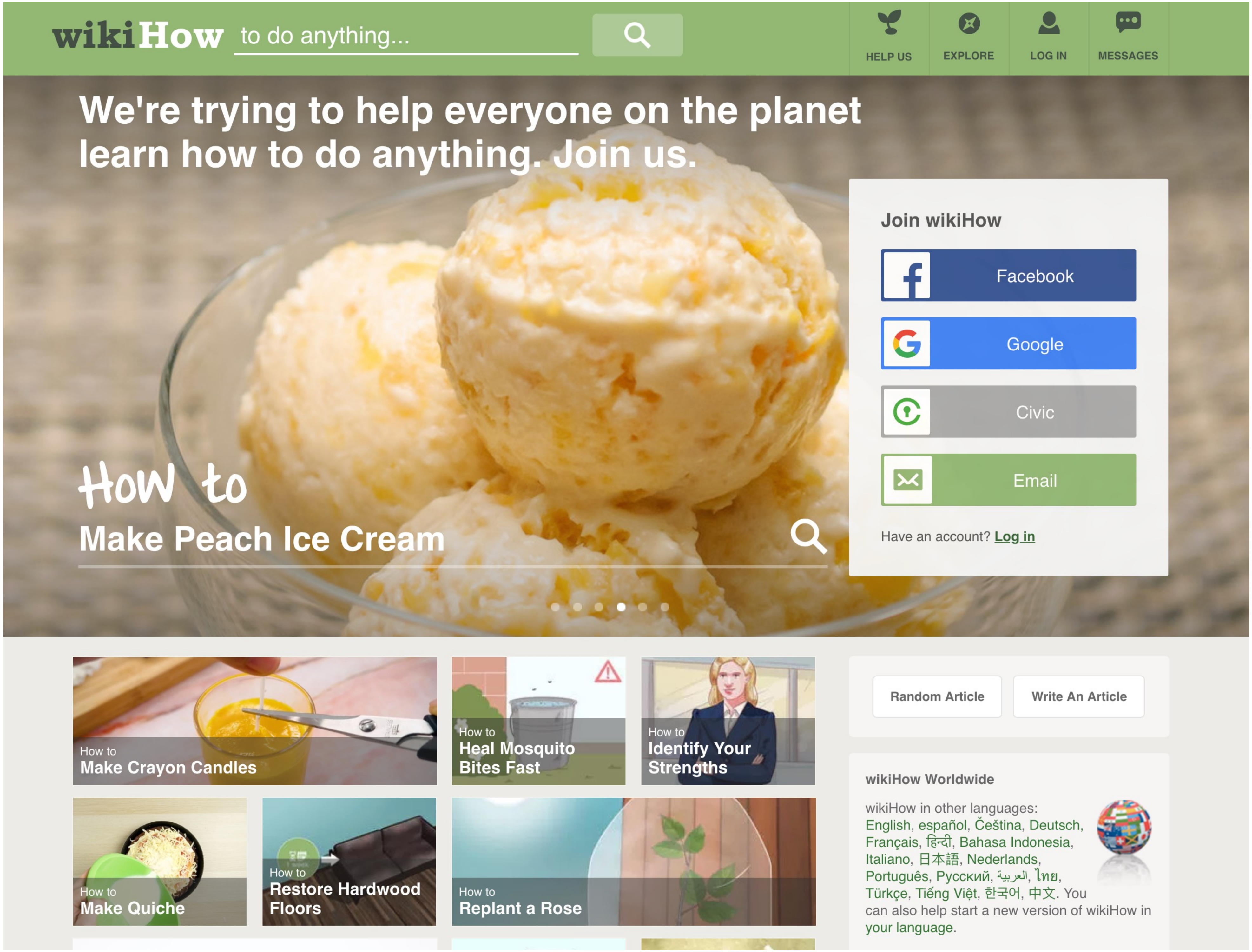
$$L = \frac{1}{B} \sum_{i=1}^{B} \sum_{j \neq i} \left[\max(0, m + S_{i,j} - S_{i,i}) + \max(0, m + S_{j,i} - S_{i,i}) \right]$$



... filled with water ...



... you just cook the meat for 3 hrs ...


wikiHow to do anything...
HELP US
EXPLORE
LOG IN
MESSAGES
We're trying to help everyone on the planet learn how to do anything. Join us.
Join wikiHow
Facebook
Google
Civic
Email
Have an account? Log in
How to
Make Peach Ice Cream
How to
Make Crayon Candles
How to
Heal Mosquito Bites Fast
How to
Identify Your Strengths
Random Article
Write An Article
How to
Make Quiche
How to
Restore Hardwood Floors
How to
Replant a Rose
wikiHow Worldwide
wikiHow in other languages:
English, español, Čeština, Deutsch, Français, हिन्दी, Bahasa Indonesia, Italiano, 日本語, Nederlands, Português, Русский, العربية, ไทย, Türkçe, Tiếng Việt, 한국어, 中文. You can also help start a new version of wikiHow in your language.

Going WikiHow scale – the HowTo100M dataset

23K tasks • 1.3M videos • 130M clip-caption pairs

[Miech, Zhukov, Alayrac, Tapaswi, Laptev and Sivic, ICCV 2019]

Going WikiHow scale

HowTo100M dataset

Dataset	Clips	Captions	Videos	Duration	Source	Year
Charades [42]	10k	16k	10,000	82h	Home	2016
MSR-VTT [52]	10k	200k	7,180	40h	Youtube	2016
YouCook2 [61]	14k	14k	2,000	176h	Youtube	2018
EPIC-KITCHENS [5]	40k	40k	432	55h	Home	2018
DiDeMo [11]	27k	41k	10,464	87h	Flickr	2017
M-VAD [46]	49k	56k	92	84h	Movies	2015
MPII-MD [37]	69k	68k	94	41h	Movies	2015
ANet Captions [22]	100k	100k	20,000	849h	Youtube	2017
TGIF [23]	102k	126k	102,068	103h	Tumblr	2016
LSMDC [38]	128k	128k	200	150h	Movies	2017
How2 [39]	185k	185k	13,168	298h	Youtube	2018
HowTo100M	**136M**	**136M**	**1.221M**	**134,472h**	Youtube	2019

23K tasks • 1.3M videos • 130M clip-caption pairs

Examples of top 4 clip retrieval results given a language query using our model on HowTo100M

Summary – Efficient visual search

1. Efficient matching of local descriptors
   - Approximate nearest neighbor search
   - k-d trees, locality-sensitive hashing (LSH)

2. Aggregate local descriptors into a single vector
   - Bag-of-visual-words, inverted files, query expansion

3. Compact representations for very large-scale search
   - Product quantization (PQ)

4. Learnable representations
   - Neural representations for large-scale visual search
   - Visual search using natural language query

Internships abroad:
ELLIS Unit Prague (ellisprague.eu)

ellis unit | PRAGUE

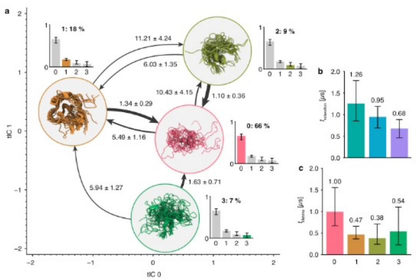
PEOPLE RESEARCH INDUSTRY 4 STUDENTS NEWS & EVENTS JOIN US

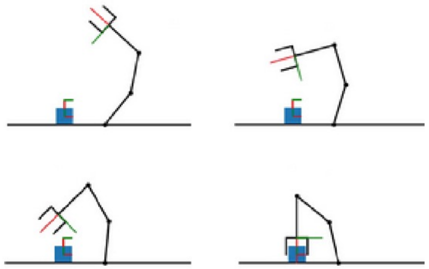
Join us

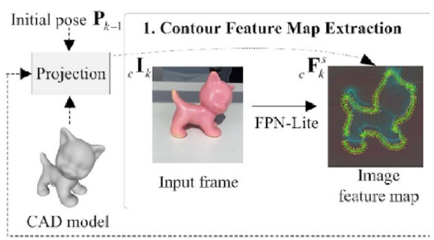
We are looking for **strongly motivated students** with an interest in applying machine learning to **computer vision, robotics** and more broadly **artificial intelligence**.

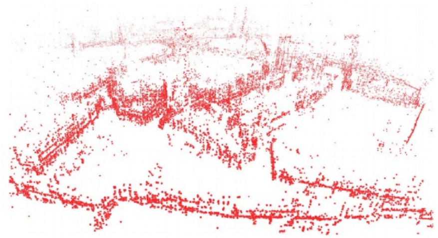
Internships and thesis projects can lead to a PhD in the ELLIS Unit at CIIRC in Prague with the possibility of **spending part of the time at other Units** in the ELLIS network.

Internship and Thesis topics







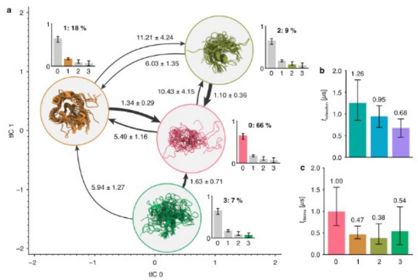


Internships abroad:
ELLIS Unit Prague (ellisprague.eu)

Join us

We are looking for **strongly motivated students** with an interest in applying

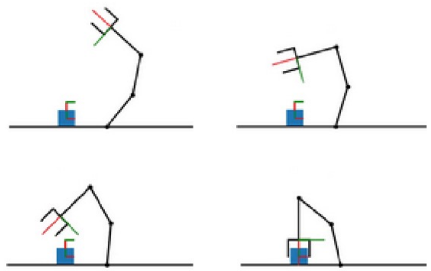
Internship and Thesis topics



Analysis of Molecular Dynamic Simulations for Alzheimer's Disease Research using VAMPnet Neural Networks

Supervisors

Jiri Sedlar, Josef Sivic, Tomas Pajdla, Torsten Sattler, Stanislav Mazurenko, Sergio



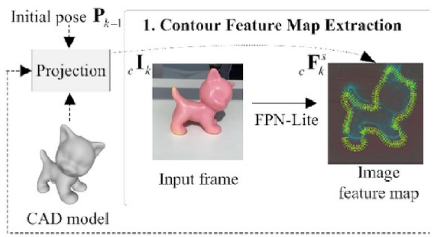
Generative models for robot motion representation

Supervisors

Mederic Fourmy, Vladimir Petrik, Josef Sivic

Motivation

The goal of this project is to use similar architecture to generate the motion of the robot given the prompt defined as the start



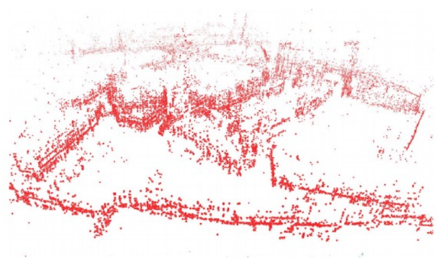
Deep active contours for object pose refinement and tracking

Supervisors

Mederic Fourmy, Vladimir Petrik, Josef Sivic

Motivation

Object 6D pose estimation and tracking are hard problems due to the wide variety of



Learning Local Features from Generative Image Models

Supervisors

Torsten Sattler

Motivation

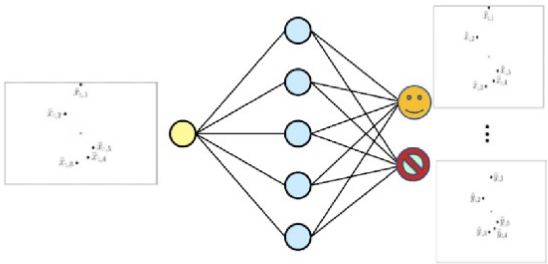
Local features play an important role in many 3D computer vision algorithms, including visual localization and Structure-

Internships abroad:
ELLIS Unit Prague (ellisprague.eu)

Generating „Realistic“ Camera Views of 3D Models

Supervisors
Torsten Sattler

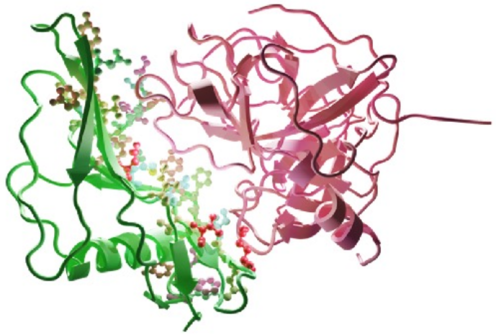
Motivation
Modern deep-based computer vision approaches, e.g., local features, 3D reconstruction, etc., need large amounts of training data. Often, such methods need annotations that are expensive to obtain from real-world images, e.g., accurate scene geometry, pixel-level correspondences



Learning to solve multiple-view geometry in Computer Vision

Supervisors
Tomas Pajdla

Motivation
We aim at using machine learning to address long-standing problems in multiple view geometry that traditional techniques cannot solve. Previous methods for computing camera geometry from image matches can coped efficiently with only



Machine learning for the design of protein-protein interactions

Supervisors
Anton Bushuiev, Roman Bushuiev, Petr Kouba, Jiri Sedlar, Jiri Damborsky, Stanislav Mazurenko, Josef Sivic

Motivation
Proteins are large molecules that drive nearly all processes in living cells. The analysis of protein-protein interactions (PPIs) and their design unlocks application

Internships abroad:
ELLIS Unit Prague (ellisprague.eu)

ellis unit | PRAGUE

PEOPLE RESEARCH INDUSTRY 4 STUDENTS NEWS & EVENTS JOIN US

Join us

We are looking for **strongly motivated students** with an interest in applying machine learning to **computer vision, robotics** and more broadly **artificial intelligence**.

Internships and thesis projects can lead to a PhD in the ELLIS Unit at CIIRC in Prague with the possibility of **spending part of the time at other Units** in the ELLIS network.

Possibility of a **joint Phd** with the Willow team in Paris or other ELLIS Units across Europe
(see “ellis.eu/units” and “ellis.eu/phd-postdoc”).
Contact: josef.sivic@cvut.cz / josef.sivic@inria.fr

ELLIS Phd with two advisors in Europe

https://ellis.eu

ELLIS Society

News Events Research People Sponsorship About Contact

e l l i s

European Laboratory for Learning and Intelligent Systems

ELLIS - the European Laboratory for Learning and Intelligent Systems - is a pan-European AI network of excellence which focuses on fundamental science, technical innovation and societal impact. Founded in 2018, ELLIS builds upon machine learning as the driver for modern AI and aims to secure Europe's sovereignty in this competitive field by creating a multi-centric AI research laboratory. ELLIS wants to ensure that the highest level of AI research is performed in the open societies of Europe and follows a three-pillar strategy to achieve that.

ELLIS Phd with two advisors in Europe

https://ellis.eu/phd-postdoc