Visual Recognition and Machine Learning Summer School

Paris 2013

Large scale visual search

Josef Sivic

http://www.di.ens.fr/~josef

INRIA, WILLOW, ENS/INRIA/CNRS UMR 8548

Departement d'Informatique, Ecole Normale Supérieure, Paris

Slides with A. Zisserman

Also with some slides from: O. Chum, K. Grauman, I. Laptev, S. Lazebnik, B. Leibe, D. Lowe, J. Philbin, J. Ponce, D. Nister, C. Schmid, N. Snavely

Outline

1. Local invariant features (C. Schmid)

2. Matching and recognition with local features (J. Sivic)

3. **Large scale visual search (J. Sivic)**

4. Very large scale visual indexing (C. Schmid)

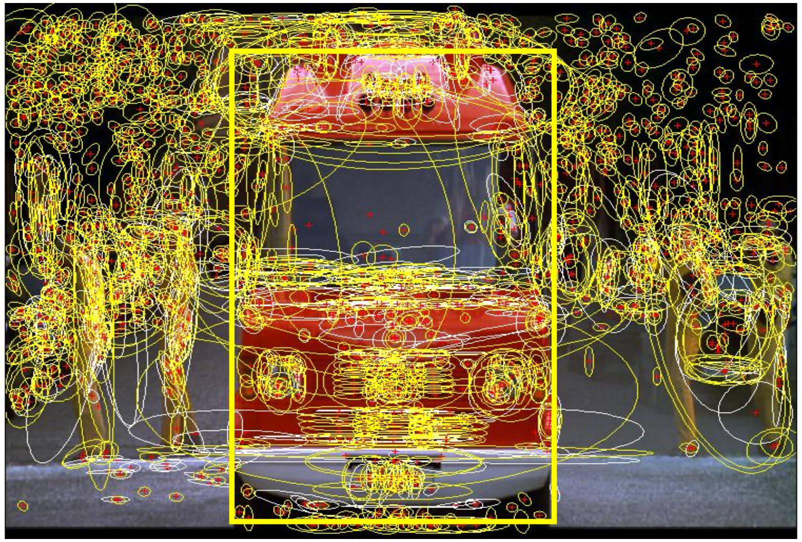
Practical session – Instance-level recognition and search

Outline

Efficient visual search

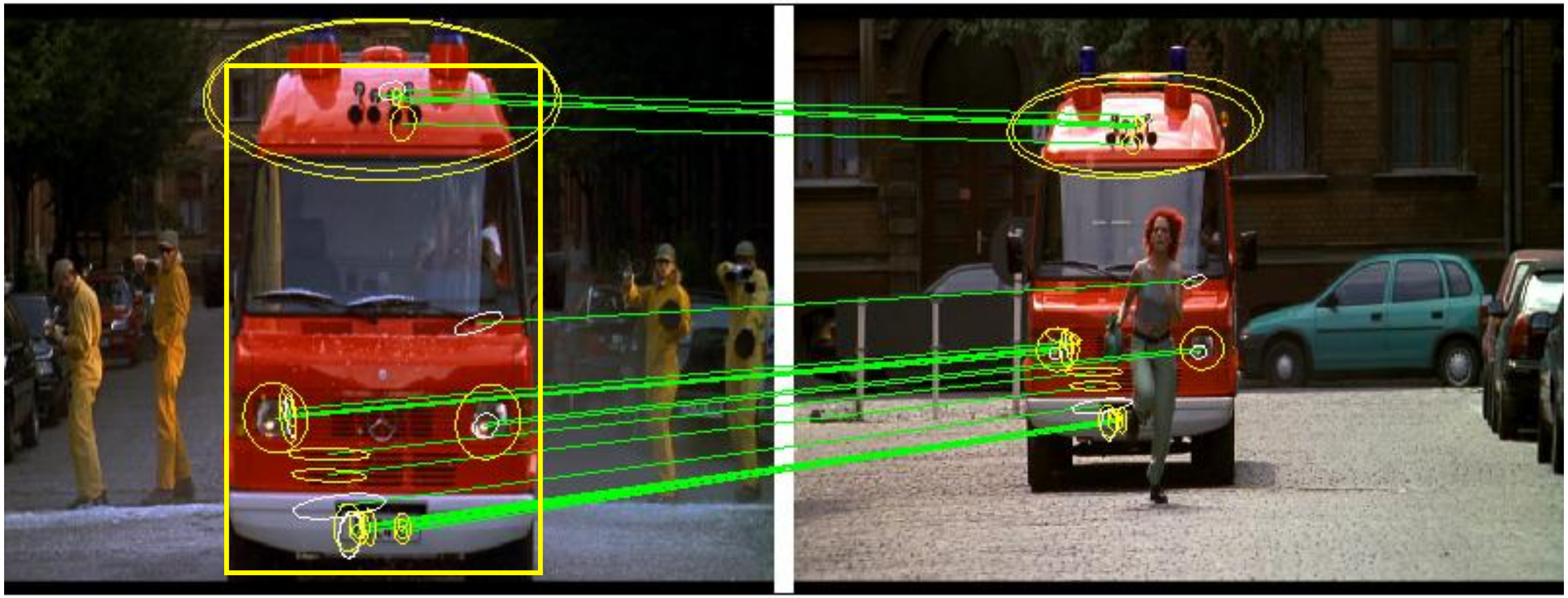
- Approximate nearest neighbour matching
- Bag-of-visual-words representation
- Efficient visual search and extensions
- Beyond bag-of-visual-words representations

Example: Two images again



1000+ descriptors per image

Match regions between frames using SIFT descriptors and spatial consistency



Multiple regions overcome problem of partial occlusion

Approach - review

1. Establish tentative (or putative) correspondence based on local appearance of individual features (now)

2. Verify matches based on semi-local / global geometric relations (You have just seen this).

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?

    - $10, 10^2, 10^3, \ldots, 10^7, \ldots 10^{10}, \ldots$ images?

History of “large scale” visual search with local regions

Schmid and Mohr ’97	– 1k images
Sivic and Zisserman’03	– 5k images
Nister and Stewenius’06	– 50k images (1M)
Philbin et al.’07	– 100k images
Chum et al.’07 + Jegou et al.’07	– 1M images
Chum et al.’08	– 5M images
Jegou et al. ’09	– 10M images
Jegou et al. ’10 and ’12	– 100M images

All on a single machine in ~ 1 second

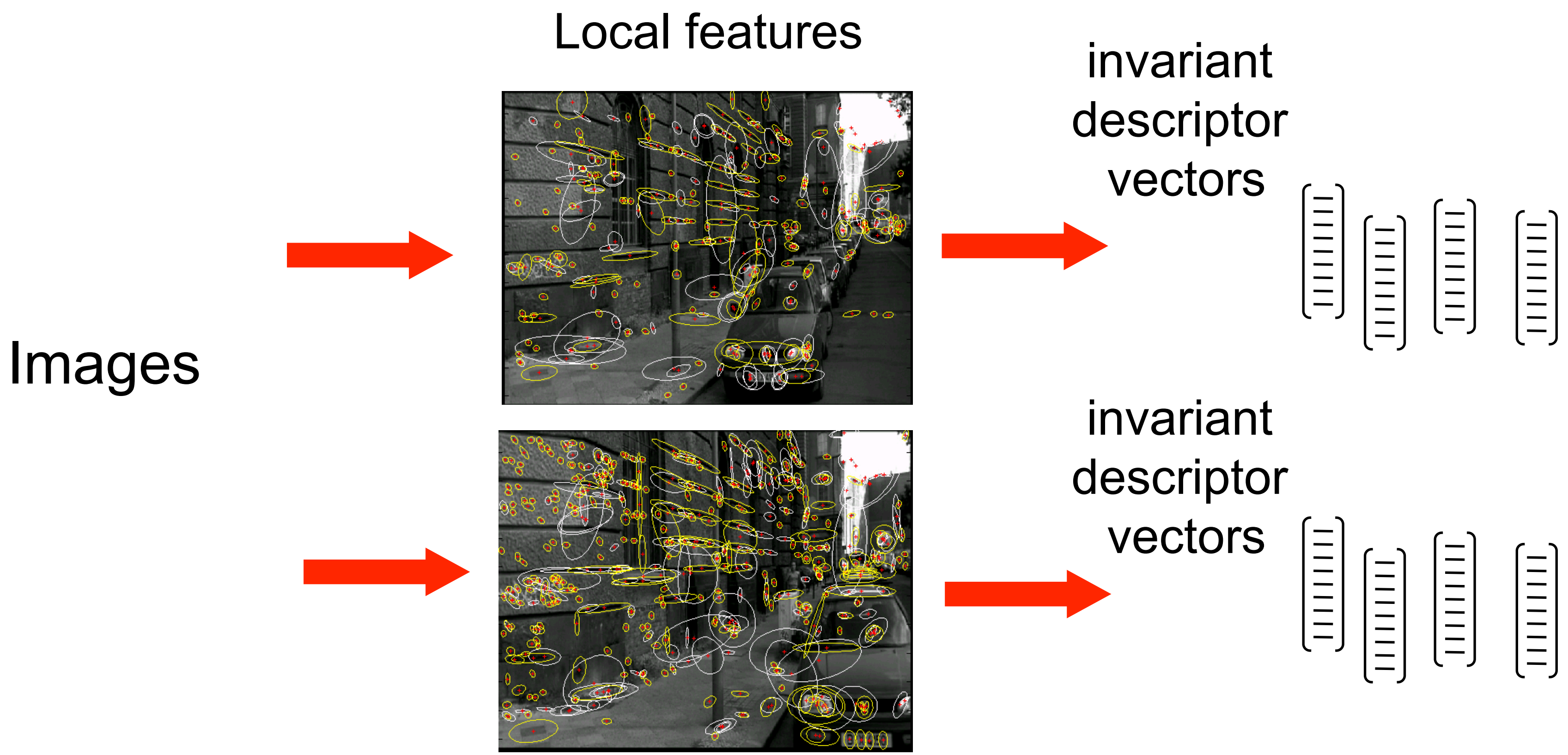
Two strategies

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

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

   (Bag-of-words representation)

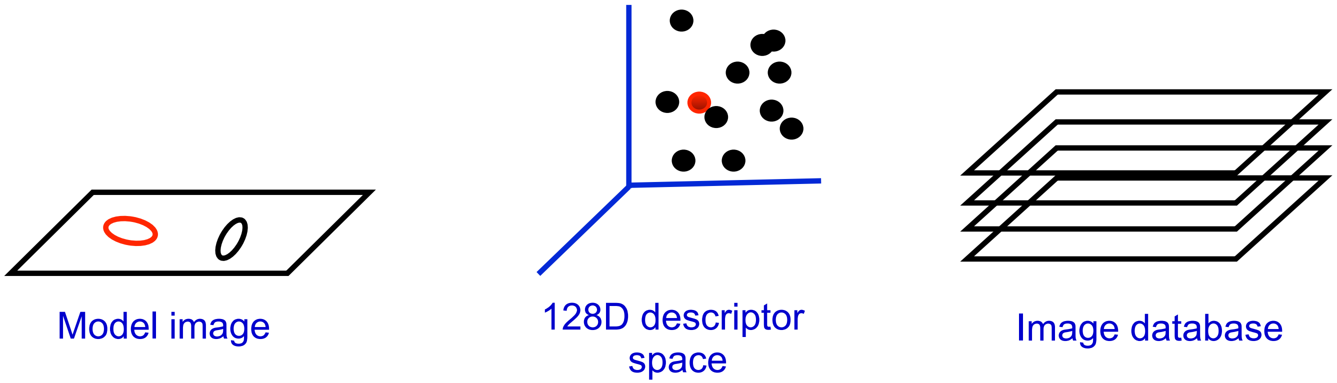
Strategy I: Efficient approximate NN search



1. Compute local features in each image independently (Part 1)
2. "Label" each feature by a descriptor vector based on its intensity (Part 1)
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 (Part 2)

Finding nearest neighbour vectors

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



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:

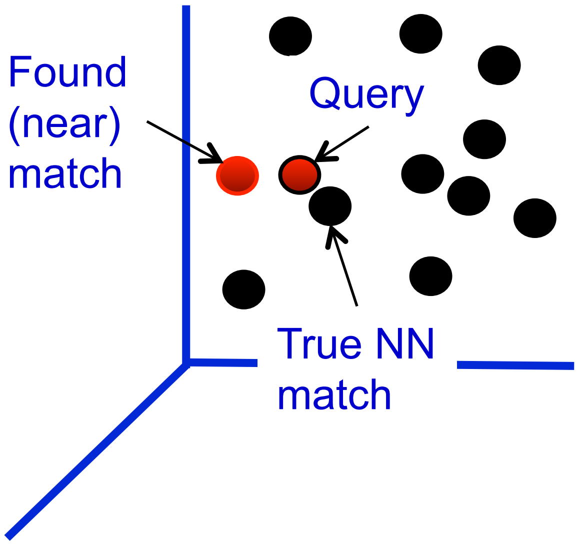
- Matching two images (N=1), each having 1000 SIFT descriptors
  Nearest neighbors search: 0.4 s (2 GHz CPU, implemenation in C)
- Memory footprint: 1000 * 128 = 128kB / image

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

Finding *approximate* nearest neighbour vectors

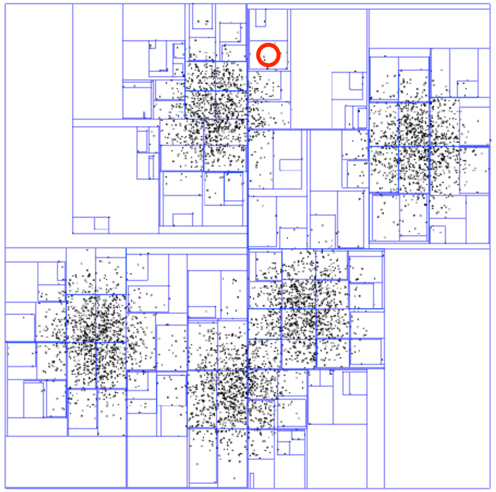
- Approximate method is not guaranteed to find the nearest neighbour.

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



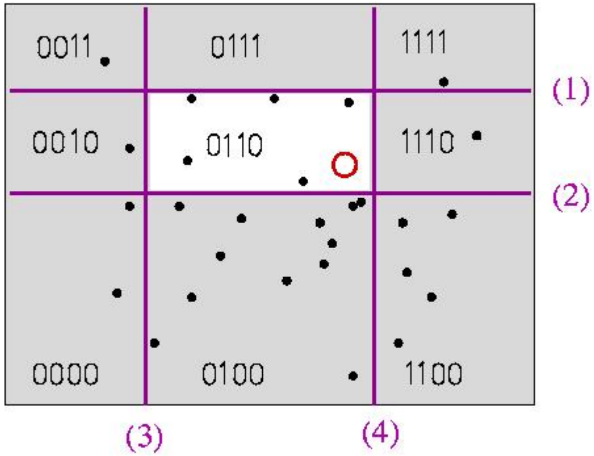
128D descriptor space

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

Comparison of approximate NN-search methods

http://www.cs.ubc.ca/~lowe/papers/09muja.pdf

FAST APPROXIMATE NEAREST NEIGHBORS WITH AUTOMATIC ALGORITHM CONFIGURATION

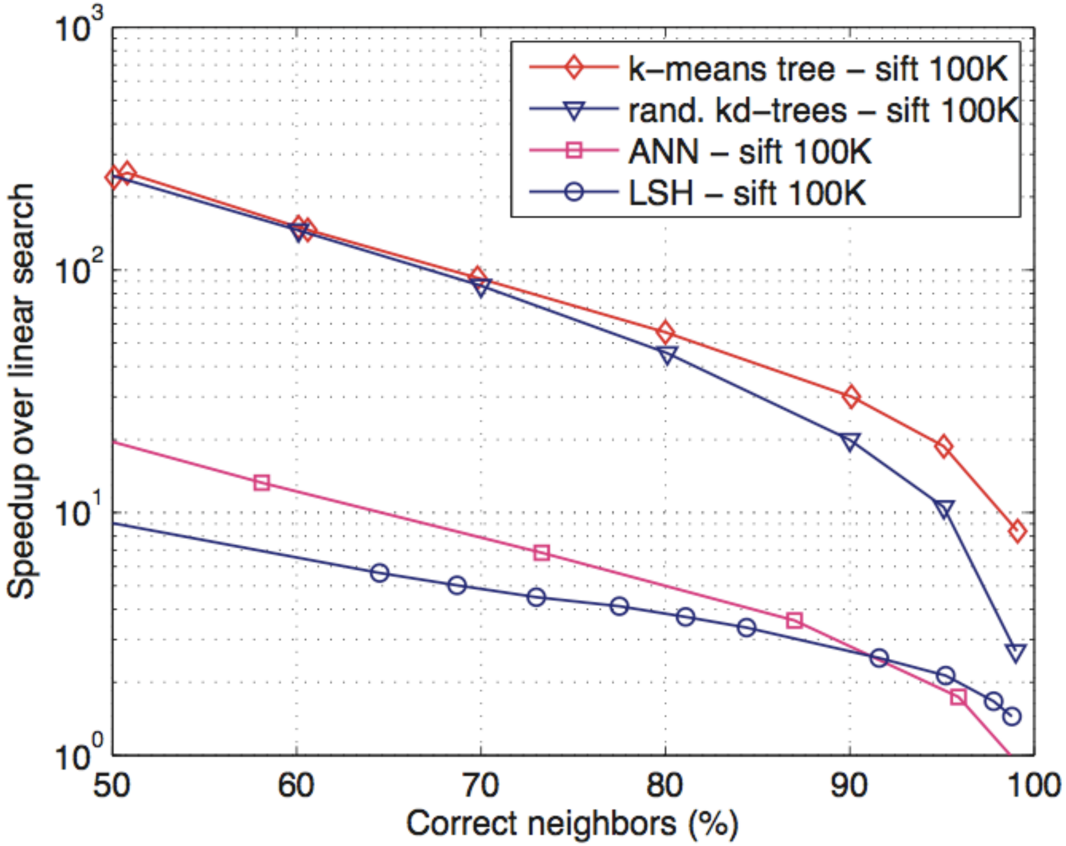
Marius Muja, David G. Lowe
Computer Science Department, University of British Columbia, Vancouver, B.C., Canada
mariusm@cs.ubc.ca, lowe@cs.ubc.ca

Keywords: nearest-neighbors search, randomized kd-trees, hierarchical k-means tree, clustering.

Abstract: For many computer vision problems, the most time consuming component consists of nearest neighbor matching in high-dimensional spaces. There are no known exact algorithms for solving these high-dimensional problems that are faster than linear search. Approximate algorithms are known to provide large speedups with only minor loss in accuracy, but many such algorithms have been published with only minimal guidance on selecting an algorithm and its parameters for any given problem. In this paper, we describe a system that answers the question, "What is the fastest approximate nearest-neighbor algorithm for my data?" Our system will take any given dataset and desired degree of precision and use these to automatically determine the best algorithm and parameter values. We also describe a new algorithm that applies priority search on hierarchical k-means trees, which we have found to provide the best known performance on many datasets. After testing a range of alternatives, we have found that multiple randomized k-d trees provide the best performance for other

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 also next lecture by C. Schmid

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

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

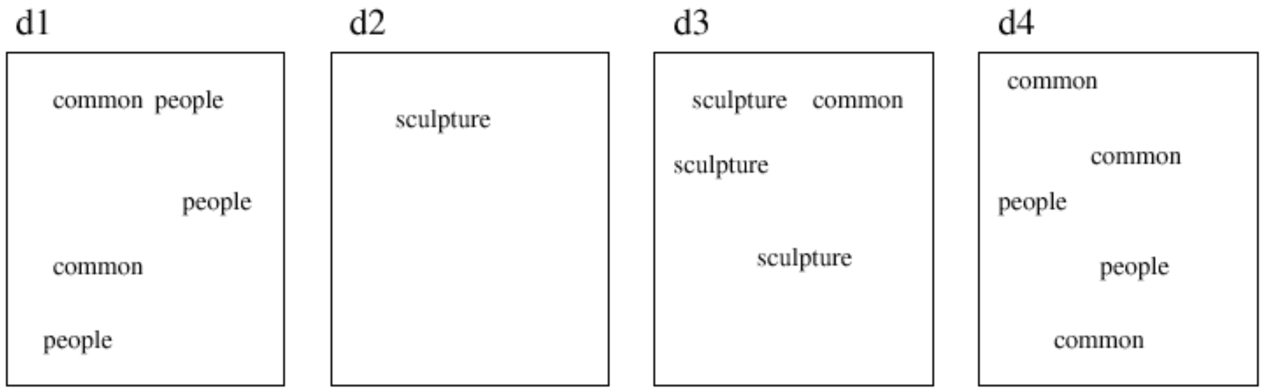
- However, memory footprint can be still high.

  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:

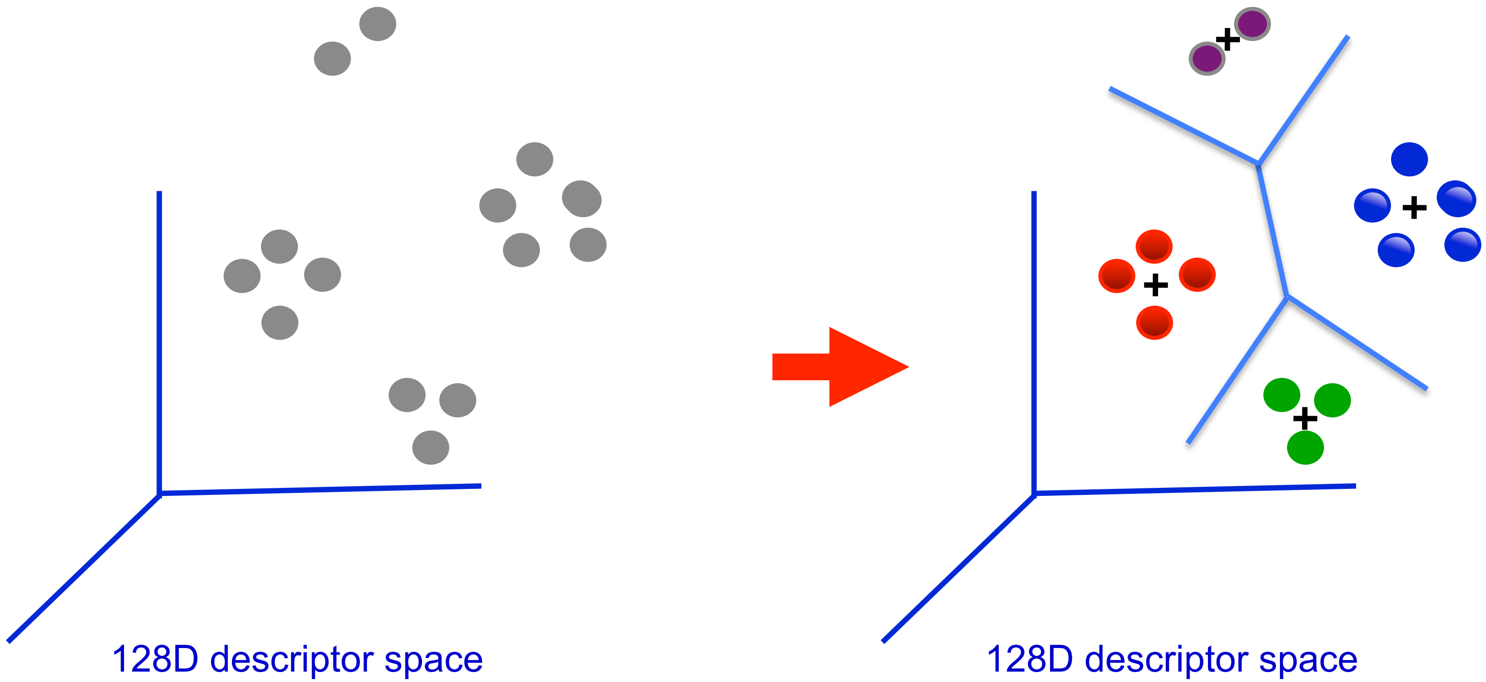


Inverted file:

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



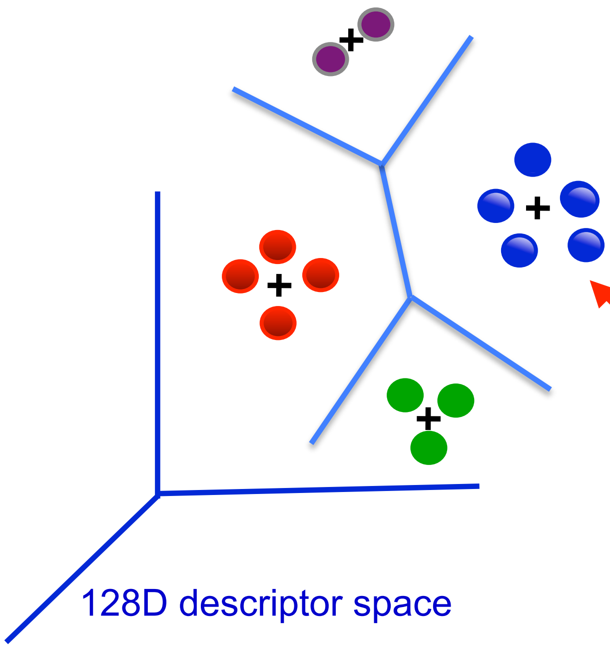
Vector quantize descriptors

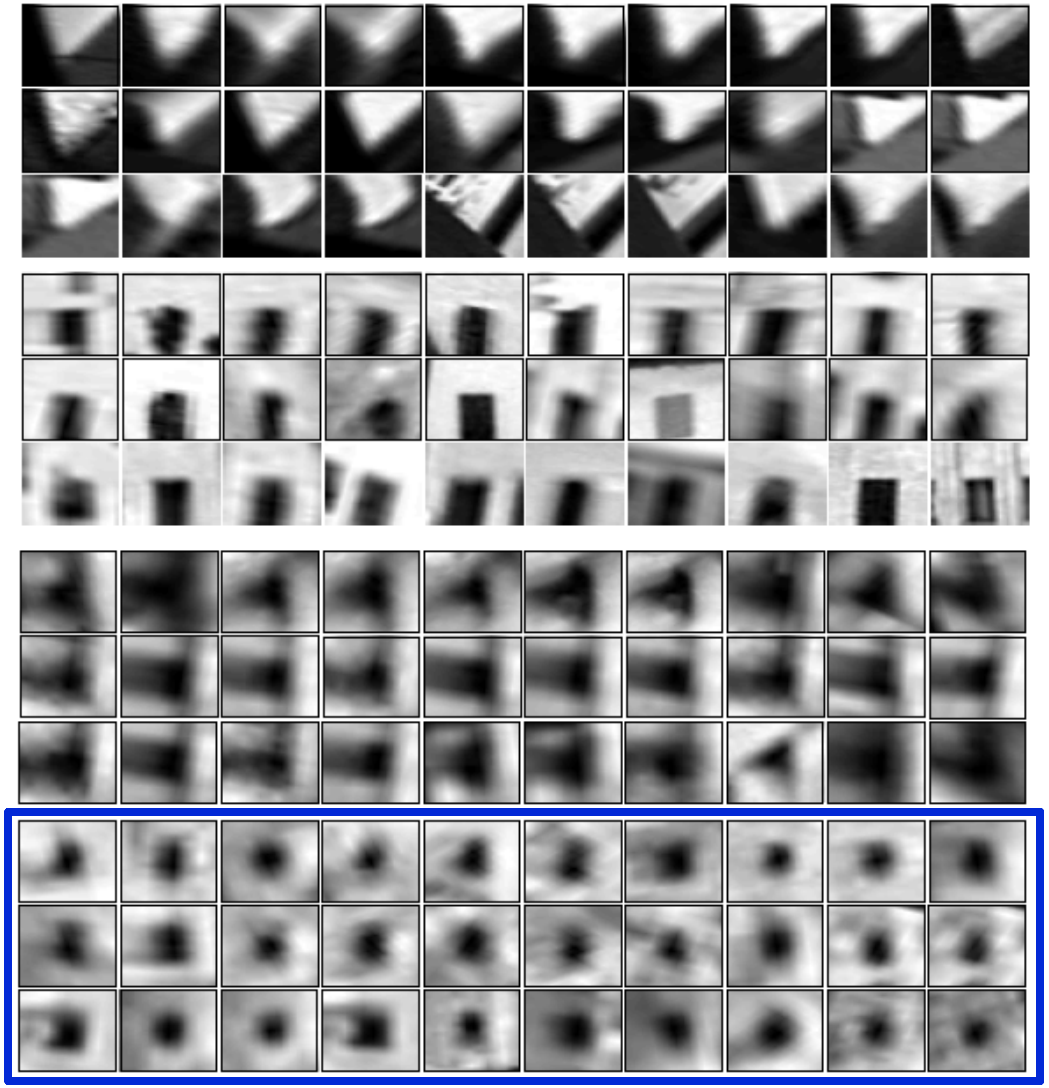
- Compute SIFT features from a subset of images
- K-means clustering (need to choose K)

[Sivic and Zisserman, ICCV 2003]

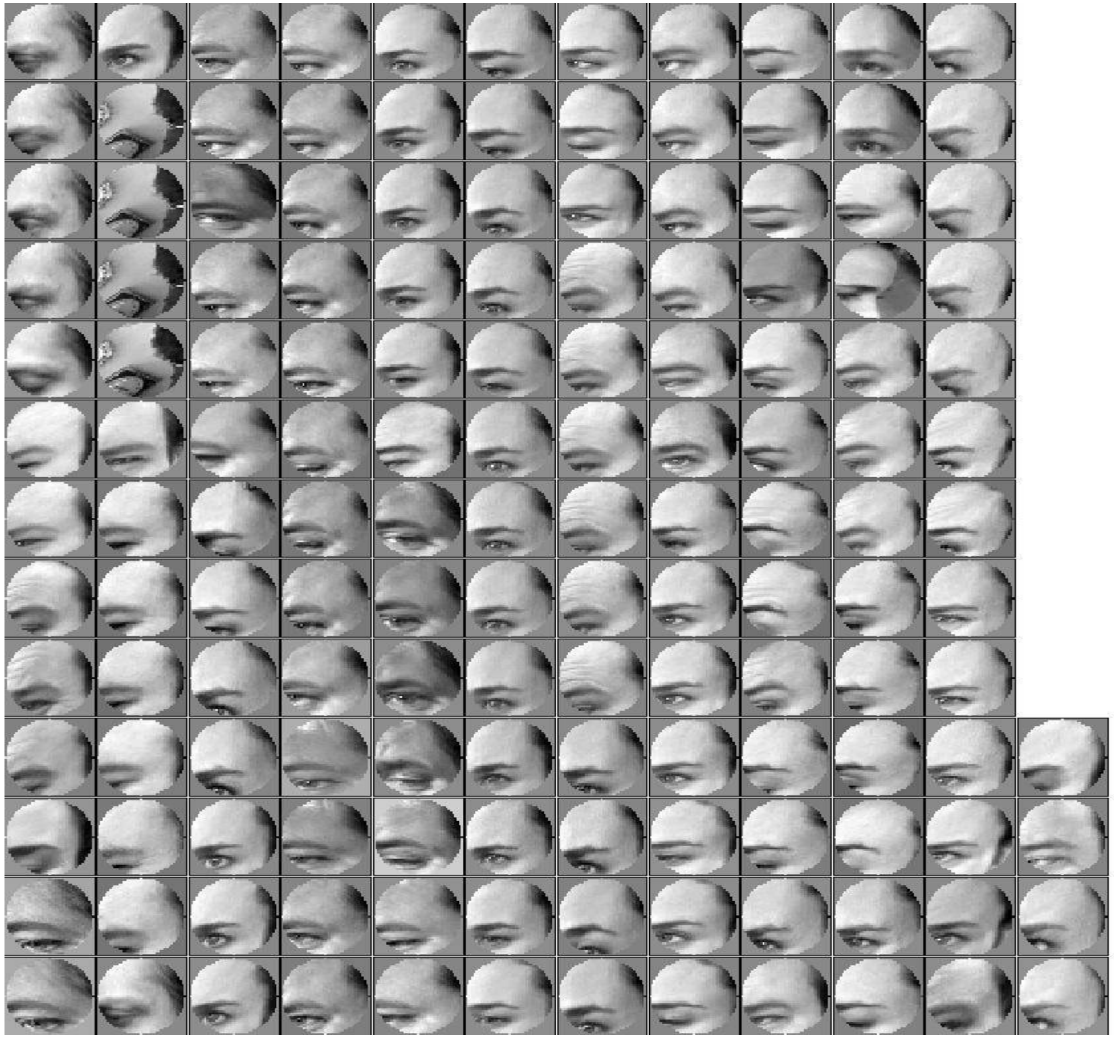
Visual words

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



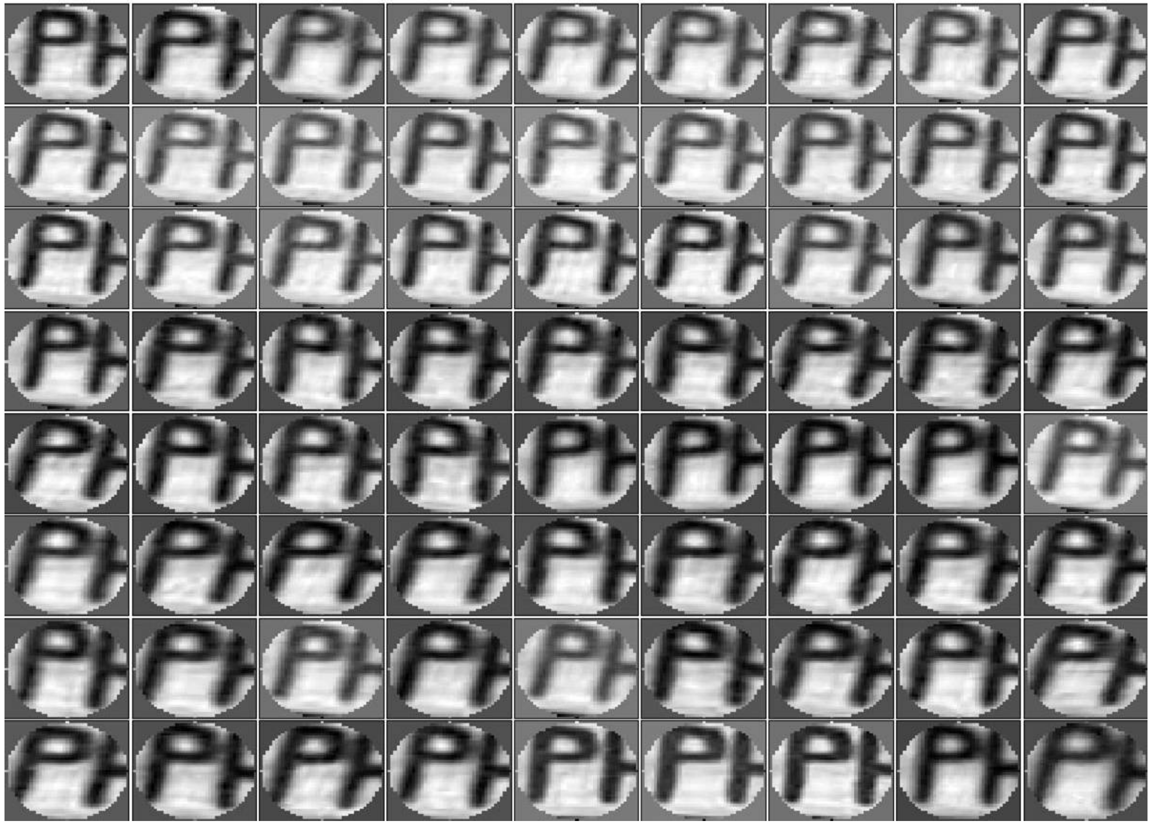


Samples of visual words (clusters on SIFT descriptors):



More specific example

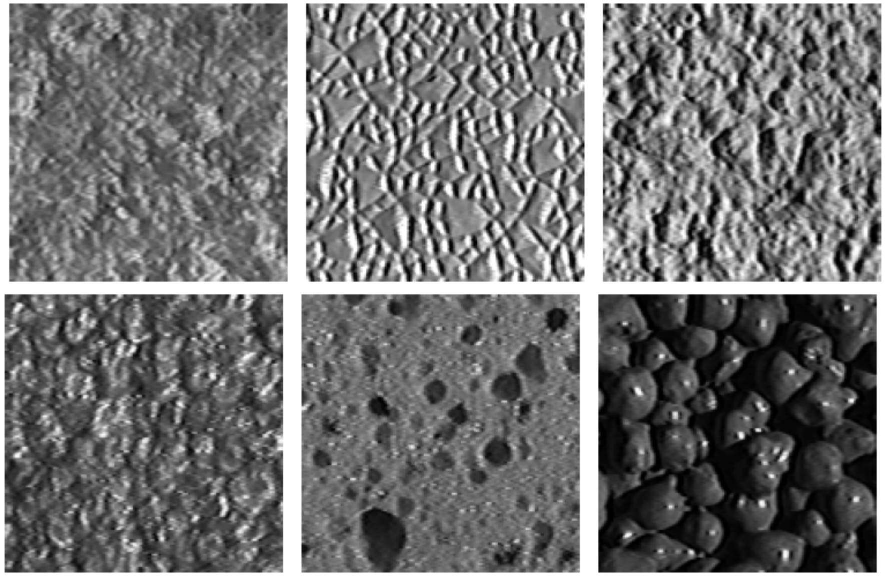
Samples of visual words (clusters on SIFT descriptors):



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;

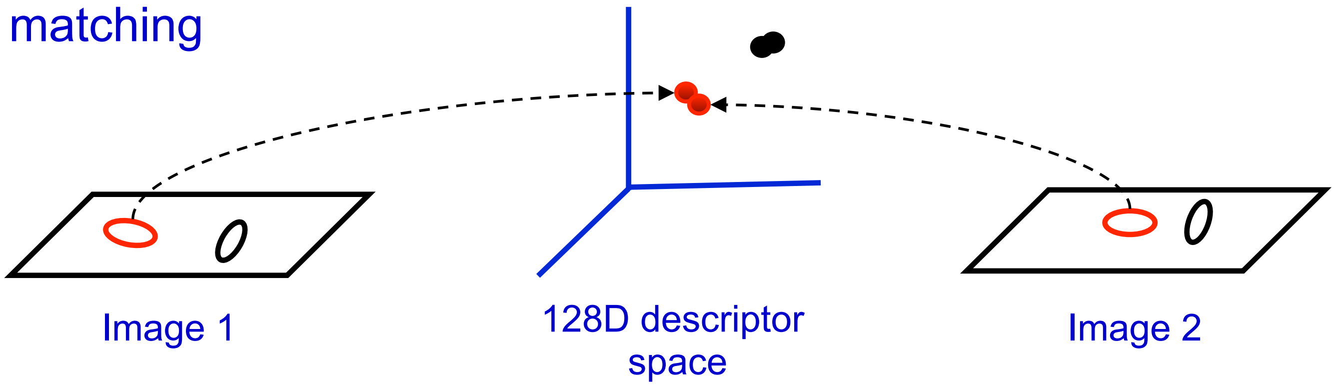
Slide: Grauman&Leibe

Visual words: quantize descriptor space

Sivic and Zisserman, ICCV 2003

Nearest neighbour matching

- expensive to
do for all frames

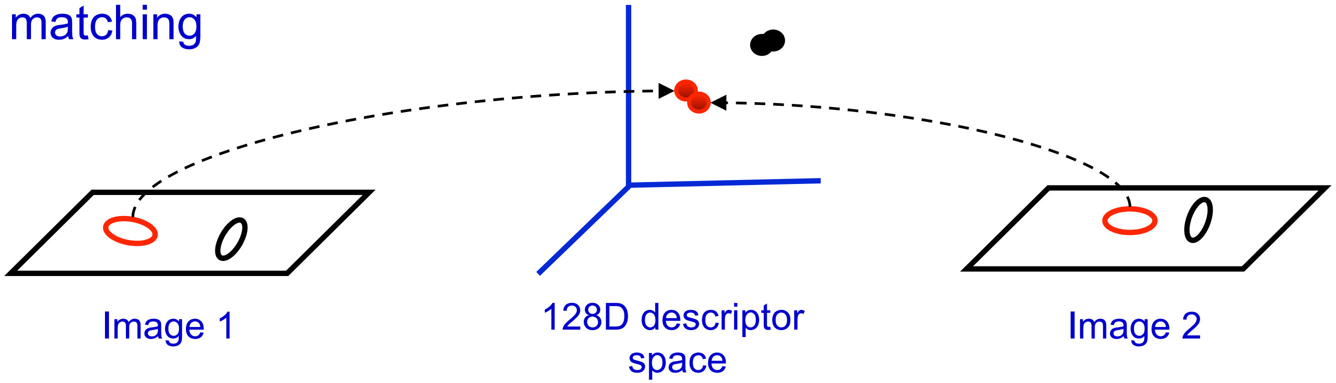


Visual words: quantize descriptor space

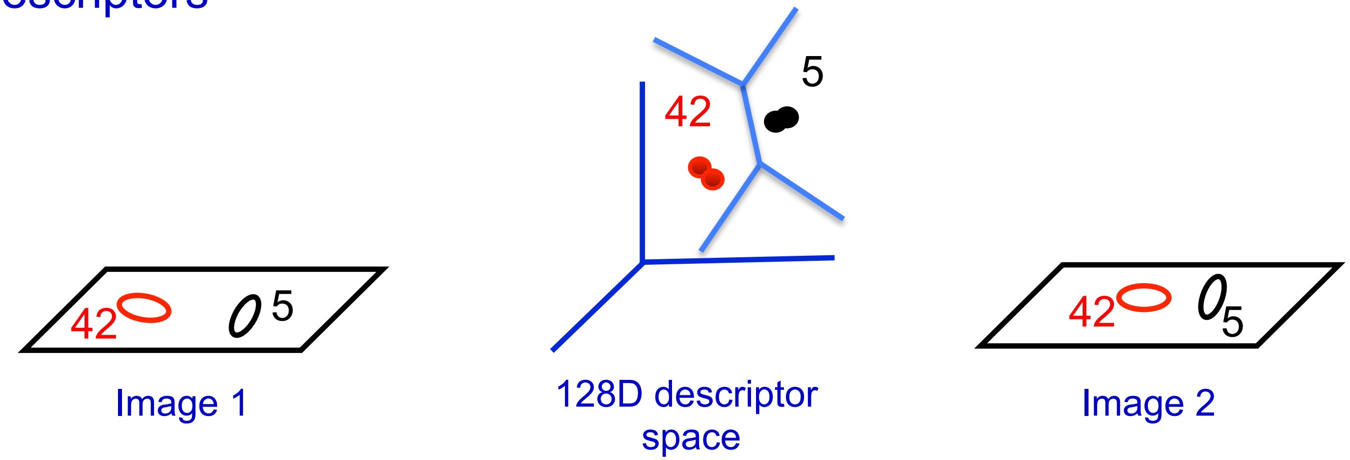
Sivic and Zisserman, ICCV 2003

Nearest neighbour matching

- expensive to do for all frames



Vector quantize descriptors

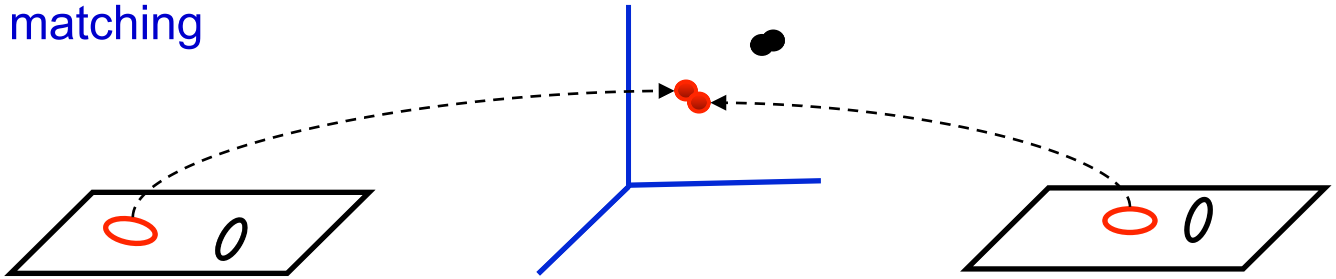


Visual words: quantize descriptor space

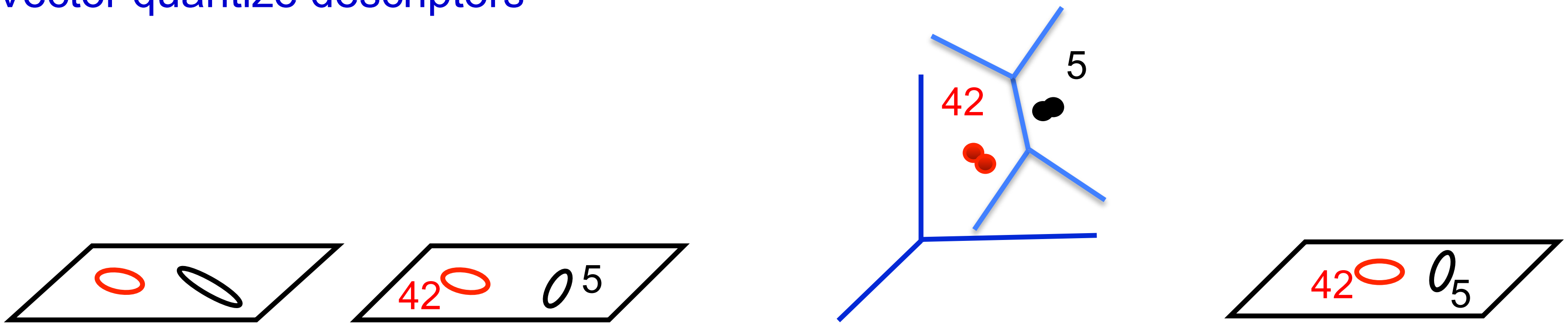
Sivic and Zisserman, ICCV 2003

Nearest neighbour matching

- expensive to do for all frames



Vector quantize descriptors

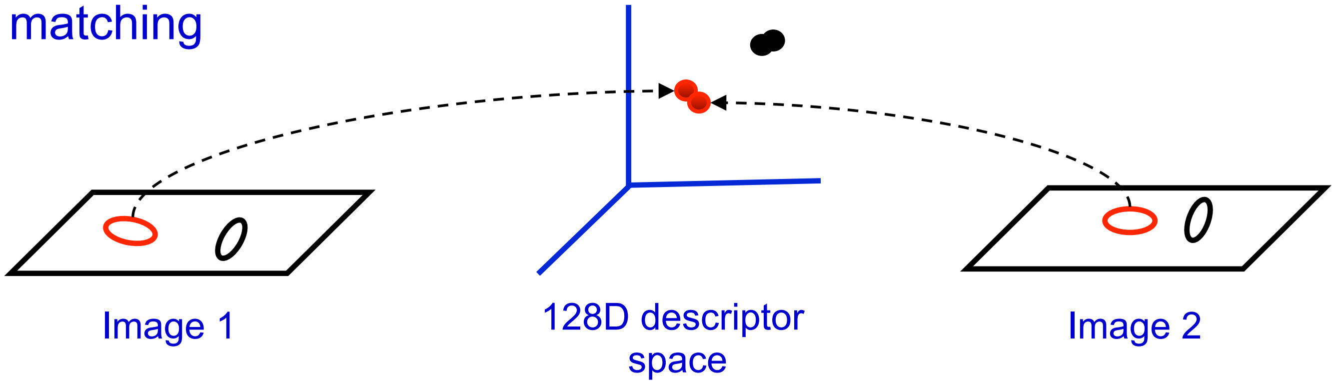


Visual words: quantize descriptor space

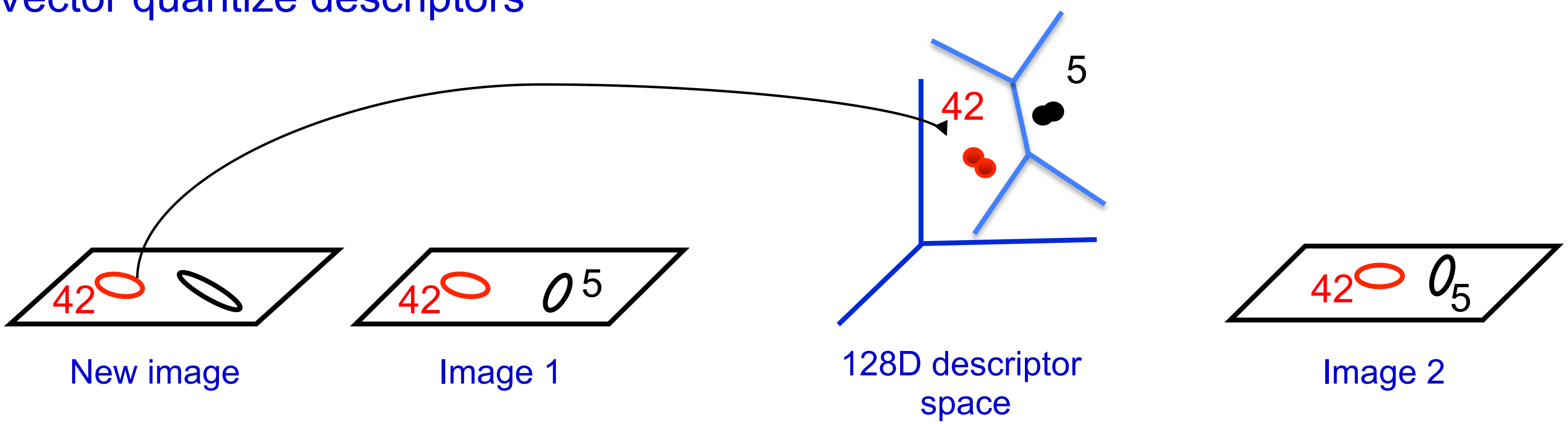
Sivic and Zisserman, ICCV 2003

Nearest neighbour matching

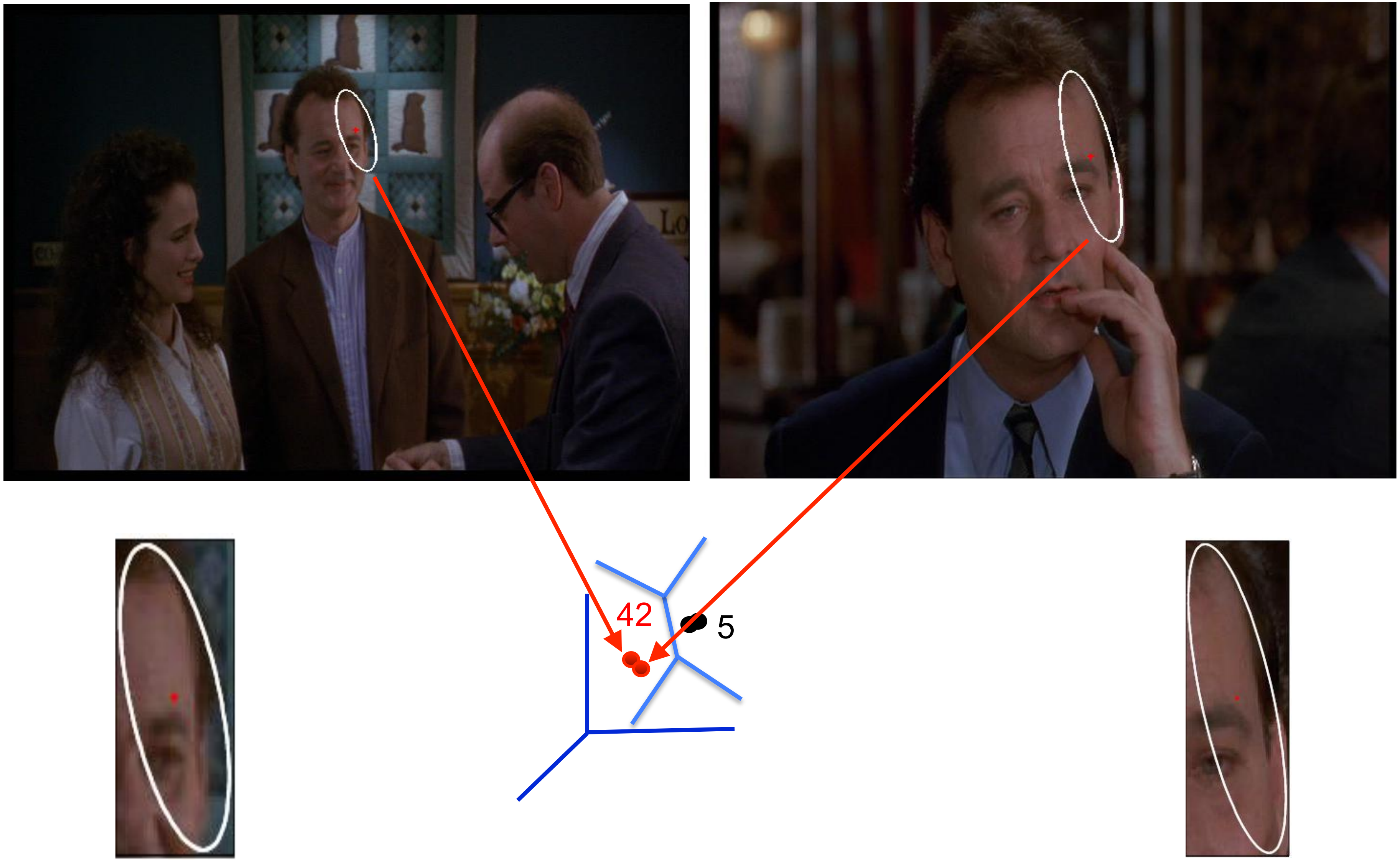
- expensive to do for all frames



Vector quantize descriptors



Vector quantize the descriptor space (SIFT)

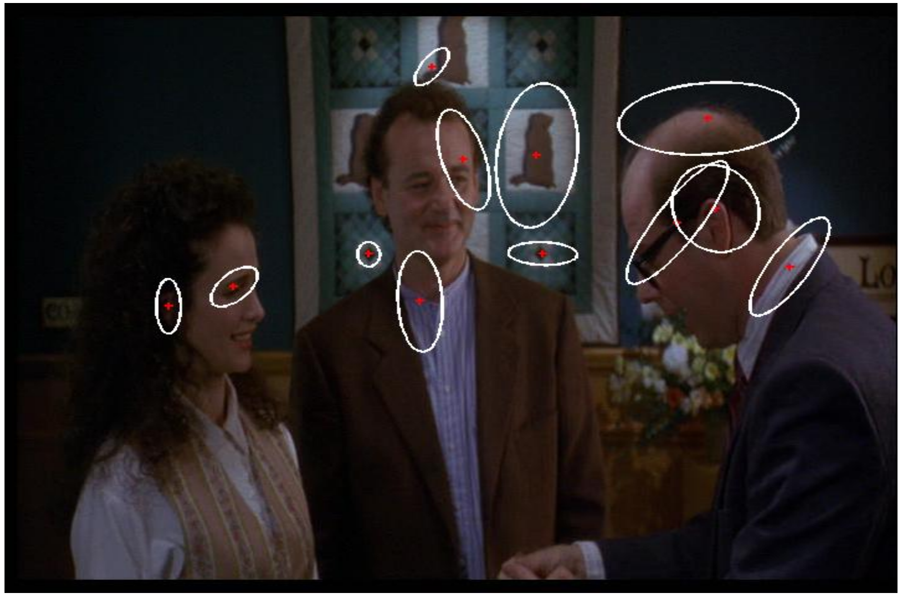


The same visual word

Representation: bag of (visual) words

Visual words are ‘iconic’ image patches or fragments

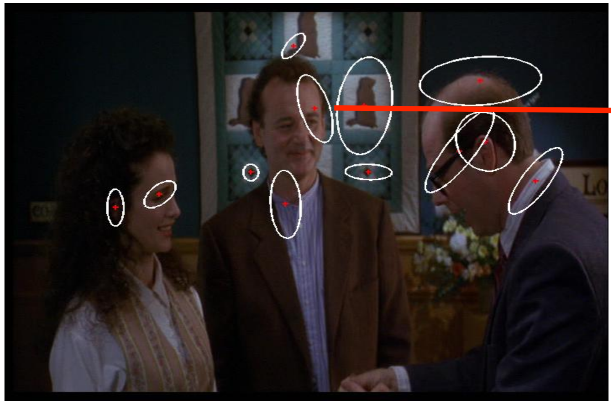
- represent their frequency of occurrence
- but not their position



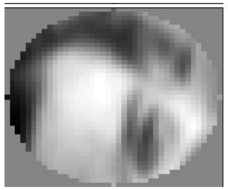
Image

Colelction of visual words

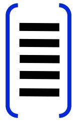
Offline: Assign visual words and compute histograms for each image



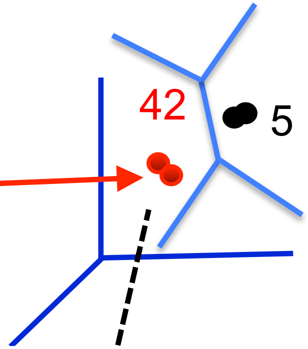
Detect patches



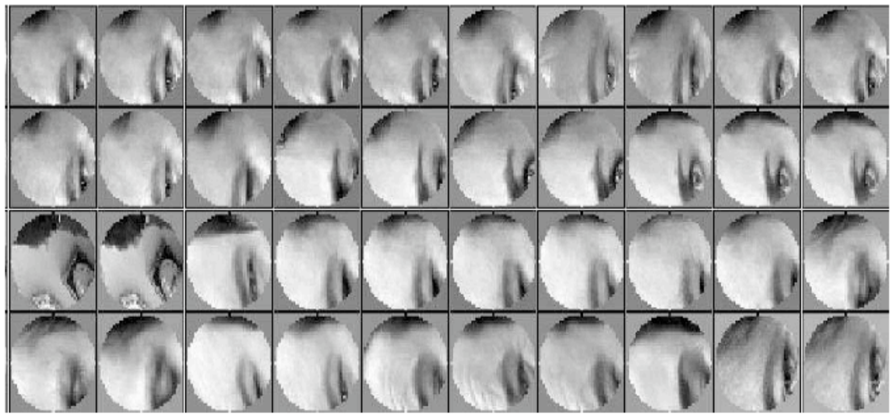
Normalize patch



Compute SIFT descriptor



Find nearest cluster center



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

Represent image as a sparse histogram of visual word occurrences

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 that share at least one word

At run time

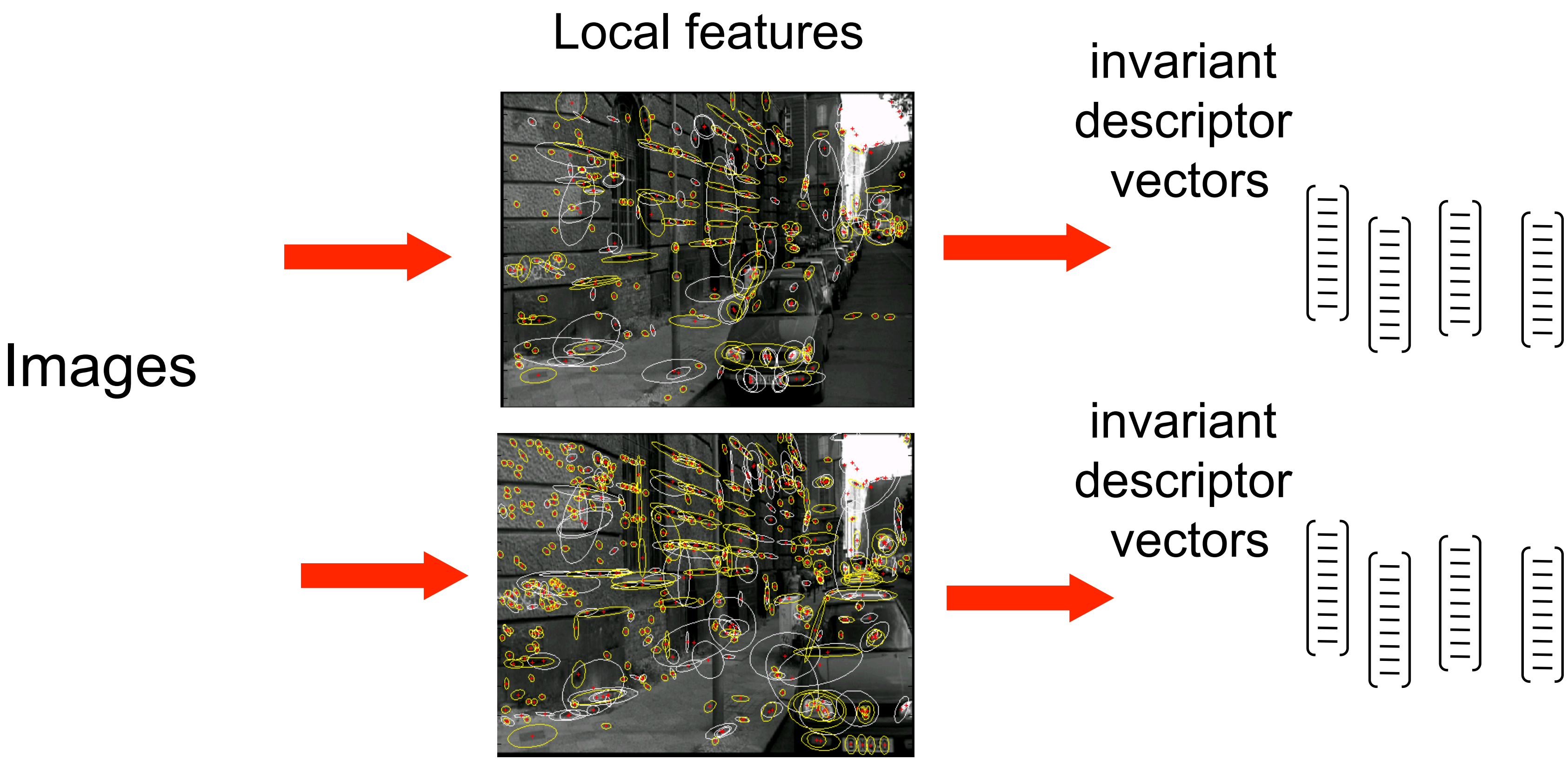
frame #5

frame #10

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

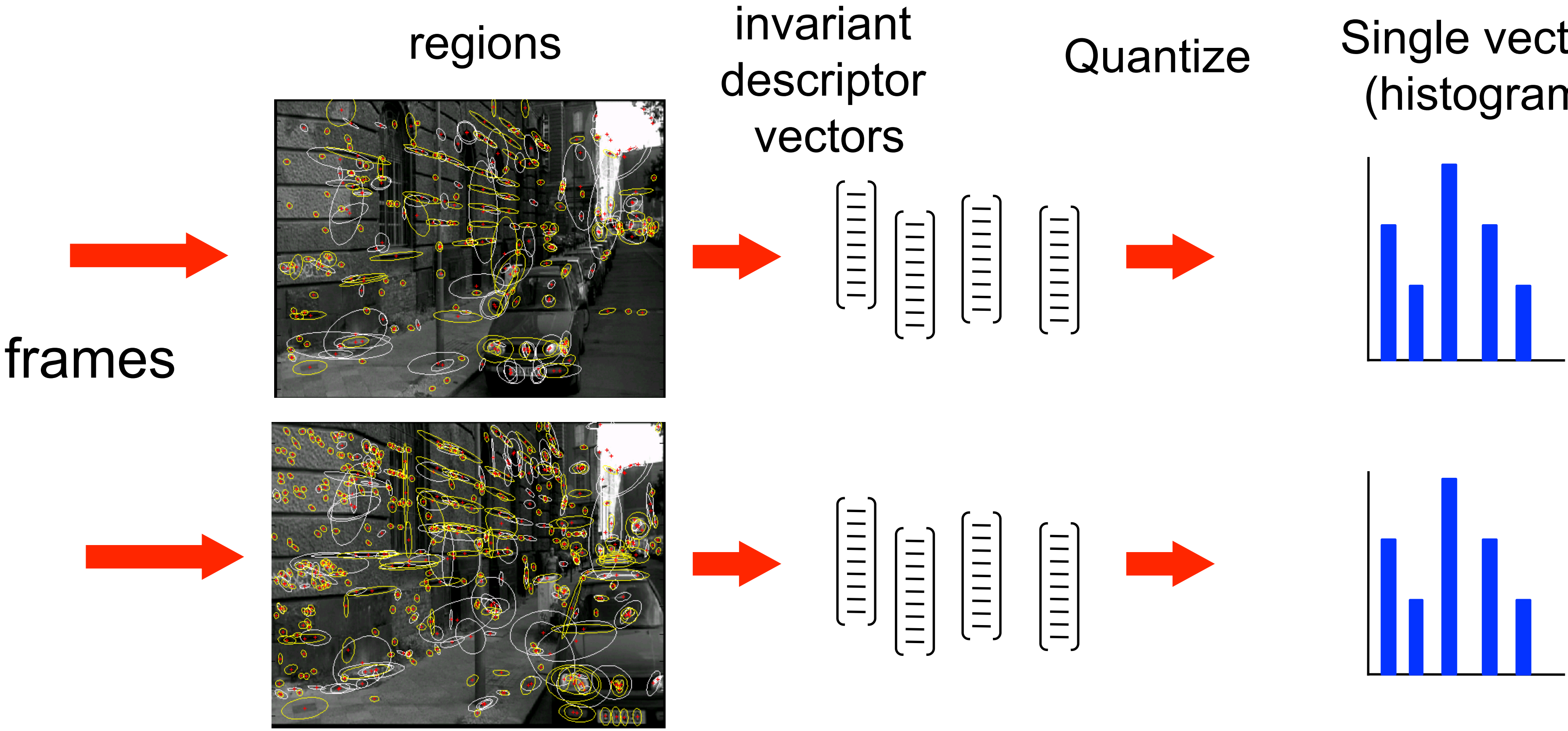
- Score each image by the (weighted) number of common visual words (tentative correspondences)
- Worst case complexity is linear in the number of images N
- In practice, it is linear in the length of the lists (<< N)

Strategy I: Efficient approximate NN search



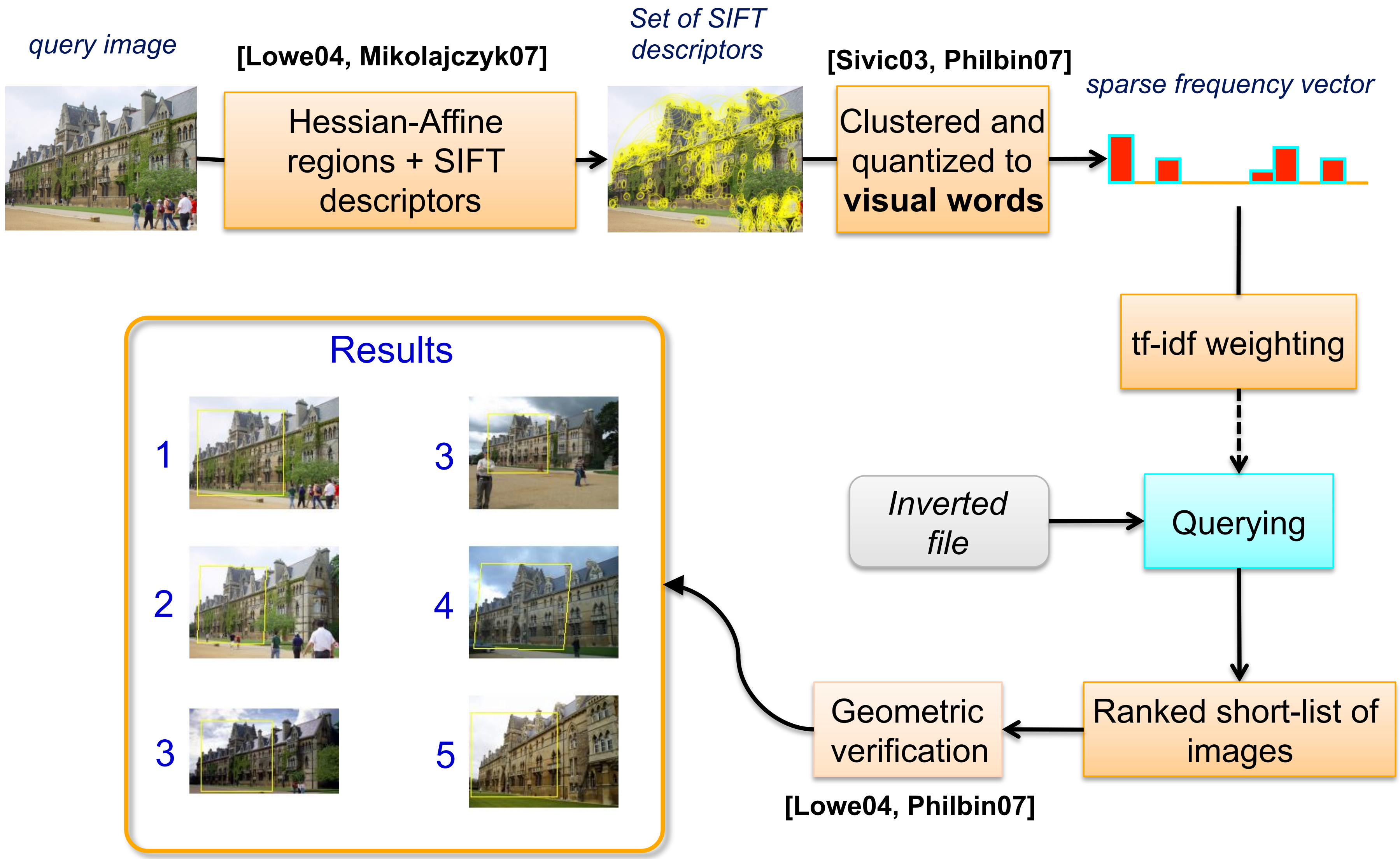
1. Compute local features in each image independently (offline)
2. "Label" each feature by a descriptor vector based on its intensity (offline)
3. Finding corresponding features is transformed to finding nearest neighbour vectors
4. Rank matched images by number of (tentatively) corresponding regions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

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 (the first part of the lecture).

Overview of the retrieval system



Visual words: discussion I.

Efficiency – cost of quantization

- Need to still assign each local descriptor to one of the cluster centers. Could be prohibitive for large vocabularies (K=1M).

- Approximate NN-search still needed
  - e.g. randomized k-d trees [Muja&Lowe 2009]

- True also for building the vocabulary
  - approximate k-means [Philbin et al. 2007]
  - Reduce k-means cost from O(NK) to O(N log K)
  - Can scale to very large K.

Visual words: discussion II.

• Need to determine the size of the vocabulary, K.

• Other algorithms for building vocabularies, e.g. agglomerative clustering / mean-shift, but typically more expensive.

• Supervised quantization?

Also give examples of images / descriptors which should and should not match.

E.g.:
Philbin et al. ECCV'10, http://www.robots.ox.ac.uk/~vgg/publications/html/philbin10b-bibtex.html

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.

And see the next lecture by C. Schmid

Efficient visual search for objects and places

Oxford Buildings Search - demo

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

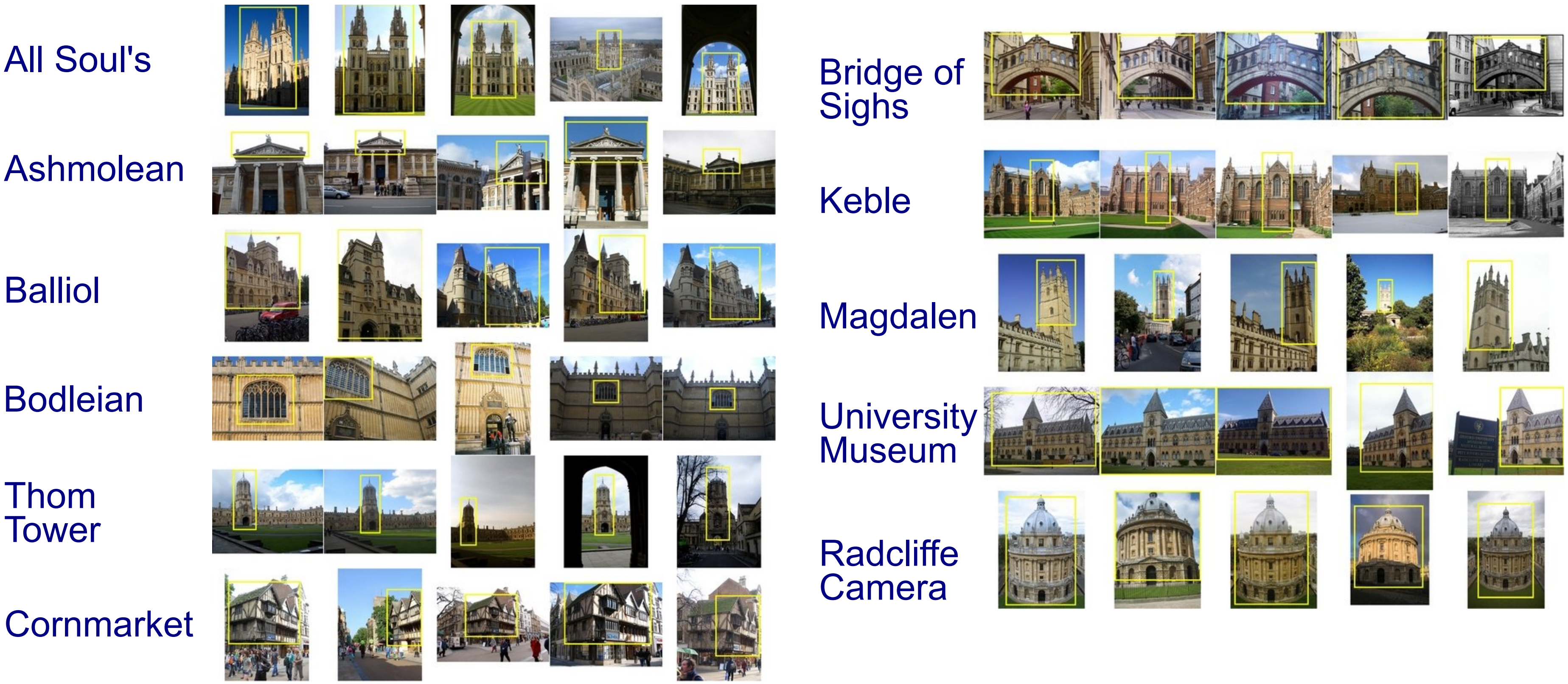
Oxford buildings dataset

- 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

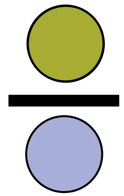
- Landmarks plus queries used for evaluation



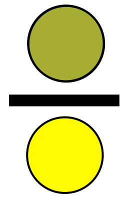
- Ground truth obtained for 11 landmarks
- Evaluate performance by mean Average Precision

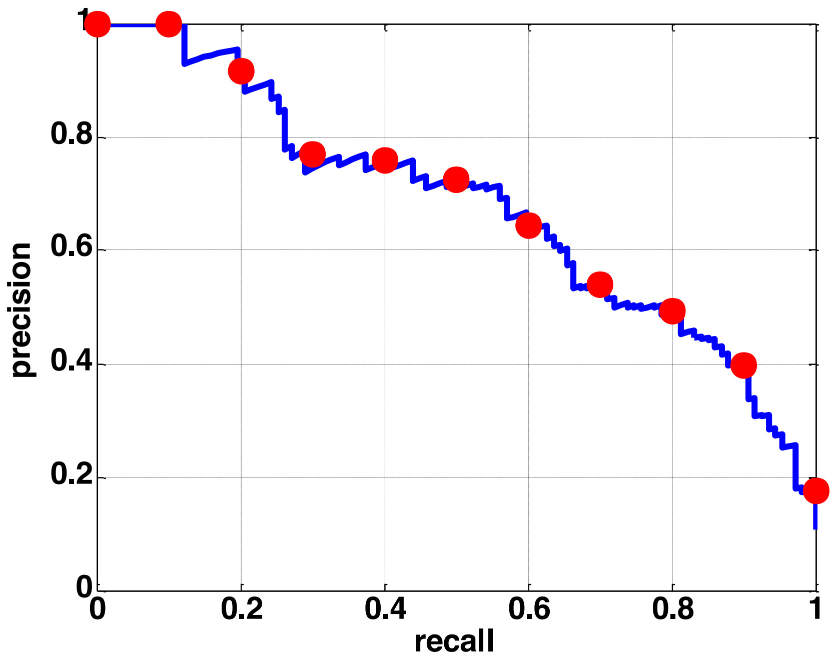
Measuring retrieval performance: Precision - Recall

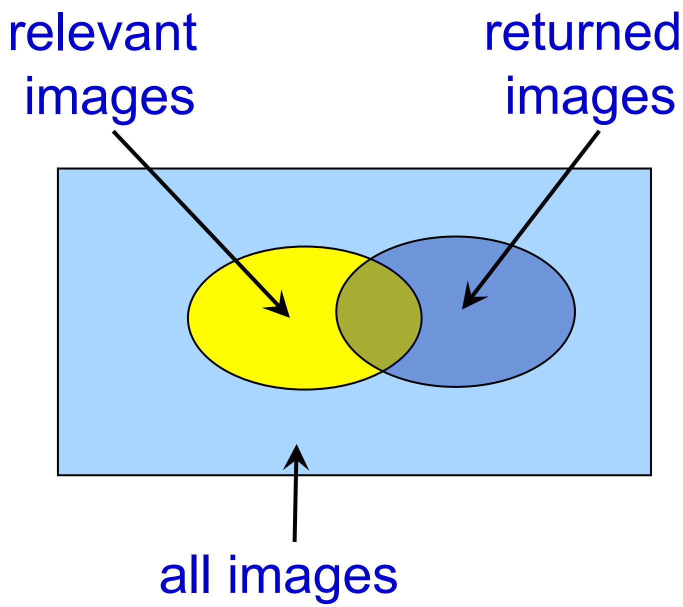
- Precision: % of returned images that are relevant



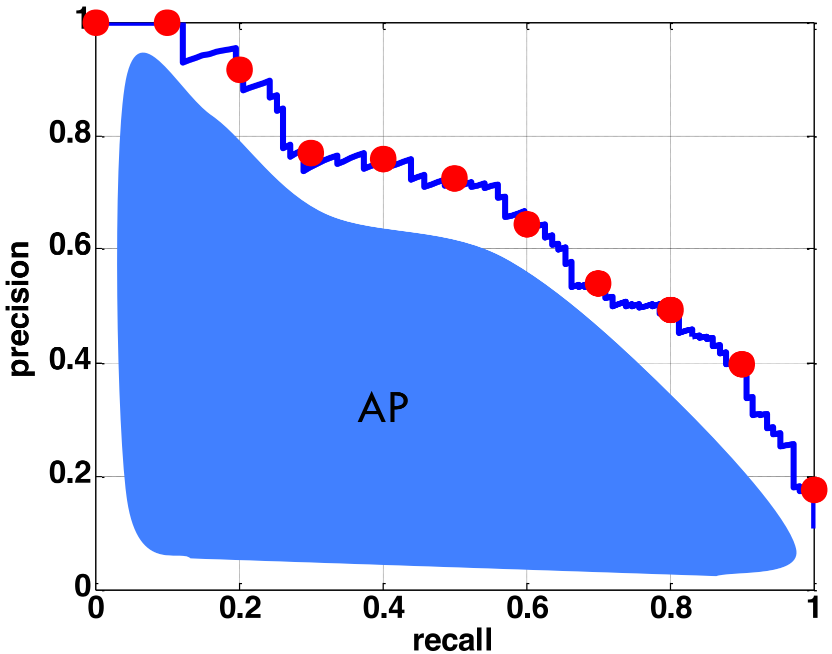
- Recall: % of relevant images that are returned





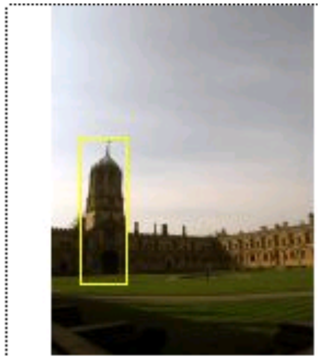


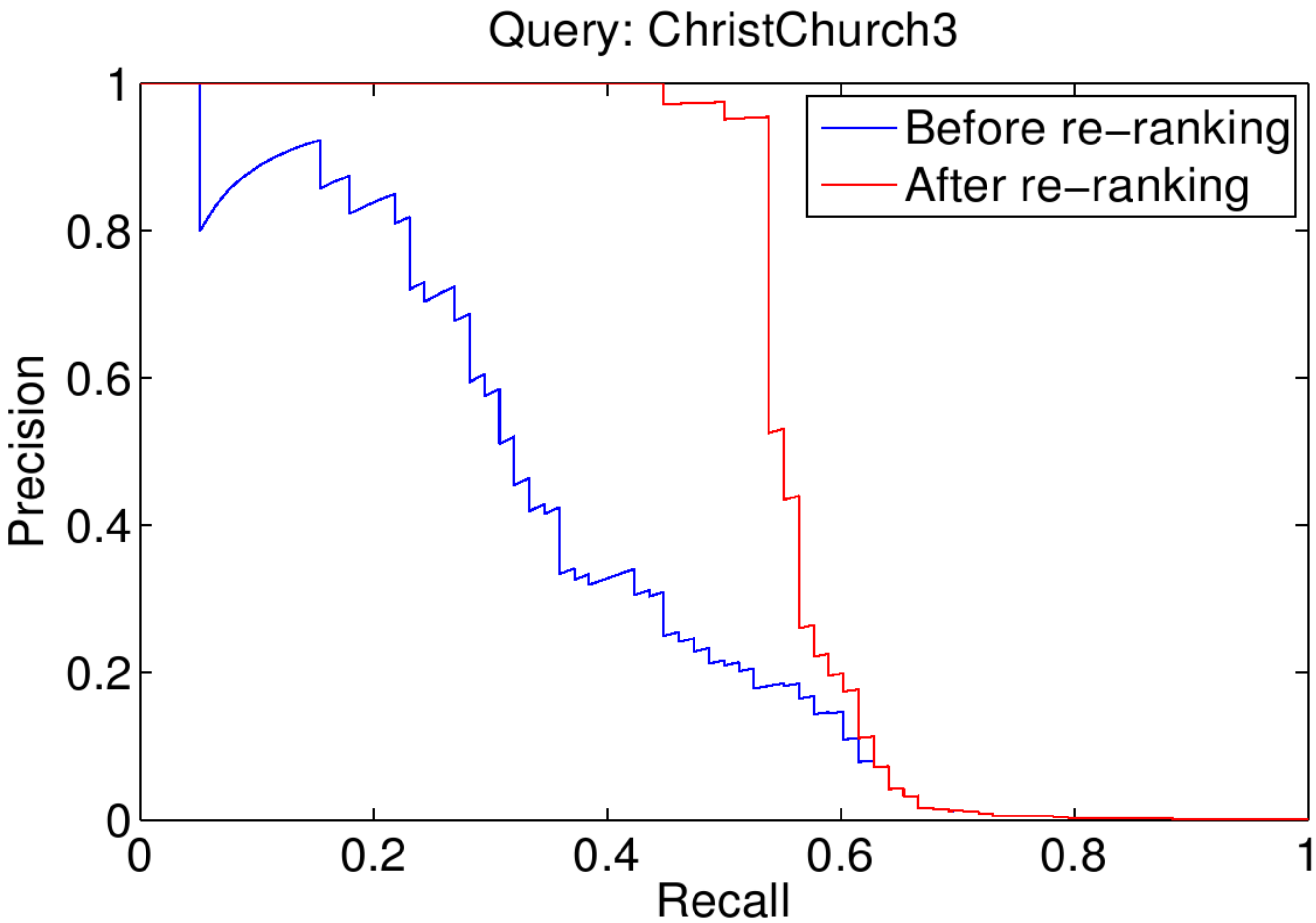
Average Precision



- 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

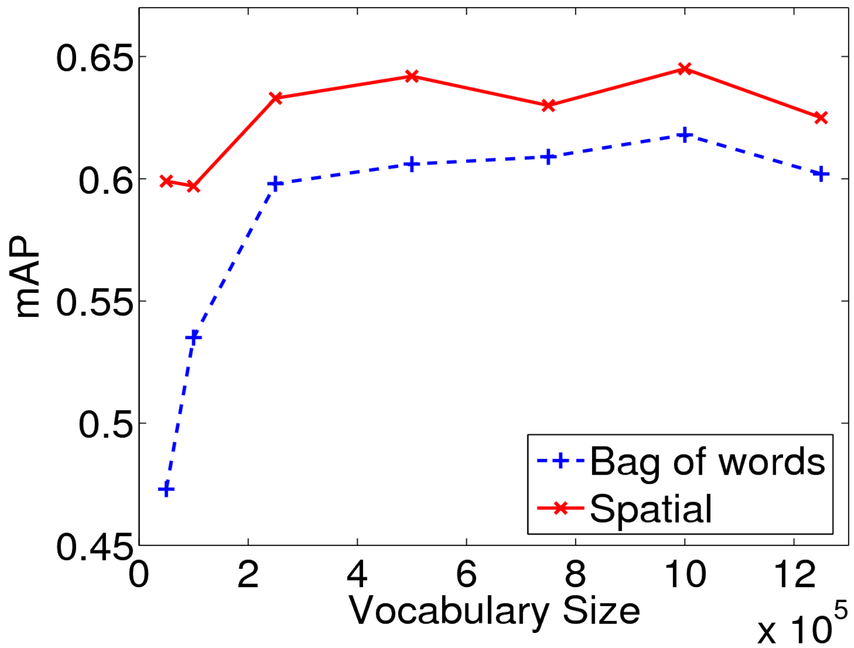




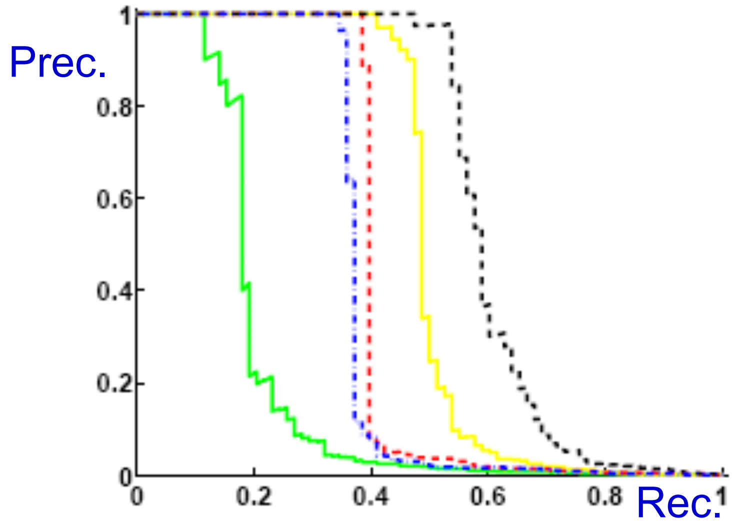
Query: ChristChurch3
Before re−ranking
After re−ranking
Precision
Recall
0
0.2
0.4
0.6
0.8
1

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

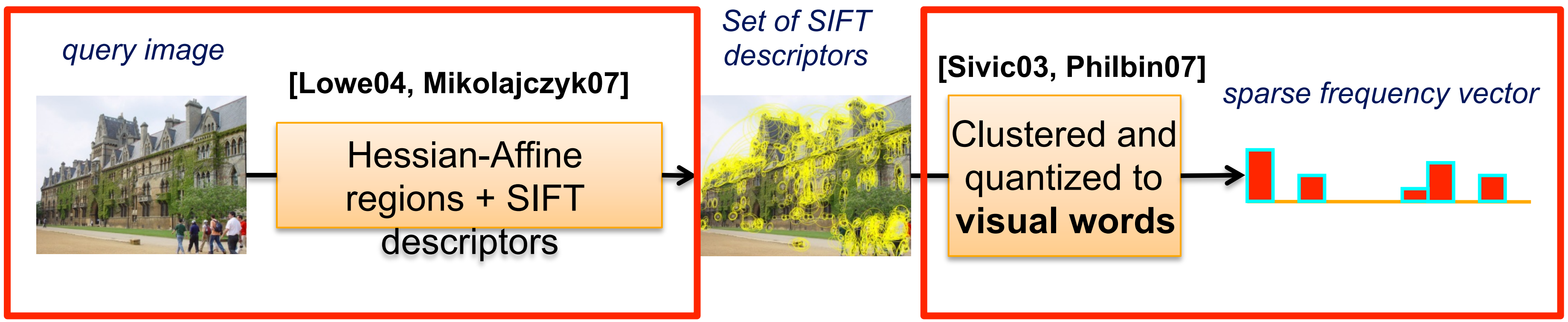


Query images



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

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

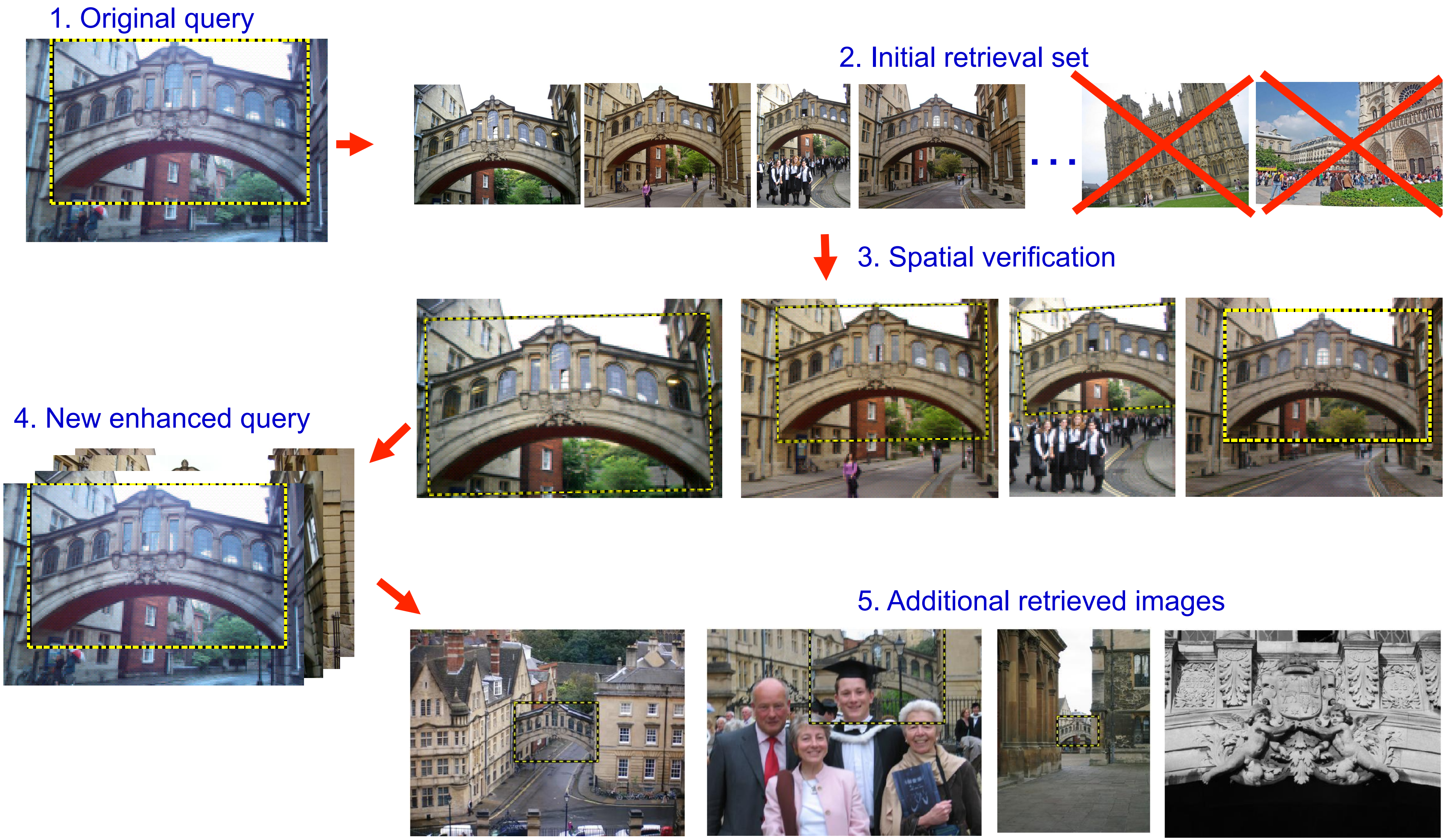
Initial returns may be used to add new relevant visual words to the query

Strong spatial model prevents 'drift' by discarding false positives

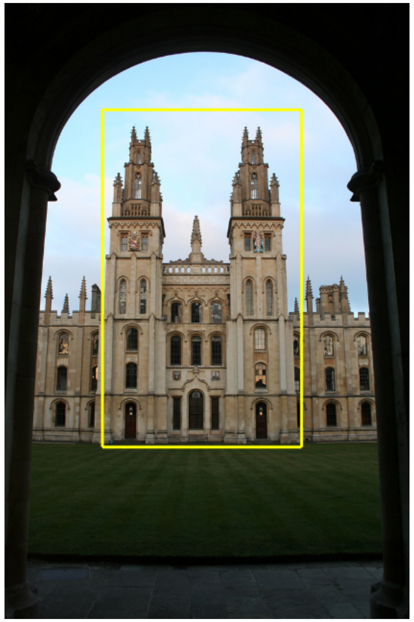
[Chum, Philbin, Sivic, Isard, Zisserman, ICCV'07;

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

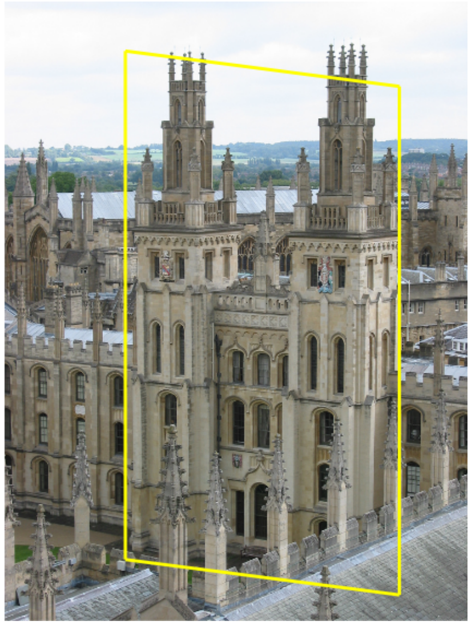
Visual query expansion - overview



Query Expansion



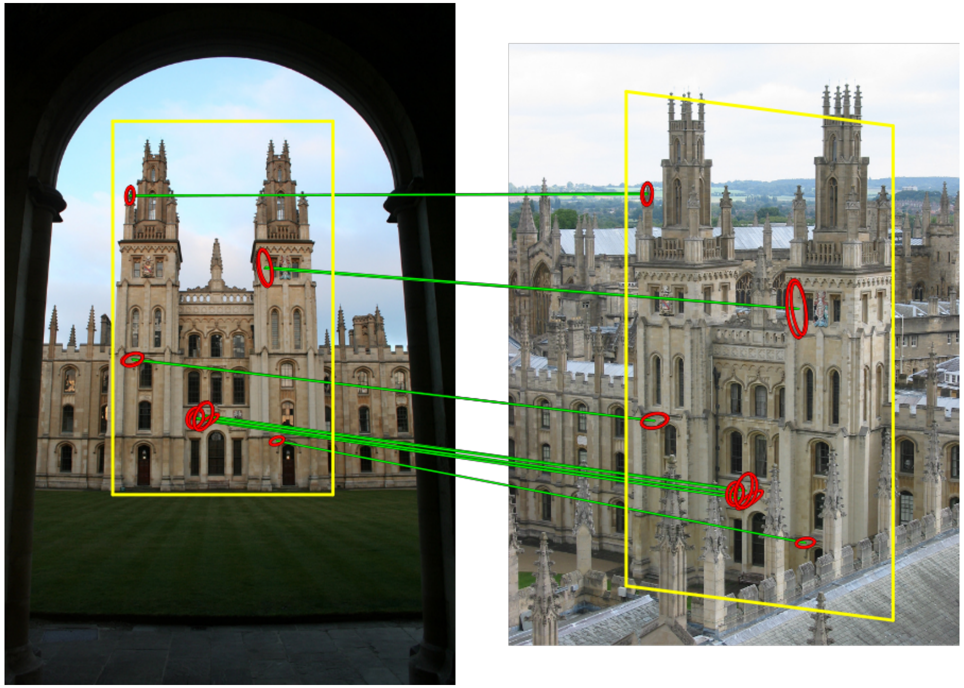
Query Image



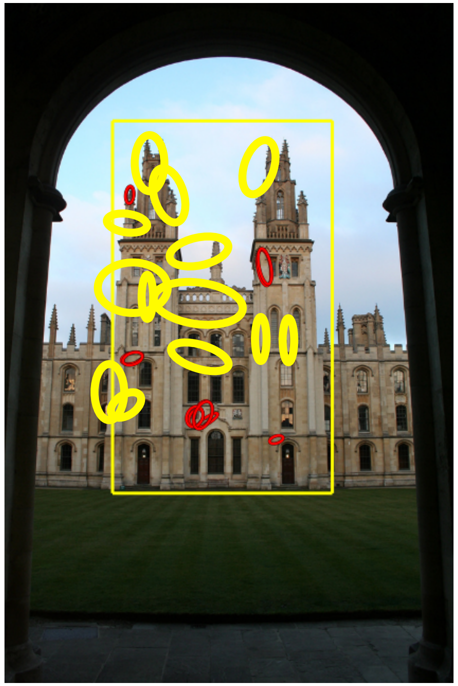
Originally retrieved image

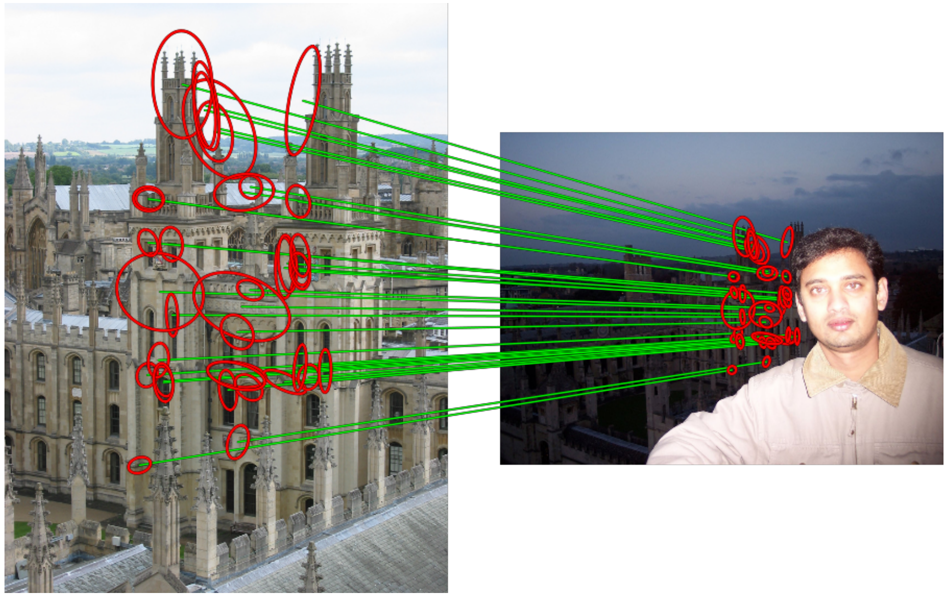
Originally not retrieved

Query Expansion

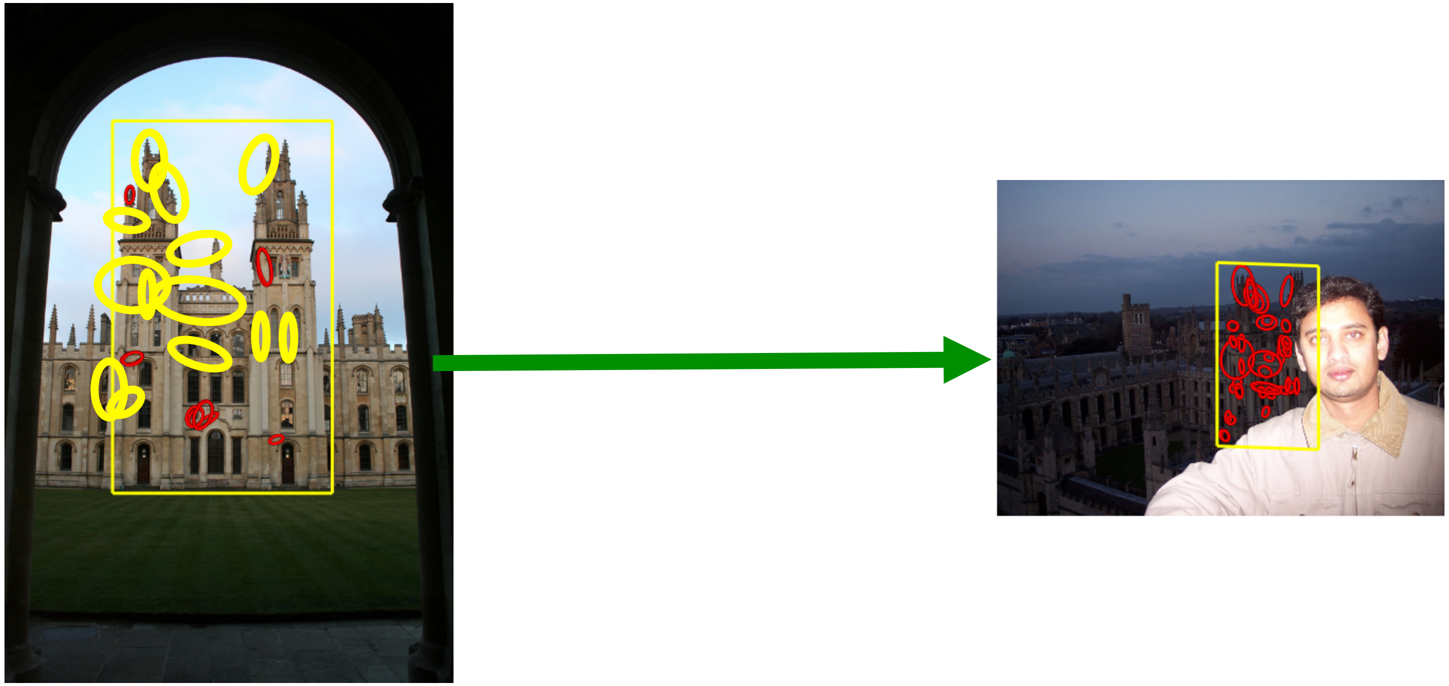


Query Expansion





Query Expansion



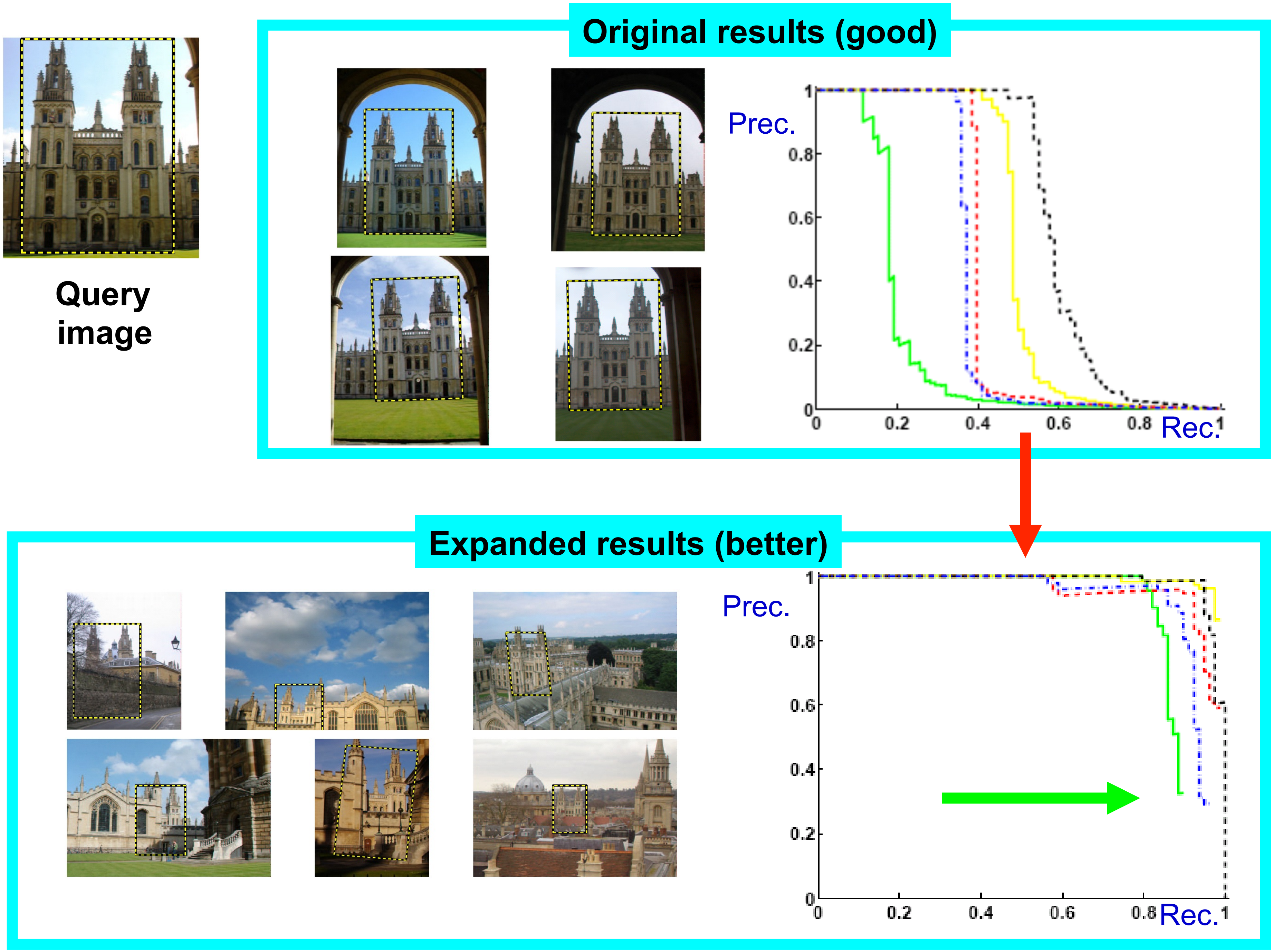
Demo

Query Expansion

Query image

Originally retrieved

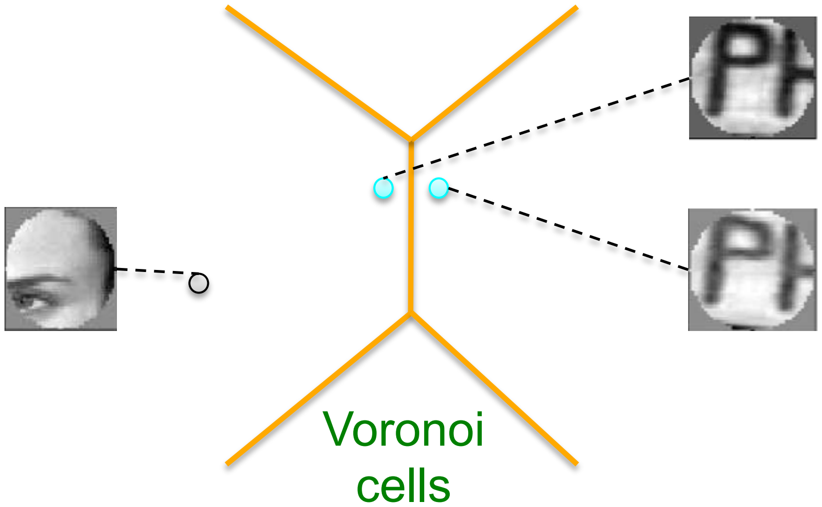
Retrieved only after expansion


Query image
Original results (good)
Prec.
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
Rec.
1
Expanded results (better)
Prec.
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
Rec.
1

Quantization errors

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

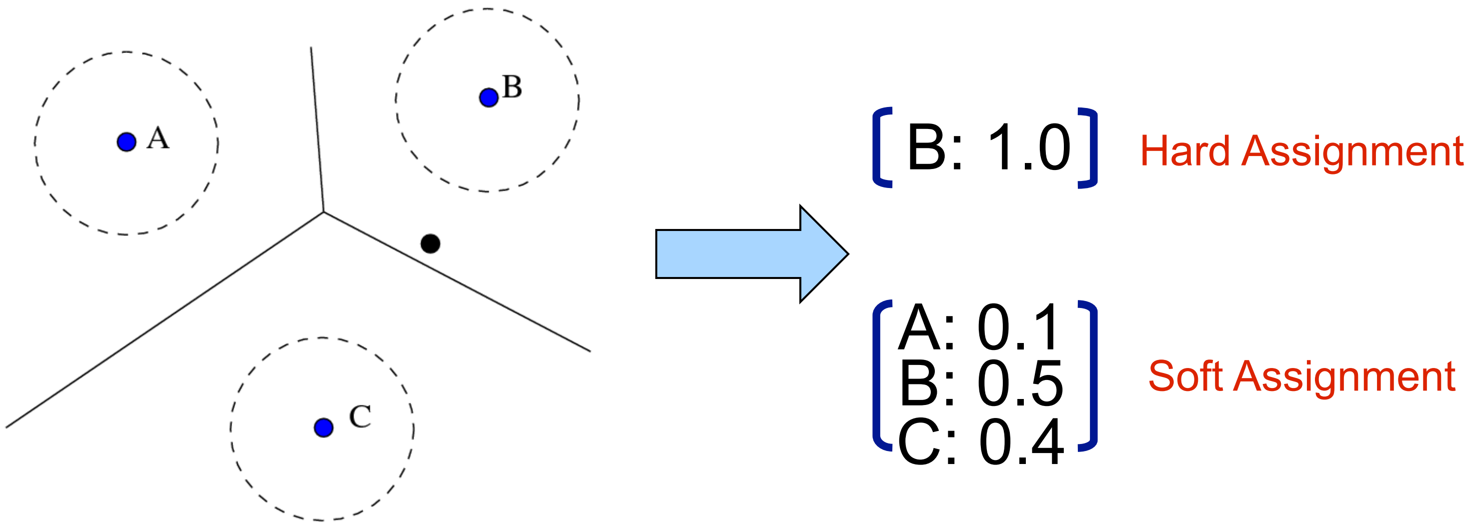
Quantization errors split features that should be grouped together and confuse features that should be separated



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]

See also next lecture.

Other recent work

Learning a vocabulary to overcome quantization errors

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

Large scale image clustering [Chum et al. CVPR 2009, Philbin et al. IJCV 2010, Li et al., ECCV 2008]

Matching in structured datasets (3D landmarks or street-view images)

[Cummins and Newman 2009, Irschara et al. CVPR 2009, Knopp et al. ECCV 2010, Zamir&Shah ECCV 2010, Li et al. ECCV 2010, Baatz et al. ECCV 2010, Chen et al. CVPR 2011, Sattler et al. CVPR 2011, Baatz et al. ECCV 2012, Torii et al. CVPR 2013, Gronat et al. CVPR 2013, Cao&Snavely CVPR 2013]

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.

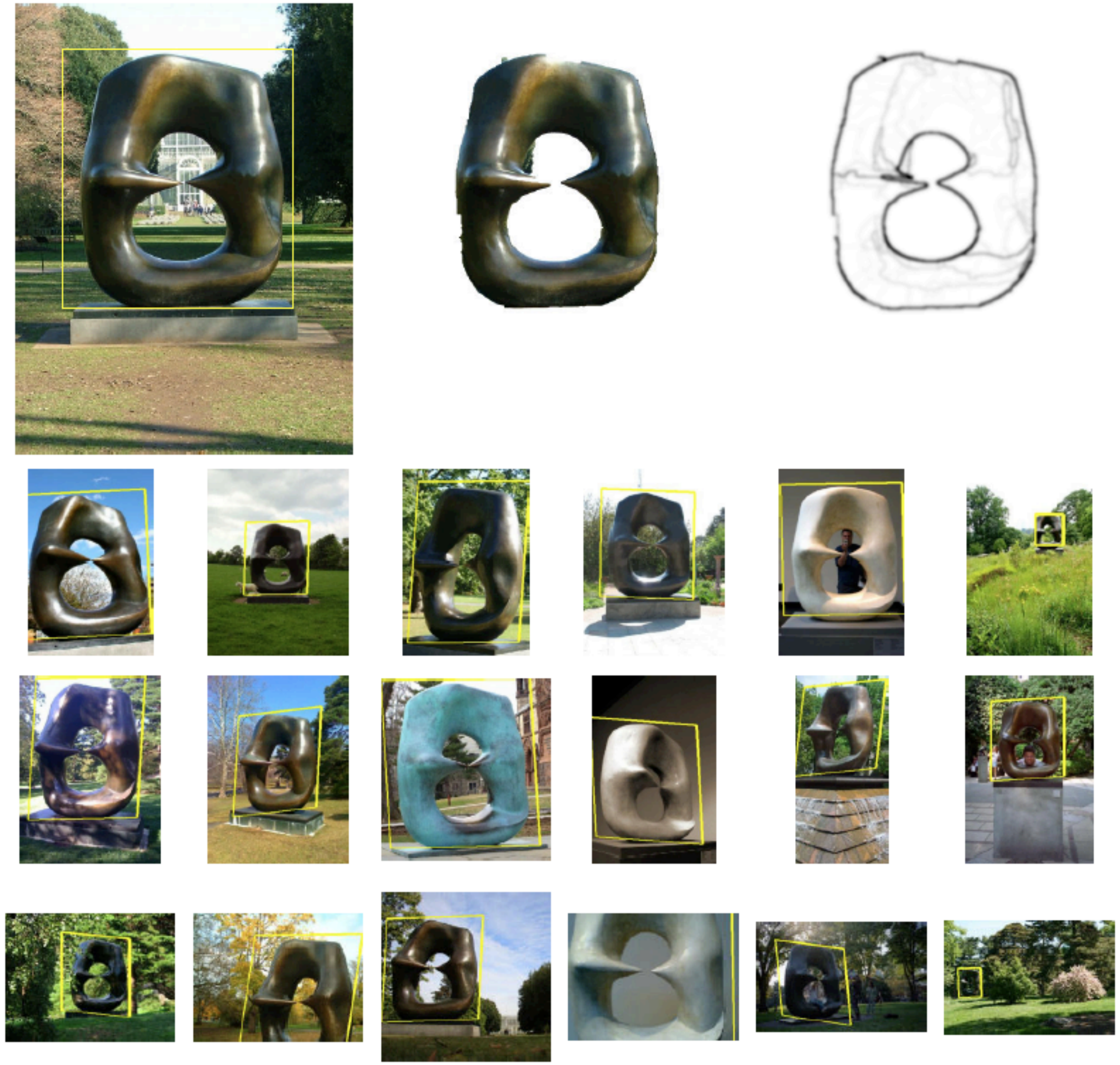
What next?

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

Example:
Smooth object retrieval using a bag of boundaries
by Arandjelovic and Zisserman, ICCV 2011

Query

Retrieved
matches



Category-level visual search [See later lectures.]

Query

same category

See also e.g. [Torresani et al. ECCV 2010]

What next?

Match objects across large changes of appearance

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




Example: Painting-to-3D model alignment via discriminative visual elements

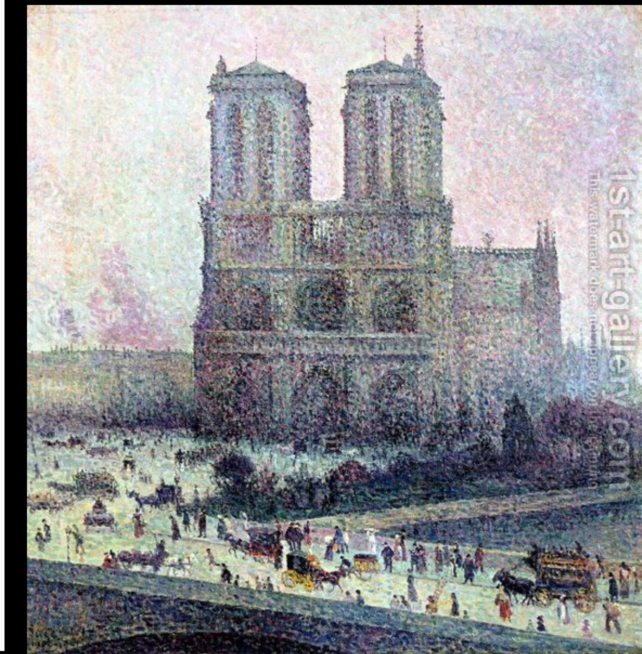
Inputs: paintings, drawings, historical photographs, reference 3D model

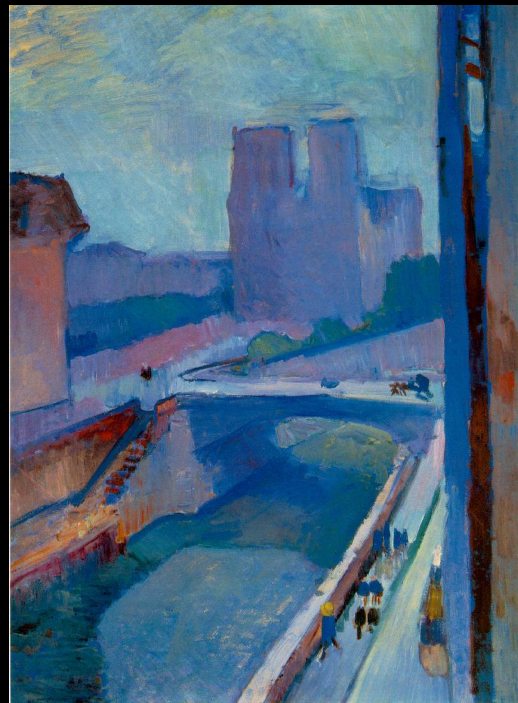
Output: recovered artist/camera viewpoints

[Aubry, Russell, Sivic, to appear in TOG 2013]

Why do this?

There are many non-photographic depictions of our world





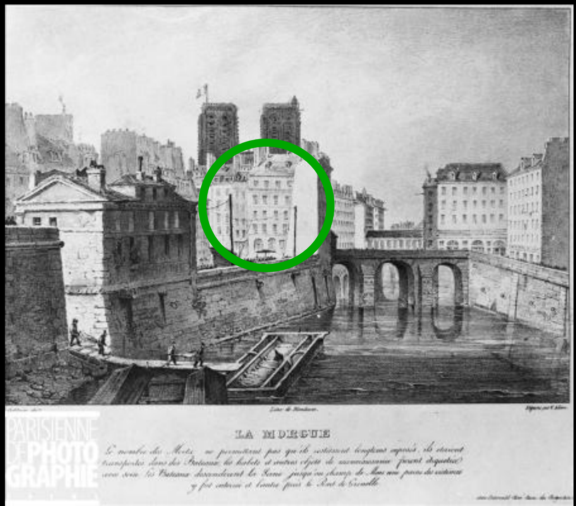
Ultimate goal: to reason about these depictions

Applications

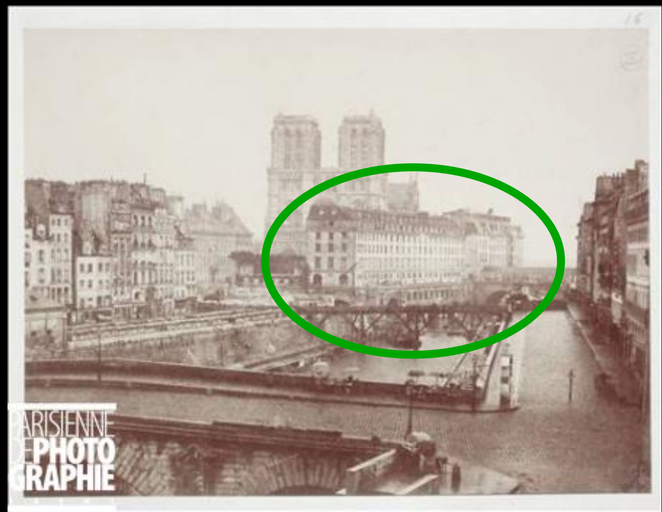
New ways to access archives for

archeology, history or architecture

Example: evolution of a particular place over time



1830



1852

1900

See also [WhatWasThere.com] with historical imagery manually aligned to a map.

Difficulty in finding correspondences

Color, geometry, illumination, shading, shadows and texture may be rendered by the artist in a realistic, but "non physical" manner

- 121 putative matches total across 563 photographs using SIFT matching
- 0 correct putative matches

Difficulty in finding correspondences

Local feature matching using SIFT:

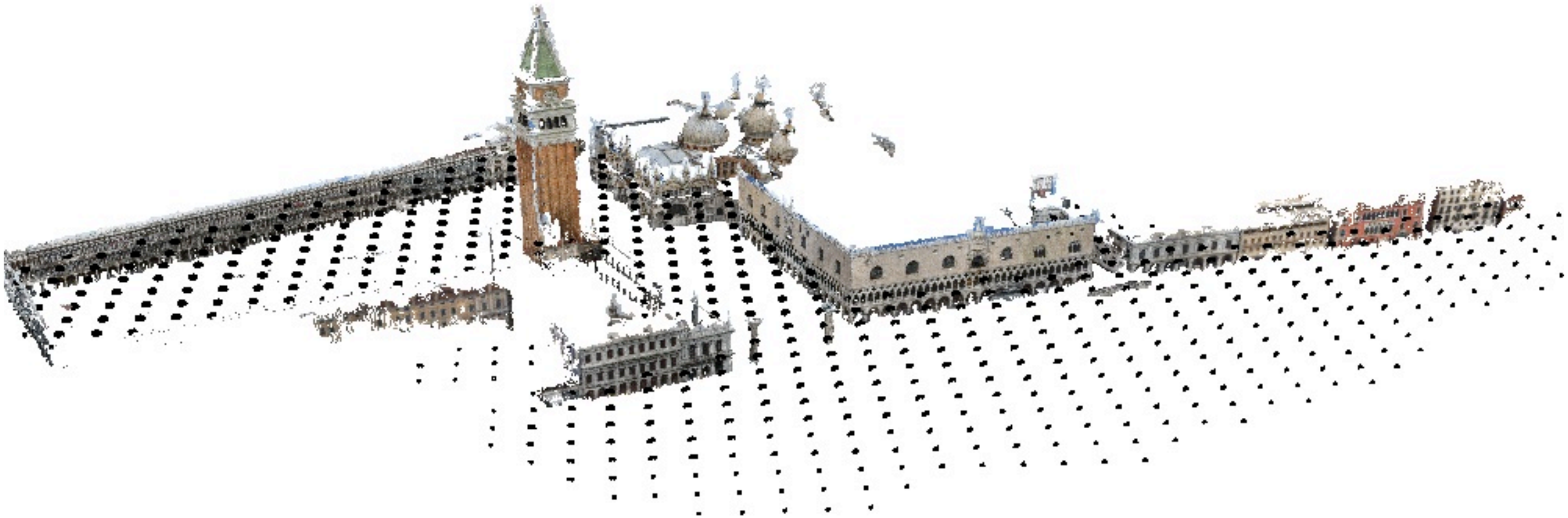
Figure from [A. Shrivastava, T. Malisiewicz, A. Gupta, A. Efros
Data-driven Visual Similarity for Cross-domain Image Matching
SIGGRAPH Asia 2011]

See also:
[Hauagge & Snavely CVPR 2012]
[Chum & Matas CVPR 2006]
[Russell, Sivic, Ponce, Dessalles 2011]

How to match a painting to a 3D model?

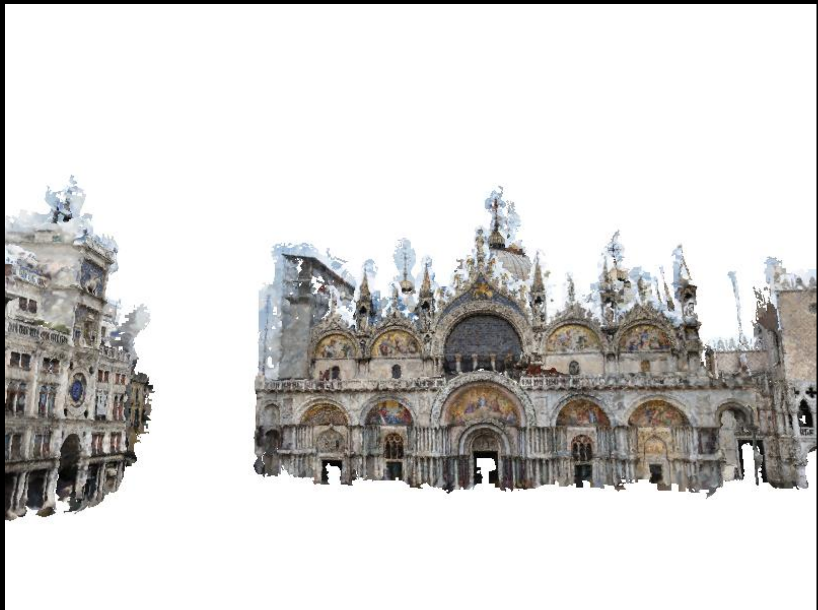
I. Use 3D model to synthesize a similar view

Synthesize ~10,000 viewpoints for an architectural site



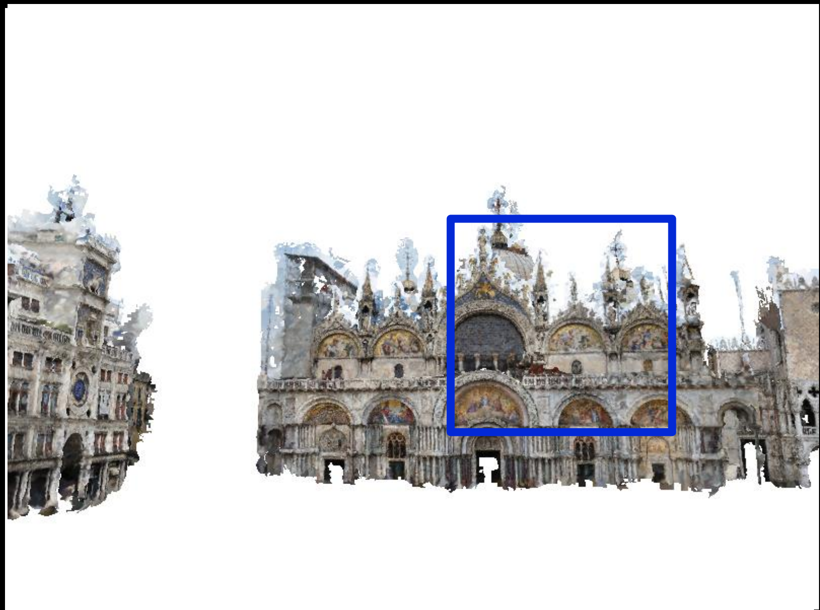
See also: [Irschara et al. CVPR 2009], [Baatz et al. ECCV 2012]

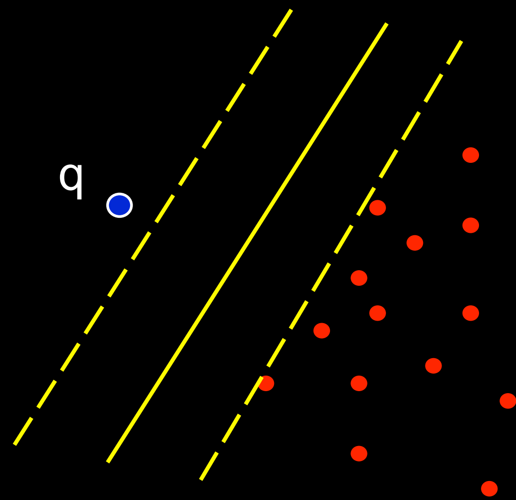
I. Use 3D model to synthesize a similar view



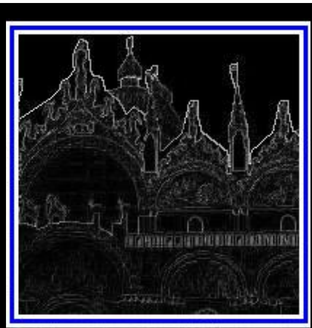
See also: [Irschara et al. CVPR 2009], [Baatz et al. ECCV 2012]

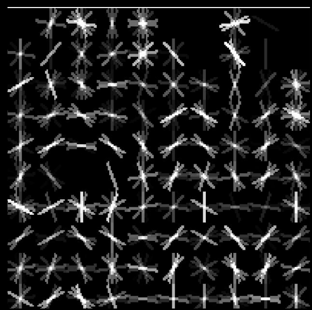
II. Matching as discriminative classification





Query region q:

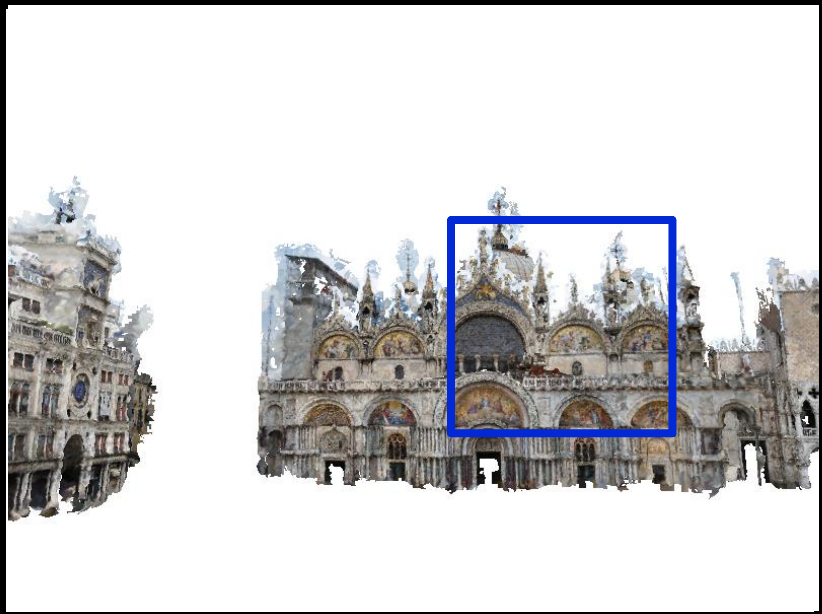




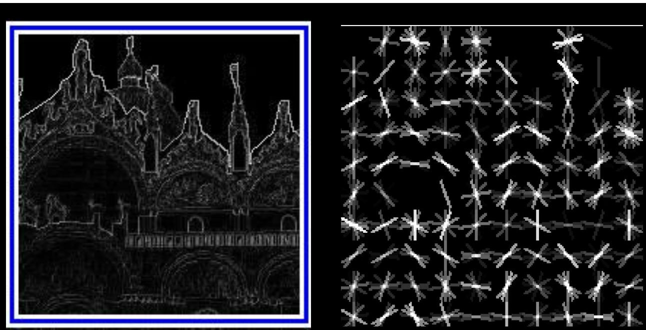
See detection lecture by A. Zisserman
See also: Exemplar SVM by [Malisiewicz et al., ICCV'11], [Shrivastava et al.'11]

1. Represent query region q using HOG descriptor
2. Train a linear classifier $f(x) = w^T x + b$ using q as a positive example and large number of negatives

II. Matching as discriminative classification

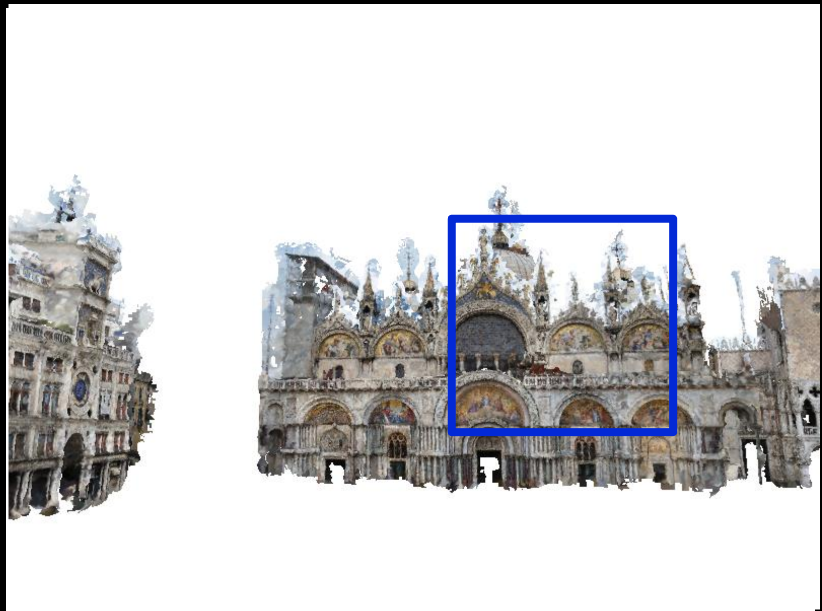


Query region q:

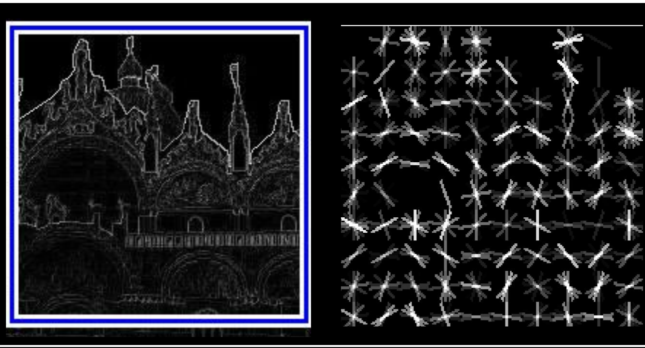


1. Represent query region q using HOG descriptor
2. Train a linear classifier $f(x) = w^T x + b$ using q as a positive example and large number of negatives

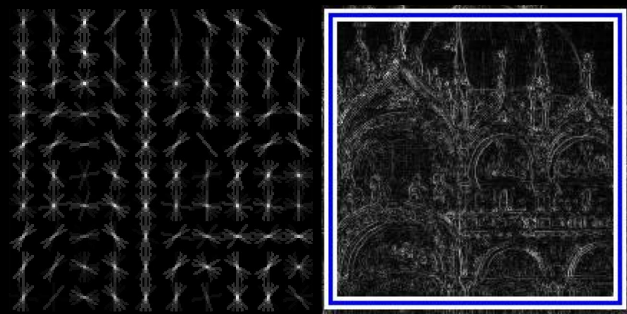
II. Matching as discriminative classification



Query region q:

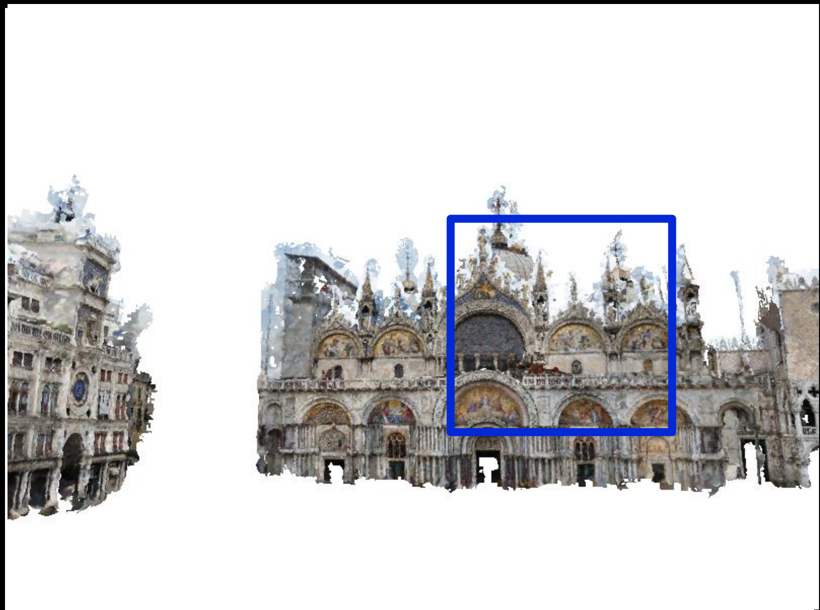


Best match:

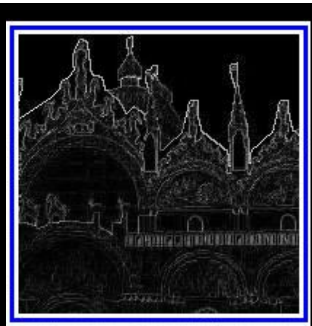


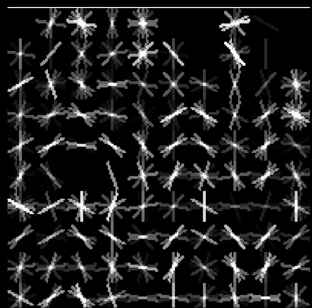
1. Represent image region using HOG descriptor x
2. Train a linear classifier $f(x) = w^T x + b$
3. Find best match in the painting maximizing the classification score f(x)

II. Matching as discriminative classification

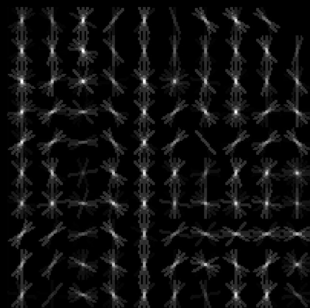


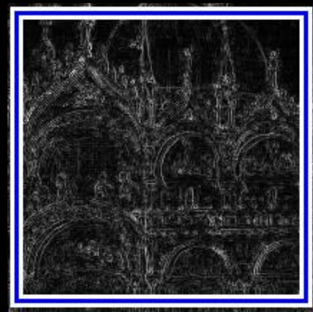
Query region q:





Best match:





Discriminative visual element: trained classifier $f(x) = w^T x + b$

How to choose discriminative visual elements representing architectural site?

See also [Doersch et al. SIGGRAPH 2012] [Singh et al. ECCV 2012], [Juneja et al. CVPR 2013]

Algorithm outline

Offline:

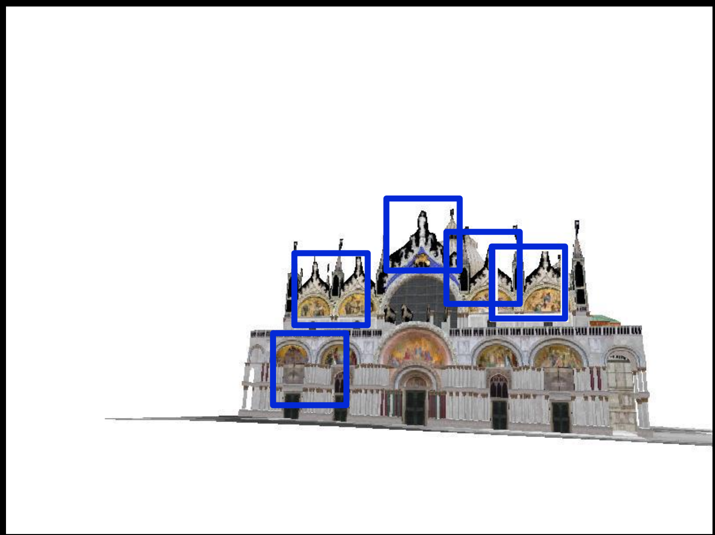
1. Sample virtual viewpoints from 3D site
2. Learn discriminative visual elements from rendered views

Given painting:

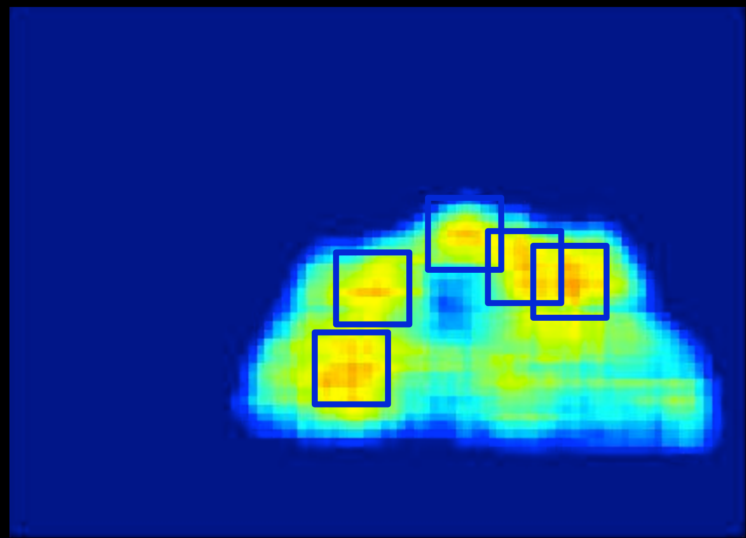
3. Obtain element detections on the painting
4. Keep only matches consistent with a single view (RANSAC)
5. Optional: fine viewpoint alignment

Offline: Learn a “vocabulary” of discriminative visual elements

- Train classifiers for all candidate regions in synthesized views
  - Can be done efficiently, see [Gharbi et al. 2012; Hariharan et al. 2012]
- Score each classifier by its training error.
- Keep only the top N most discriminative visual elements.



Original image

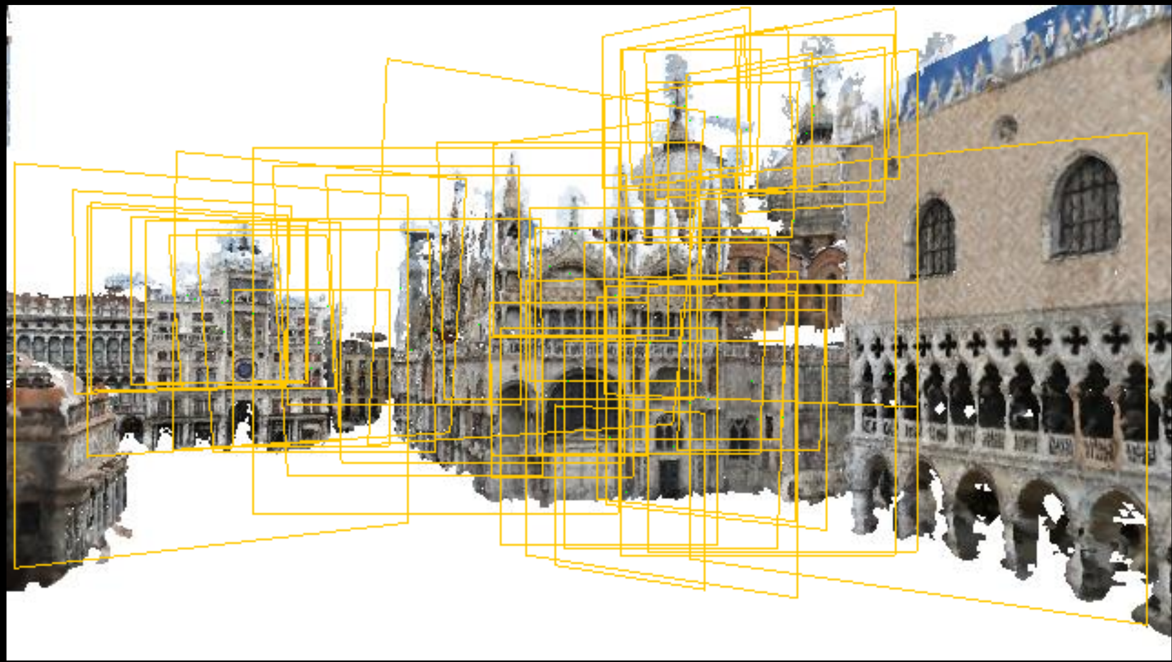


Discriminative score: 1 / training error

Note: Can be thought of as a generalization of local feature detection.

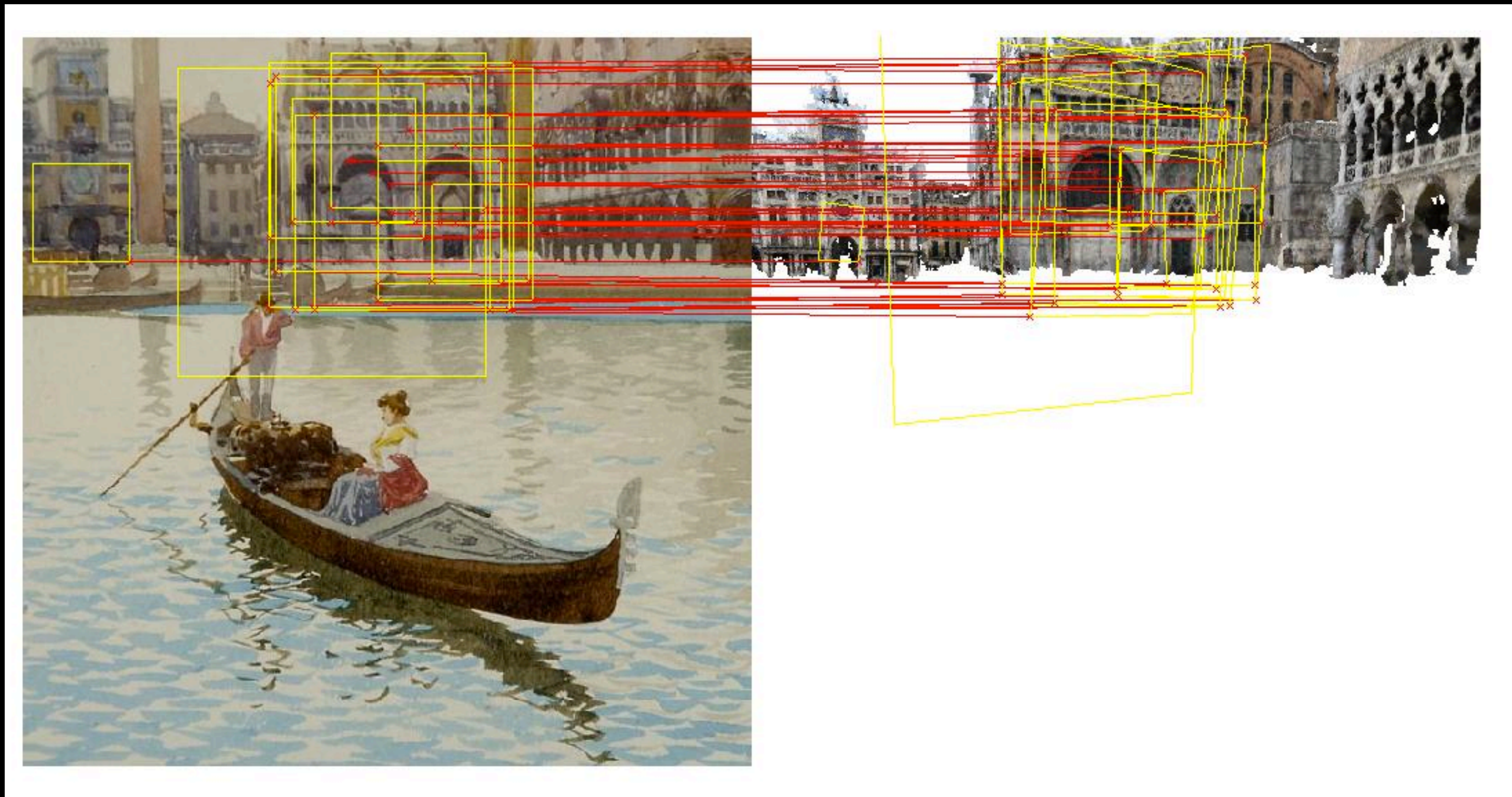
Offline: Learn a “vocabulary” of discriminative visual elements

- Back-project learnt discriminative elements onto the 3D model

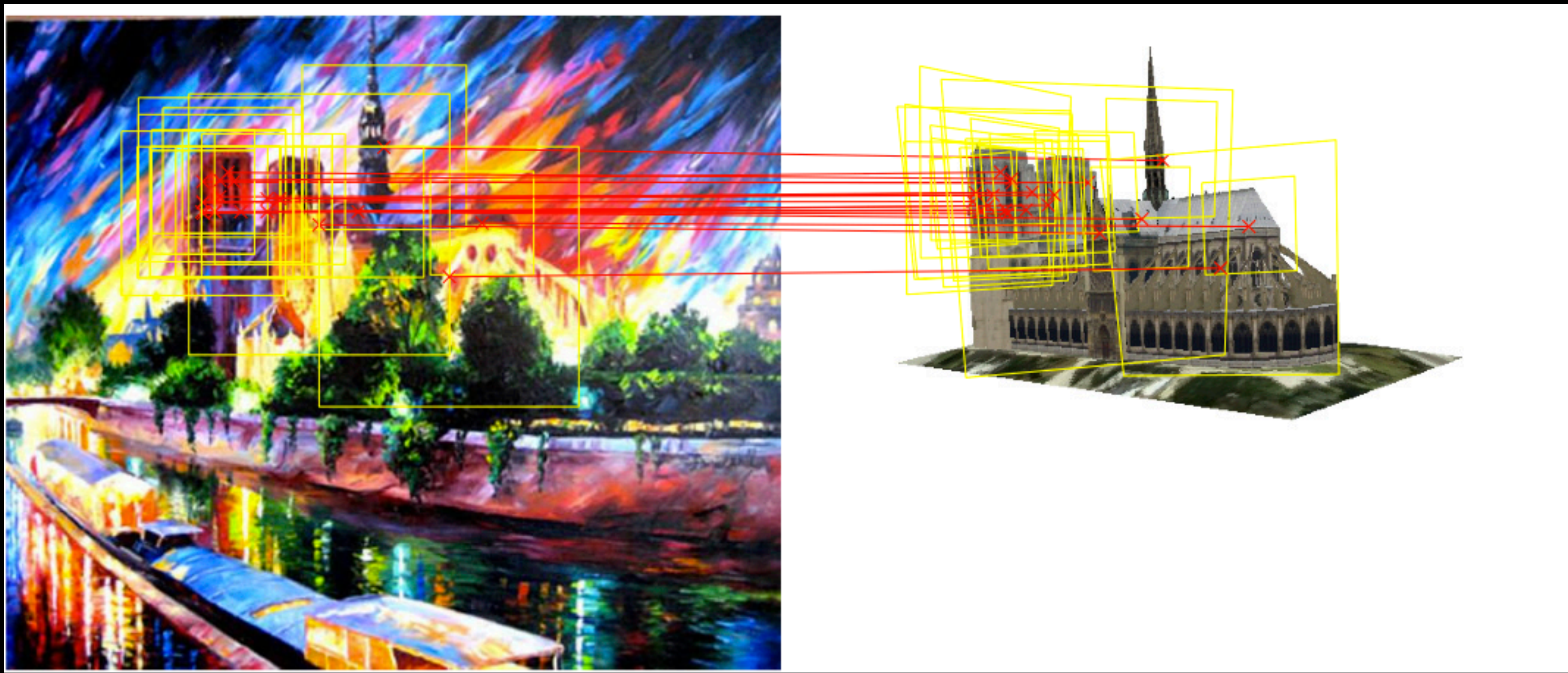


See also [Doersch et al. SIGGRAPH 2012] [Singh et al. ECCV 2012], [Juneja et al. CVPR 2013]

Given a painting:
Obtain visual element detections and verify matches with RANSAC



Example II.

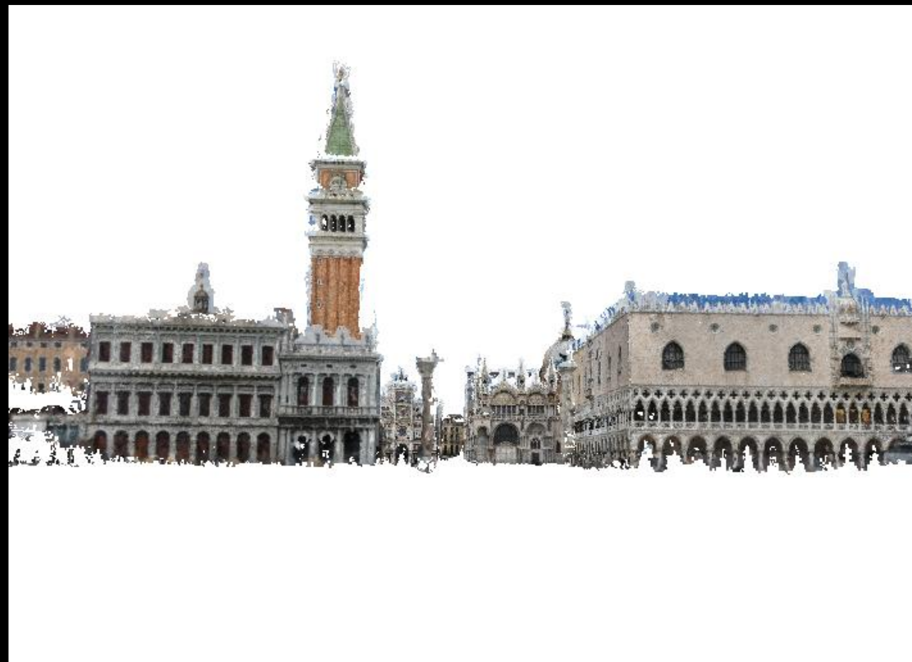


Experiments

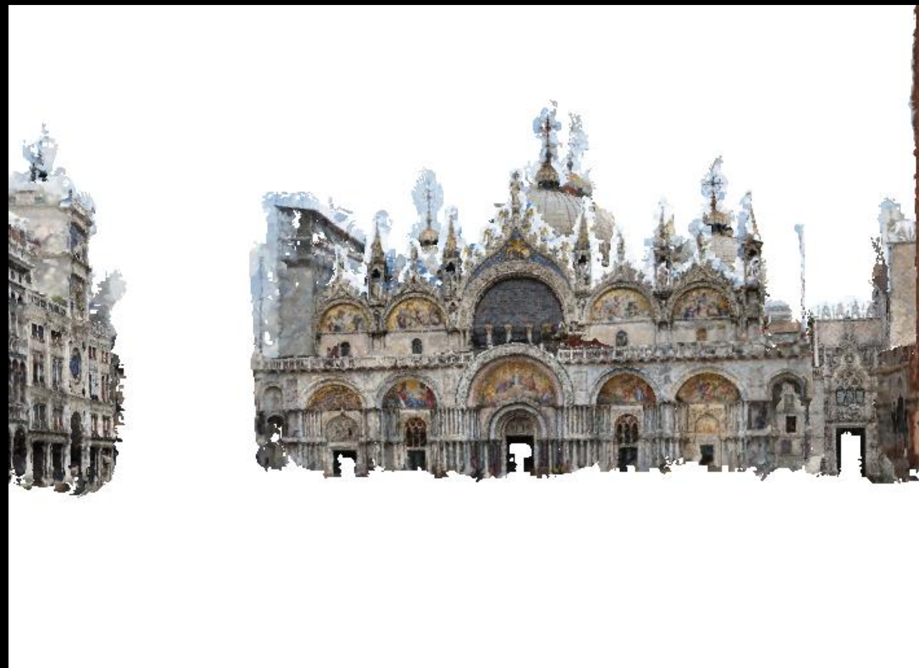
- 3D architectural sites
  - Venice (PMVS reconstruction from “Rome in a day” photographs)
  - Venice (3D CAD model)
  - Trevi Fountain (3D CAD model)
  - Notre Dame of Paris (3D CAD model)

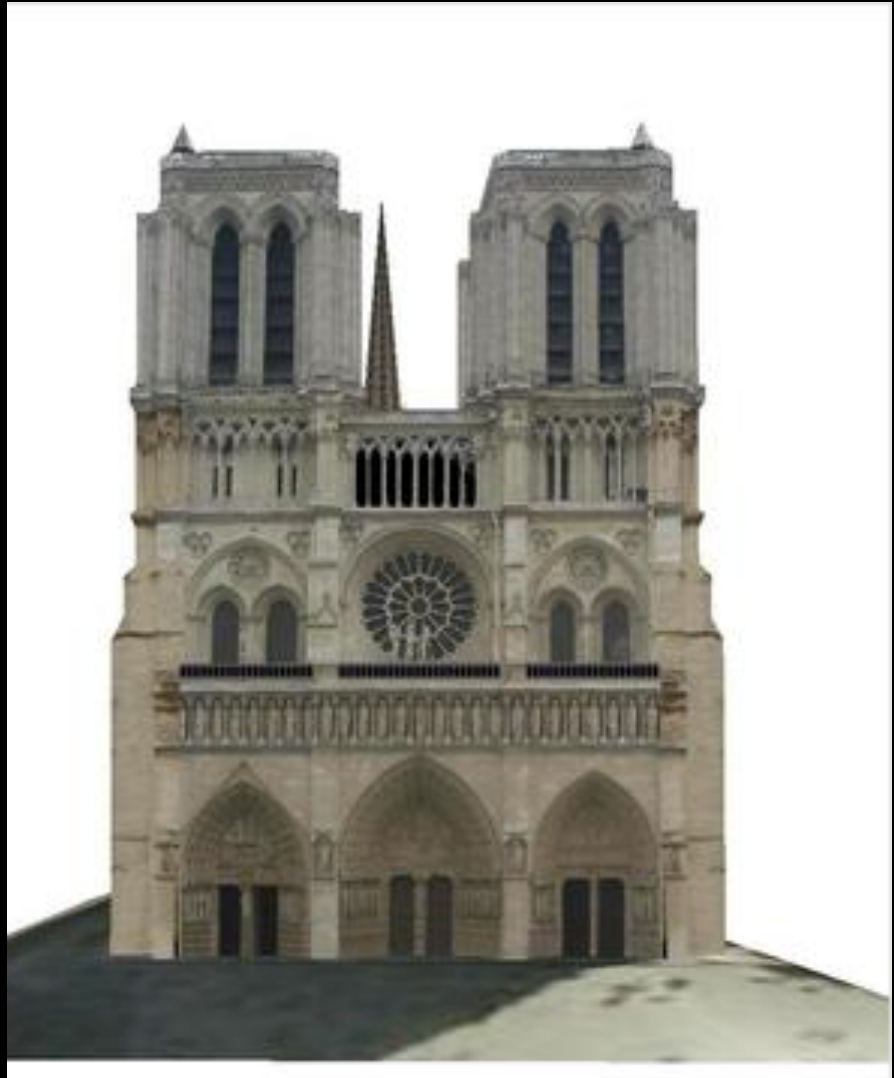
- “Test queries”
  - 50 historical photographs
  - 150 paintings/drawings

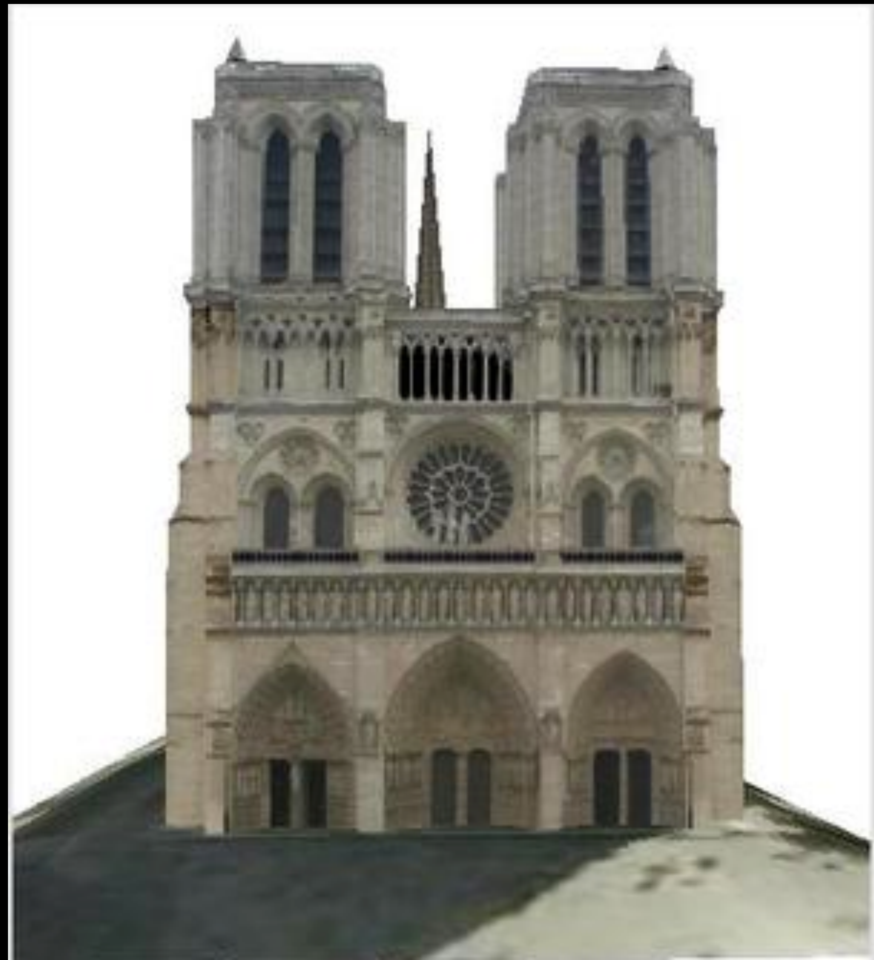
Results: historical photographs



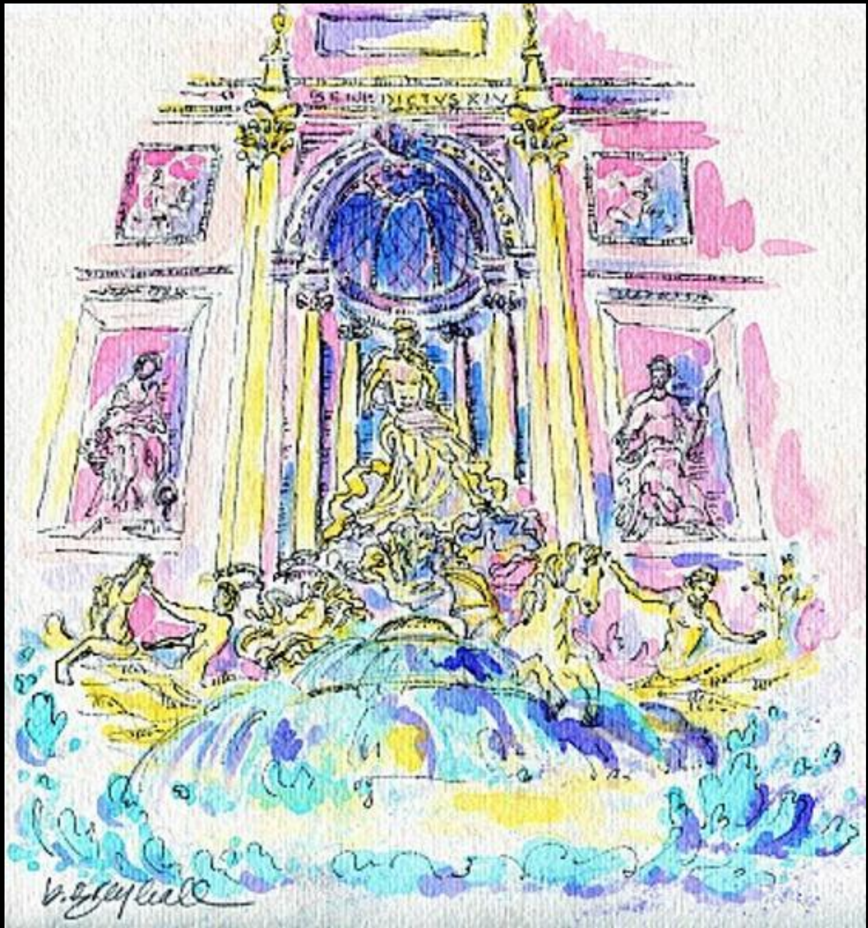


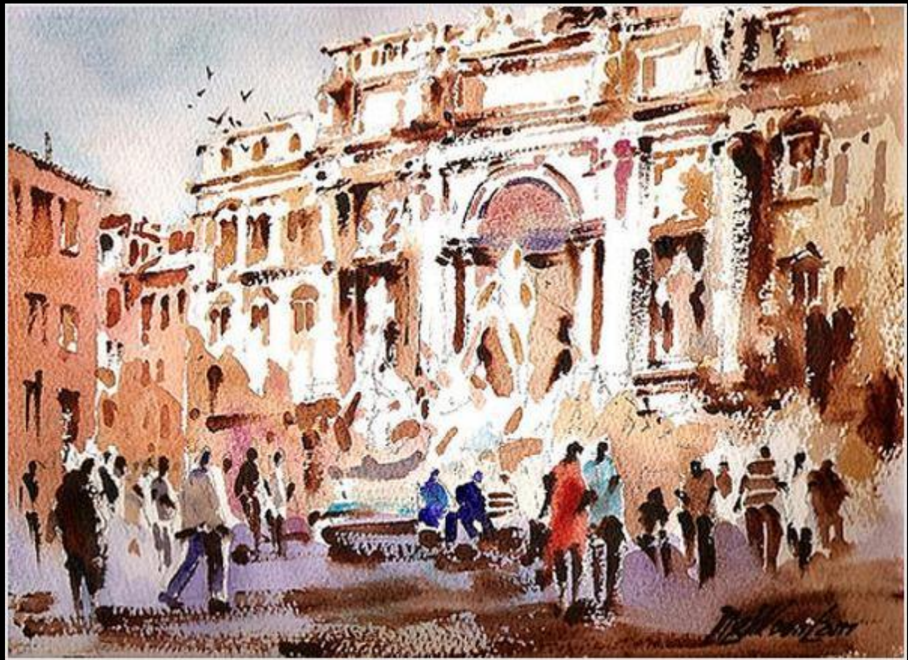


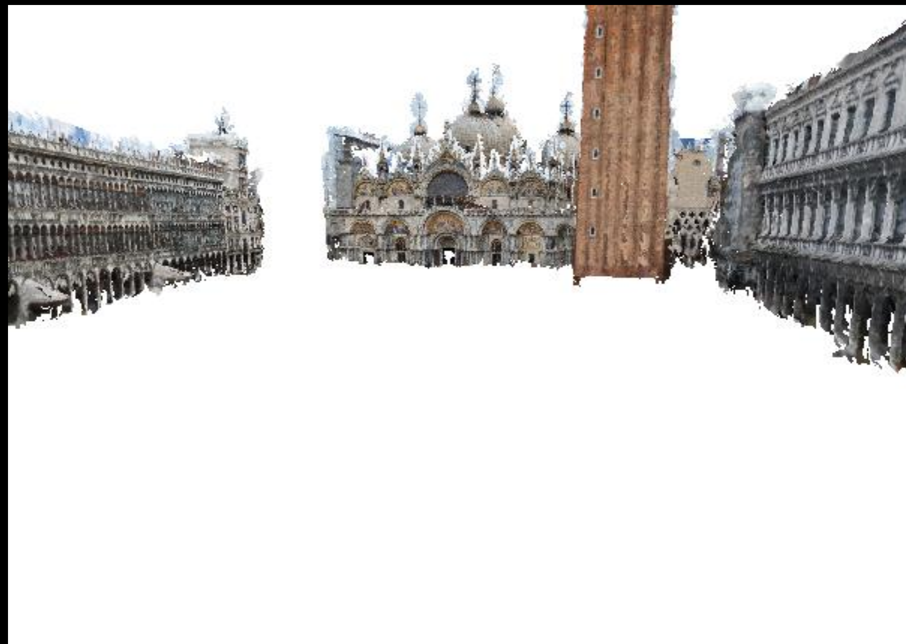


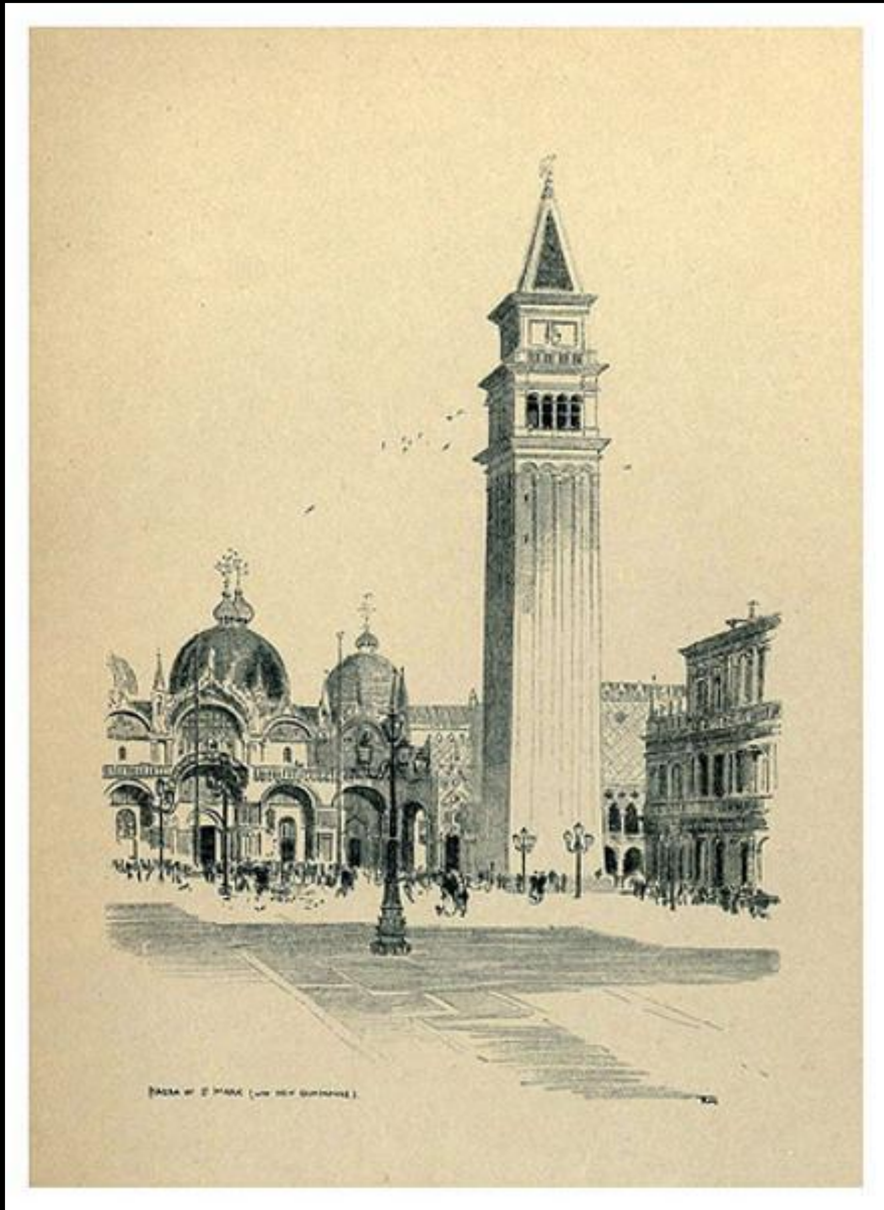


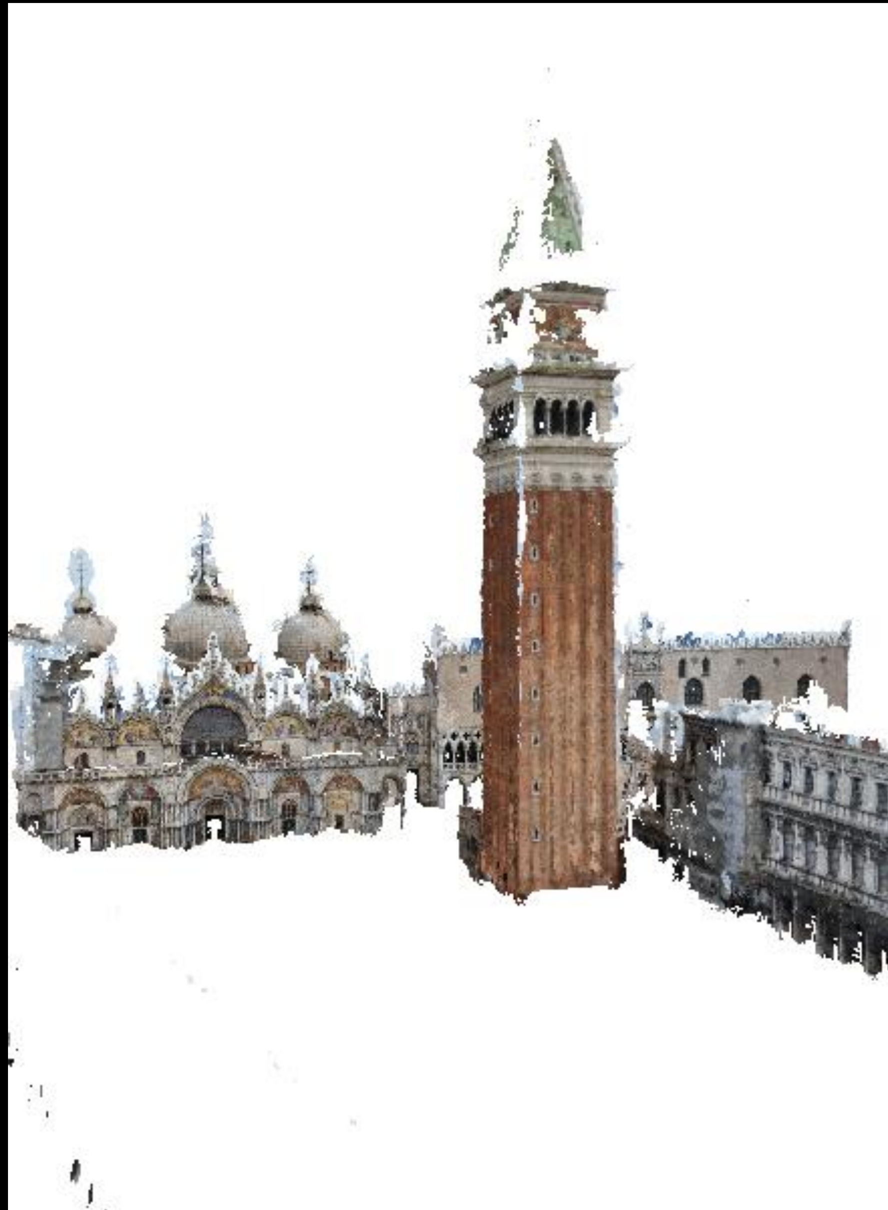
Results: paintings and drawings



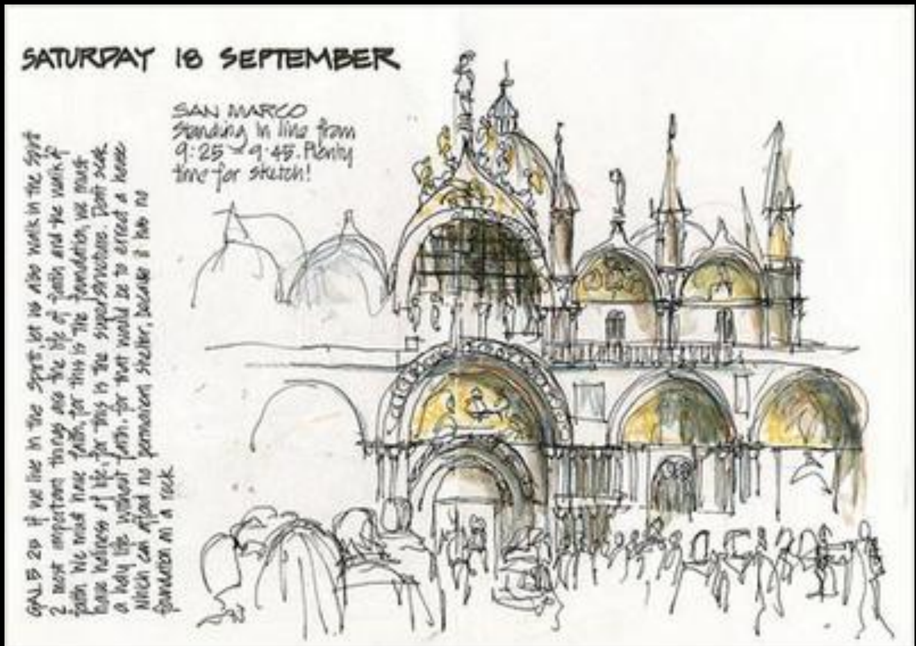






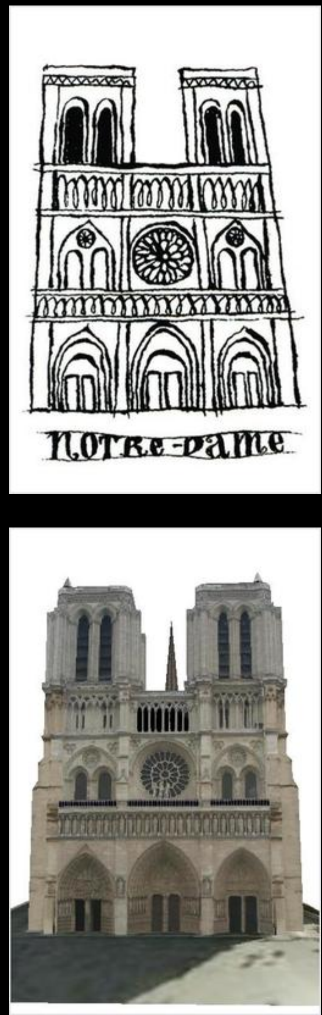



1st-art-gallery.com
The watermark does not appear on the painting

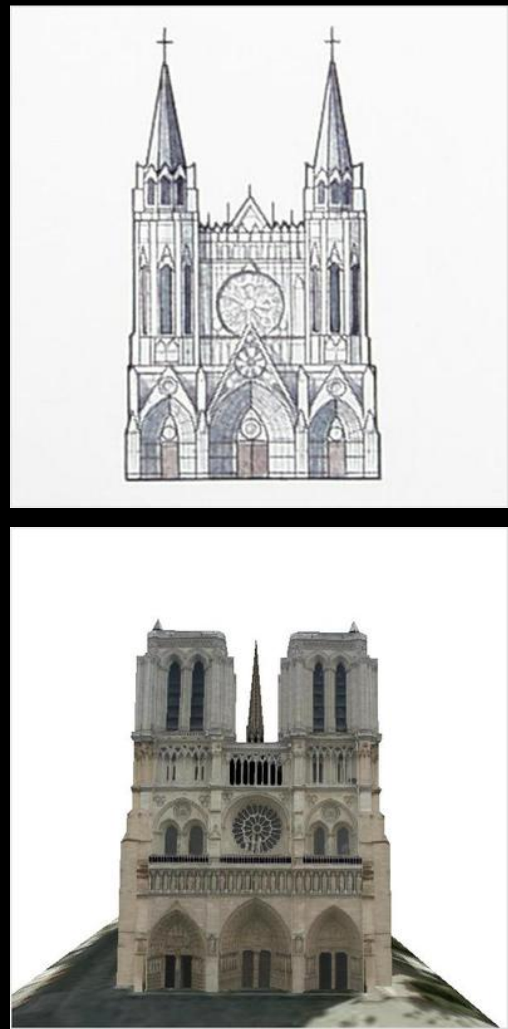

SATURDAY 18 SEPTEMBER
SAN MARCO
Standing in line from
9:25 - 9:45. Plenty
time for sketch!

Challenging examples

Scene distortion



Drawing errors



Different scene

Failures

Extreme change in depiction styles (smeared watercolor)

Part of the architectural site not covered by 3D model

Extreme geometric distortion

Viewing frusta in 3D

Fly-through video

Outline

1. Local invariant features (C. Schmid)

2. Matching and recognition with local features (J. Sivic)

3. Efficient visual search (J. Sivic)

4. **Very large scale visual indexing (C. Schmid)**

Practical session – Instance-level recognition and search