Reconnaissance d’objets et vision artificielle 2010

Instance-level recognition III.
Visual search: extensions and applications

Josef Sivic

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

INRIA, WILLOW, ENS/INRIA/CNRS UMR 8548

Laboratoire d’Informatique, Ecole Normale Supérieure, Paris

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

Announcements

Class web-page:

http://www.di.ens.fr/willow/teaching/recvis10/

Email list: Please add your name and email.

Assignment 1 deadline was extended to
Next Tuesday, Nov 2nd 2010!

Assignment 2: Stitching photo-mosaics
http://www.di.ens.fr/willow/teaching/recvis10/assignment2/
is due next Tuesday, Nov 2nd 2010

Lecture plan

Lecture 2:

- Local invariant features (C.Schmid)

Lecture 3:

- Camera geometry – review (J. Ponce)
- Correspondence, matching and recognition with local features, efficient visual search (J. Sivic)

Lecture 4: (C. Schmid):

- Very large scale visual indexing
- Bag-of-feature models for category-level recognition

Lecture 5 (today):

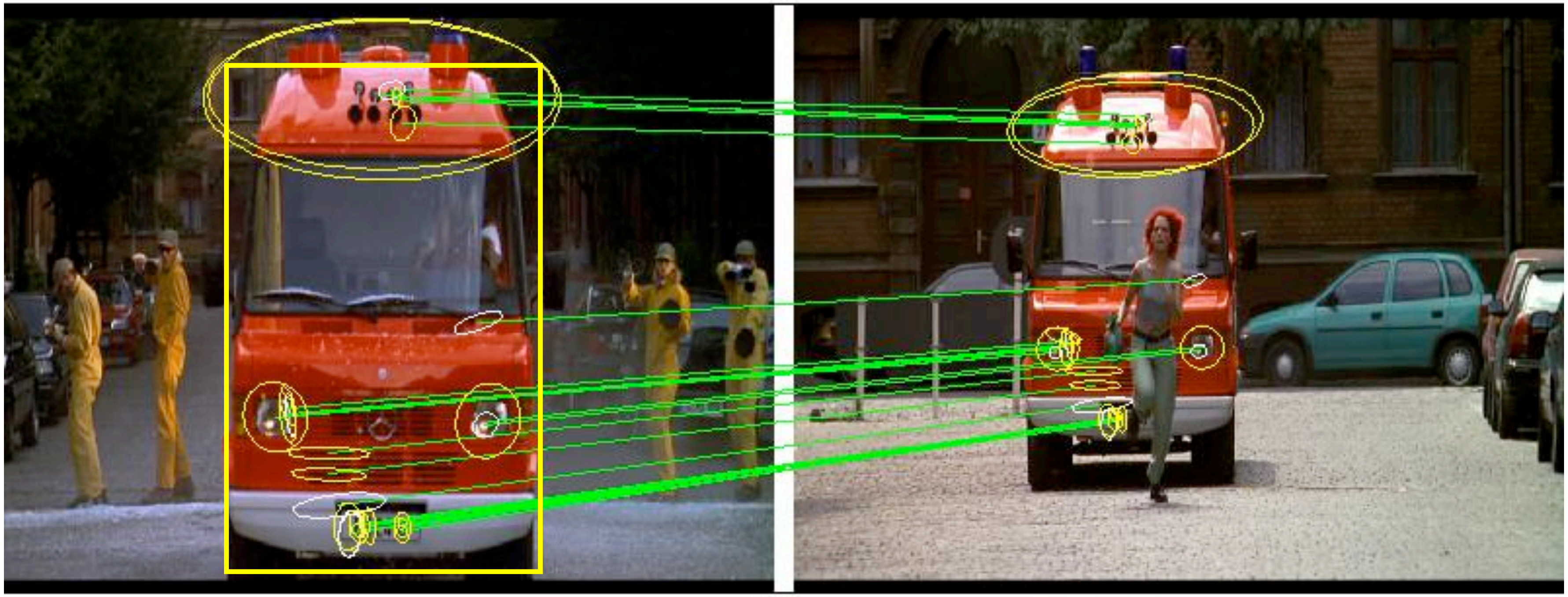
- Sparse coding and dictionary learning (J. Ponce)
- Visual search – extensions and applications (J. Sivic)
- Category-level localization (J. Sivic)

1. Review: Large-scale recognition with local features

Review: recognition with local features

1000+ descriptors per image

Match regions between frames using SIFT descriptors and spatial consistency



Multiple regions overcome problem of partial occlusion

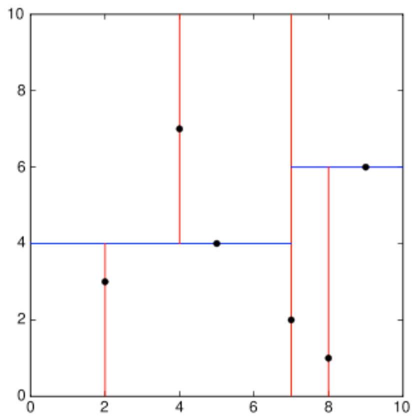
Fast descriptor search

Complexity

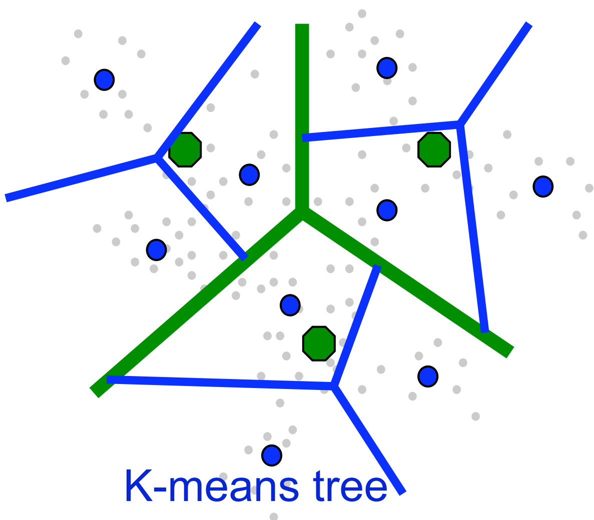
- O(nd) for n features and d dimensions
- Linear in the number of features / images

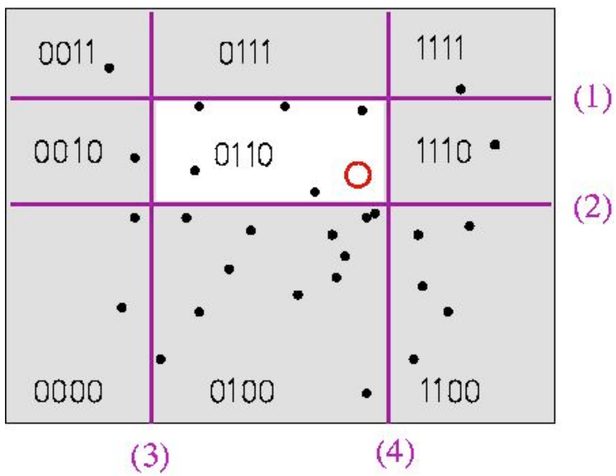
Speed up individual descriptor vector search

- kd-trees (k dim. tree), approximate nearest neighbor search
- K-means tree
- Locality sensitive hashing (LSH)



kd-tree decomposition

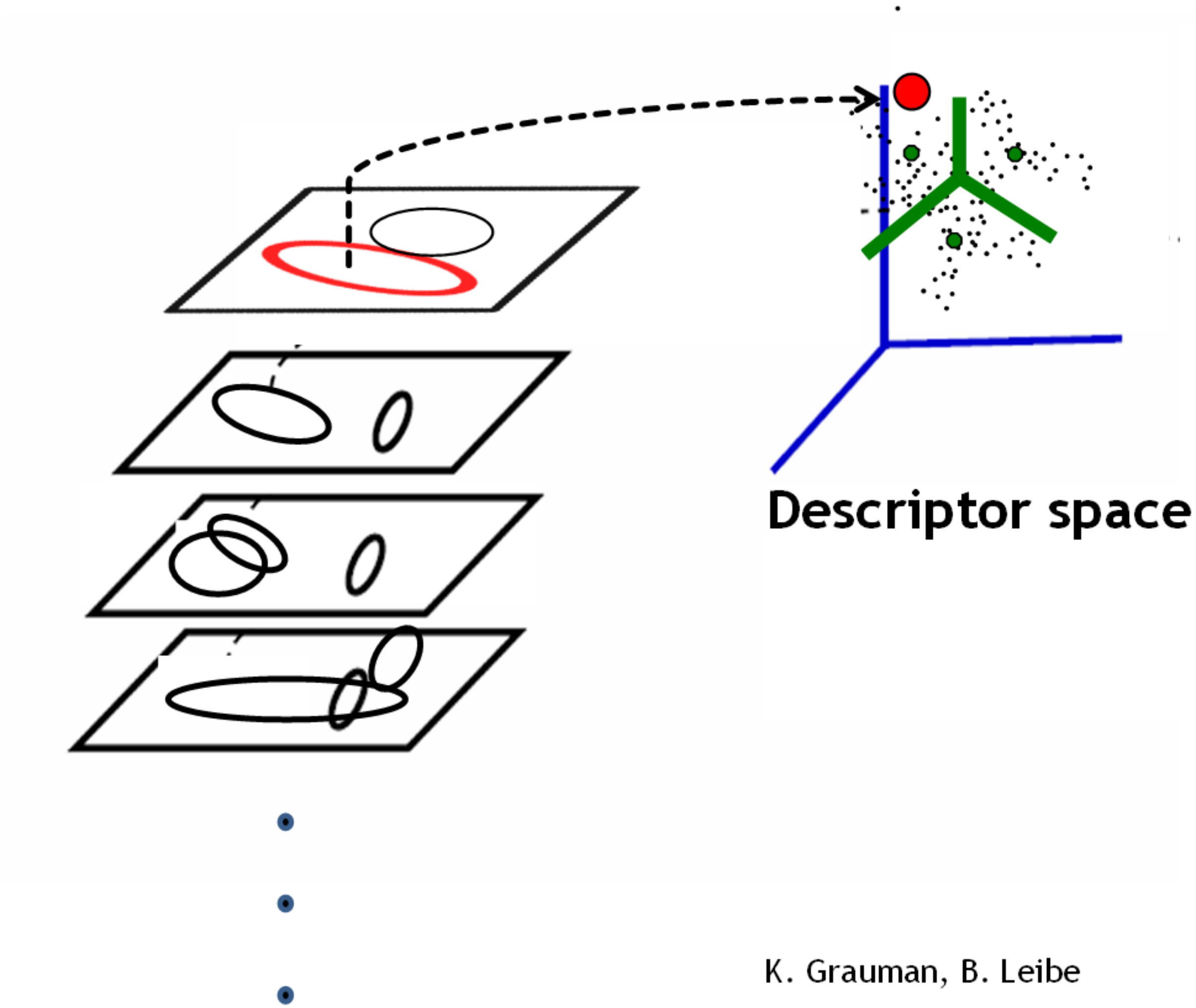

K-means tree



Locality sensitive hashing (LSH)

Visual words: main idea

Map high-dimensional descriptors to tokens/words by quantizing the feature space



- Determine which word to assign to each new image region by finding the closest cluster center.

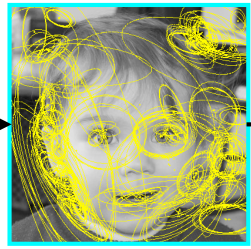
K. Grauman, B. Leibe

Bag-of-features / Bag-of-visual-words [Sivic&Zisserman'03]

Query image

Harris-Hessian-Laplace regions + SIFT descriptors

Set of SIFT descriptors



centroids (visual words)

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

sparse frequency vector

Inverted file

querying

ranked image short-list

Geometric verification

[Chum & al. 2007]

Re-ranked list

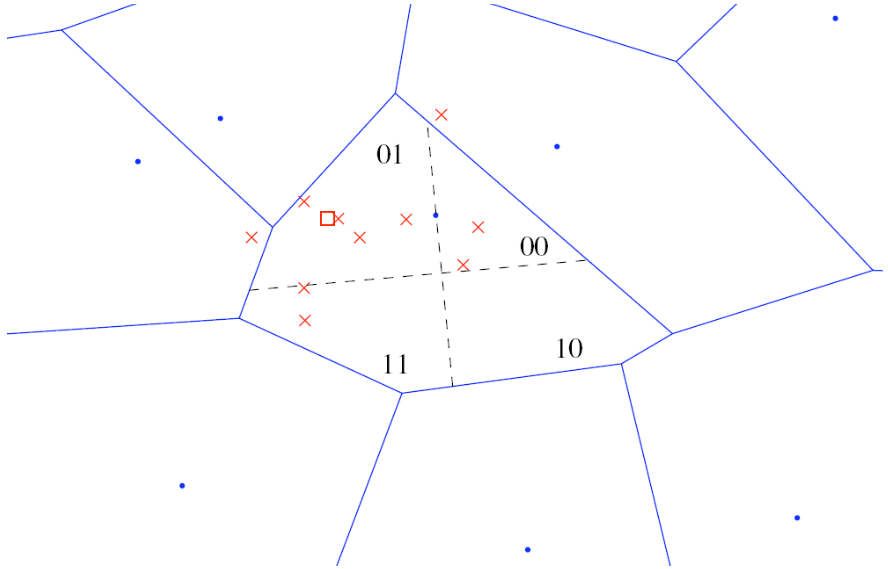
"visual words":

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

=> 8 GB fits!

Beyond visual words: Hamming Embedding

[Jegou et al. ECCV'08]



Representation of a descriptor *x*

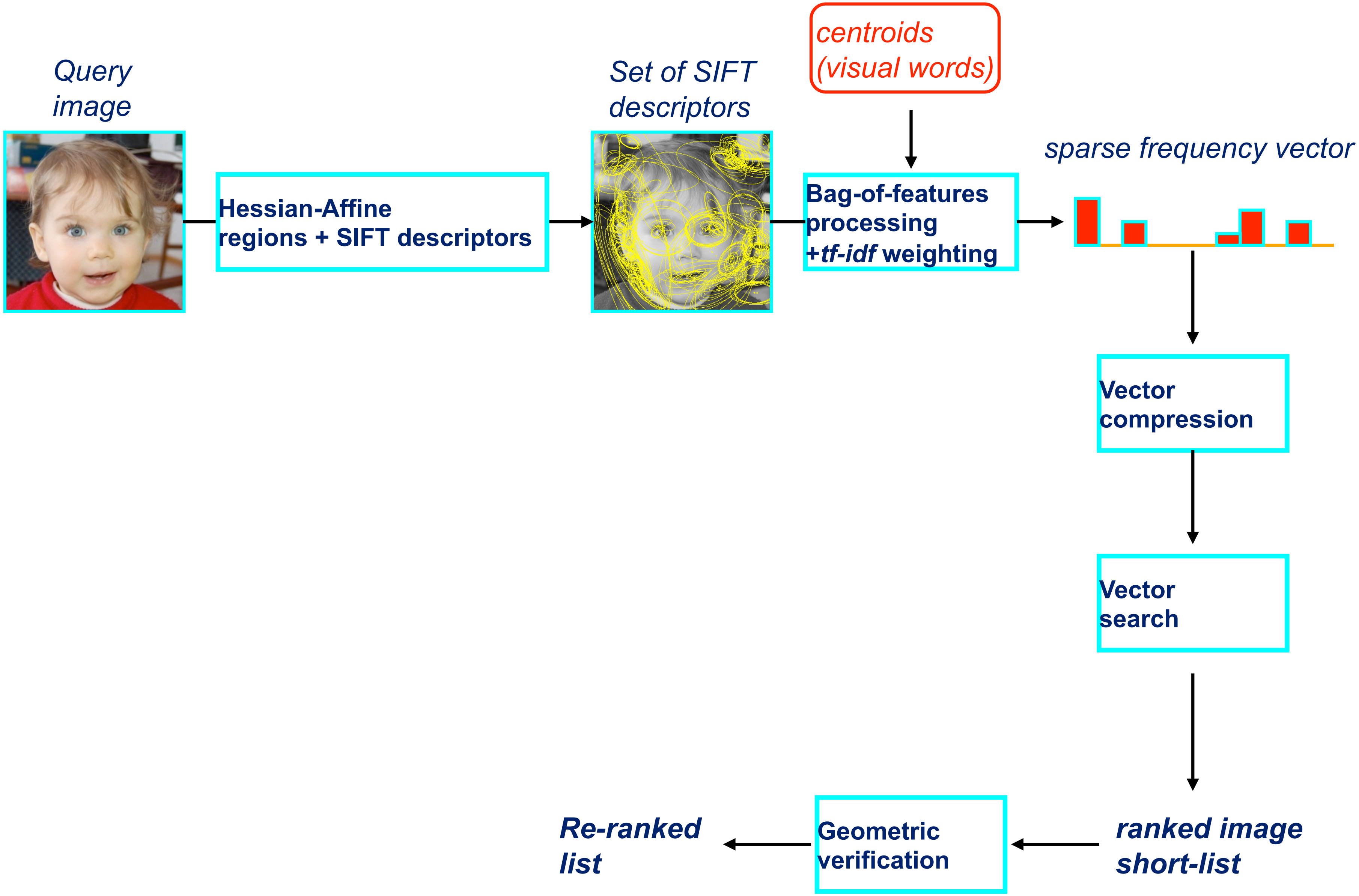
- Vector-quantized to *q(x)* as in standard BOF
- \+ short binary vector *b(x)* for an additional localization in the Voronoi cell

Two descriptors x and y match iff

$$f_{\mathrm{HE}}(x,y) = \begin{cases} (\text{tf-idf}(q(x)))^2 & \text{if } q(x) = q(y) \\ & \quad \text{and } h\left(b(x), b(y)\right) \leq h_t \\ 0 & \text{otherwise} \end{cases}$$

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

Recent approaches for very large scale indexing



VLAD : vector of locally aggregated descriptors

- Simplification of Fisher kernels

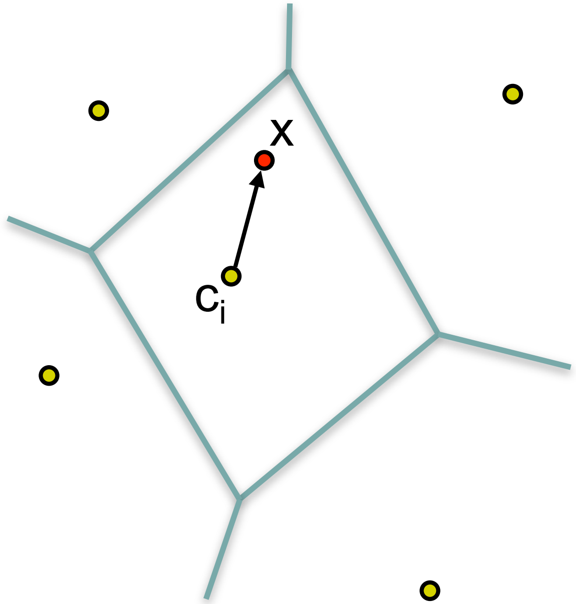
- Learning: a vector quantizer (*k*-means)
  - output: *k* centroids (visual words): $c_1,\ldots,c_i,\ldots c_k$
  - centroid c_i has dimension *d*

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

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

- VLAD (dimension $D = k \times d$)

- The vector is L2-normalized



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

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

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

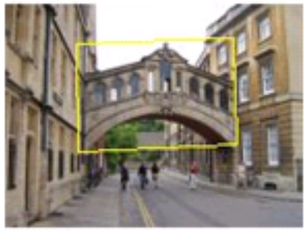
1

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

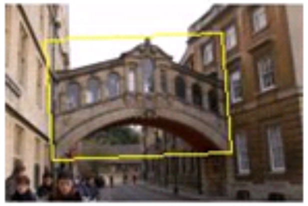
ID: oxc1_all_souls_000075
Score: 352.000000
Putative: 645
Inliers: 352
Hypothesis: 1.162245 0.041211 -70.414459 -0.012913 1.146417 91.276093
Detail

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

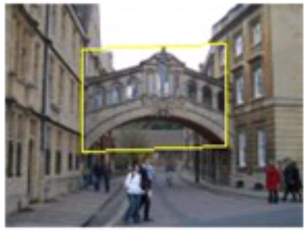


ID: oxc1_oxford_001612
Score: 252.000000
Putative: 451
Inliers: 252
Hypothesis: 1.046026 0.069416 51.576881 -0.044949 1.046938 76.264442
Detail

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

2. Visual search - extensions

- Query expansion
- Pre-computing matching graph
- Overcoming quantization errors
- Retrieval in structured databases

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

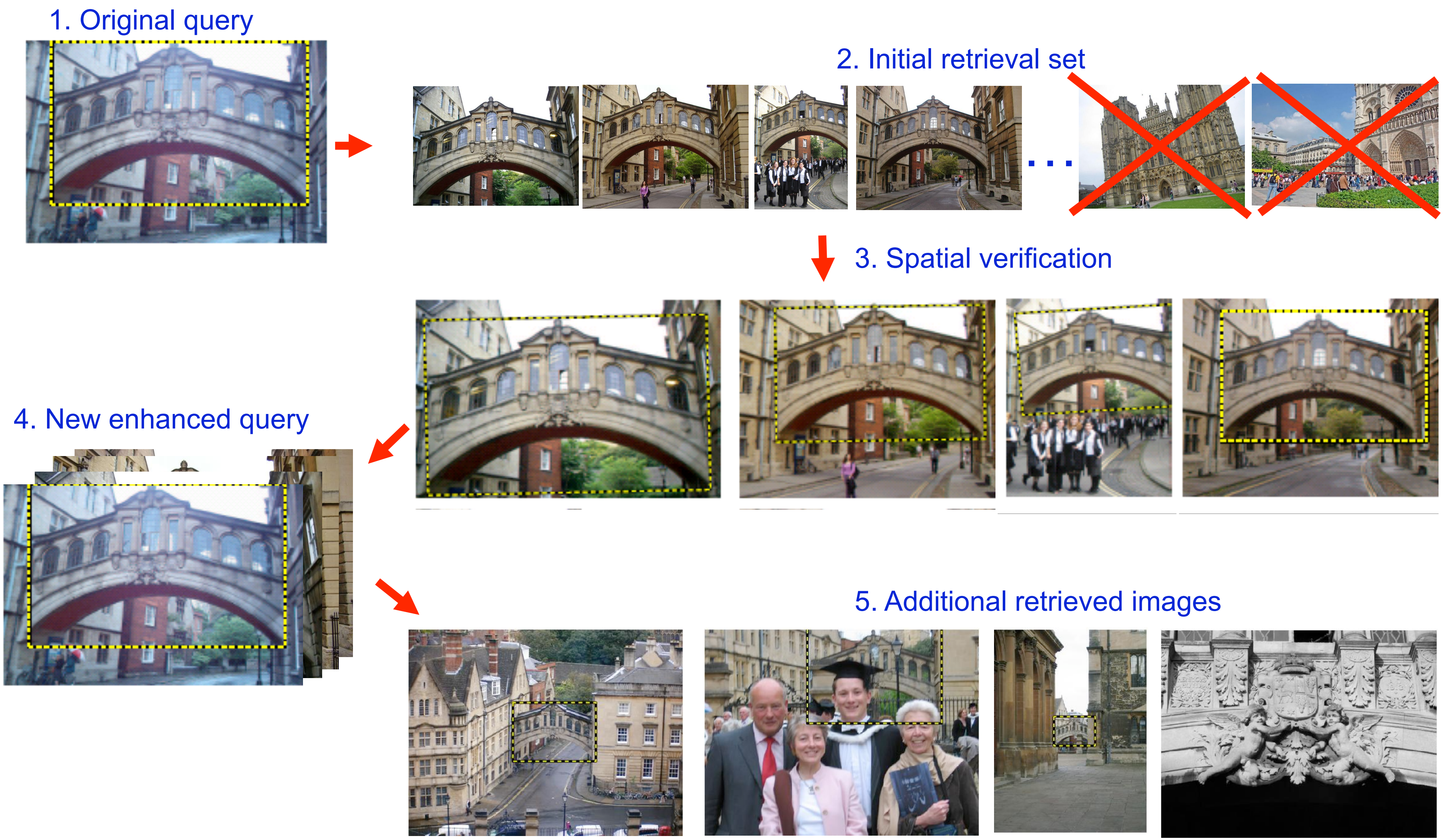
Visual word representations of two images of the same object may differ (due to e.g. detection/quantization noise) resulting in missed returns

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

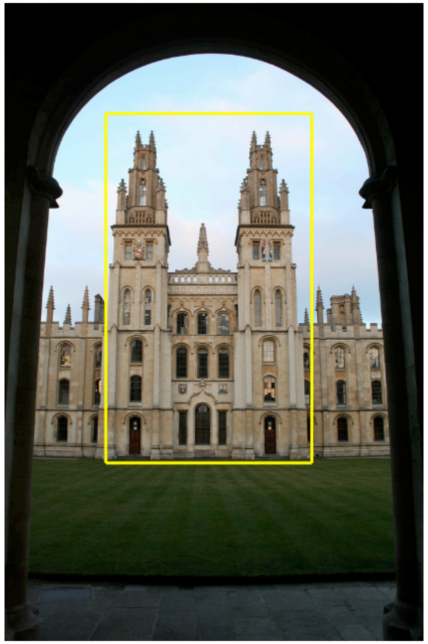
Strong spatial model prevents ‘drift’ by discarding false positives

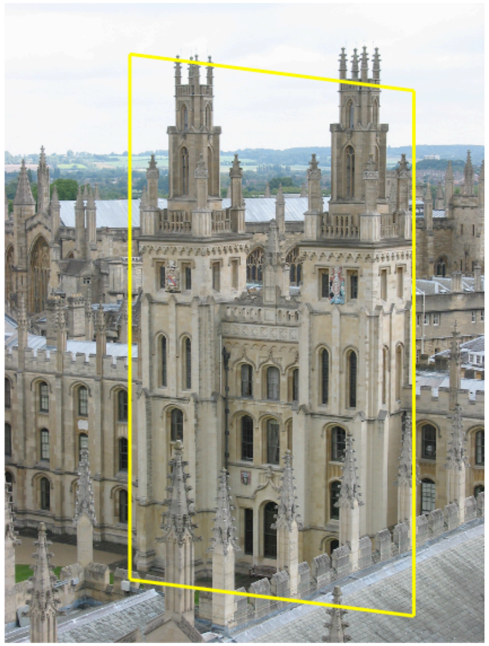
[Chum, Philbin, Sivic, Isard, Zisserman, ICCV’07]

Visual query expansion - overview



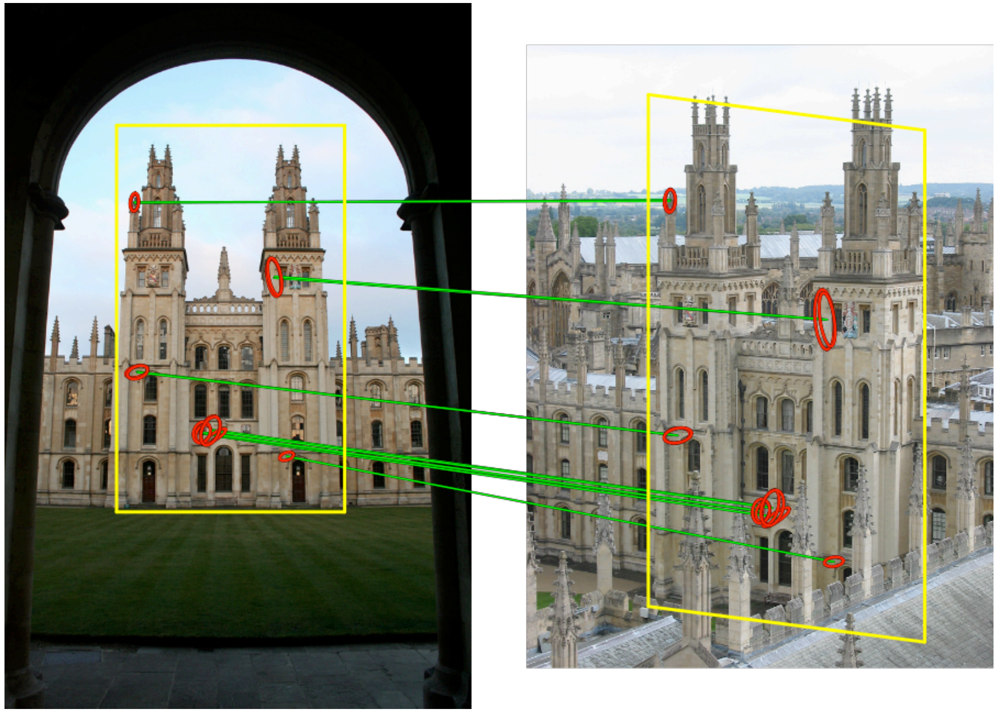
Query Expansion


Query Image

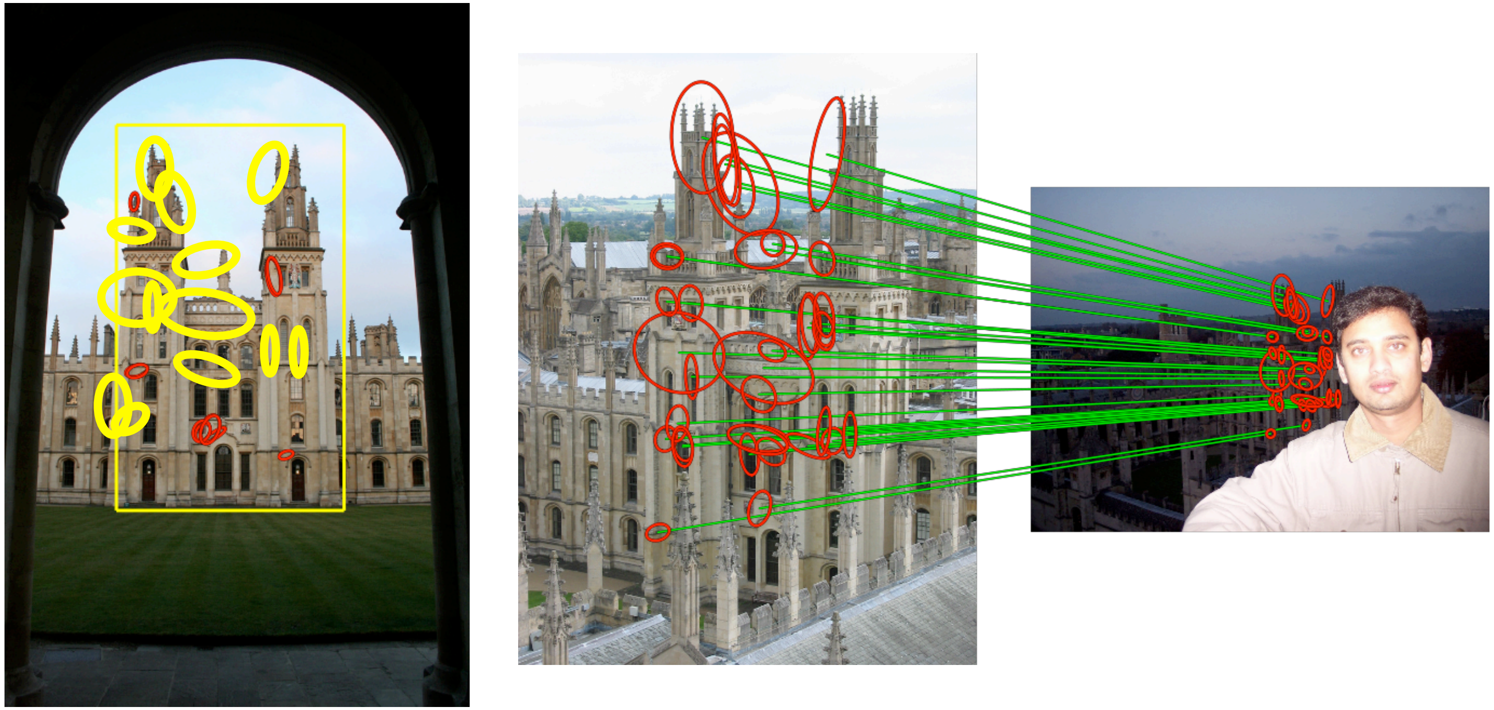

Originally retrieved image


Originally not retrieved

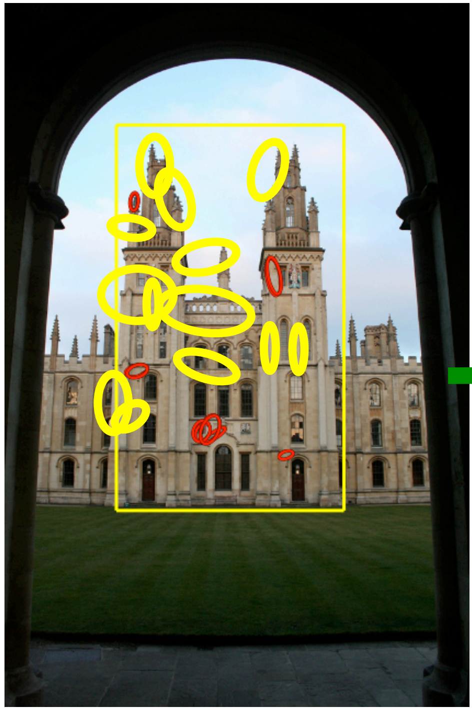
Query Expansion

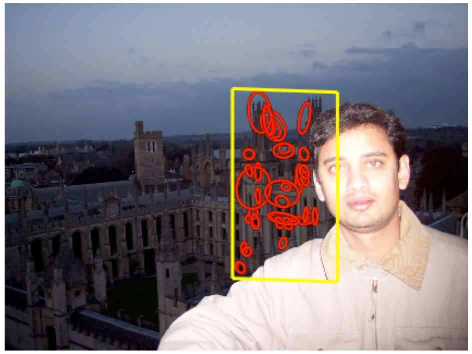


Query Expansion



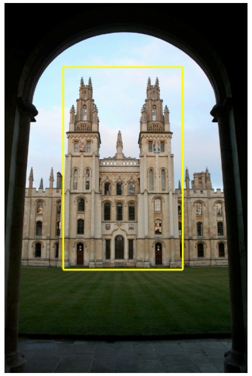
Query Expansion



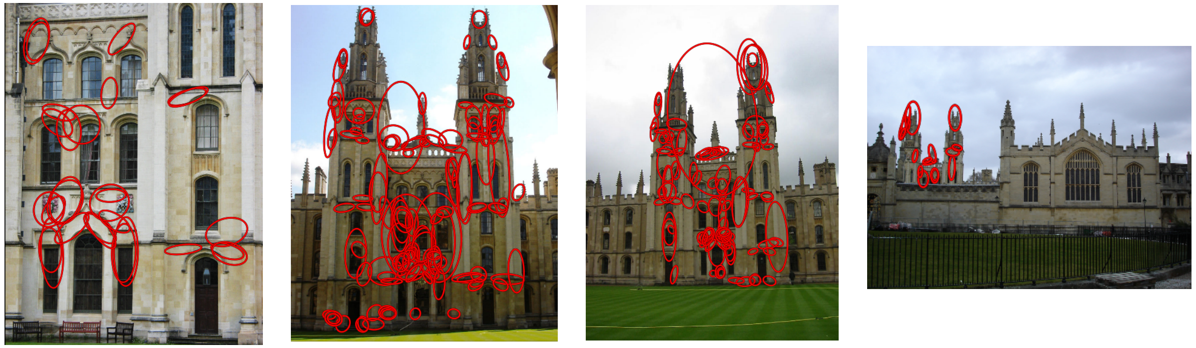


Query Expansion

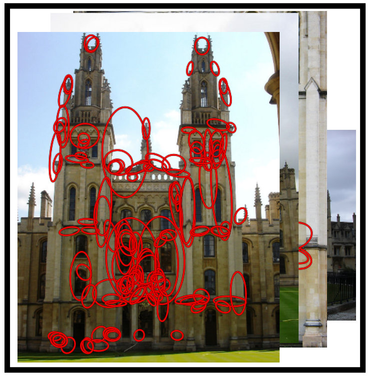
Query Image



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

Efficient visual search for objects and places

Oxford Buildings Search - demo

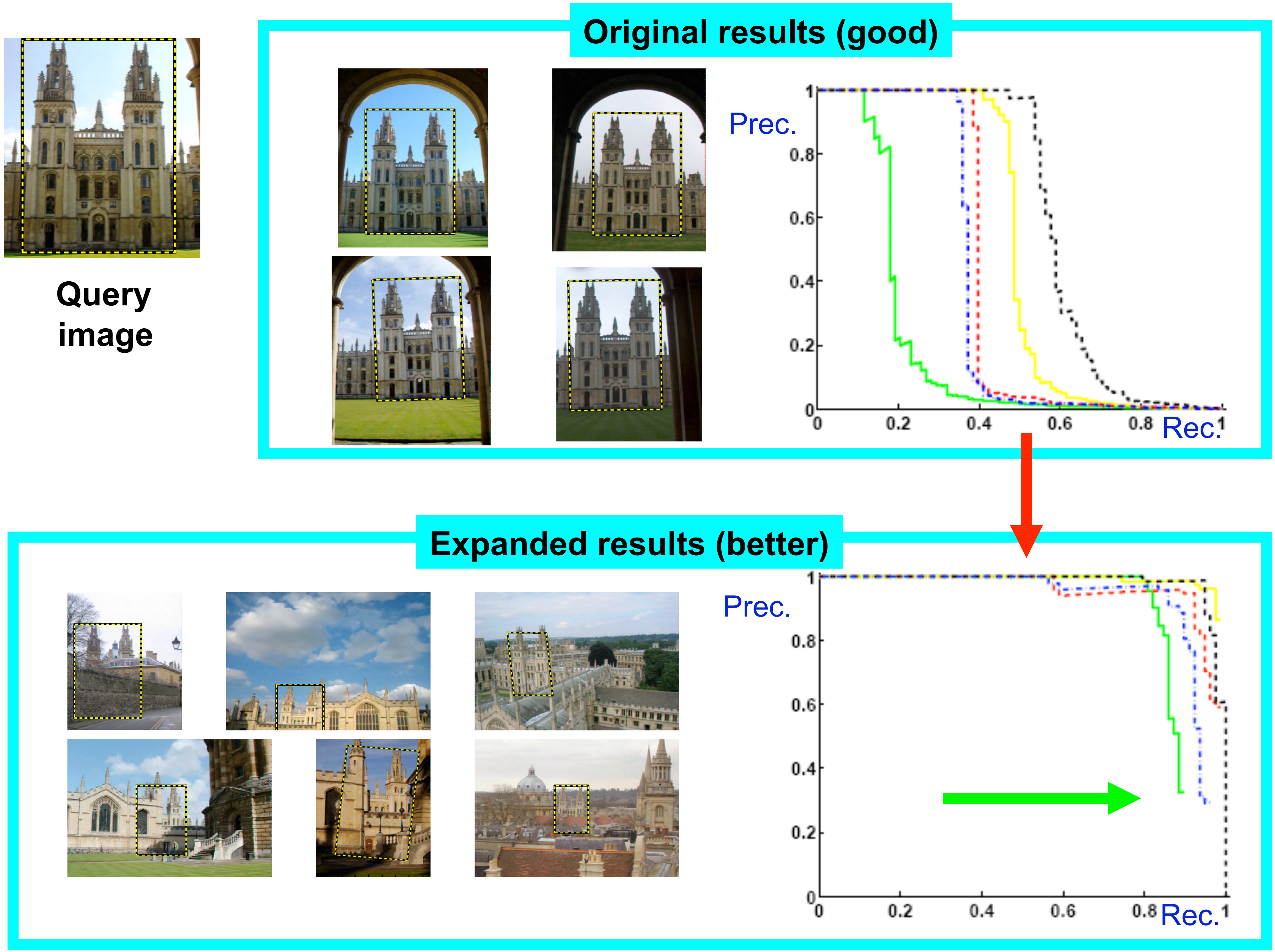
http://www.robots.ox.ac.uk/~vgg/research/oxbuildings/index.html

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

Pre-compute query expansion?

- Query expansion works well, however, at an additional cost at the query time.

- Can we offline pre-process the database and pre-compute the query expansion?

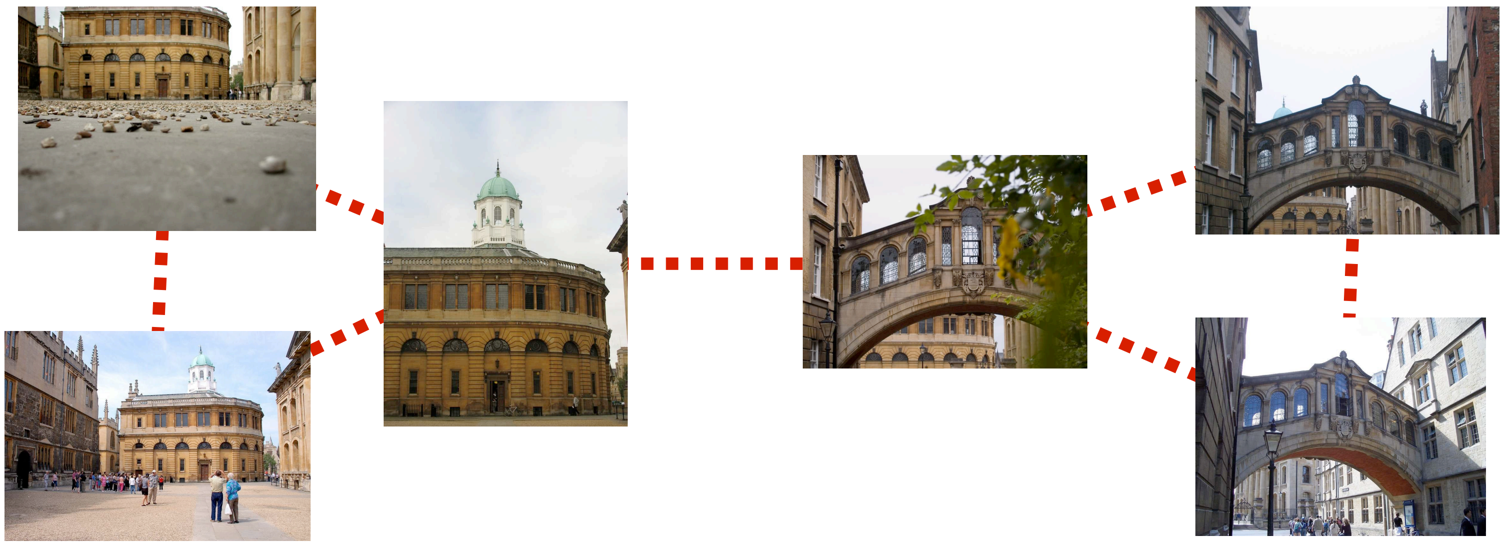
Solution: Compute and build a matching graph.

Matching graph

Build a ‘matching graph’ over all the images in the dataset

Each image is a node and a link represents two images having some object in common

Instead of expanding the query, traverse links of this graph



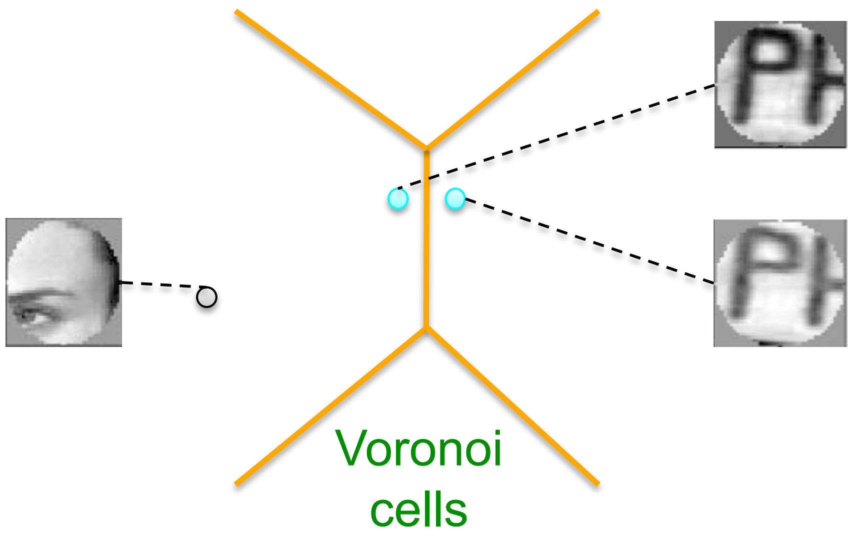
[Chum et al. 2008, Philbin et al. IJCV 2010, Turcot and Lowe 2009]

Example:

Quantization errors

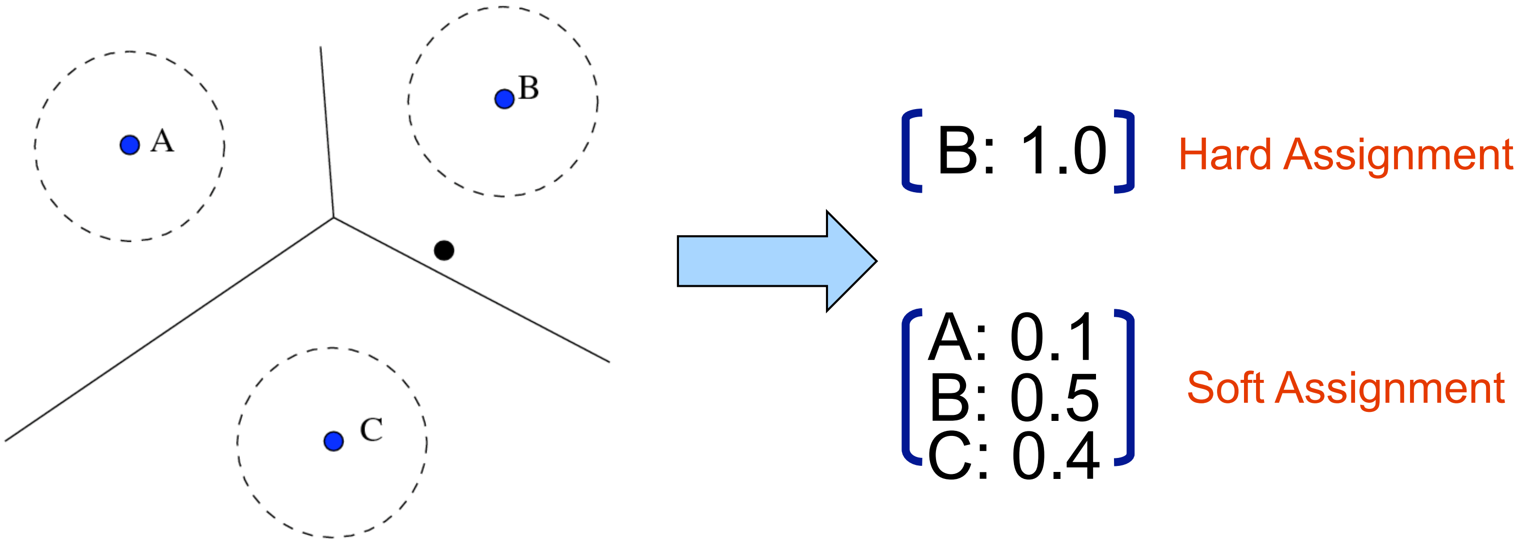
Typically, quantization has a significant impact on the final performance of the system **[Sivic03,Nister06,Philbin07]**

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



Overcoming quantization errors

- Query expansion. [Chum et al. 2007]
- Soft-assignment. [Philbin et al. 2008]
- Hamming embedding / VLAD [Jegou&Schmid ’08, ‘10]



Overcome errors **given** a quantization.
Have cost in terms of space and/or time complexity at query-time

Descriptor learning for efficient retrieval

The aim of this work is to reduce these errors ***at source***, by learning a projection function that actively reduces this error:

$$T(x;W) \qquad T : \mathbb{R}^D \to \mathbb{R}^M$$

$$d_W(x,y) = \|T(x;W) - T(y;W)\|_2$$

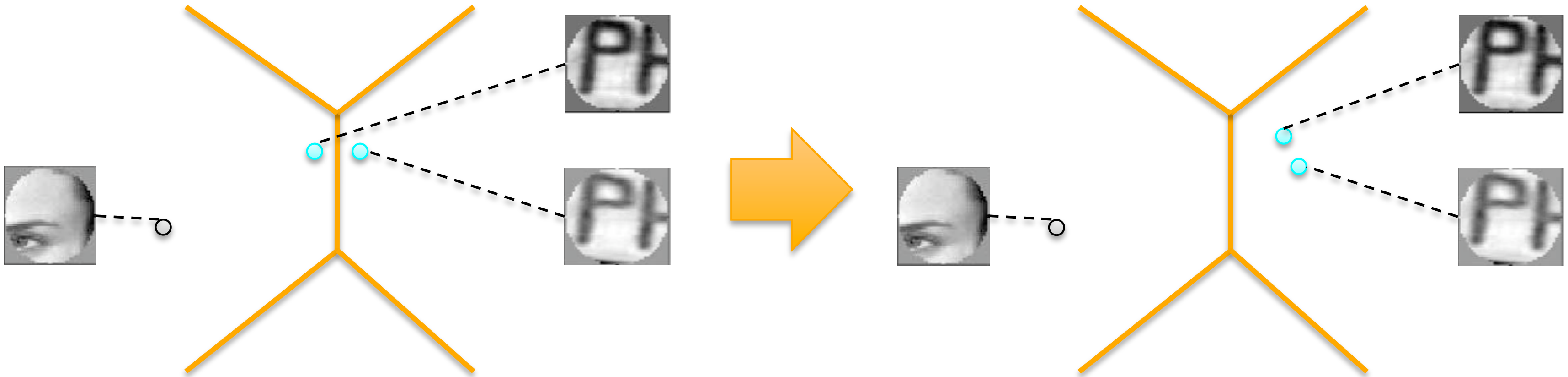
- T can be linear or non-linear and we can choose keep the descriptor dimensionality the same or reduce it
- After this projection, use the same visual words architecture

[Philbin, Isard, Sivic, Zisserman, ECCV 2010]

Descriptor learning for efficient retrieval

- No additional query-time cost over BOW

- For particular object retrieval, we can leverage the spatial consistency between object instances to automatically generate large amounts of training data (matched / non matched point pairs)



Confusion and splitting

No confusion or splitting

Descriptor learning for efficient retrieval

Choose form of $T(x;W)$:

- Can be linear: $T(x;W) = Wx$

- Or non-linear (DBN-style formulation):

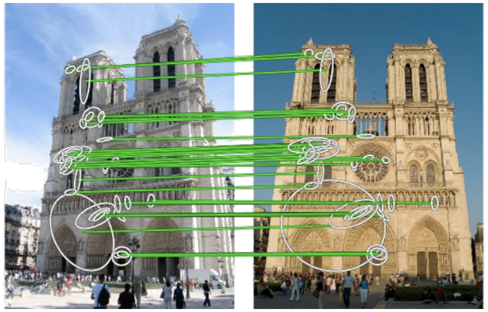
$$T(x;W_1,W_2,W_3,h_0,h_1,h_2) =$$

$$W_3\sigma(W_2\sigma(W_1\sigma(x+h_0)+h_1)+h_2)$$

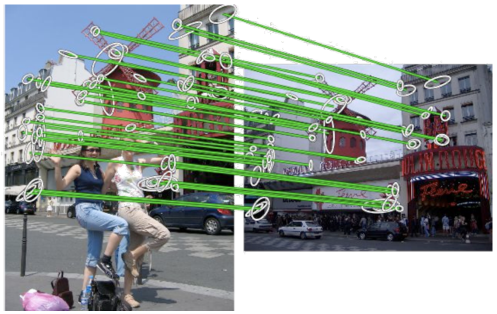
Non-linear model gives better results.

Results: Spatial Verification

26 inliers



38 inliers

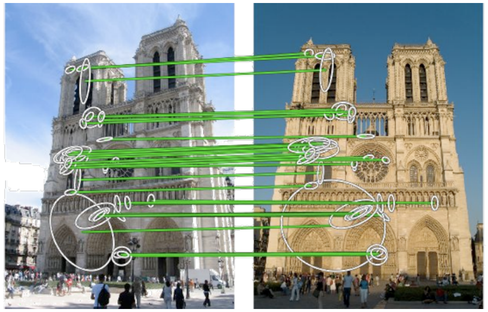


49 inliers

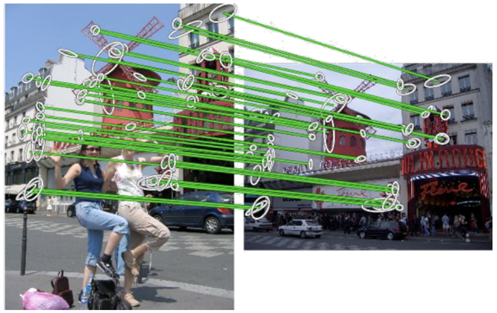
Quantized 128-D SIFT descriptors (K=1M)

Results: Spatial Verification

26 inliers

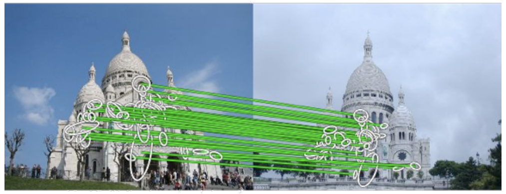


38 inliers

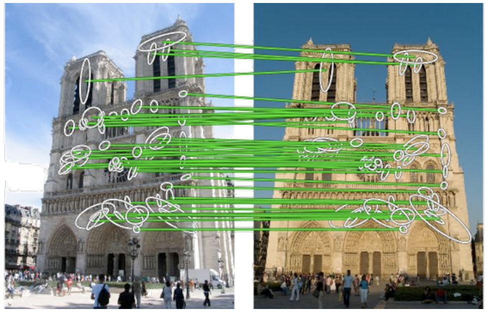


49 inliers

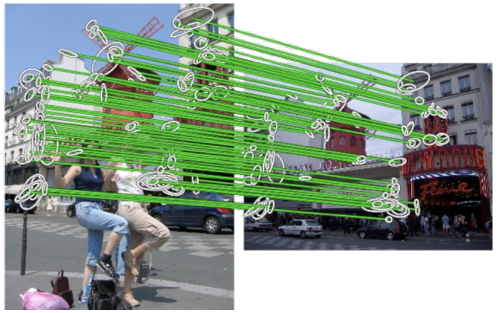
Quantized 128-D SIFT descriptors (K=1M)



48 inliers



61 inliers

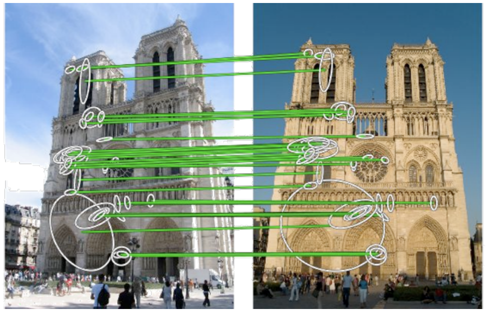


114 inliers

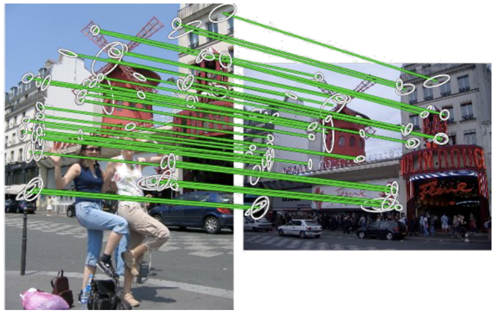
Raw 128-D SIFT

Results: Spatial Verification

26 inliers

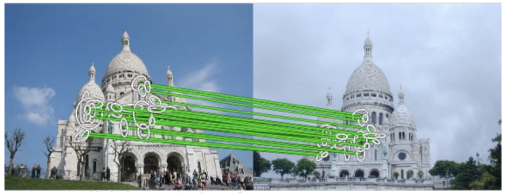


38 inliers

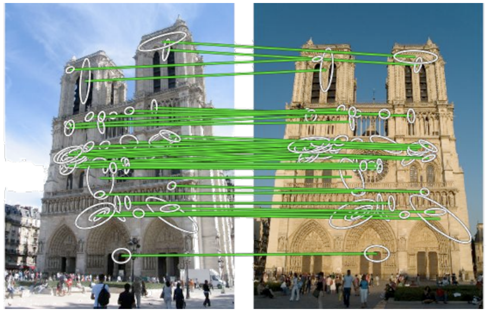


49 inliers

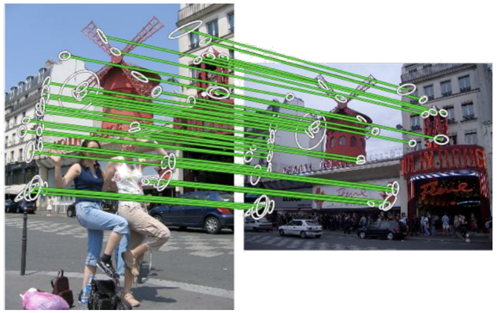
Quantized 128-D SIFT descriptors (K=1M)



37 inliers



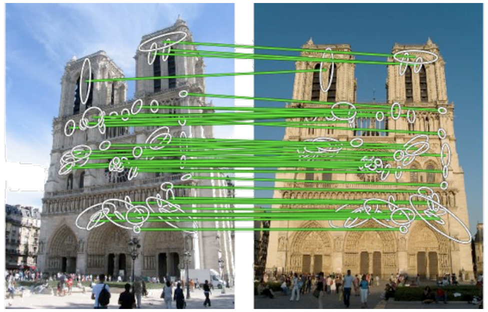
56 inliers



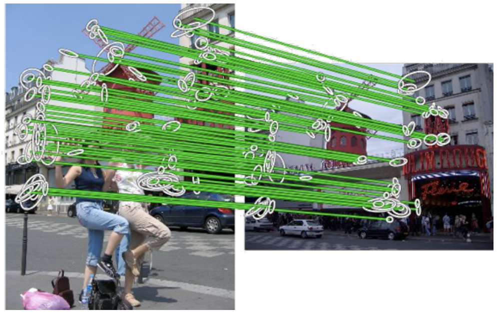
64 inliers

Quantized 32-D learnt descriptors (K=1M)

48 inliers



61 inliers



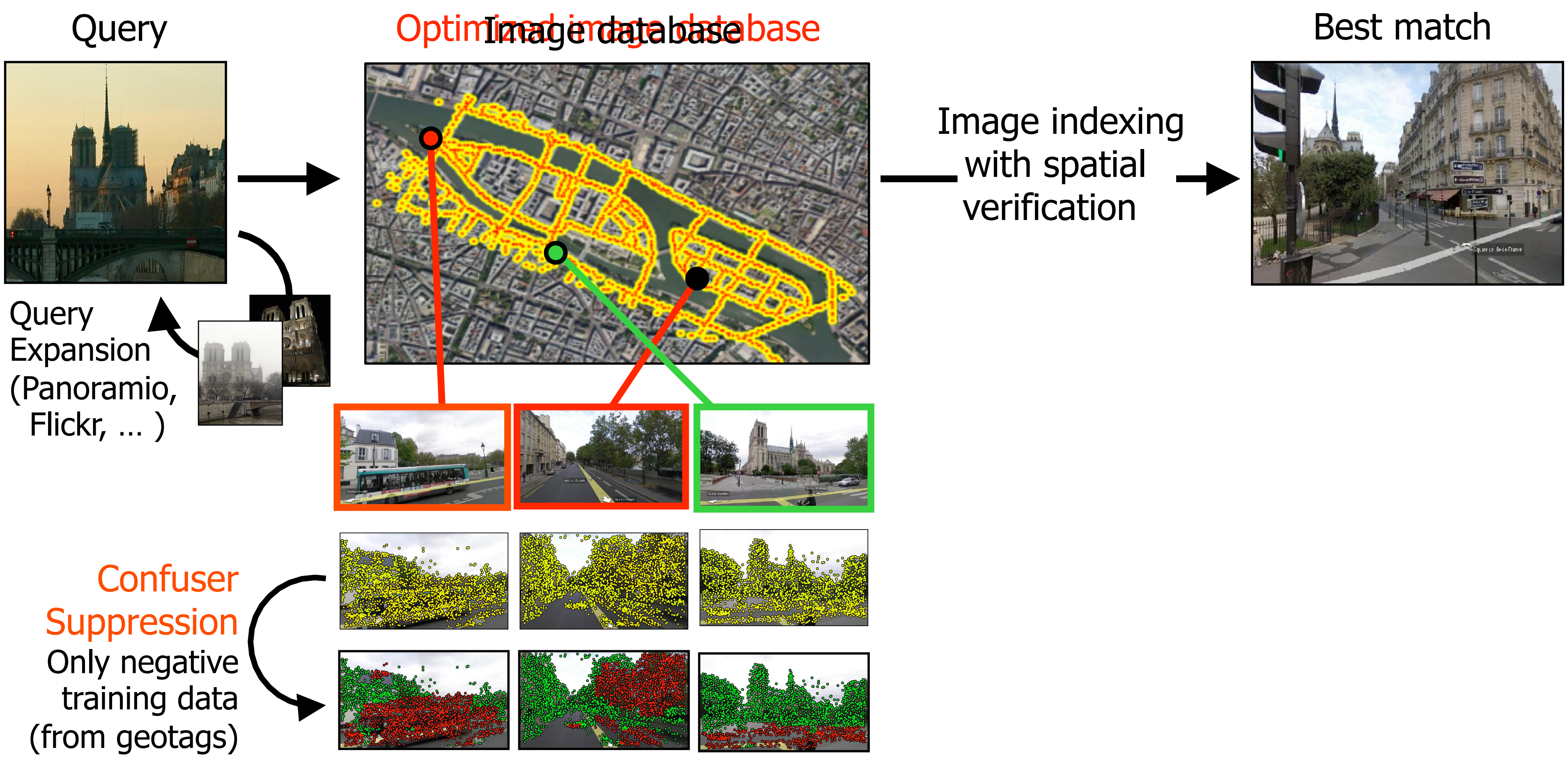
114 inliers

Raw 128-D SIFT

Results: Baseline to State of the Art

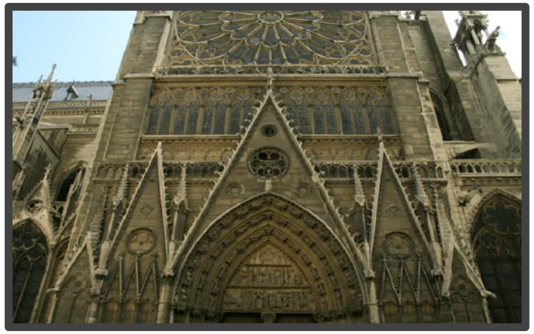
	Mean Average Precision
1. Baseline Method K = 10K	**0.389**
2. Large Vocabulary K=1M	**0.618**
3. Spatial Re-ranking	**0.653**
4. Soft Assignment (SA)	**0.731**
Learnt descriptors	**0.707**
5. Query Expansion (QE)	**0.801**
6. SA & QE	**0.825**

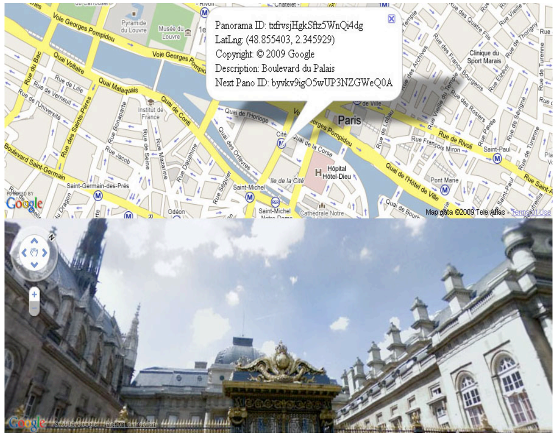
Place recognition:
retrieval in a **structured** (on a map) database

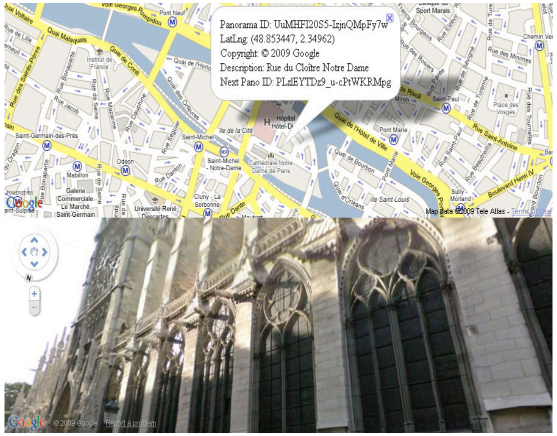


[Knopp, Sivic, Pajdla, ECCV 2010]

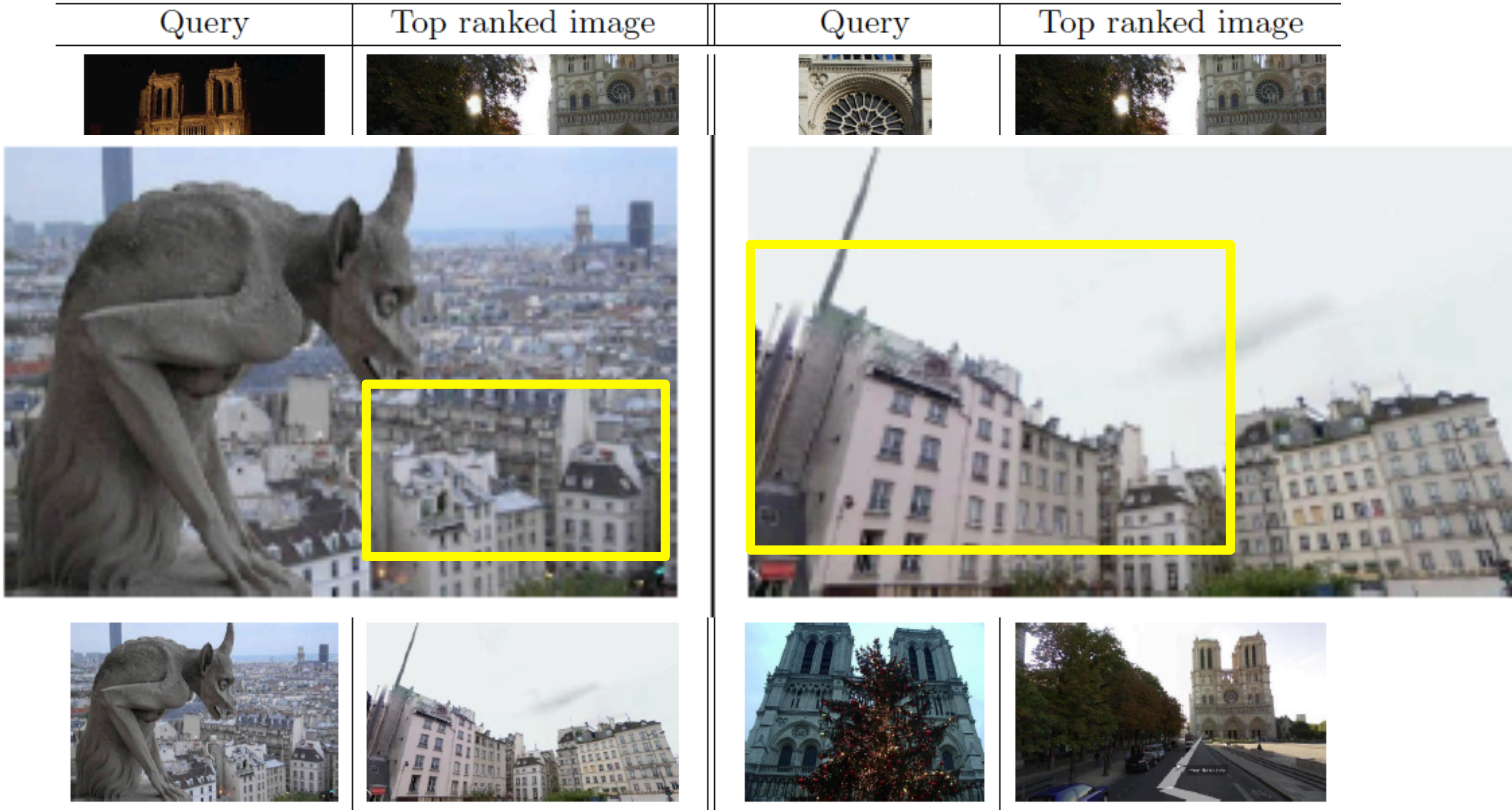
Correctly recognized examples







More correctly recognized examples



Quantitative evaluation

- 200 challenging test queries downloaded from Panoramio
- ~17,000 geotagged images downloaded from Google Street View

Method	**% correct** *initial retrieval*	**% correct** *with spatial verification*
a. Baseline place recognition	20.96	29.34
b. Query expansion	26.35	41.92
c. Confuser suppression	29.94	37.72
d. Confuser suppression+Query expansion	32.93	**47.90**

Table 1. Percentage of correctly localized test queries for different place recognition approaches.

Other recent work

Learning a vocabulary to overcome quantization errors
[Mikulik et al. ECCV 2010]

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

Very large scale retrieval -- towards 1 billion images
[Jegou et al. CVPR 2010] Last lecture!

Matching in structured datasets (3D landmarks or street-view images)
[Knopp et al. ECCV 2010, Zamir&Shah ECCV 2010, Li et al. ECCV 2010, Baatz et al. ECCV 2010]

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.

3. Example applications of large scale visual search and matching

Sony Aibo (Evolution Robotics)

SIFT usage

- Recognize docking station
- Communicate with visual cards

Other uses

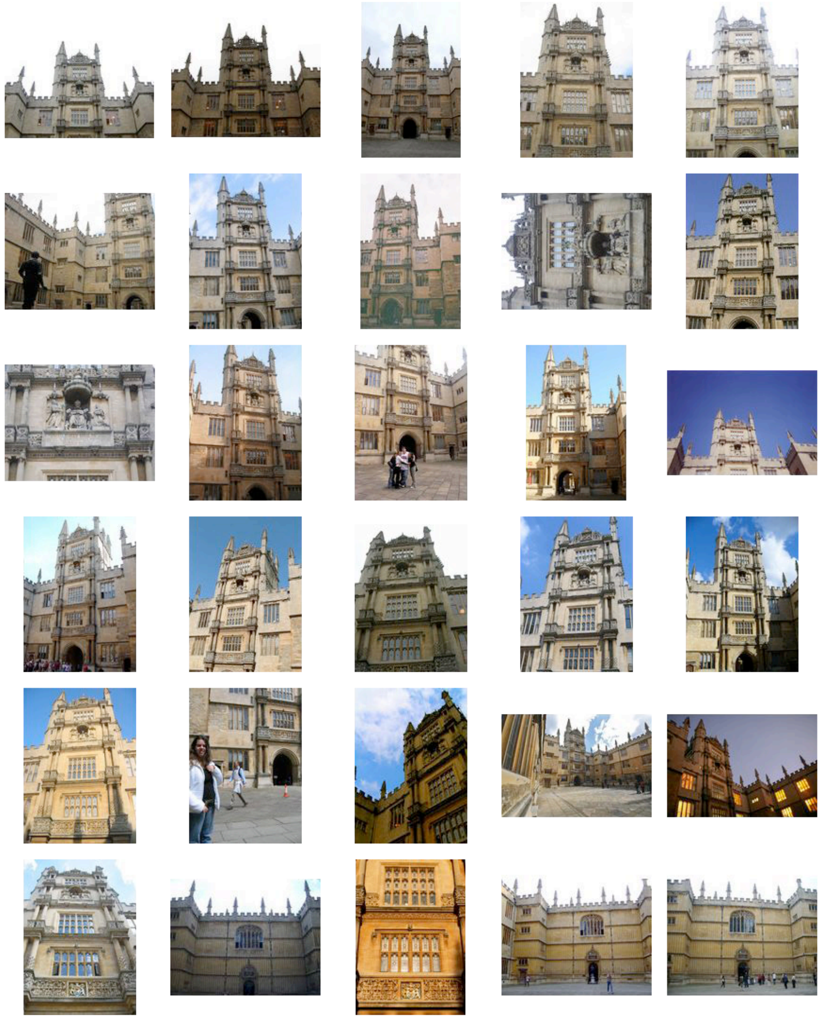
- Place recognition
- Loop closure in SLAM

Slide credit: David Lowe

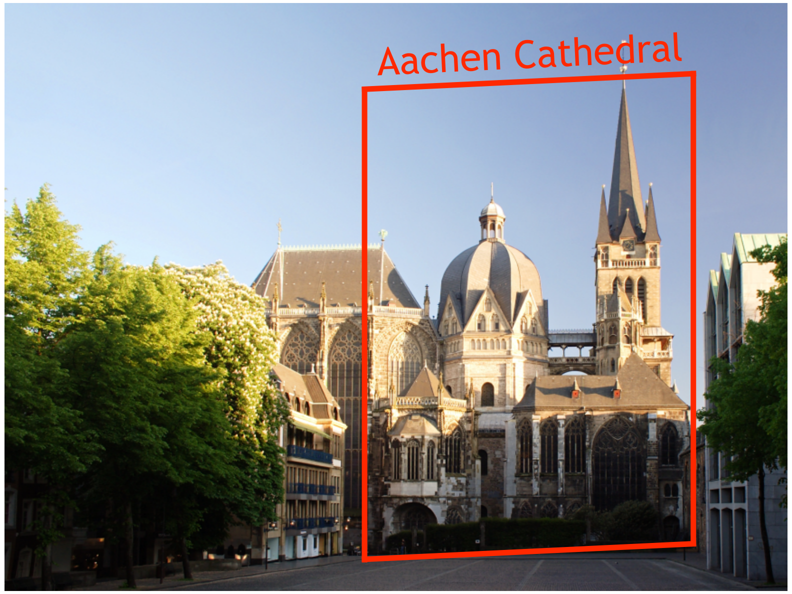
Application: Internet-based inpainting

Photo-editing using images of the same place

[Whyte, Sivic and Zisserman, 2009]



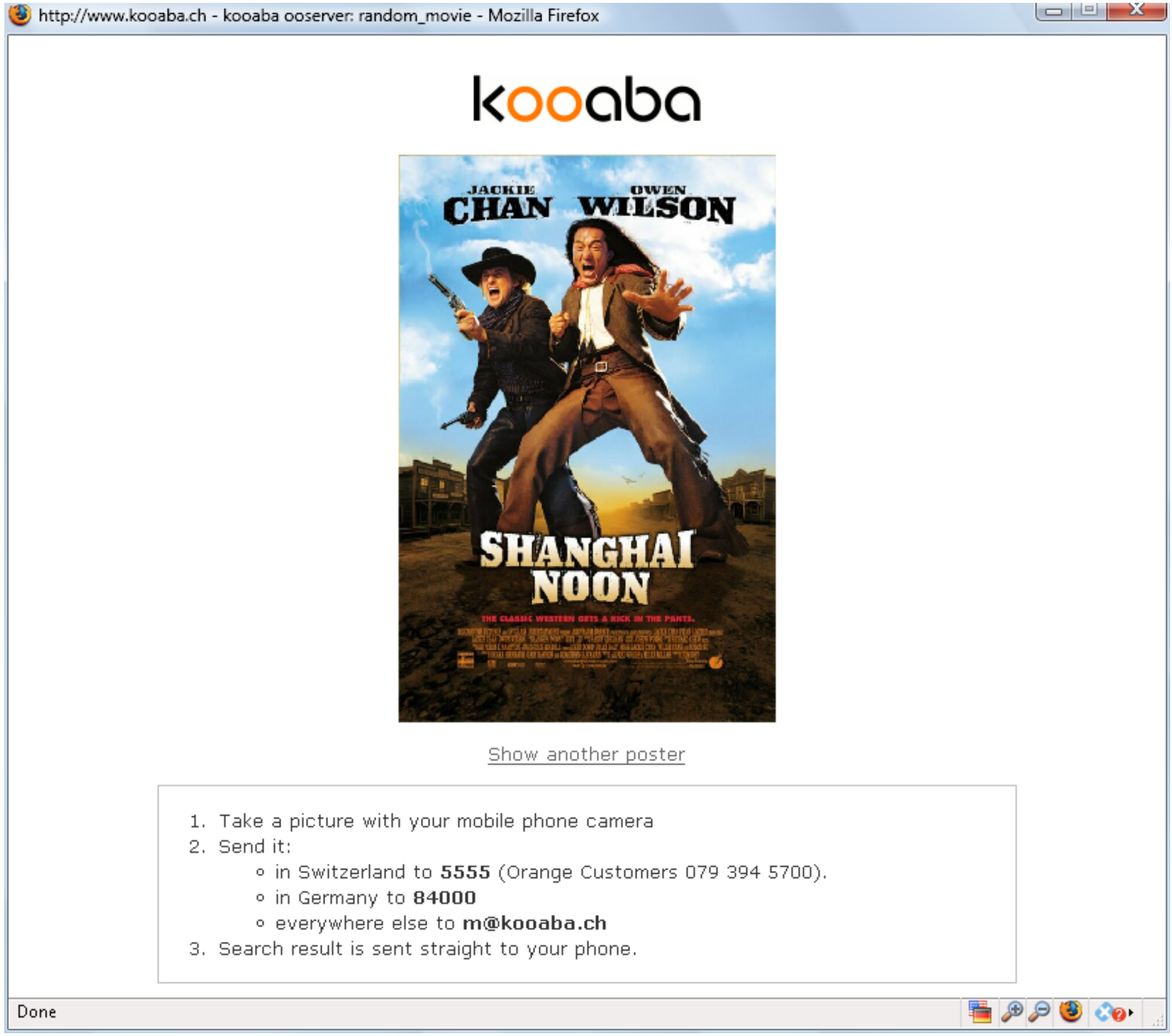
Mobile tourist guide



Mobile tourist guide
- **Self-localization**
- **Object/building recognition**
- **Photo/video augmentation**

[Quack, Leibe, Van Gool, CIVR'08]

Web Demo: Movie Poster Recognition



50'000 movie posters indexed

Query-by-image from mobile phone available in Switzer-land

http://www.kooaba.com/en/products_engine.html#

K. Grauman, B. Leibe

Image Auto-Annotation

Left: Wikipedia image
Right: closest match from Flickr

[Quack CIVR'08]

Visual search in your pocket

Google Goggles

Use pictures to search the web. ▷ Watch a video

Google Goggles

Use pictures to search the web. ▷ Watch a video

but it doesn't work well yet on things like food, cars, plants, or animals.

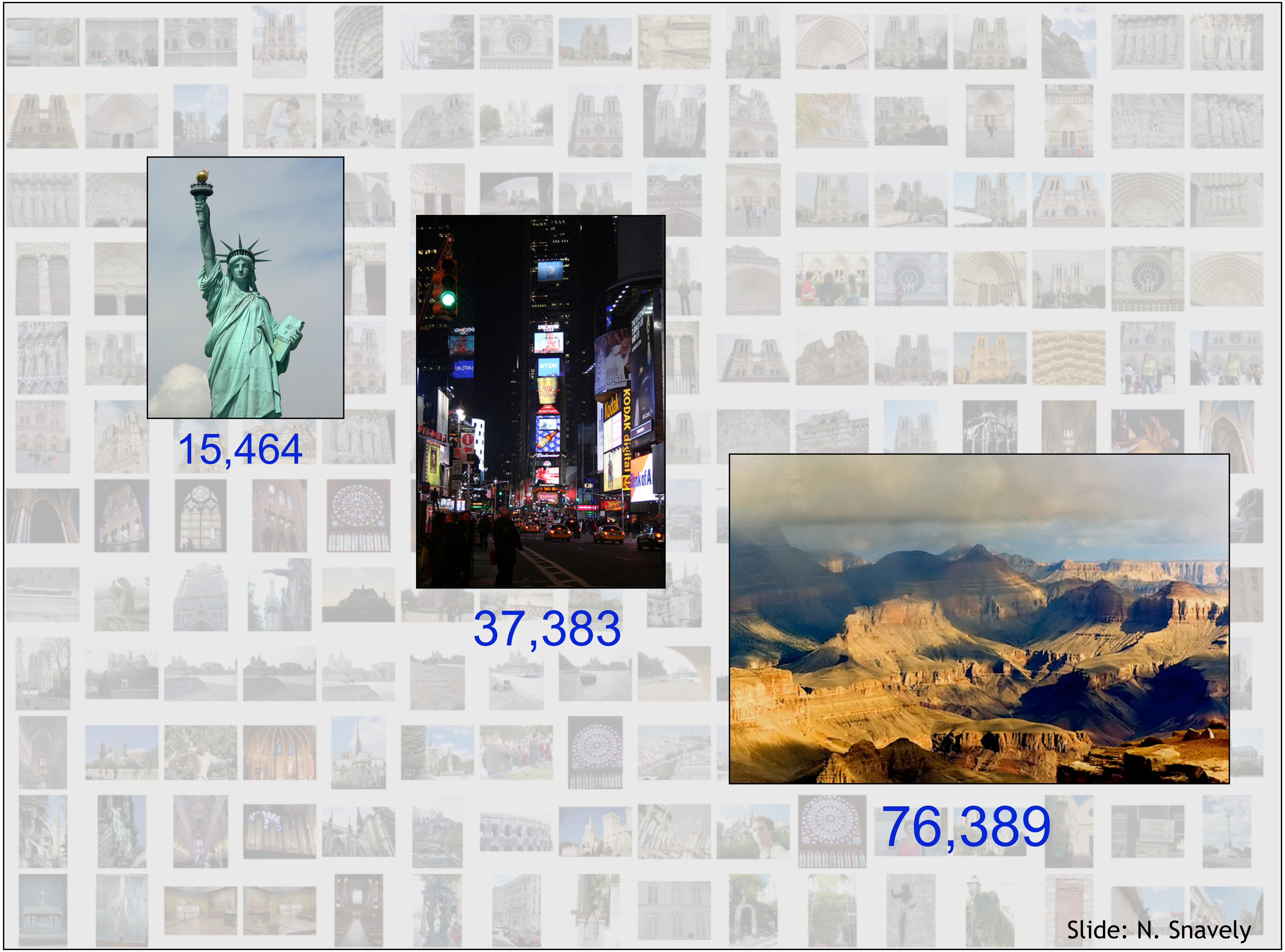
Building Rome in a Day – or –

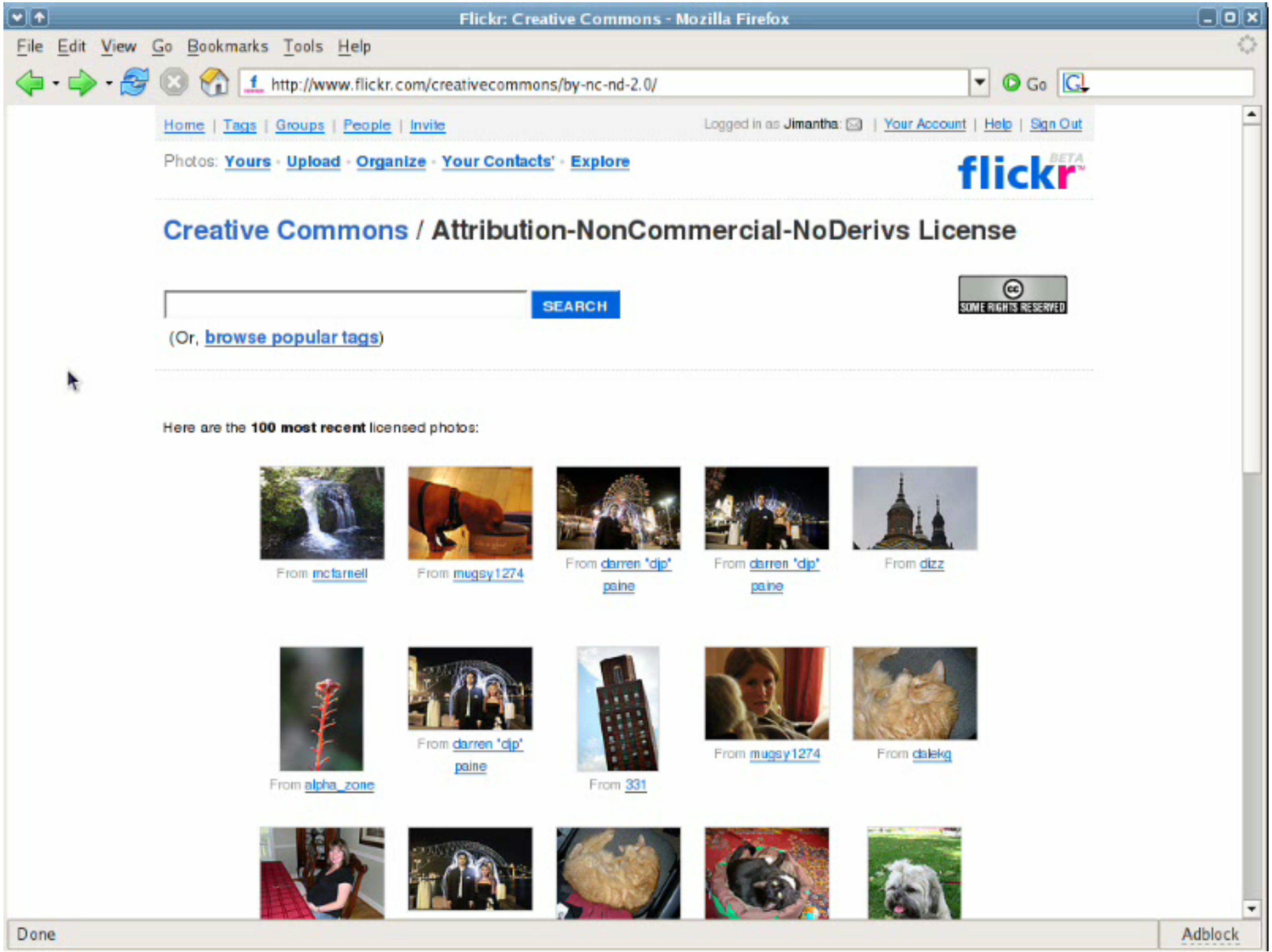
matching and 3D reconstruction in large unstructured datasets.

Goal: Build a 3D model of a city from
a large collection of images downloaded from the Internet

Use a cluster with 500 CPU cores.

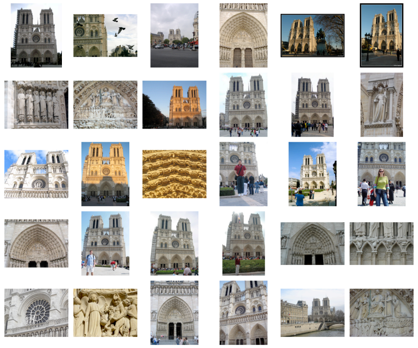
Building Rome in a Day, Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz and Richard Szeliski,
International Conference on Computer Vision, 2009
http://grail.cs.washington.edu/rome/


15,464
37,383
76,389
Slide: N. Snavely



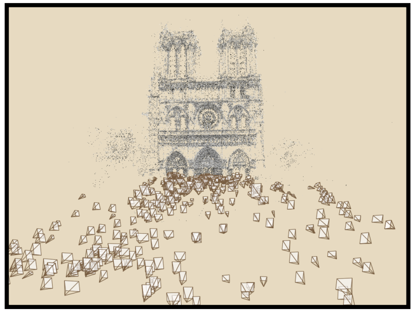
Slide: N. Snavely

Photo Tourism overview



Input photographs

Scene reconstruction



Relative camera positions and orientations

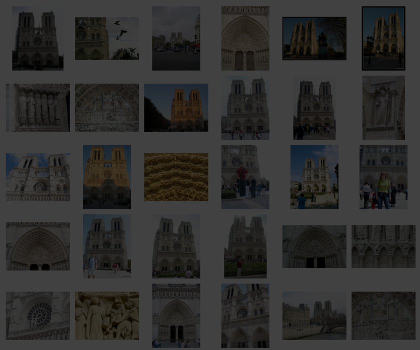
Point cloud

Sparse correspondence

Photo Explorer

Slide: N. Snavely

Photo Tourism overview



Input photographs

Scene reconstruction

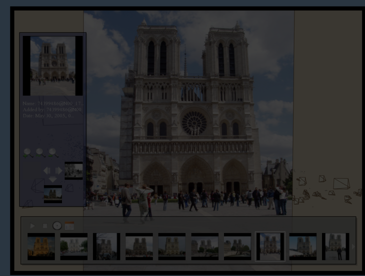


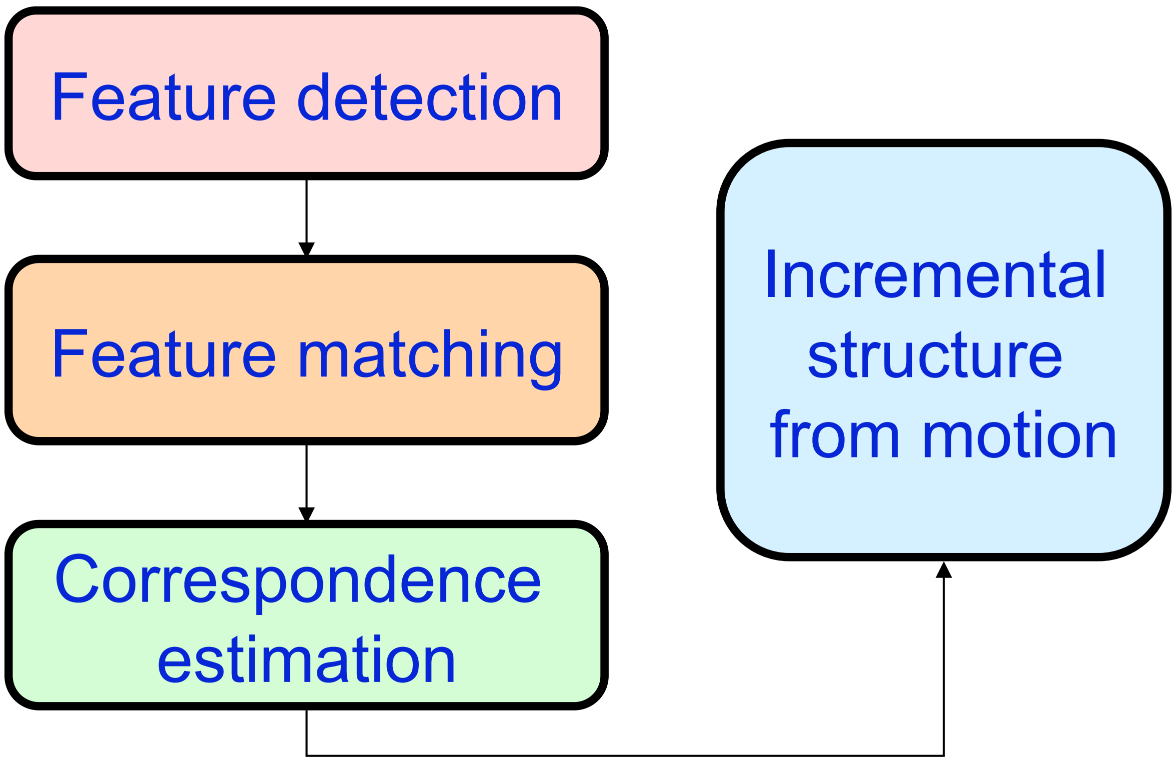
Photo Explorer

Slide: N. Snavely

Scene reconstruction

Automatically estimate

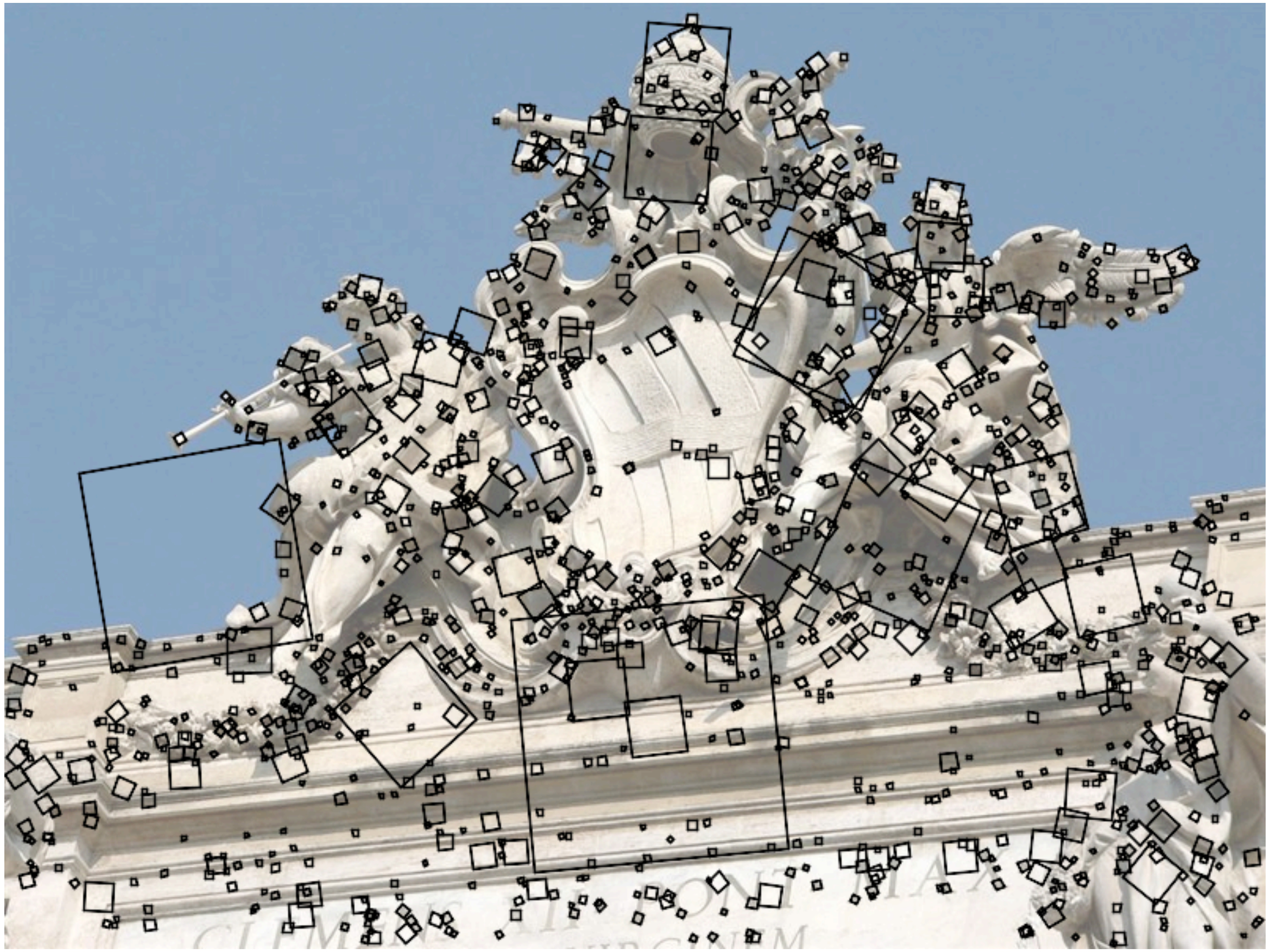
- position, orientation, and focal length of cameras
- 3D positions of feature points



Slide: N. Snavely

Feature detection

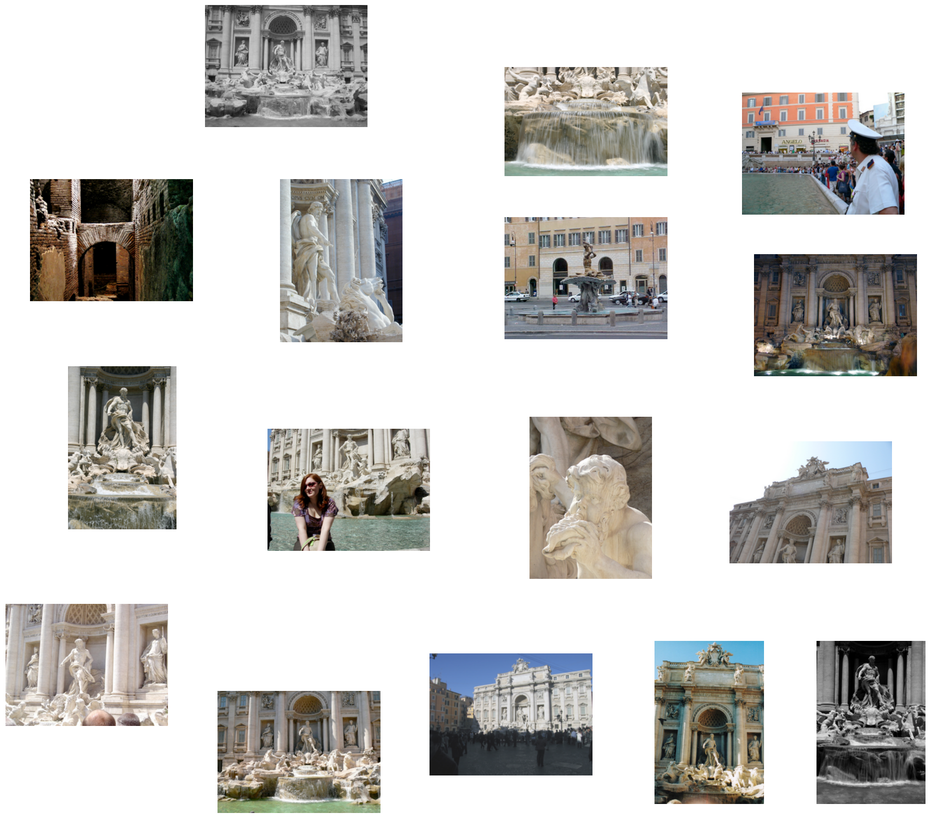
Detect features using SIFT [Lowe, IJCV 2004]



Slide: N. Snavely

Feature detection

Detect features using SIFT [Lowe, IJCV 2004]



Slide: N. Snavely

Feature detection

Detect features using SIFT [Lowe, IJCV 2004]

Slide: N. Snavely

Feature matching

Complexity of matching:

Unfortunately, even with a well optimized implementation of the matching procedure described above, it is not practical to match all pairs of images in our corpus. For a corpus of 100,000 images, this translates into 5,000,000,000 pairwise comparisons, which with 500 cores operating at 10 image pairs per second per core would require about 11.5 days to match. Furthermore, this does not even take into account the network transfers required for all cores to have access to all the SIFT feature data for all images.

From Agarwal et al. “Building Rome in a Day”, ICCV’09

Feature matching

Obtain candidate pairs of images to match using
visual vocabulary matching based on k-means tree

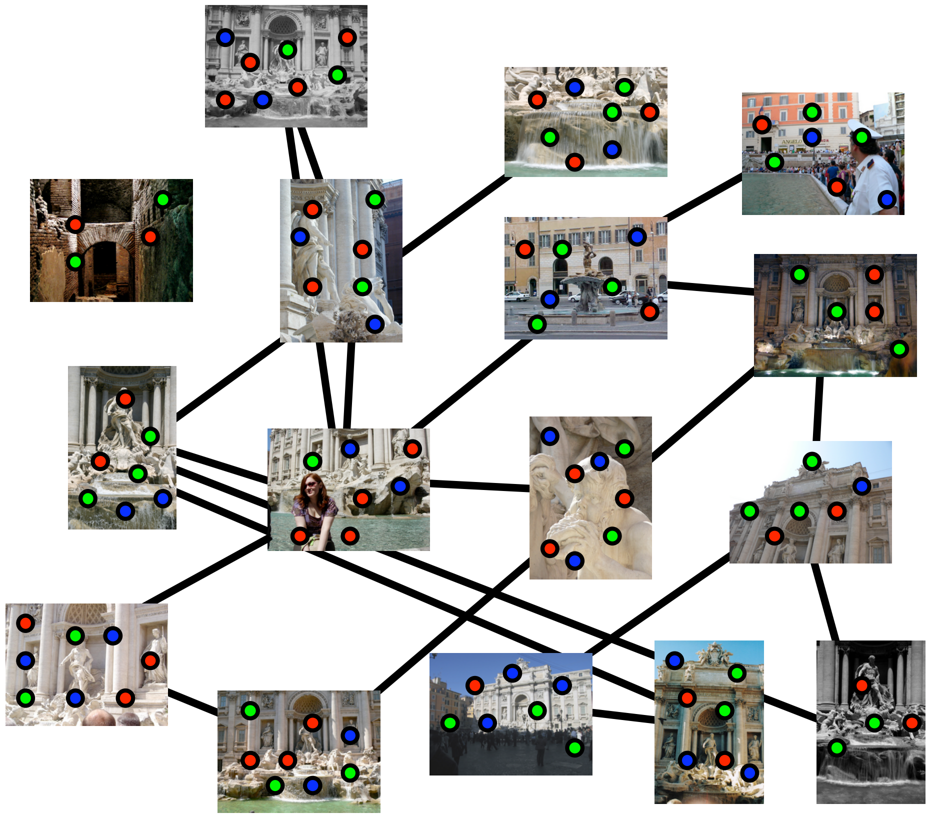


Figure: N. Snavely

Feature matching

Match features between candidate pairs using K-d trees built on SIFT descriptors.

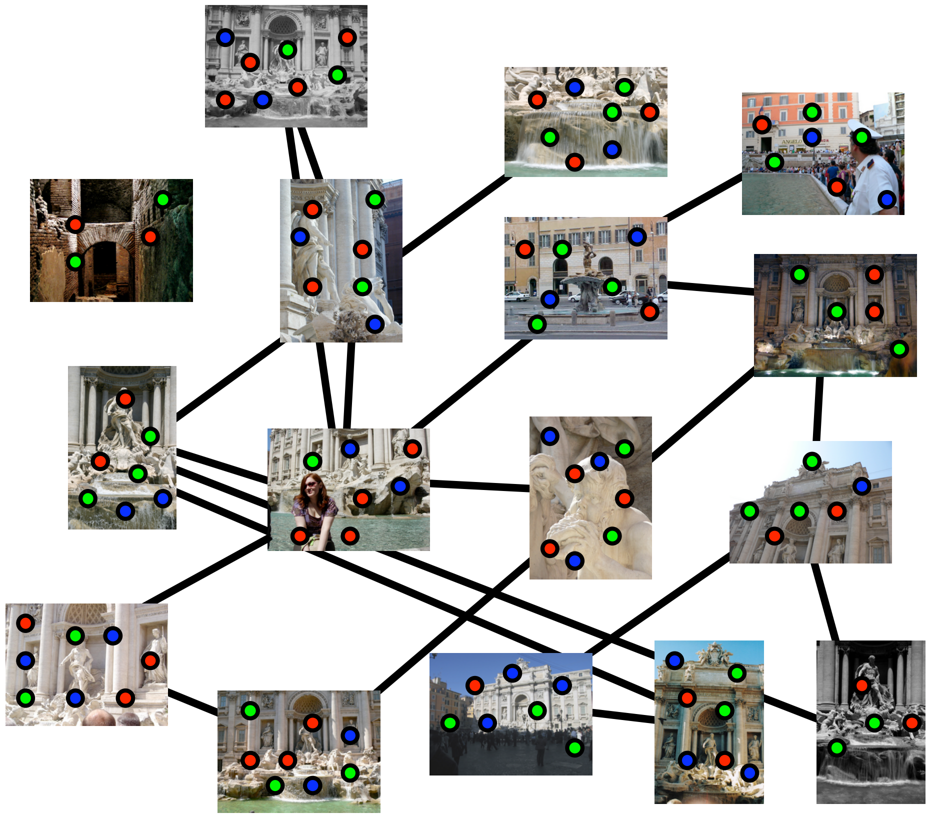
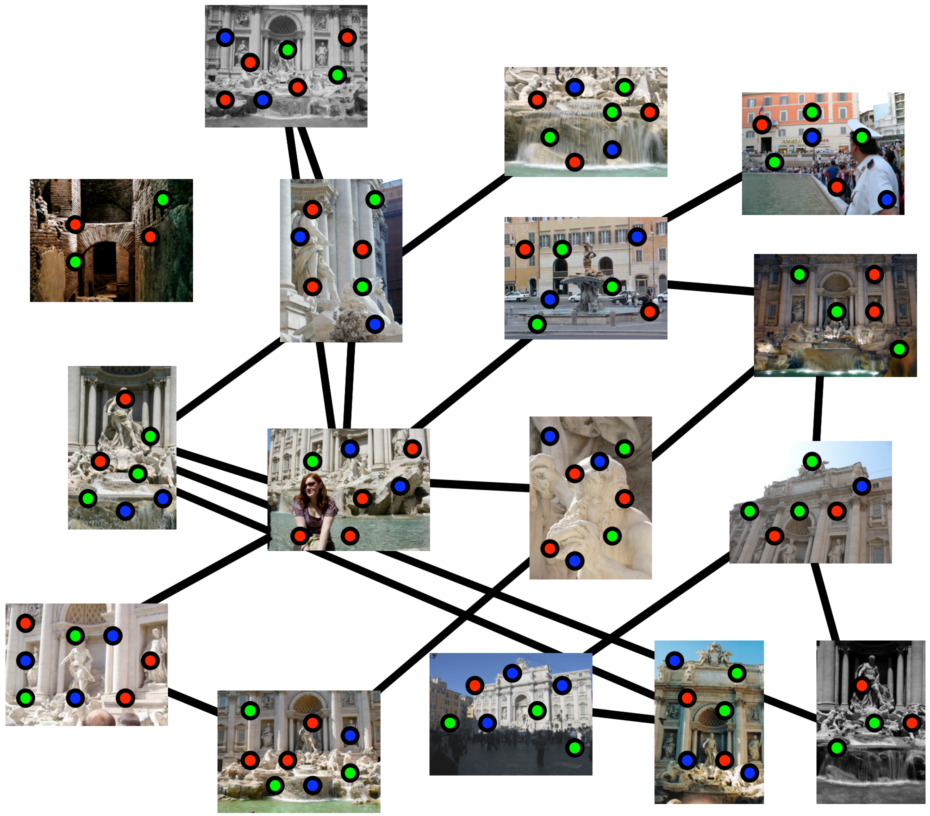


Figure: N. Snavely

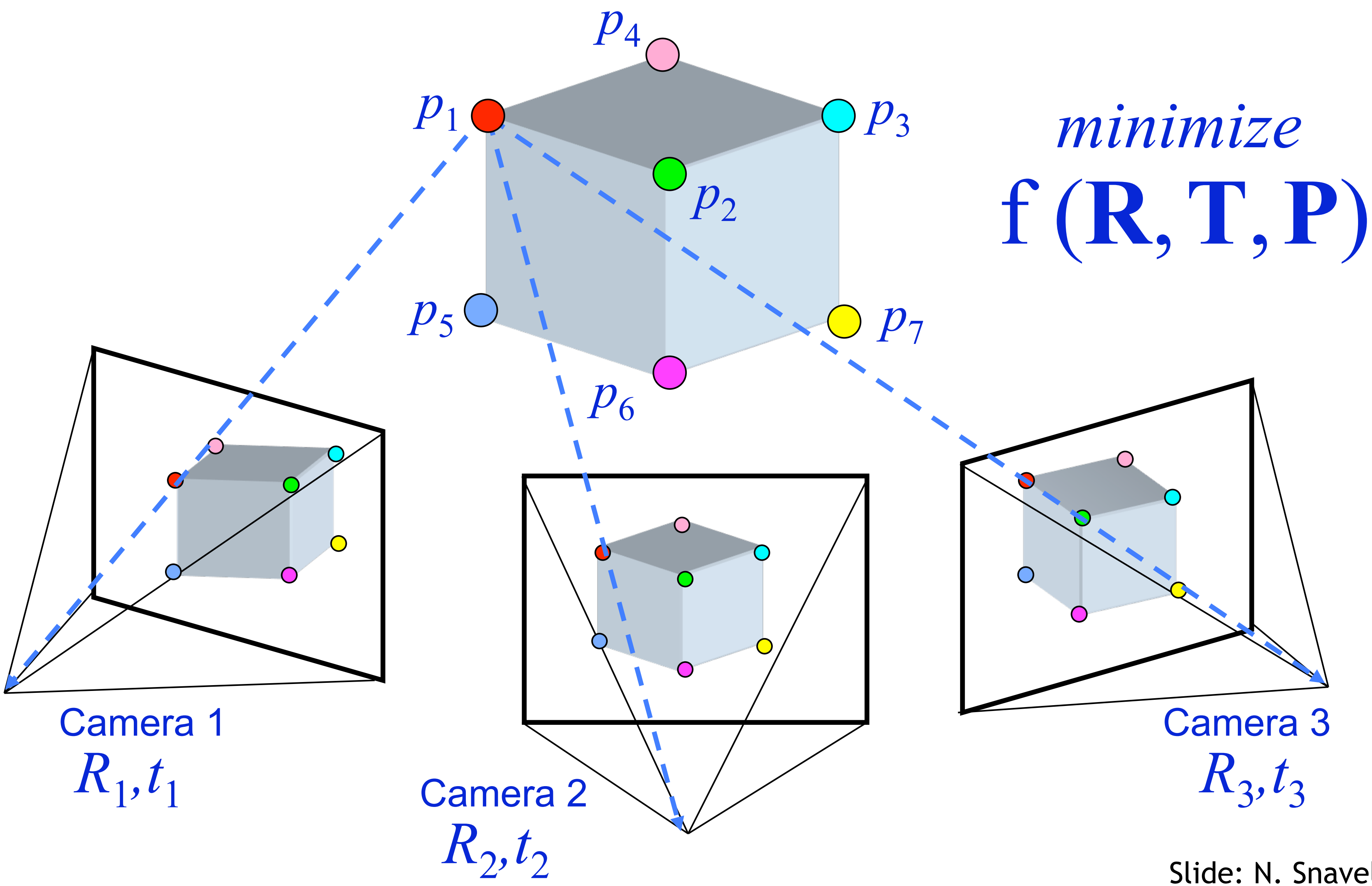
Feature matching

Refine matching using RANSAC [Fischler & Bolles 1987] to estimate fundamental matrices between pairs



Slide: N. Snavely

Structure from motion (R. Keriven's class)



Slide: N. Snavely

Example of the final 3D point cloud and cameras

57,845 downloaded images, 11,868 registered images. This video: 4,619 images.