Visual Object Retrieval

Andrew Zisserman, Ondrej Chum, Michael Isard
James Philbin, Josef Sivic

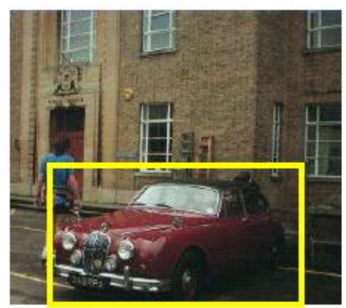
Visual Geometry Group
Dept of Engineering Science
University of Oxford

Presented by Jizhou Gao

Introduction

- Query by visual example:

near duplicate



same object

same category

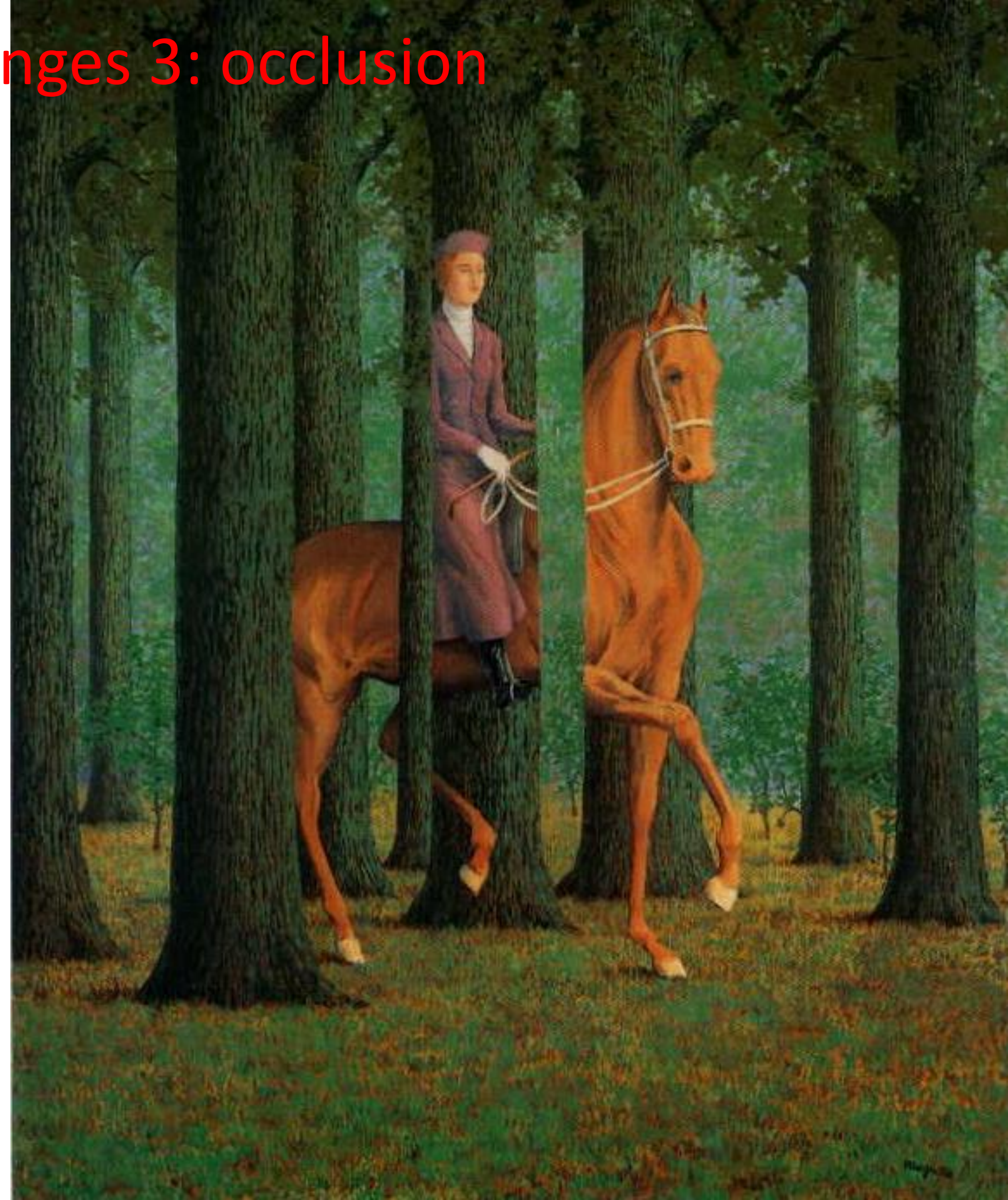
Challenges 1: view point



Michelangelo 1475-1564

Challenges 2: illumination

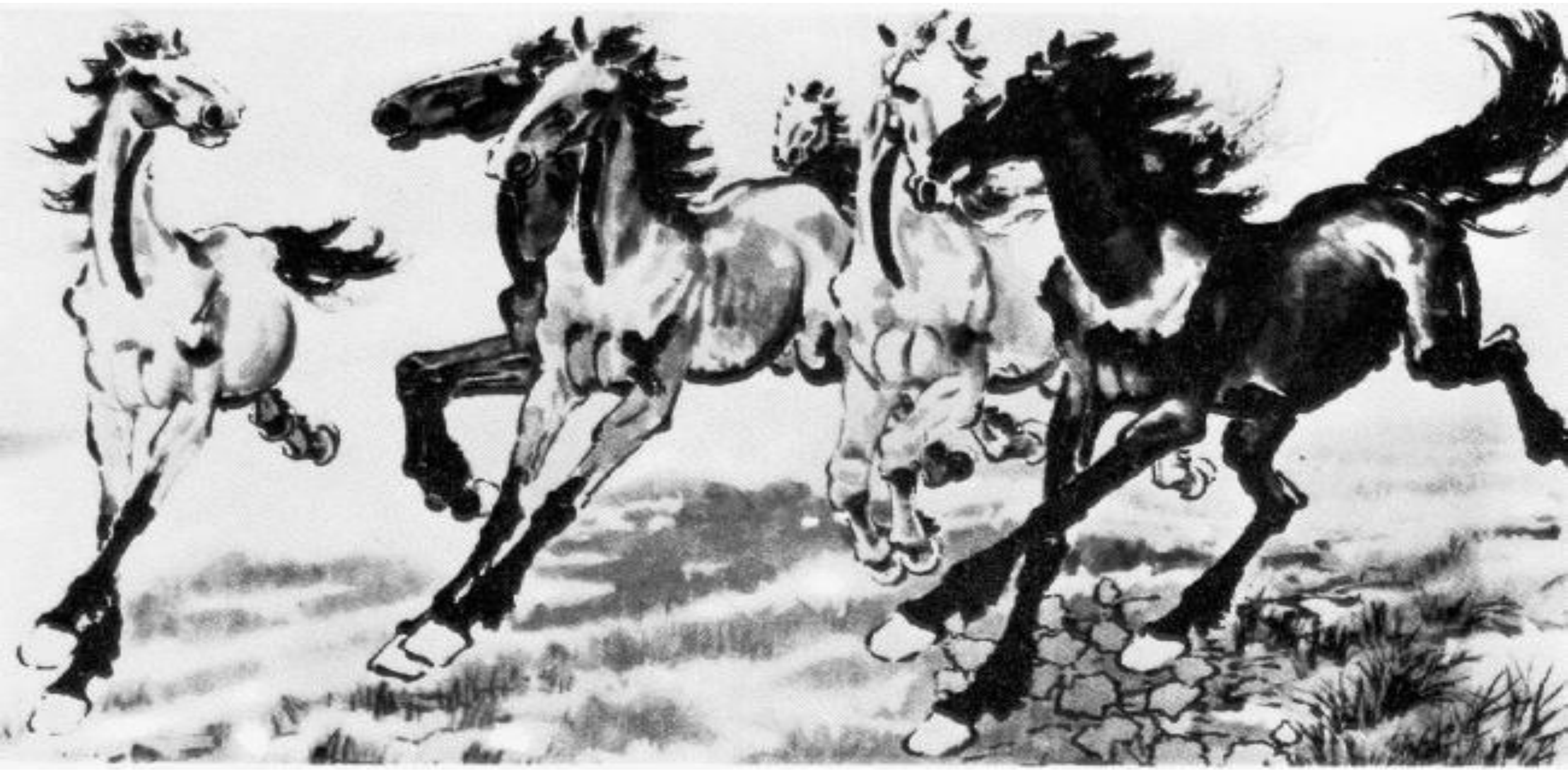
Challenges 3: occlusion



Magritte, 1957

Challenges 4: scale

Challenges 5: deformation

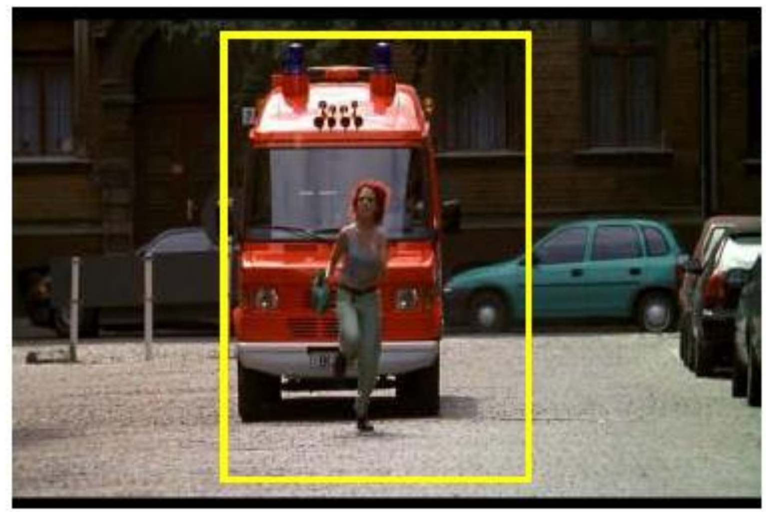


Xu, Beihong 1943

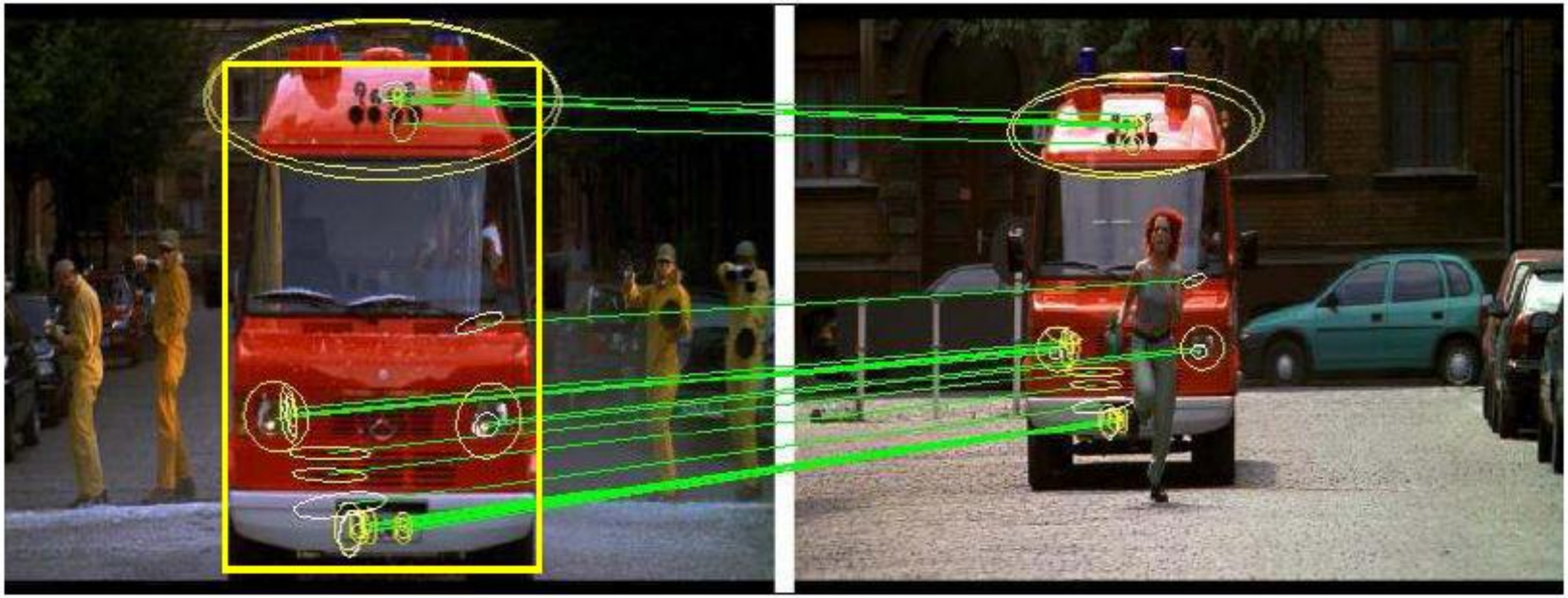
Challenges 6: background clutter

Klimt, 1913

Particular objects, not entire images



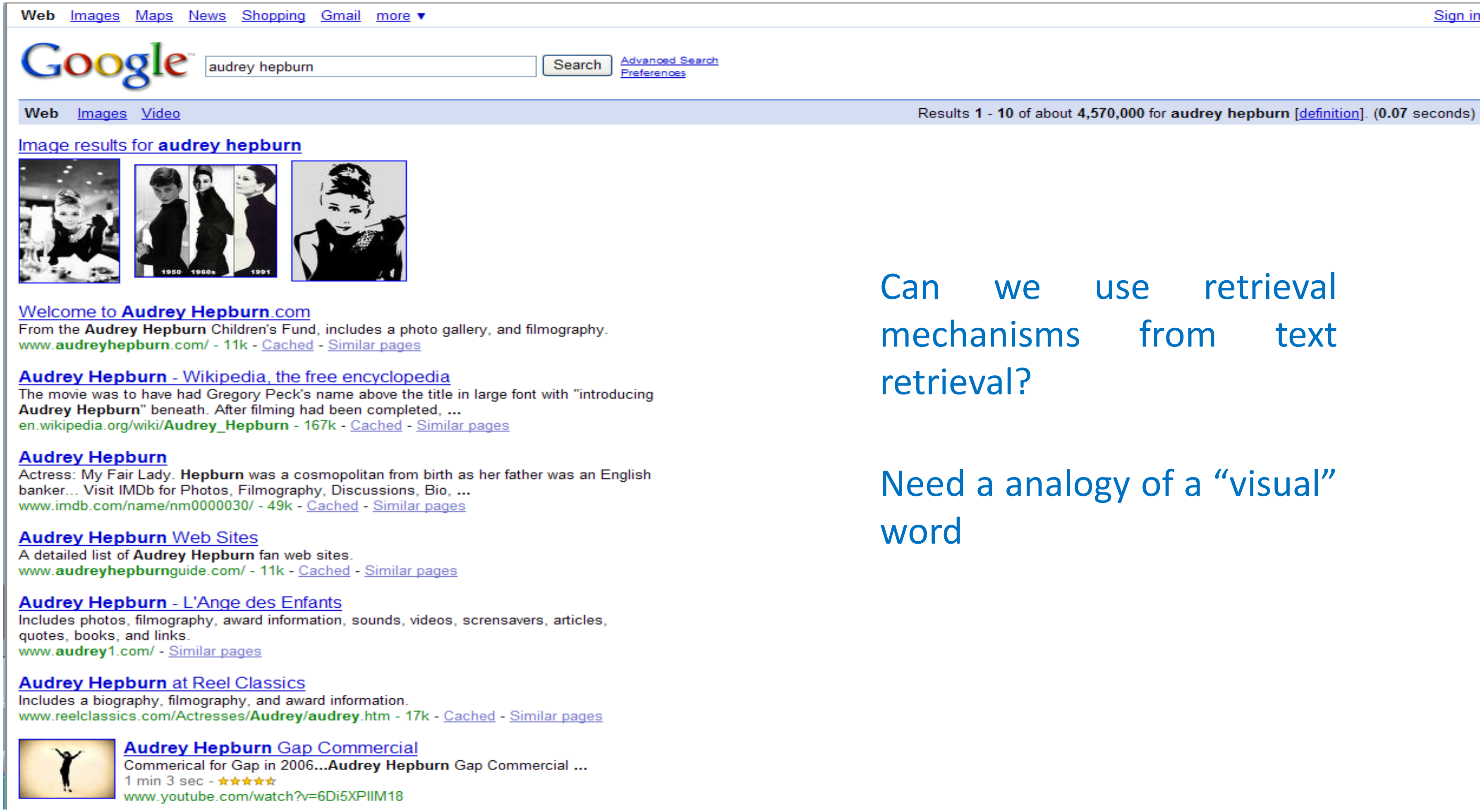
When do objects match?



Two requirements:

1. “patches” (parts) correspond, and
2. Configuration (spatial layout) corresponds

Success of text retrieval



Can we use retrieval mechanisms from text retrieval?

Need a analogy of a “visual” word

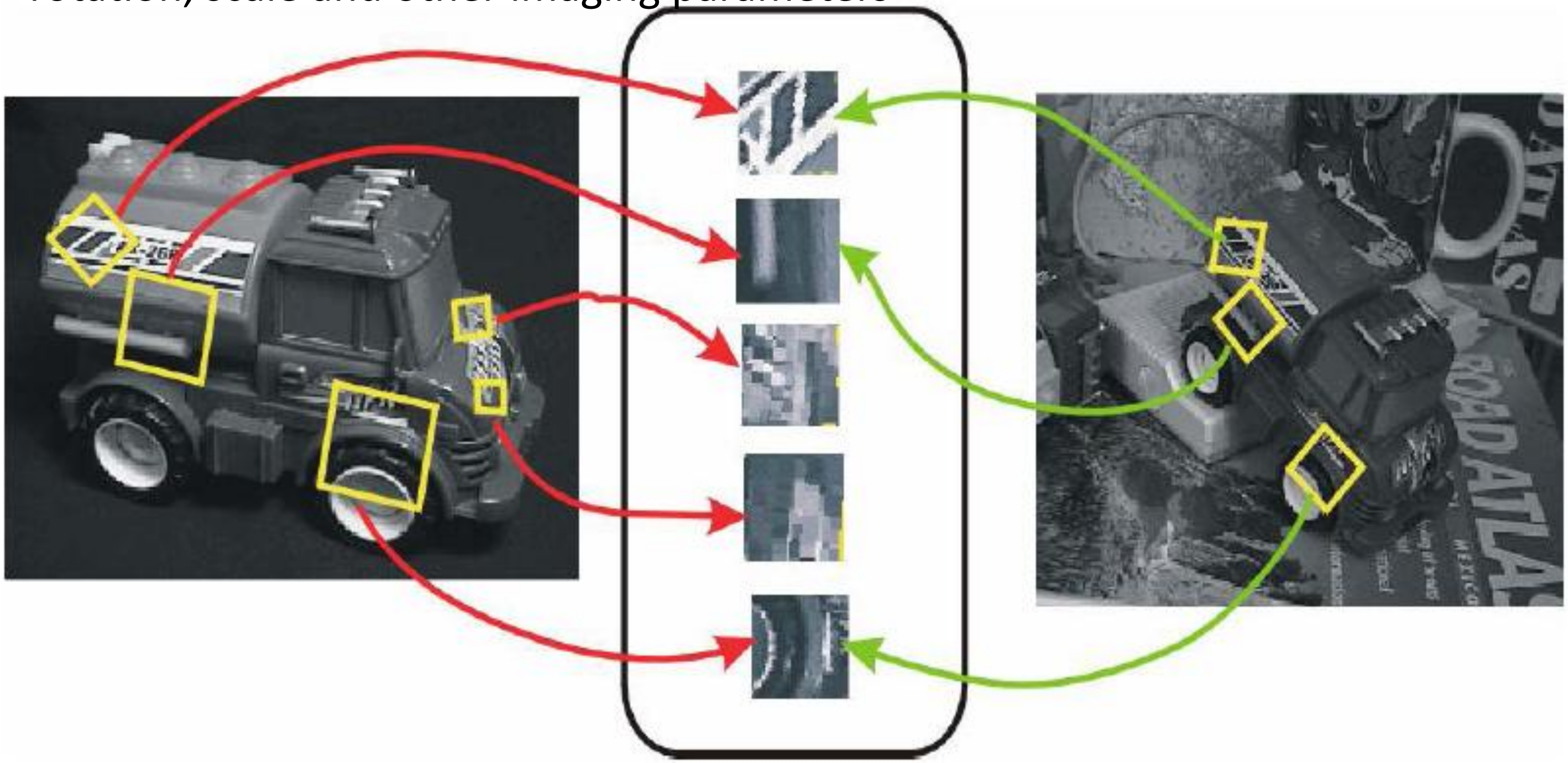
Feature detector & descriptor

- Determine regions and vector descriptors in each image/frame which are invariant to camera viewpoint changes

- Match descriptors between frames using invariant vectors

Example of visual fragment (feature)

- Image content is transformed into local fragments that are invariant to translation, rotation, scale and other imaging parameters

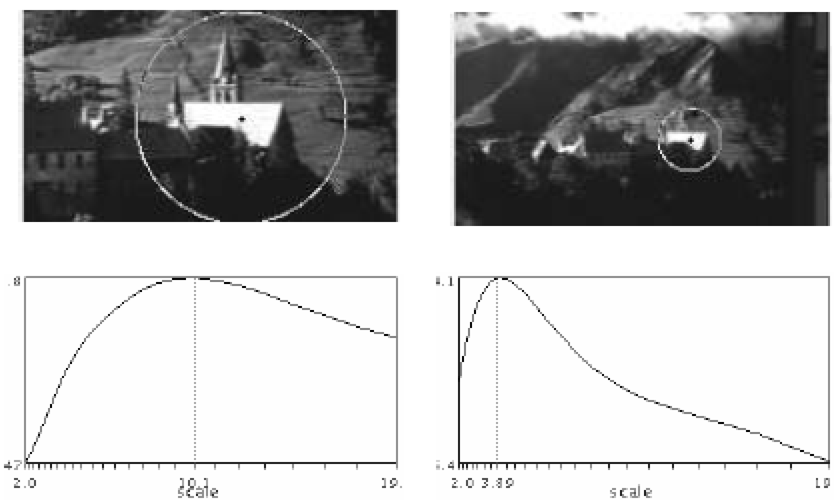


Lowe ICCV 1999

Scale invariance

- Multi-scale extraction of Harris interest points
- Selection of points at characteristic scale in scale space

Laplacian



Chacteristic scale :

- maximum in scale space
- scale invariant

Mikolajczyk and Schmid ICCV 2001

Viewpoint covariant region detectors

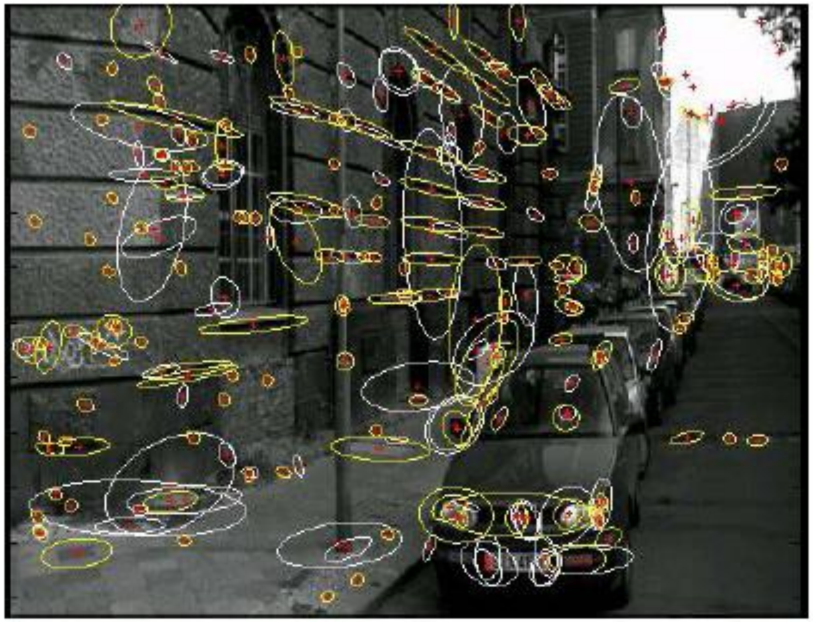
- Characteristic scales (size of region)
  - Lindeberg and Garding ECCV 1994
  - Lowe ICCV 1999
  - Mikolajczyk and Schmid ICCV 2001

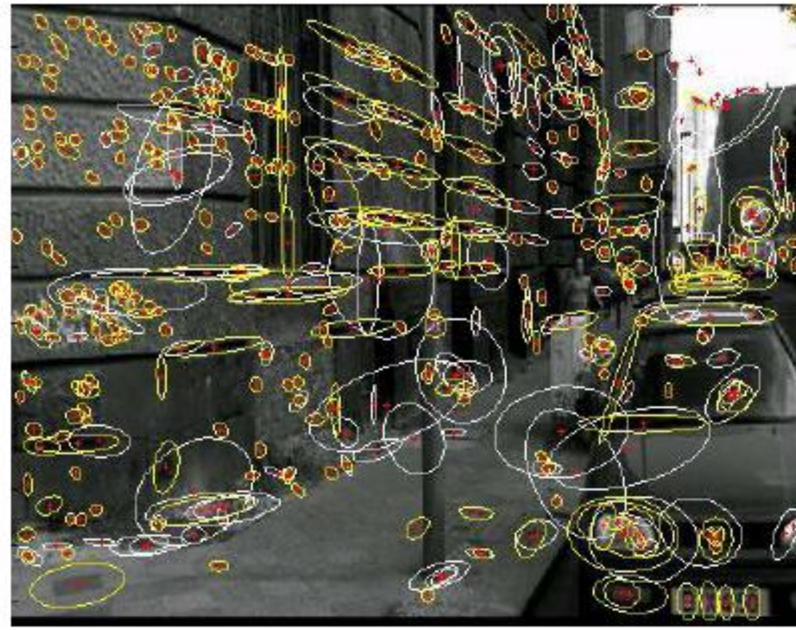
- Affine covarance (shape of region)
  - Baumberg CVPR 2000
  - Matas et al BMVC 2002
  - Mikolajczyk and Schmid ECCV 2002
  - Schaffalitzyk and Zisserman ECCV 2002
  - Tuytelaars and Van Gool BMVC 2000

Maximally stable regions

Shape adapted regions
"Harris affine"

Example of affine covariant regions





1000+ regions per image

Harris-affine

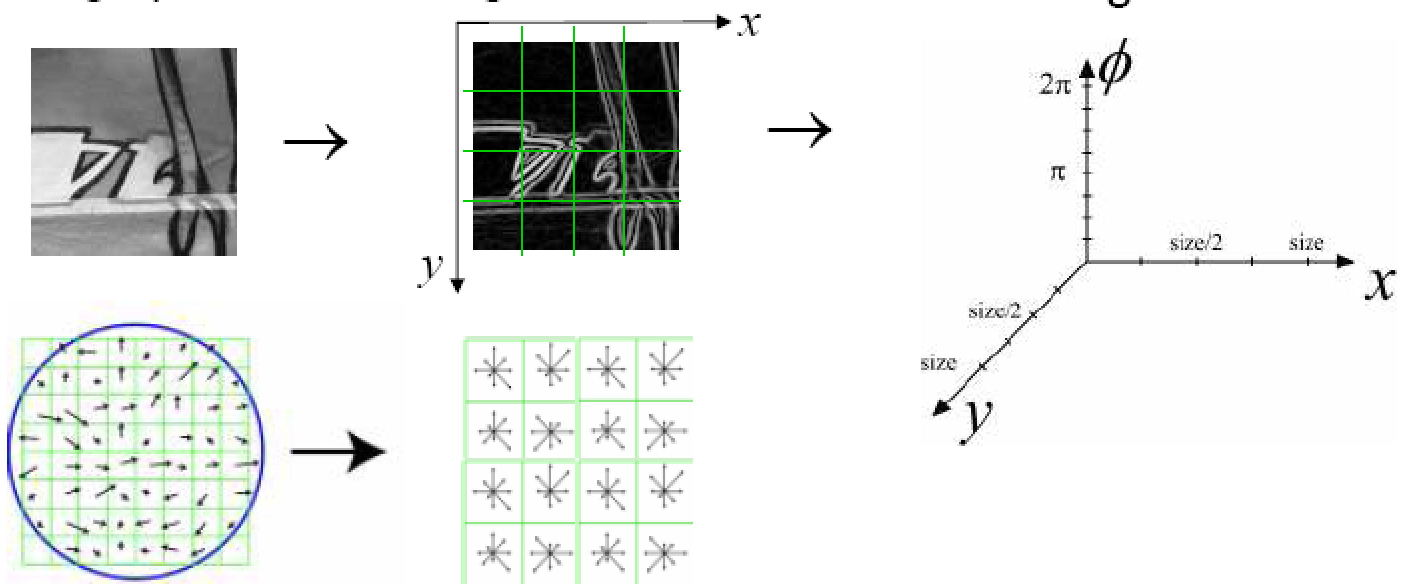
Maximally stable regions

Represent each region by SIFT descriptor (128-vector) [Lowe 1999]

Descriptors – SIFT [Lowe 1999]

distribution of the gradient over an image patch

image patch | gradient | 3D histogram

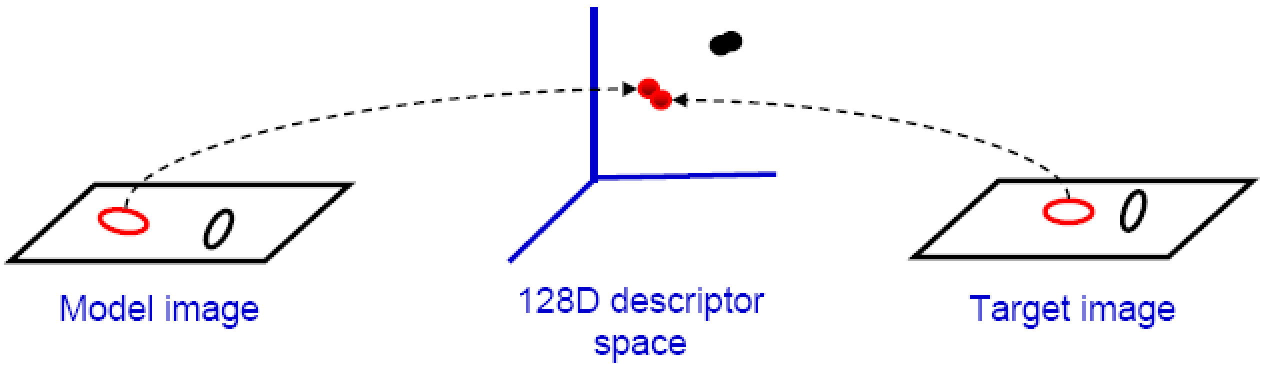


4x4 location grid and 8 orientations (128 dimensions)

very good performance in image matching [Mikolaczyk and Schmid’03]

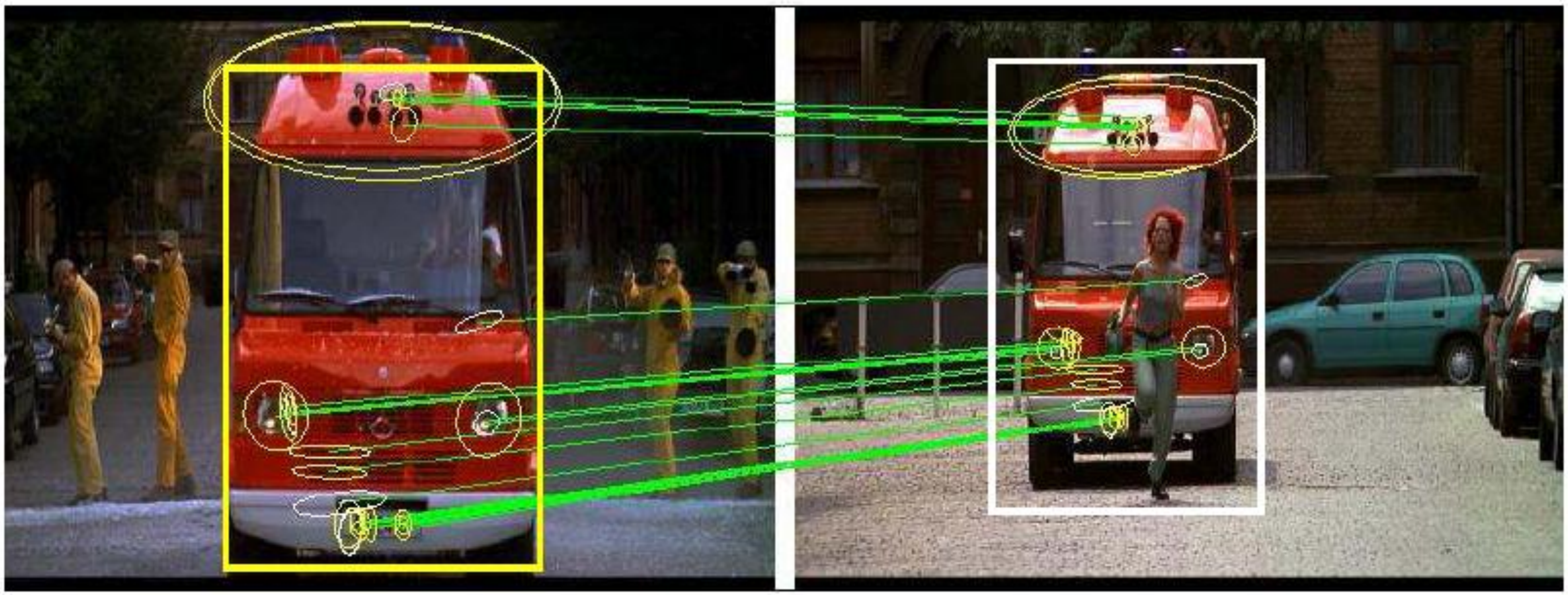
SIFT in object recognition

Establish correspondences between object model image and target image by nearest neighbour matching on SIFT vectors

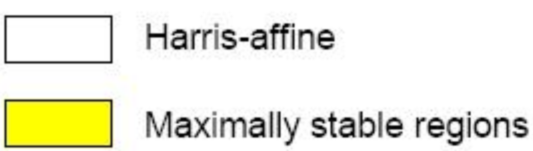


Euclidean Distance
or
Angle between 2 sift vectors

Match regions between frames using SIFT descriptors



- Multiple fragments overcomes problem of partial occlusion
- Transfer query box to localize object

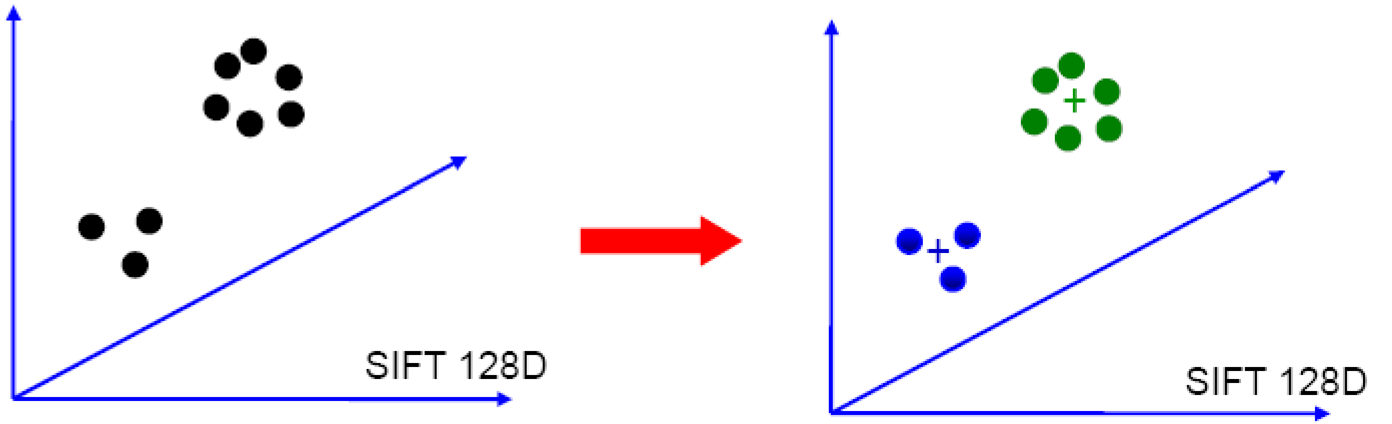


Now, convert this approach to a text retrieval representation

Build a visual vocabulary for a movie

Vector quantize descriptors

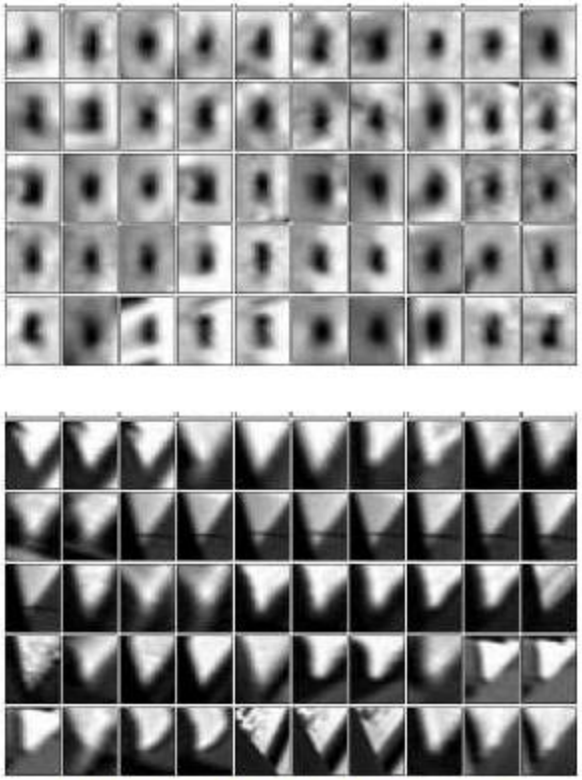
- k-means clustering



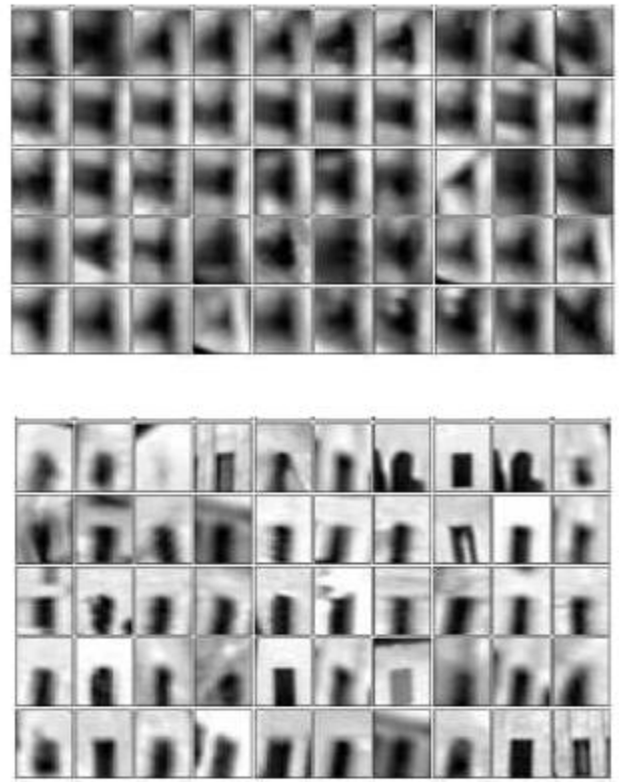
Implementation

- compute SIFT features on frames from 48 shots of the film
- 6K clusters for Shape Adapted regions
- 10K clusters for Maximally Stable regions

Samples of visual words (clusters on SIFT descriptors):

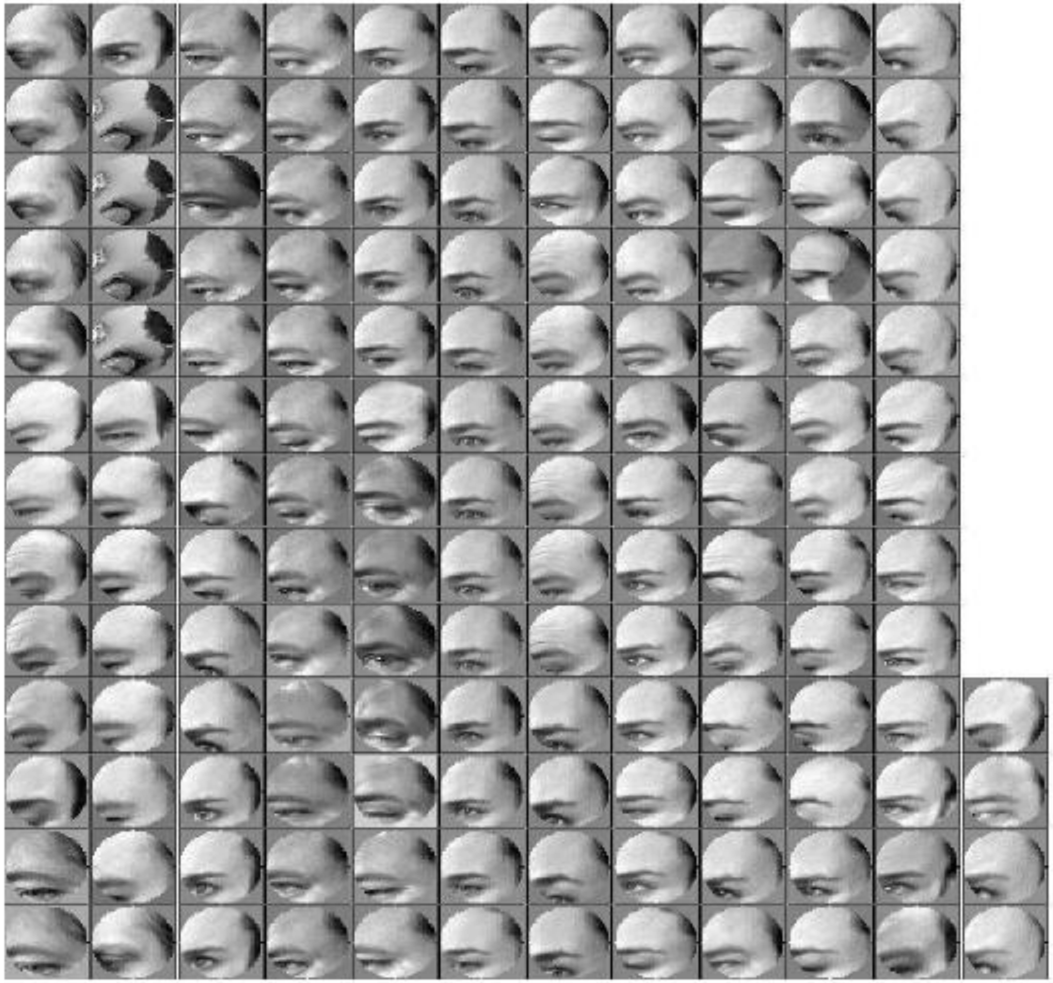


Shape adapted regions

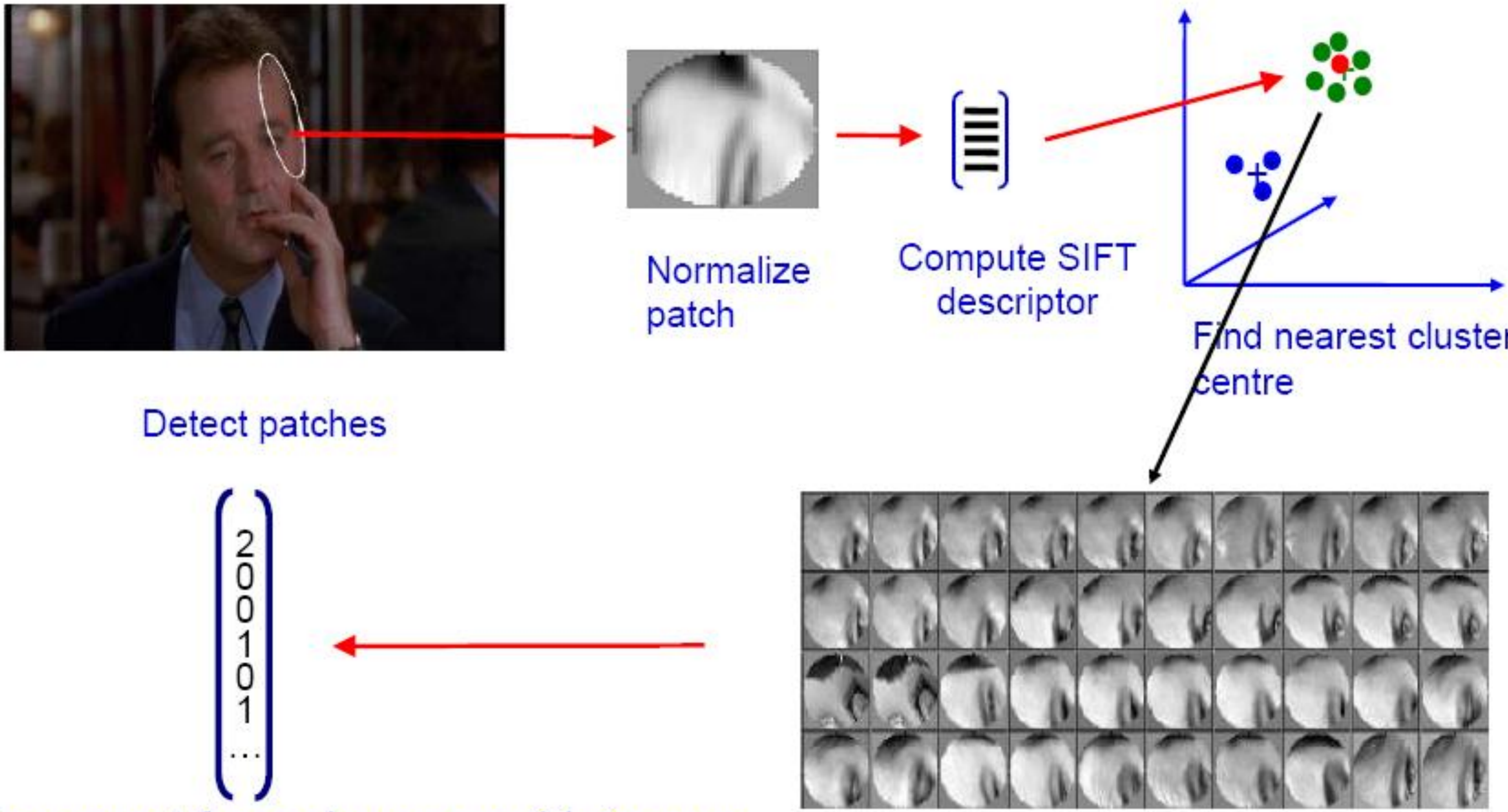


Maximally stable regions

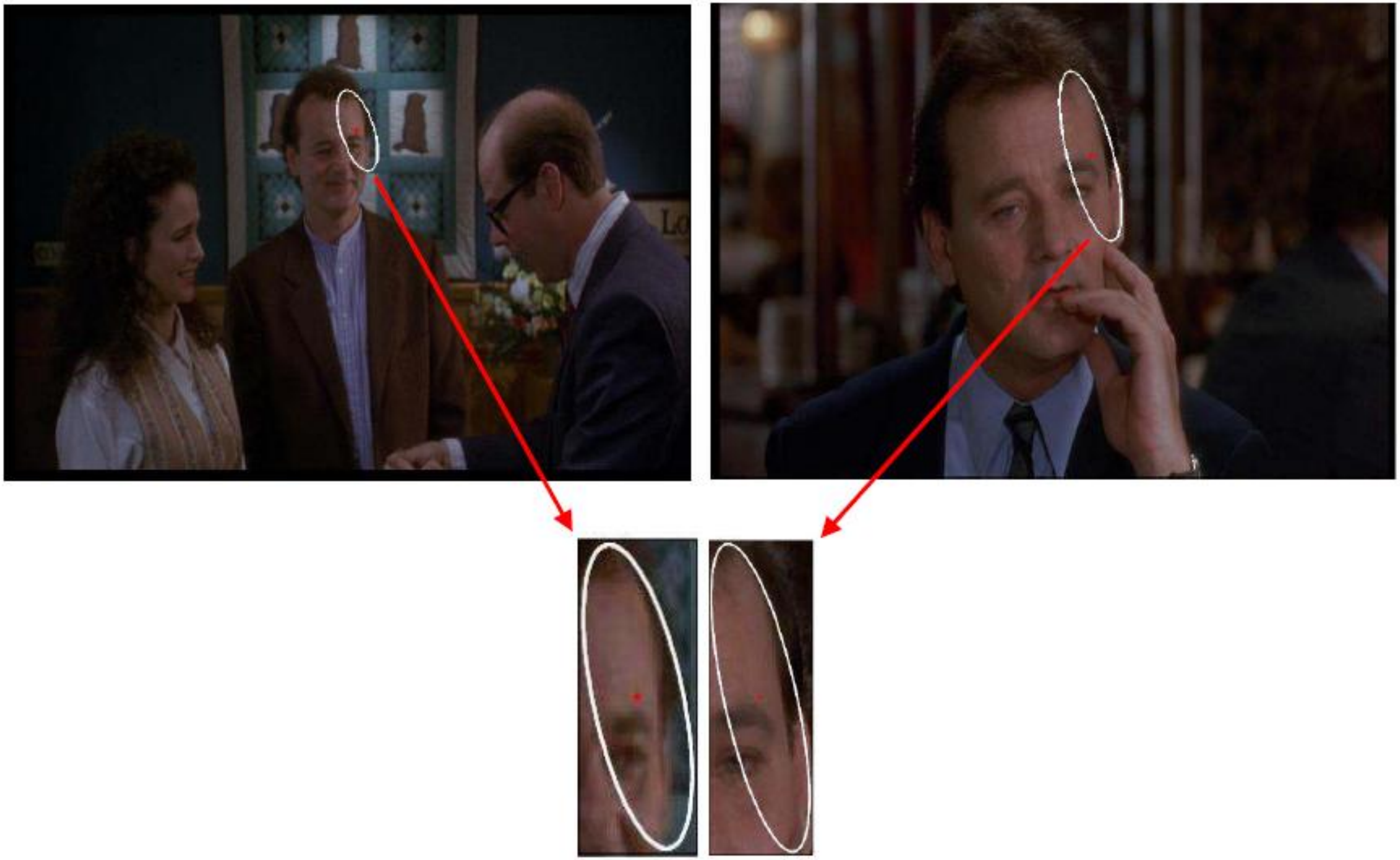
Samples of visual words (clusters on SIFT descriptors):



Assign visual words and compute histograms for each key frame in the video



Represent frame by sparse histogram of visual word occurrences

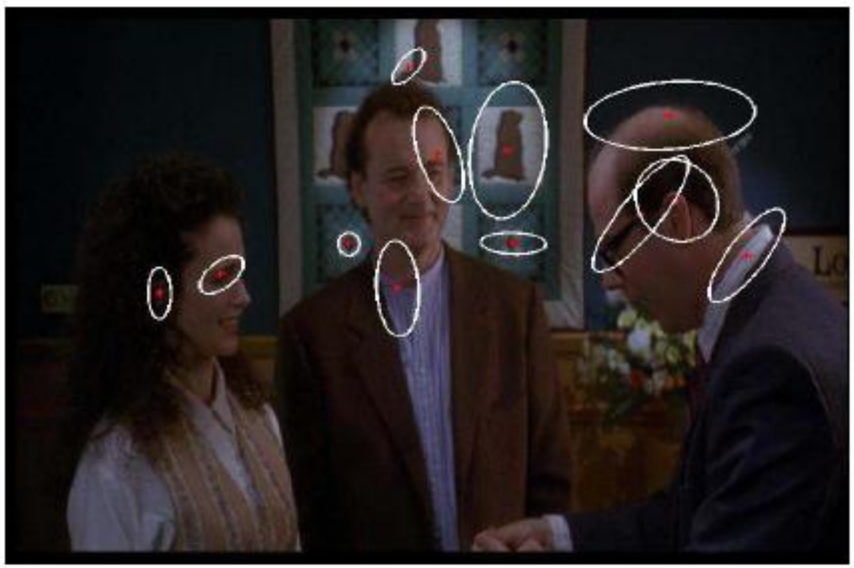


The same visual word

Representation: bag of (visual) words

Visual words are 'iconic' image patches or fragments

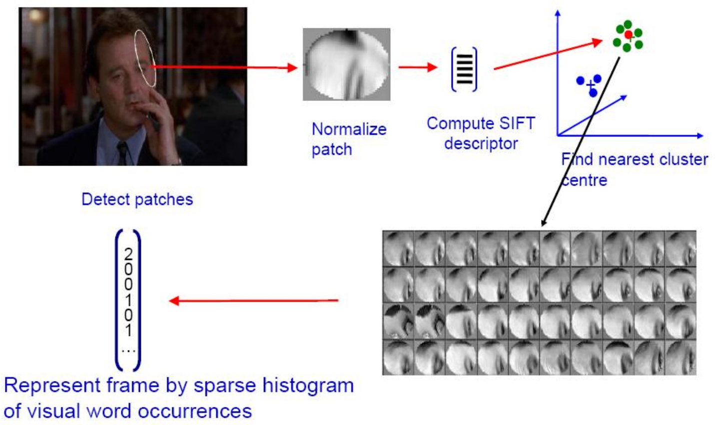
- represent the frequency of word occurrence
- but not their position



Image

Collection of visual words

tf-idf



- Recall from previous slide:

$$V_d = \begin{bmatrix} t_1 \\ \vdots \\ t_i \\ \vdots \\ t_k \end{bmatrix}$$

#word i in doc d #doc's

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

#words in doc d #word i in all doc's

- Word frequency weights words occurring often in a particular document, and thus describe it well; while the inverse document frequency down-weights words that appear often in the database.

Search

- For fast search, store a “posting list” for the dataset
- This maps word occurrences to the documents they occur in

frame #5

frame #10

Posting list

1 → 5,10, ...
2 → 10,...
... ...

inverted file in text retrieval

Matching a query region

Stage 1: generate a short list of possible frames using bag of visual word representation:

1. Accumulate all visual words within the query region
2. Use “book index” to find other frames with these words
3. Compute similarity for frames which share at least one word

frame #5

frame #10

Posting list

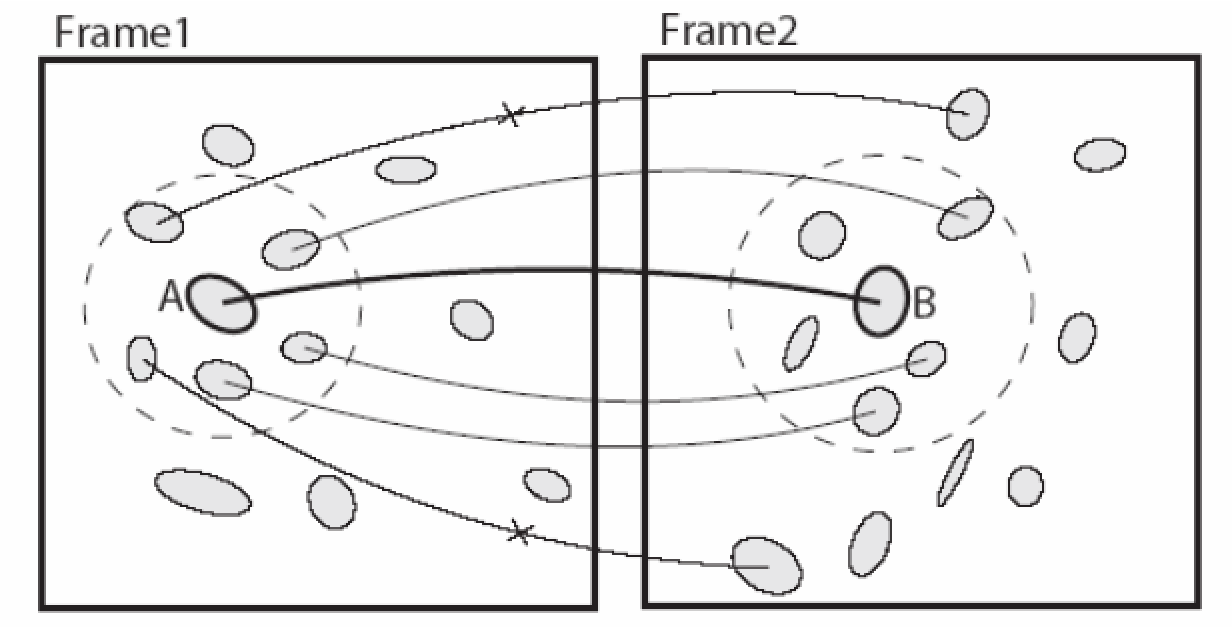
1 → 5,10, ...

2 → 10,...

... ...

- Generates a tf-idf ranked list of all the frames in dataset

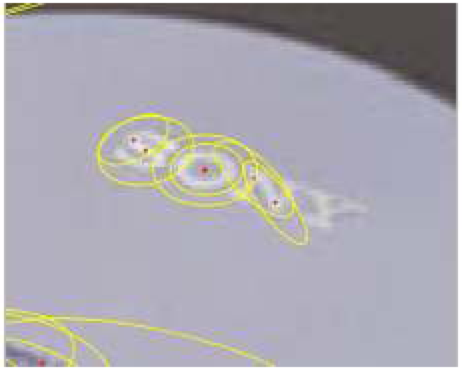
Stage 2: re-rank short list on spatial consistency



- Discard mismatches
  - require spatial agreement with the neighbouring matches
- Compute matching score
  - score each match with the number of agreement matches
  - accumulate the score from all matches
- Also matches define correspondence between target and query region

Example application I – product placement

Sony logo from Google image search on `Sony'



Retrieve shots from Groundhog Day

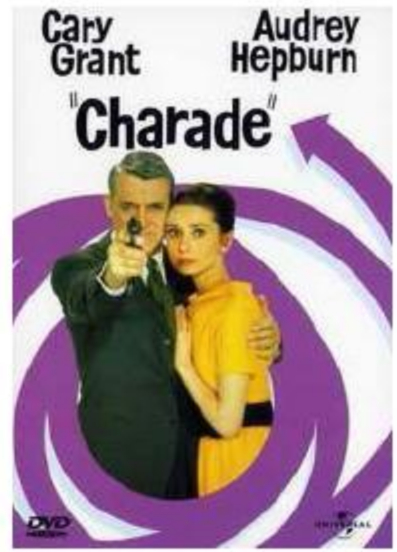
Retrieved shots in Groundhog Day for search on Sony logo

Example II - finding photos in a personal collection

Notre Dame from Google image search on `Notre Dame'

Query image

Charade (6,503 keyframes)



Retrieve shots from Charade

First (correctly) retrieved shot

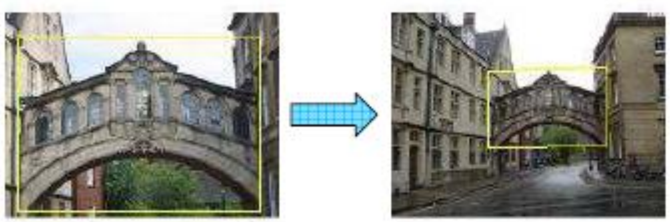
Particular object search

Find these landmarks ...in these images

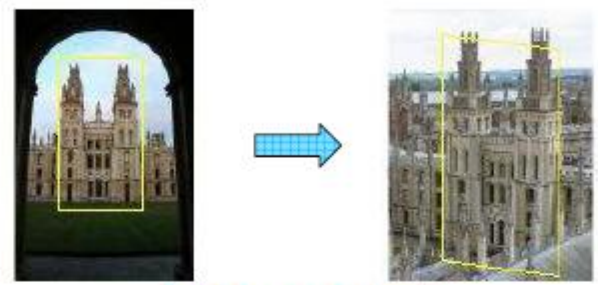
Particular Object Search

- Problem: find particular occurrences of an object in a very large dataset of images

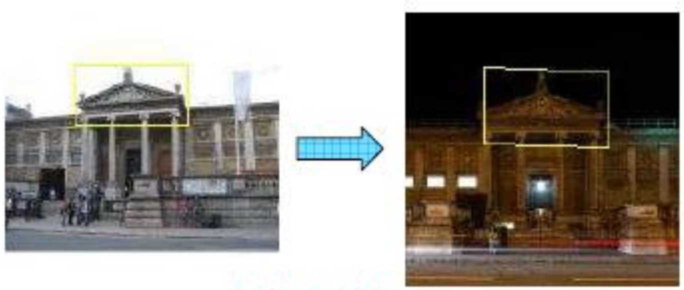
- Want to find the object despite possibly large changes in scale, viewpoint, lighting and partial occlusion



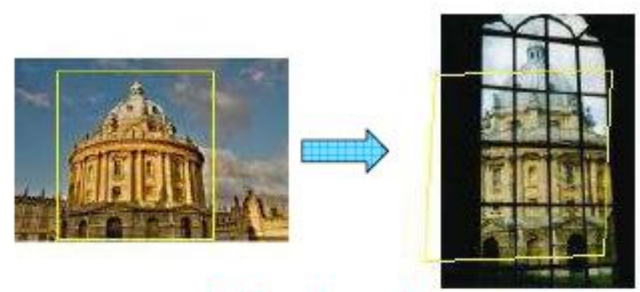
Scale



Viewpoint



Lighting



Occlusion

Representation & Similarity

- Text retrieval approach to visual search ("Video Google")

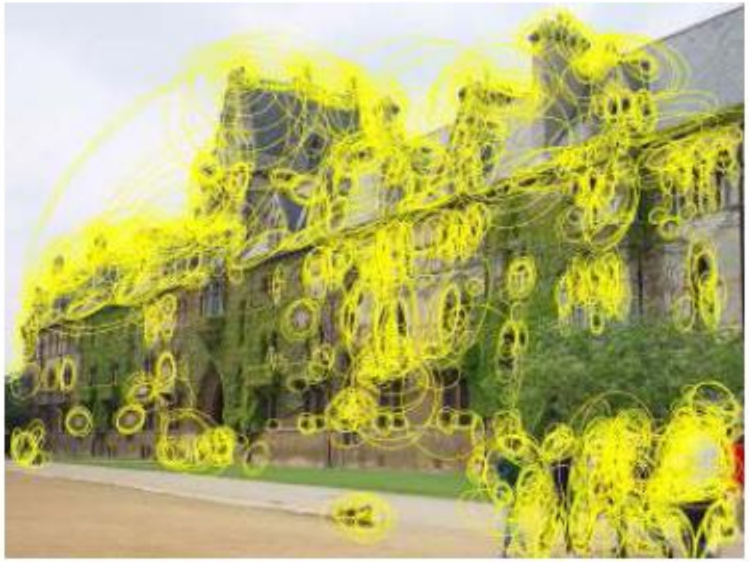
- Representation is a sparse histogram for each image

- Similarity measure is L_2 distance between tf-idf weighted histograms

Investigate …

Vocabulary size: number of visual words in range 10K to 1M

Use of spatial information to re-rank



Oxford buildings dataset

- Landmarks plus queries used for evaluation

- Ground truth obtained for 11 landmarks over 5062 images

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

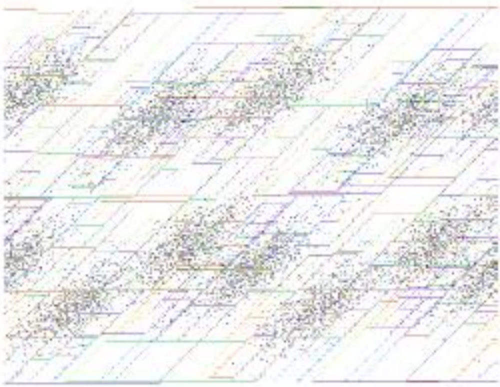
- Dataset (i) crawled by searching for Oxford landmarks
- Datasets (ii) and (iii) from other popular Flickr tags. Acts as additional distractors

Quantization / Clustering

- K-means usually seen as a quick + cheap method
- But far too slow for our needs – D~128, N~20M+, K~1M

Approximate K-means

- Use multiple, randomized k-d trees for search



- A k-d tree hierarchically decomposes the descriptor space

- Points nearby in the space can be found (hopefully) by backtracking around the tree some small number of steps

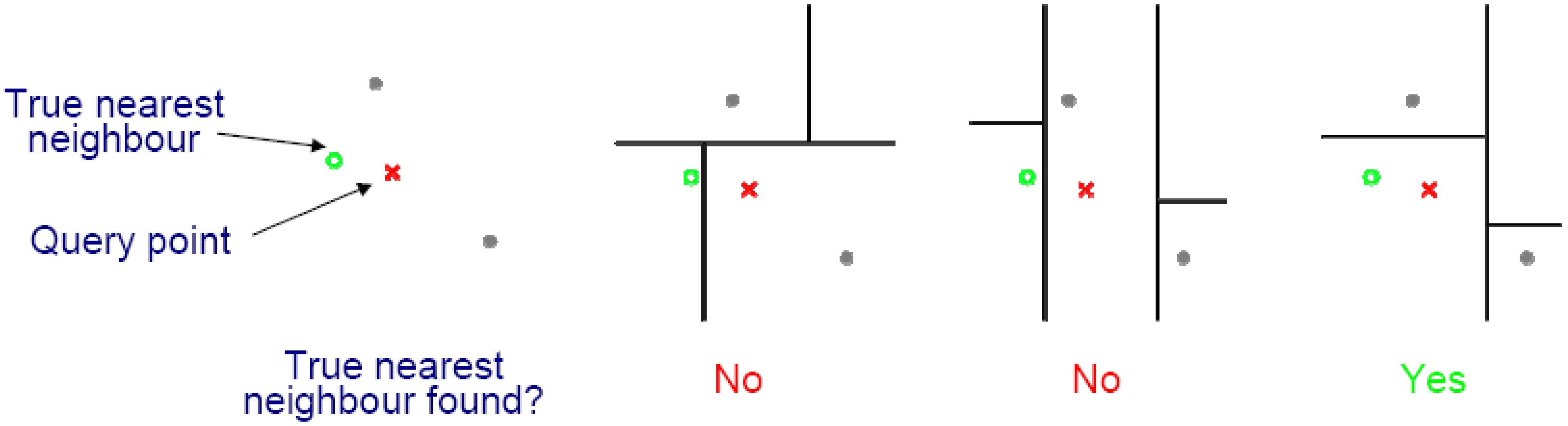
- Original K-means complexity = O(N K)

- Approximate K-means complexity = O(N log K)

- This means we can scale to very large K

Approximate K-means

- Multiple randomized trees increase the chances of finding nearby points



Approximate K-means

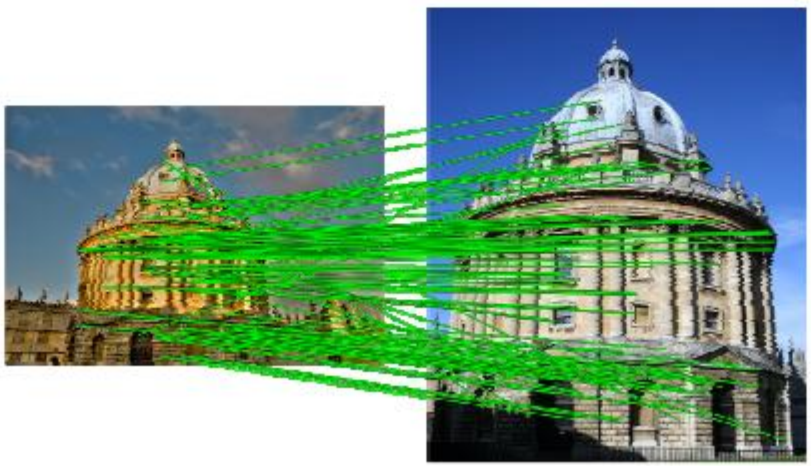
- How accurate is the approximate search?
- Performance on 5K image dataset for a random forest of 8 trees

Clustering parameters		mAP	
# of descr.	Voc. size	k-means	AKM
800K	10K	0.355	0.358
1M	20K	0.384	0.385
5M	50K	0.464	0.453
16.7M	1M		0.618

- Allows much larger clusterings than would be feasible with standard K-means: N~17M points, K~1M
  - AKM – 8.3 cpu hours per iteration
  - Standard K-means - estimated 2650 cpu hours per iteration

Beyond Bag of Words

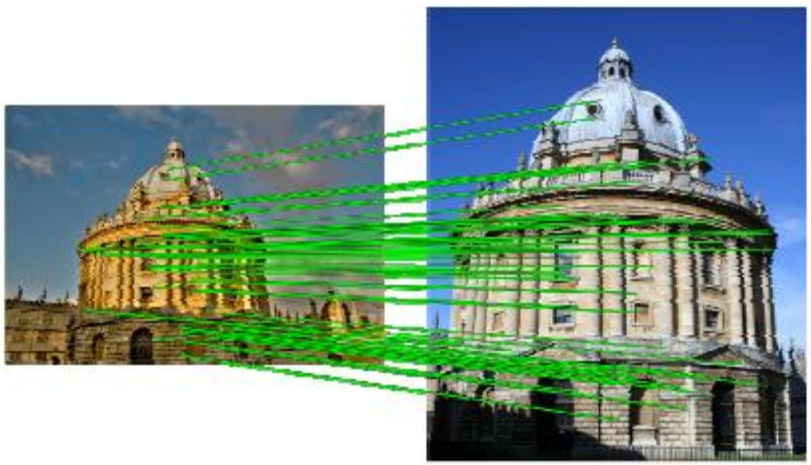
- Use the **position** and **shape** of the underlying features to improve retrieval quality



- Both images have many matches – which is correct?

Beyond Bag of Words

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

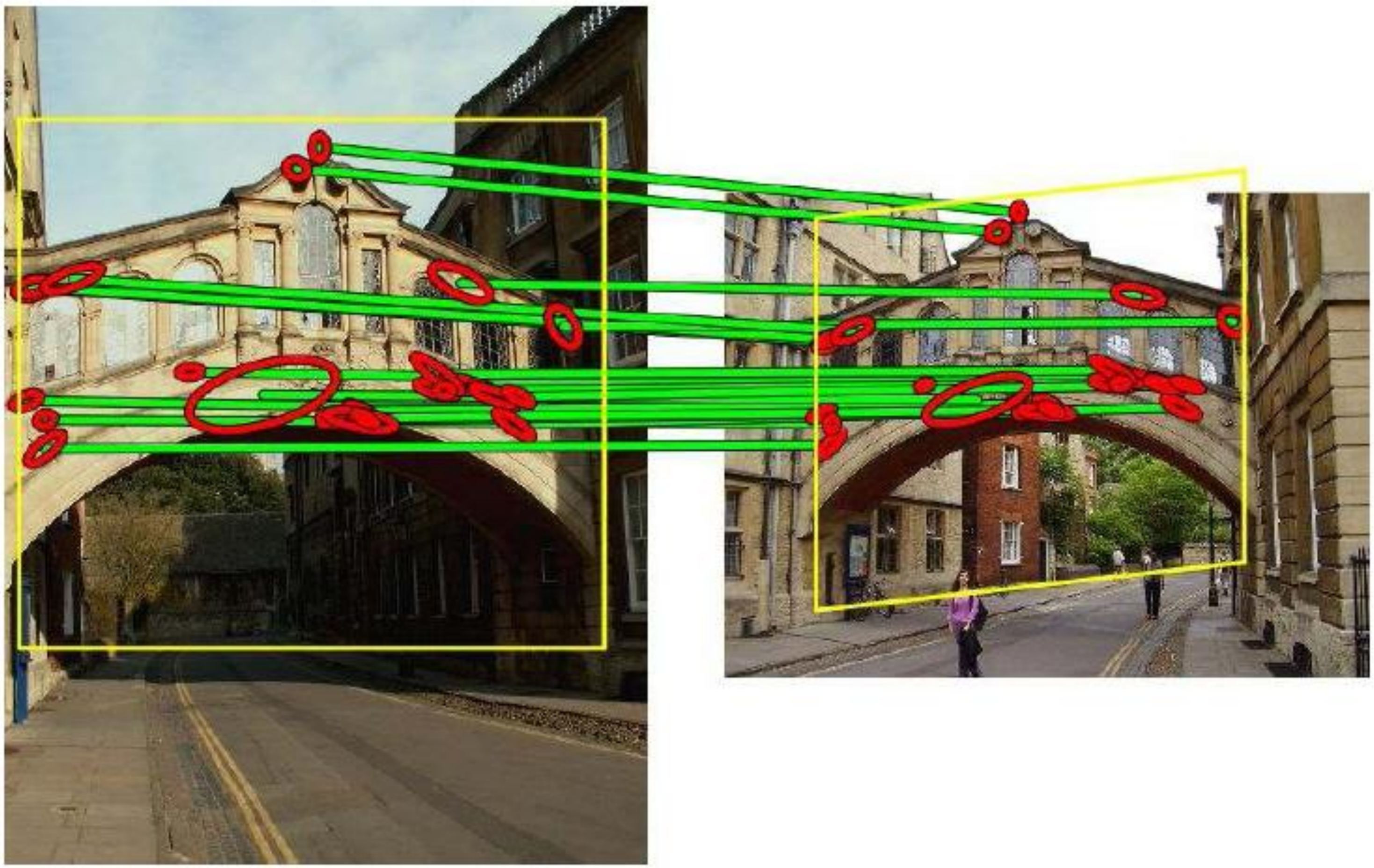


Many spatially consistent matches – **correct result**

Few spatially consistent matches – **incorrect result**

Estimating spatial correspondences

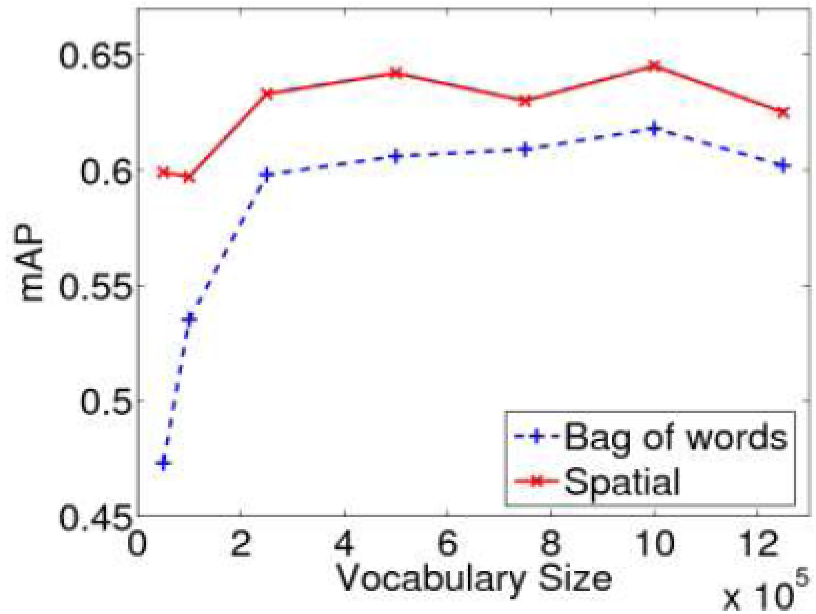
Score by number of consistent matches



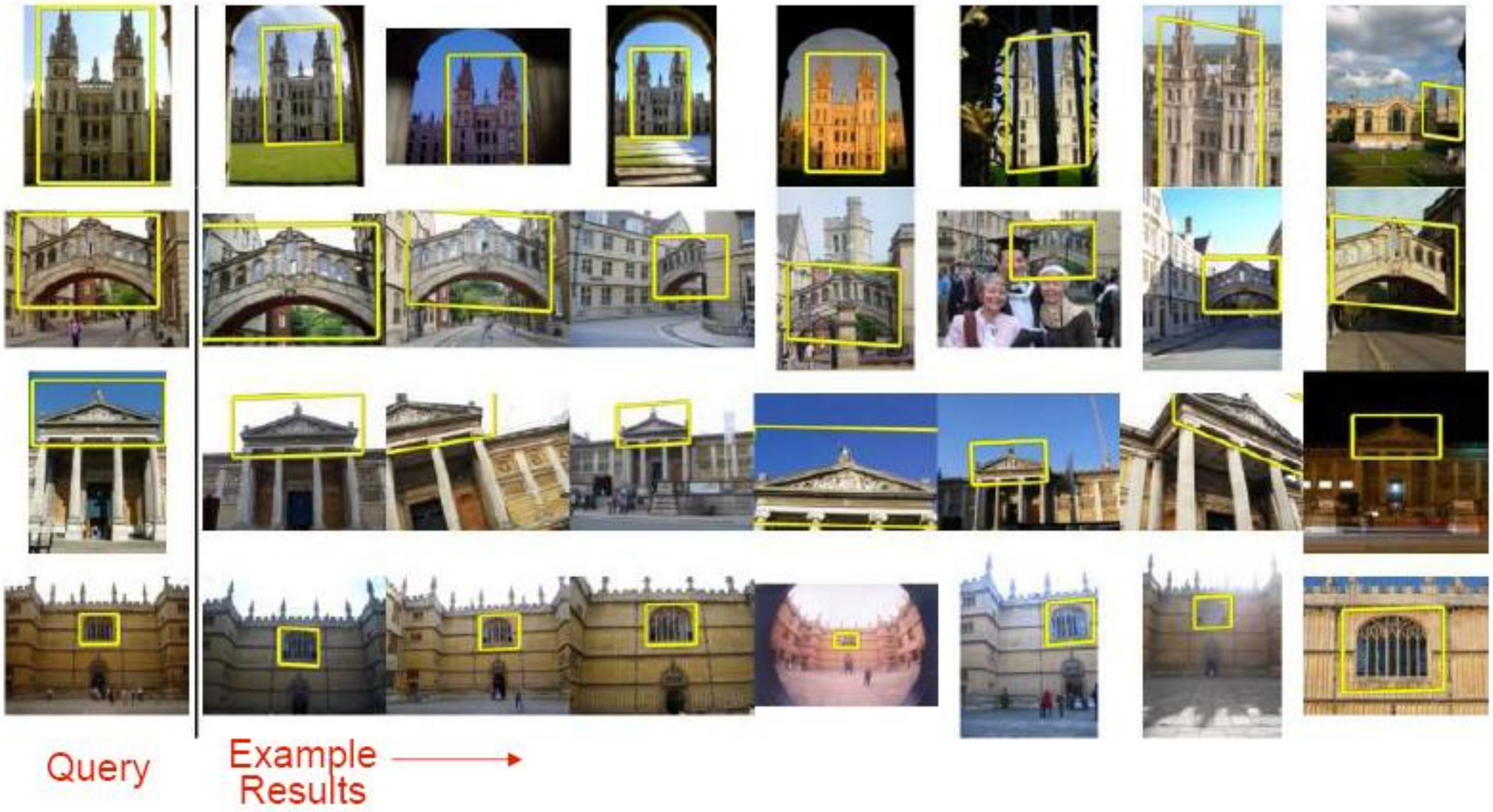
Use RANSAC on full affine transformation (6 dof)

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



Example Results



Summary and Extensions

Have successfully ported methods from text retrieval to the visual domain:

- Visual words enable posting lists for efficient retrieval of specific objects
- Spatial re-ranking improves precision

Outstanding problems:

- Include spatial information into index
- Universal vocabularies

Papers and Demo

Sivic, J. and Zisserman, A.
Video Google: A Text Retrieval Approach to Object Matching in Videos
Proceedings of the International Conference on Computer Vision (2003)
http://www.robots.ox.ac.uk/~vgg/publications/papers/sivic03.pdf

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

Philbin, J., Chum, O., Isard, M., Sivic, J. and Zisserman, A.
Object retrieval with large vocabularies and fast spatial matching
Proceedings of the Conference on Computer Vision and Pattern Recognition(2007)
http://www.robots.ox.ac.uk/~vgg/publications/papers/philbin07.pdf