From similarity to scalability in content-based image and video retrieval

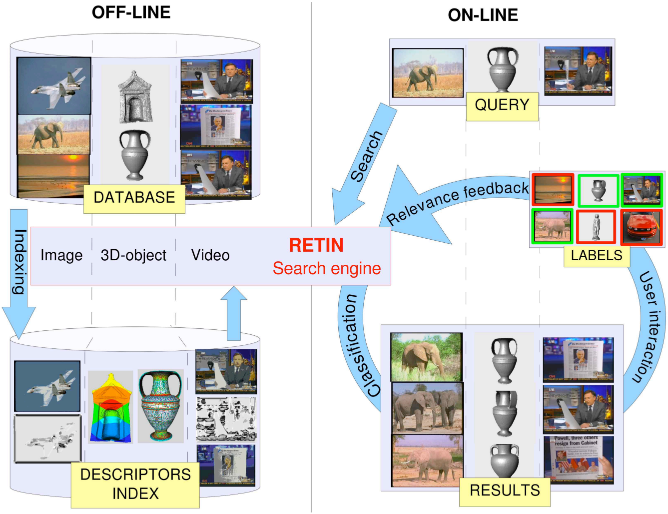
Matthieu Cord
LIP6, Univ. UPMC-PARIS VI, FRANCE
matthieu.cord@lip6.fr

December 13, 2011

Introduction

- Content based image and video retrieval systems
- Kernel design and data representation for actor retrieval
- Approximate k-NN for fast similarity approximation
- Optimization active learning on large scale databases

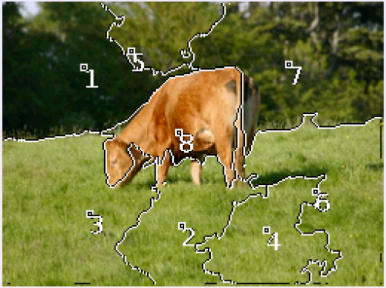
Content Based retrieval system:



Visual data representation

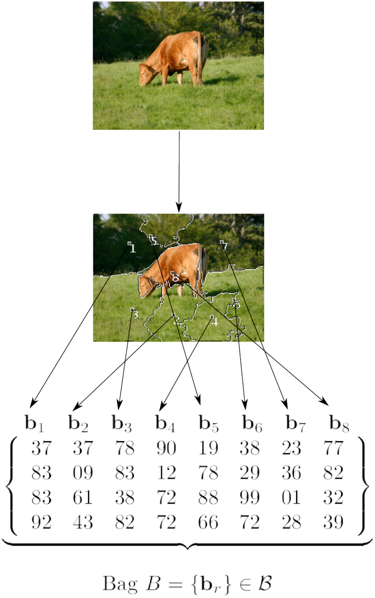
Image features

- Pixels, Points of Interest, RoI, Regions, Blobs

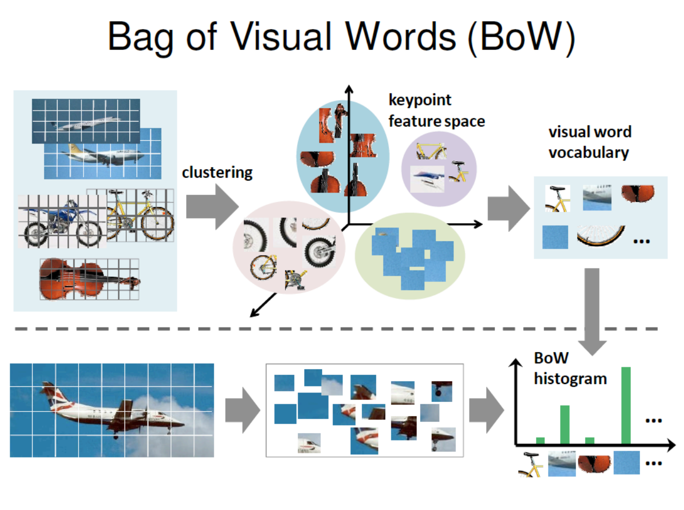


Indexing process

1. Feature extraction
   - $\Rightarrow$ Regions, points of interest, ...
2. Descriptors
   - $\Rightarrow$ Color, texture, SIFT, ...
3. Bags of feature $B_i = \{\mathbf{b}_{ri}\}_r$
   - $\Rightarrow$ $\mathbf{b}_{ri}$: region/poi r of image i
4. Similarity $S(B_i, B_j)$?



SoA model



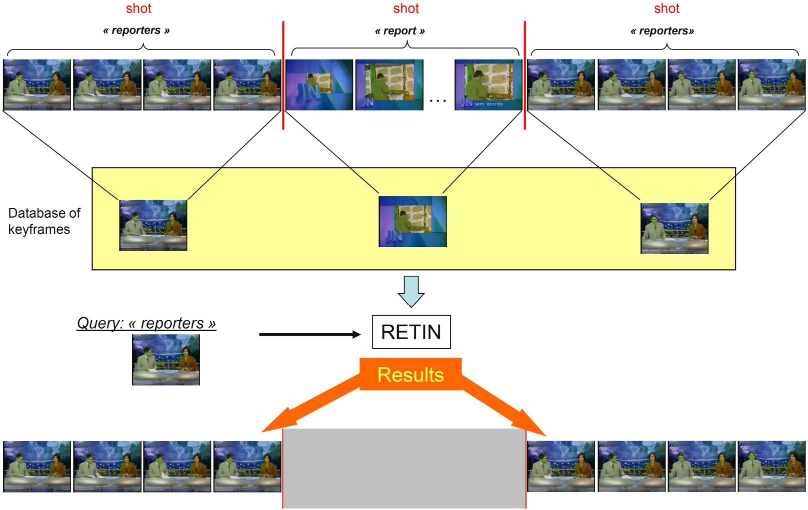
Credit: Prof. Shih-Fu Chang

Similarity $S(B_i, B_j)$ using Visual Dictionary

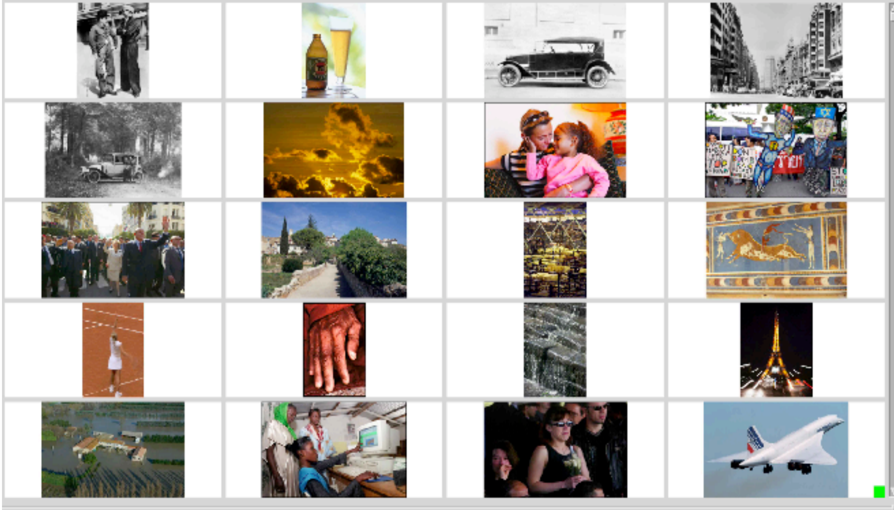
- 2 steps:
  1. Explicit mapping of B_i into a vector space
  2. Similarity on vectors
- Computation of the Visual Dictionary over the database
- Strategies to cluster all the feature data, like k-means

Image index: distribution on the Visual dictionary

Content-based video shot retrieval



Example of search by similarity



Example of search by similarity

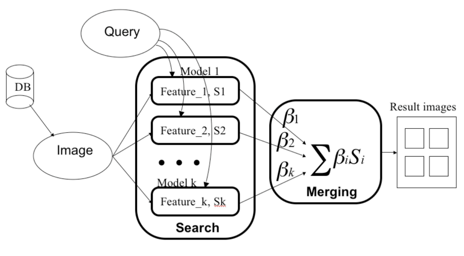
Online learning

Optimisation of the ranking using $\mathcal{A}$

Online learning

Optimisation of the ranking using $\mathcal{A}$

- similarity function updating $f(\mathbf{x})$



Online learning

Optimisation of the ranking using $\mathcal{A}$

- similarity function updating $f(\mathbf{x})$
- classification scheme

Online learning

Optimisation of the ranking using $\mathcal{A}$

- similarity function updating $f(\mathbf{x})$
- classification scheme

$\mathcal{A}$ Enhancement

- Show the top rank examples

Online learning

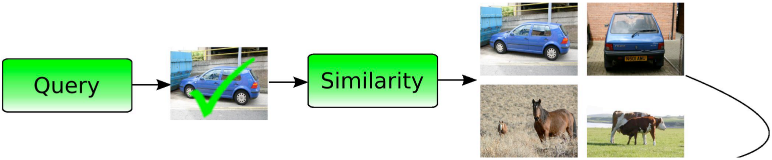
Optimisation of the ranking using $\mathcal{A}$

- similarity function updating $f(\mathbf{x})$
- classification scheme

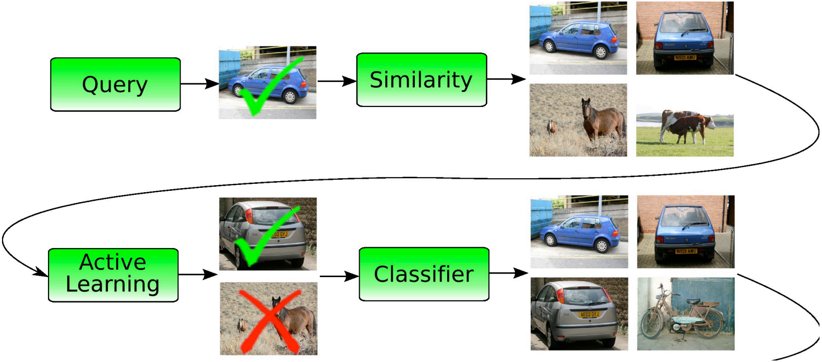
$\mathcal{A}$ Enhancement

- Show the top rank examples
- or show the best ones to enhance ranking: Active learning strategy

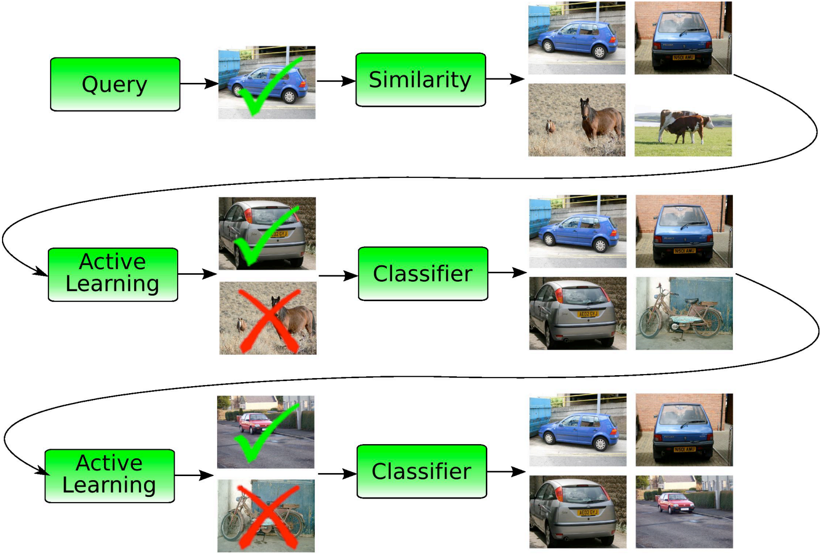
Online learning



Online learning



Online learning



Content-based Video Retrieval: Query

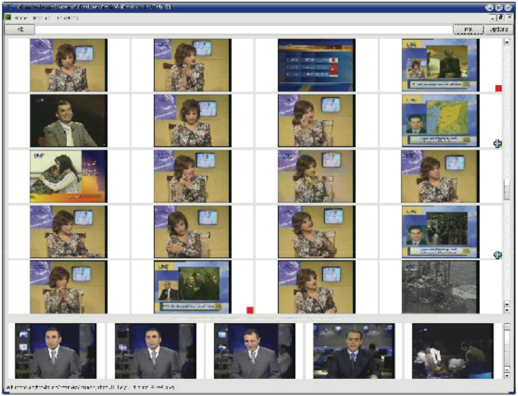


Figure: Query

Content-based Video Retrieval: Result

Figure: Top ranked

Content-based Video Retrieval: Result

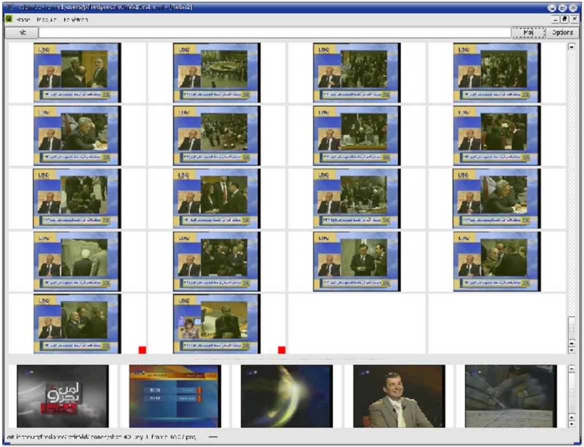
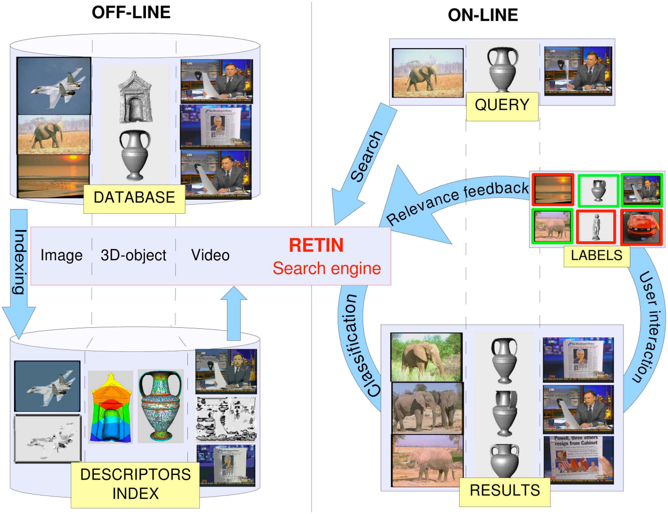


Figure: Bottom ranked

Introduction



Introduction

One step further to track a needle in a haystack

- Dictionary-based approaches $=>$ Vectors as index
- Other index ? more discriminant ?

Similarity functions $S(B_i, B_j)$

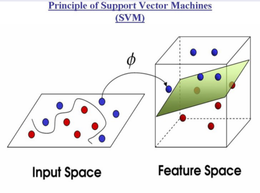
- Alternatives to dictionary-based approaches:
  1. Copy detection approach :
     - Signature $= B_i$ the set of vectors b_{ik}
     - Similarity retrieval using NN search and voting systems
  2. Kernels on bags

Kernel as similarities

Definition of a kernel function $K : \mathcal{X} x \mathcal{X} \rightarrow \mathcal{R}$

K is a kernel *iff* $\exists \Phi | \forall (x, y), K(x, y) =< \Phi(x), \Phi(y) >$
with Φ an injection into a Hilbert $\mathcal{H}$ space (explicit or not)

$$\begin{aligned} \Phi : \mathcal{X} & \rightarrow \mathcal{H} \\ x & \rightarrow \Phi(x) \end{aligned}$$



Advantages:

- Integration with Machine Learning techniques (Neural networks, SVM, ...)
- Allow to build similarities on non vector input spaces

Kernel functions for bags of vectors

Framework:

Soft maximum kernel function [Shawe-Taylor book02]:

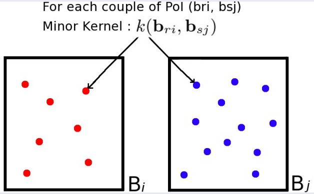
$$K_{softmax}(B_i, B_j) = \sum_{\mathbf{b}_{ri} \in B_i} \sum_{\mathbf{b}_{sj} \in B_j} k(\mathbf{b}_{ri}, \mathbf{b}_{sj})$$

- Nice property:
  k is a kernel function $\Rightarrow$ $K_{softmax}$ is a kernel function
- Not enough discriminant ?

Kernel on Bags of Features

Improvement [LyuCVPR 05, CordCIVR 07]

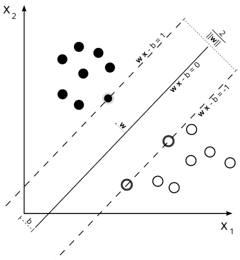
$$K(B_i, B_j) \triangleq \left(\sum_r \sum_s \left(k(\mathbf{b}_{ri}, \mathbf{b}_{sj})\right)^q \right)^{\frac{1}{q}}$$



Classifier

Training set

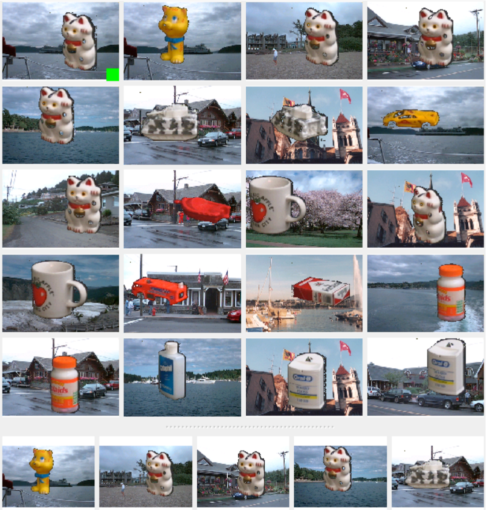
- $\mathcal{A} = \{(\mathbf{x}_i, y_i)_{i=1,N} \mid y_i \neq 0\}$
- $\mathcal{U} = \{(\mathbf{x}_i, y_i)_{i=1,N} \mid y_i = 0\}$



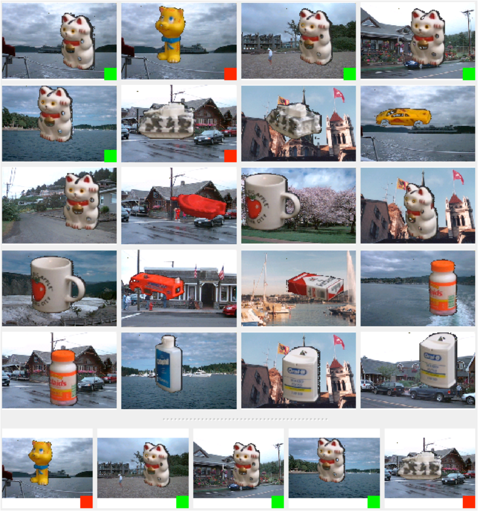
SVM :

- Minimize $\frac{||\mathbf{w}||^2}{2}$ s.t. $y_i(<\mathbf{w}, \Phi(\mathbf{x_i})> + b) \geq 1, \forall i \in [1, N]$
- Classifier : $f_{\mathcal{A}}(\mathbf{x}) =< \mathbf{w}, \Phi(\mathbf{x}) > + b$

Example of retrieval session



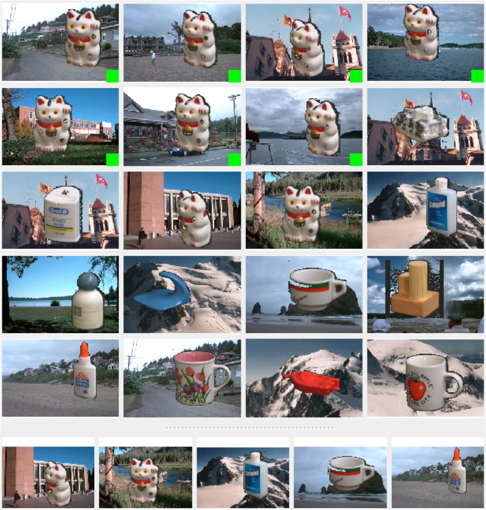
Example of retrieval session



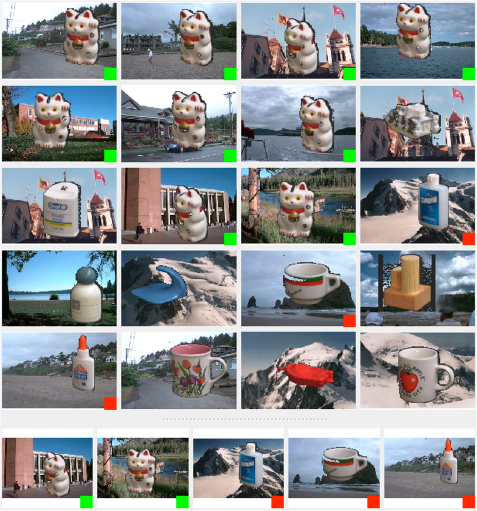
Example of retrieval session

Example of retrieval session

Example of retrieval session



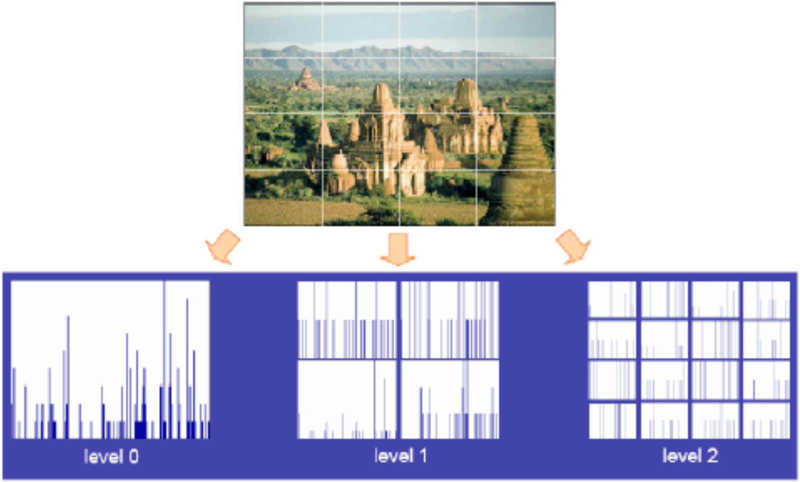
Example of retrieval session



Example of retrieval session

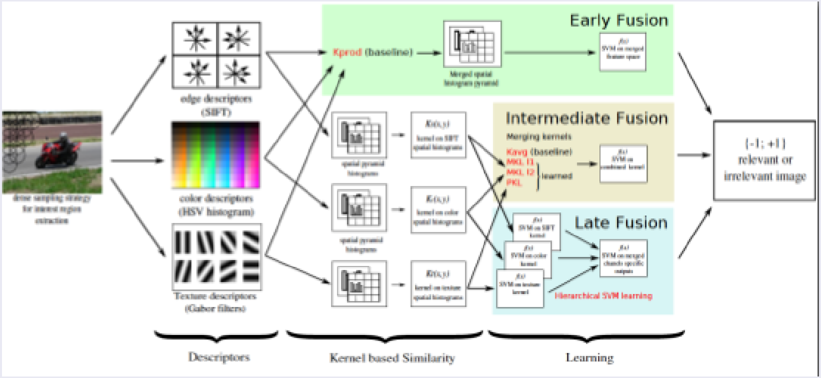
Any Extension ?

Extension : integration of spatial constraints

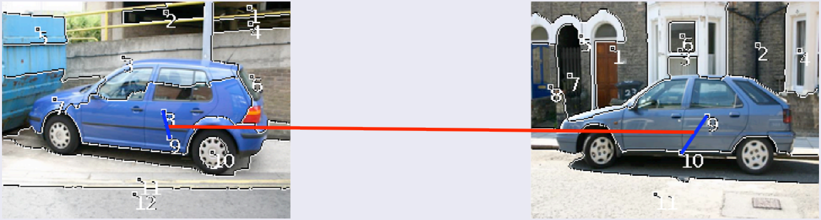


Credit: Dr. S. Lazebnik

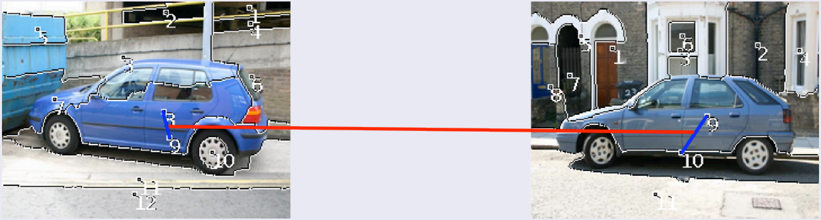
Extension : integration of spatial constraints



Extension : integration of spatial constraints



Extension : integration of spatial constraints

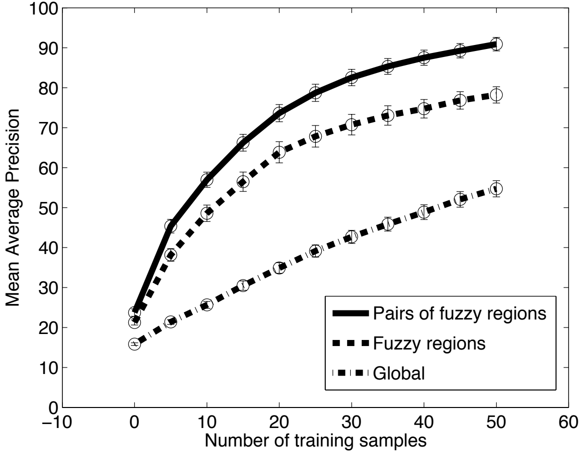


Kernel function on bag $\mathcal{P}_i$ of bags of pairs P_{vi}:

$$K_{pairs}(\mathcal{P}_i, \mathcal{P}_j) = \left(\sum_{P_{vi} \in \mathcal{P}_i} \sum_{P_{wj} \in \mathcal{P}_j} K_{single}(P_{vi}, P_{wj})^q \right)^{\frac{1}{q}}$$

For each region $\mathbf{b}_{ri}$, we build 3 pairs with its 3 closest regions.
K_{pairs} may be connected to kernel on graphs [Kashima]

Evaluation



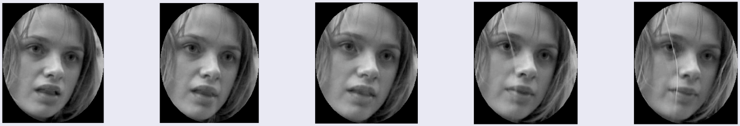
RETIN Active learning with 5 labels/feedback, 10 feedbacks.

Extension (2): application to video actor retrieval

Video object extraction and description

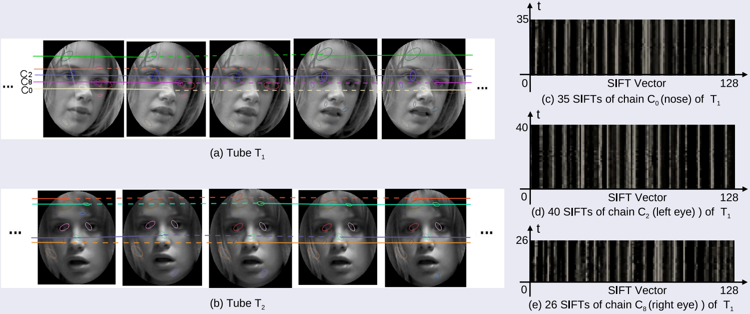
- RoI = face tubes
  - Frame face detection
  - Face region grouping in shots

Example of a tube:



Data representation

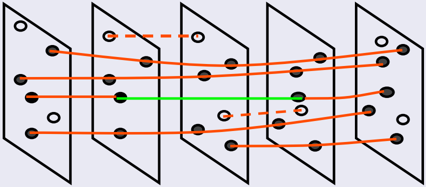
Temporal stability of SIFT points: Intra-tube chain tracking



SIFT points along the same chain in same color (scale and orientation of ellipses representing the scale and orientation of SIFT)

Representation optimization

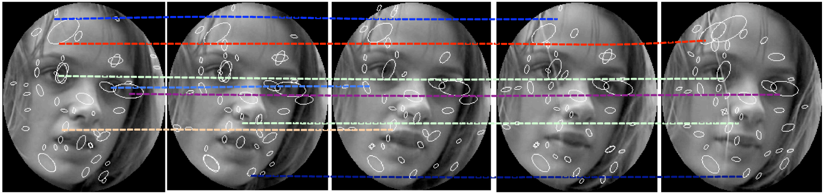
- Intra-tube chain tracking
- Consistent chain extraction:



Solid lines: consistent chains, dash lines: noise, green lines: linking two short chains

- Tube T_i: a set of chains C_{ri} of SIFT descriptors:
- $T_i = \{C_{1i}, \ldots, C_{ki}\}$ and $C_{ri} = \{SIFT_{1ri}, \ldots, SIFT_{pri}\}$

Kernel design for actor retrieval



The major kernel on tubes is then defined as:

$$K'_{pow}(T_i, T_j) = \left(\sum_r \sum_s \frac{|C_{ri}|}{\sqrt{|T_i|}} \frac{|C_{sj}|}{\sqrt{|T_j|}} k'(C_{ri}, C_{sj})^q \right)^{\frac{1}{q}} \tag{1}$$

with the following minor kernel on chains:

$$k'(C_{ri}, C_{sj}) = exp\left(-\frac{1}{2\sigma^2} \chi^2 \left(\overline{C}_{ri}, \overline{C}_{sj} \right) \right) e^{-\frac{\left(\overline{x}_{ri} - \overline{x}_{sj}\right)^2 + \left(\overline{y}_{ri} - \overline{y}_{sj}\right)^2}{2\sigma_2^2}}$$

Kernel design

And so what ?

Kernel design

And so what ? Actually, all the work is done !

Kernel design

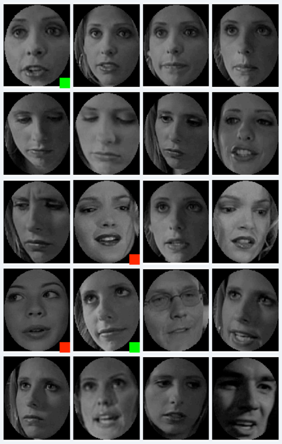
And so what ? Actually, all the work is done !
It is now RETIN compatible: online actor retrieval

Experiments on a french movie ”L’esquive”

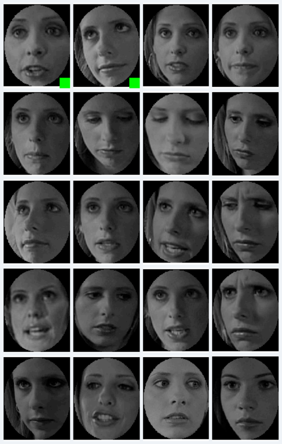
Experiments on a french movie ”L’esquive”

Experiments on a french movie ”L’esquive”

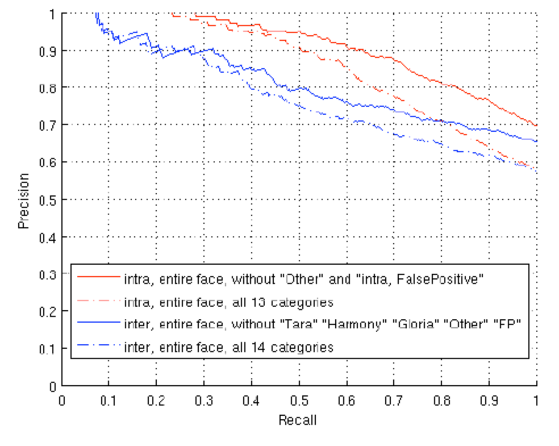
Experiments for multi-class actor retrieval on videos ”Buffy” [Zisserman&Sivic database]



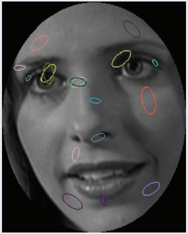
Experiments for multi-class actor retrieval on videos ”Buffy” [Zisserman&Sivic database]

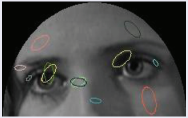


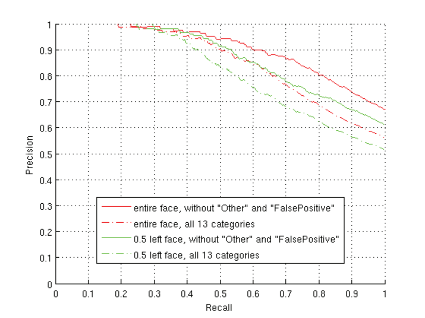
Experiments for multi-class actor retrieval on videos ”Buffy” [Zisserman&Sivic database]



Experiments on system robustness:

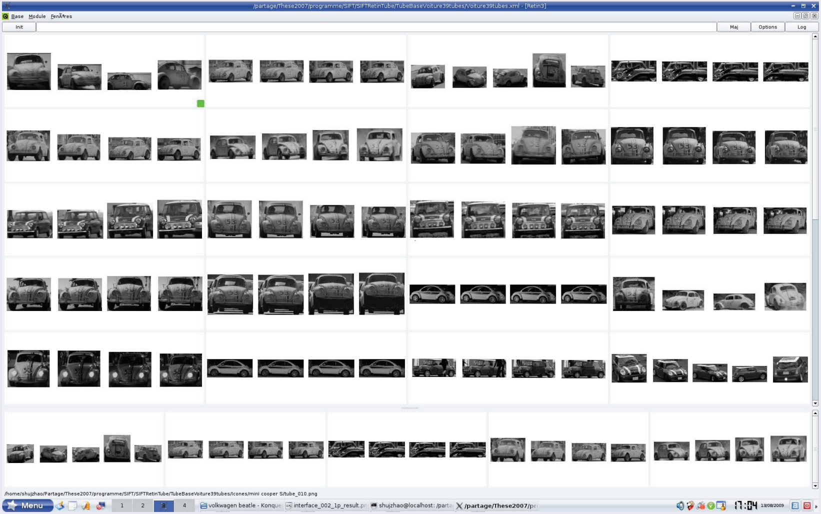






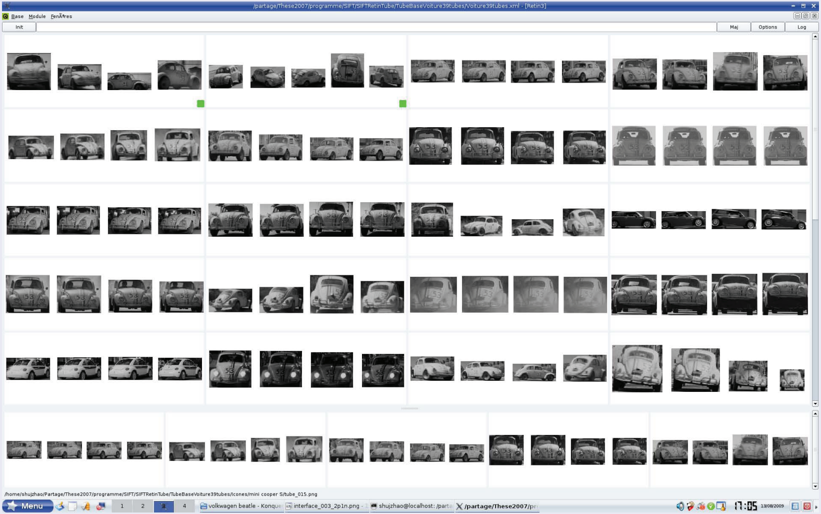
Experiments on system generalization:

Experiments on system generalization:



Experiments on system generalization:

Experiments on system generalization:



Introduction to fast retrieval scheme

Computation optimization pb

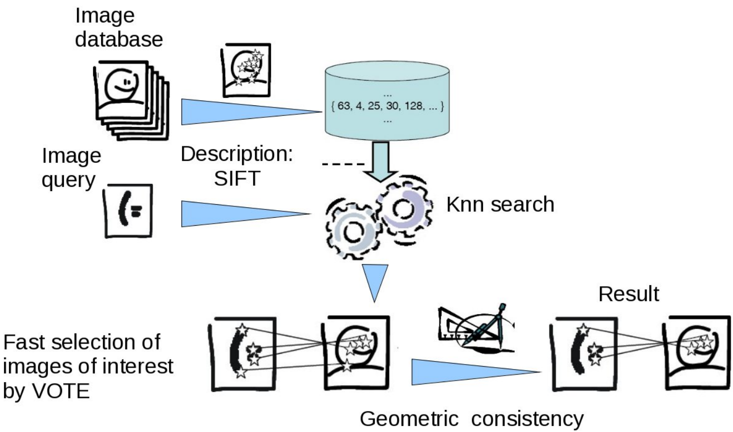
Control of search complexity when the size of the database becomes huge
Problem even more crucial when the number and the size of the descriptors increase

Computational pb of similarity functions $S(B_i, B_j)$

- All the Alternatives to dictionary-based approaches are time consuming
  1. Copy detection approach :
     - Signature $= B_i$ the set of vectors b_{ik}
     - Similarity retrieval using NN search and voting systems
  2. Kernels on bags

$\Rightarrow$ Need optimization scheme !

Copy Detection scheme [Lowe04]



Copy Detection scheme

Optimization scheme

- **Fast NN search (1)** to quickly retrieve near duplicate or most similar images (TOPN) to a given query
- Need to structure the database $\Rightarrow$ indexing scheme

Database indexing schemes

- Classical indexing schemes fail with high dimensional data
- Approximate search approaches
  - Tree techniques (BBFirst Kd-tree,...)
  - Projections (NV Tree, VA files, Space Filing Curves, **Locality Sensitive Hashing**)

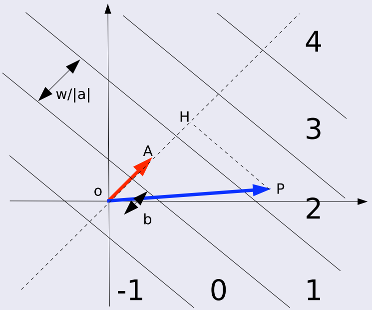
Implementing Locality Sensitive Hashing

[datar 2004]

$f_i()$: function of the hash table i and $h_{a,c}()$ the hash function:

$f_i(\mathbf{b}) = \left(h^i_{a_1,c_1}(\mathbf{b}), \ldots, h^i_{a_k,c_k}(\mathbf{b})\right)$

$h_{\mathbf{a},c}(\mathbf{b}) = \left\lfloor \frac{\mathbf{a}.\mathbf{b}+c}{w} \right\rfloor$



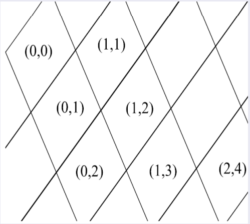
Implementing Locality Sensitive Hashing

[datar 2004]

$f_i()$: function of the hash table i and $h_{a,c}()$ the hash function:

$f_i(\mathbf{b}) = \left(h^i_{a_1,c_1}(\mathbf{b}), \ldots, h^i_{a_k,c_k}(\mathbf{b})\right)$

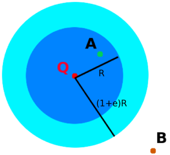
$h_{\mathbf{a},c}(\mathbf{b}) = \left\lfloor \frac{\mathbf{a}.\mathbf{b}+c}{w} \right\rfloor$



Locality-Sensitive Function [indyk 1998]

[datar 2004]

Under conditions, the $h_{a,c}()$ family is LSH:



DEFINITION

$\mathcal{H} = \{h : S \to U^1\}$ is (R, ϵ, p_1, p_2) Locality-Sensitive if, $\forall(A, B, Q)$:

$$A \in B(Q, R) \Rightarrow Pr_{\mathcal{H}}[h(Q) = h(A)] \geq p_1, \tag{3}$$

$$B \notin B(Q, (1+\epsilon)R) \Rightarrow Pr_{\mathcal{H}}[h(Q) = h(B)] \leq p_2. \tag{4}$$

Implementing Locality Sensitive Hashing (2)

Implementation depending on the representation space

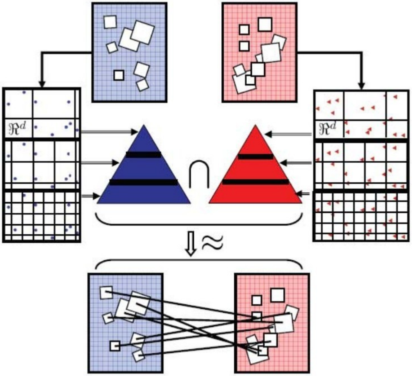
- in Hamming space $\mathcal{H}^d$ or $\mathcal{Z}^d$: LSH random permutation [Indyk98]
- in $\mathcal{R}^d$ normalized: cosine similarity [Charikar02]
- in $\mathcal{R}^d$: distance L2 or L1
  - [Gionis99] projection of $\mathcal{R}^d$ in $\mathcal{H}^d$ + [Indyk98]
  - [Datar04] splitting along 1 dimension
  - [Lv07] (multi-probe) extension of [Datar04]
  - [Andoni06] 24 lattice, [Jegou08] E8 lattice
- Implementation available for a vector representation of images and distances aforementioned
- Extension to other similarities and to non vector spaces ?

Fast kernel on Bags Pyramid Match Hashing [Grauman07]

- Each image is described by a bag of SIFT
- Injection with a function Φ in a space of high dimension
- The injection is explicit:
  - Projection into SIFT space
  - Multi-scale grid
  - Projection into Hamming space

$\Rightarrow$ Each image becomes a unique Vector

- An explicit induced space allows to use LSH
- The resulting kernel allows to get a similarity from the matching between PoIs (Points of Interest) of the 2 images



LSH on other kernels ?

- Pyramid Match Hashing $\Rightarrow$ good performances
- BUT cannot be extended to kernels where Φ is not explicit
- If the class of kernels is different:

$$\text{ex: } K(B_i, B_j) = \left(\sum_r \sum_s \left(k(\mathbf{b}_{ri}, \mathbf{b}_{sj}) \right)^q \right)^{\frac{1}{q}}$$

- Can we speed up the computation?

Approx. scheme [ICPR 2008]

- Model:
  - consider each image as a bag of unordered features
  - similarity : class of kernels on bags

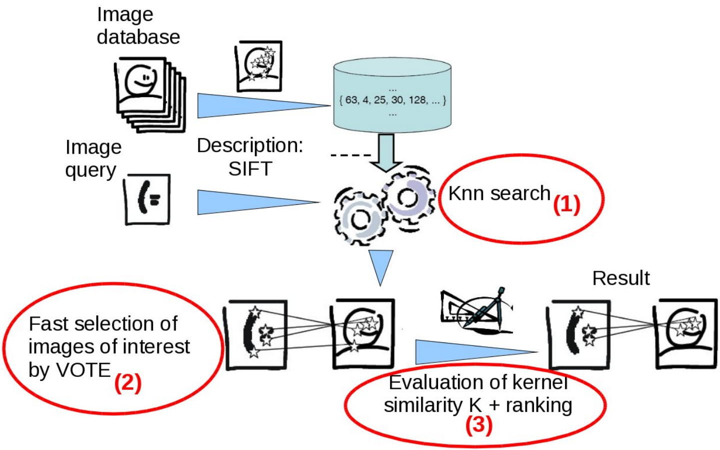
$$K(B_i, B_j) = \left(\sum_r \sum_s \left(k(\mathbf{b}_{ri}, \mathbf{b}_{sj})\right)^q \right)^{\frac{1}{q}}$$

- Objective:
  - fast computation of the topN from a ranking of the database with similarity kernel K

  $\Rightarrow$ decrease the kernel computational complexity
- Principle (inspired from copy detection):
  - (1) (2) Quick selection of database subset (LSH scheme)
  - (3) Kernel computation only on this relevant subset

  $\Rightarrow$ resulting scheme is an approximation of the exact similarity ranking of the whole database

Principle for fast retrieval



Pre-processing: Hashing of the database

For each image B_i

For each attribute b_{si}

For each hash table k

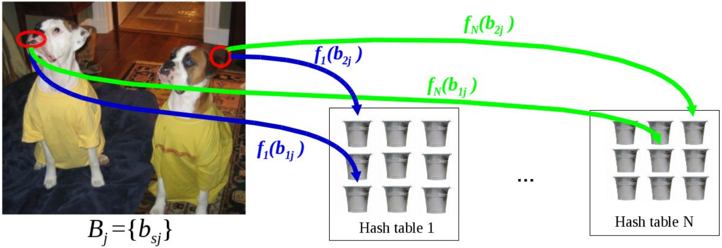
- selection of a bucket with hashing function: $f_k(b_{si})$
- put b_{si} in the selected bucket

Locality Sensitive Hashing [datar 2004]
Notation : $f_i()$: function of the hash table i

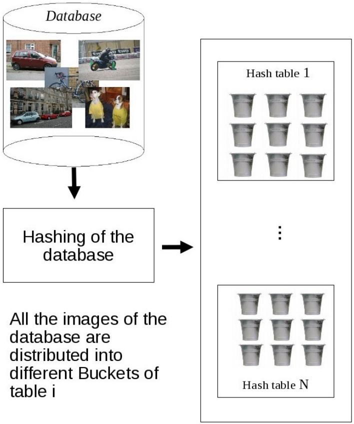
$$f_i(\mathbf{b}) = \left(h^i_{a_1,c_1}(\mathbf{b}), \ldots, h^i_{a_k,c_k}(\mathbf{b})\right)$$

$h_{a,c}()$: hash function

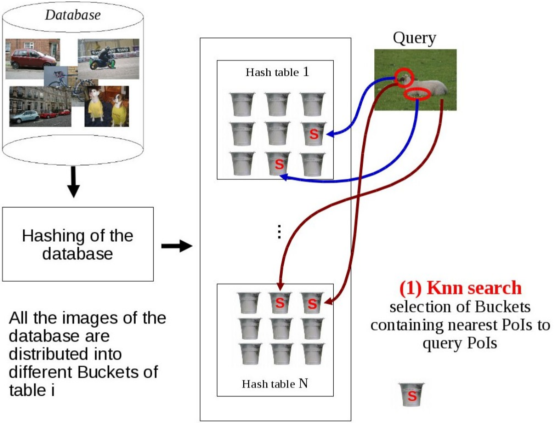
$$h_{\mathbf{a},c}(\mathbf{b}) = \left\lfloor \frac{\mathbf{a}.\mathbf{b}+c}{w} \right\rfloor$$



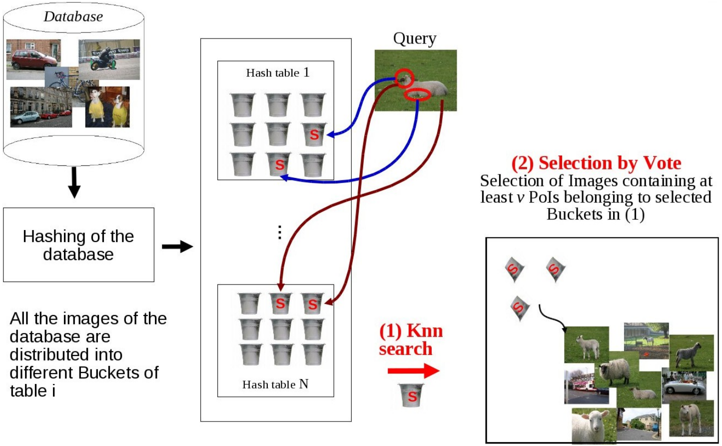
Retrieval Algorithm



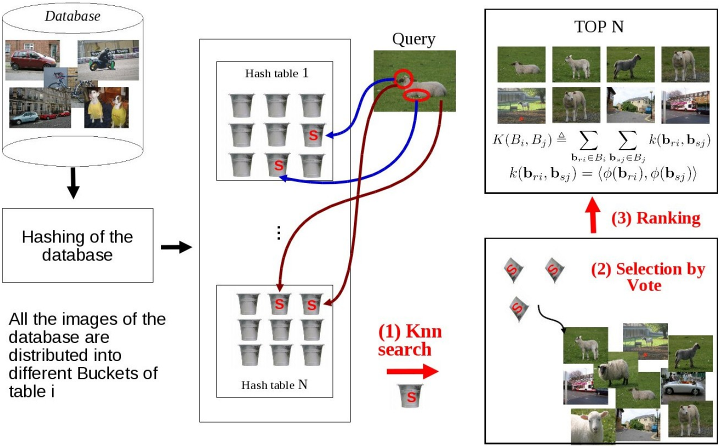
Retrieval Algorithm



Retrieval Algorithm



Retrieval Algorithm

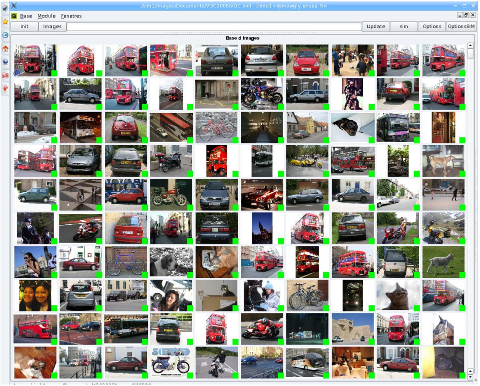


Experiments

- VOC2006 database : 5,304 images
- Indexing : $\sim$100 PoI per Image
  - MSER region detectors
  - SIFT descriptors
- Variance normalization
- E2LSH parameters
  - radii between 150 and 250 (4.0 and 6.0 after normalization)
  - L = 50 hash tables
  - K = 20 projections
- Image selection VS whole database
  - TOP100 deterioration
  - computational time reduction

Example

Fast Selection + Ranking by Vote

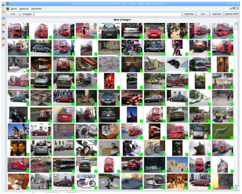


372 / 5304 images (7,1% of the database)

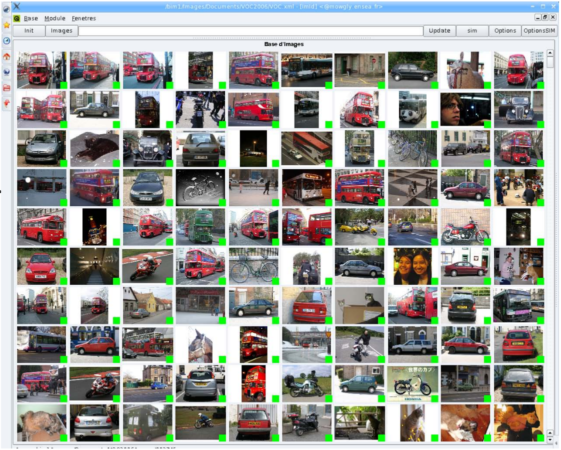
Example

Ranking of the selection by Similarity K

Fast selection



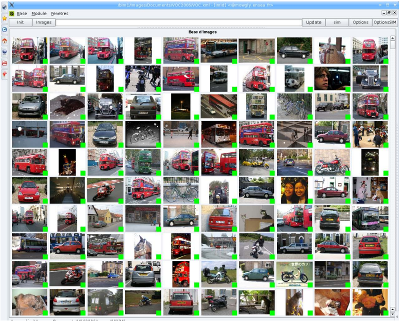
$\Rightarrow$



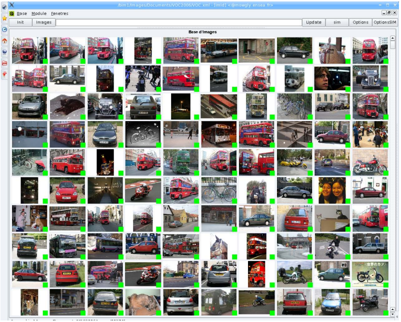
372 / 5304 images (7,1% of the database)

Example

Selection ranking

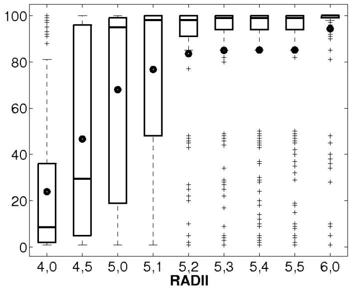


Ground truth for K : Ranking of the whole database

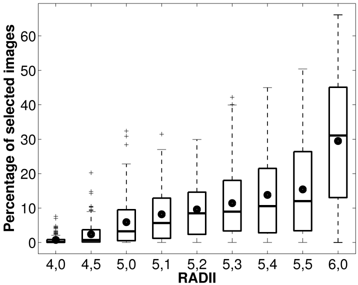


96% of images of TOP100 obtained from our fast selection are identical to TOP100 on the whole database

Results



Accuracy of TOP100 for various radii of search around query points



Percentage of selected images for various radii

radius	4.0	5.0	5.2	6.0
factor	122.17	14.85	10.03	3.19

Speed improvement factor regarding the true search

Discussion

Fast similarity scheme

- Fast similarity search working with non explicit kernels and with all fast knn search methods
- Good trade-off between Precision and Speed for R=5.2: 10 time faster and median precision 99%

But ...

TOPN not good enough for category retrieval

Discussion

Fast similarity scheme

- Fast similarity search working with non explicit kernels and with all fast knn search methods
- Good trade-off between Precision and Speed for R=5.2: 10 time faster and median precision 99%

But ...

TOPN not good enough for category retrieval
Is it RETIN compatible ?

Discussion

Fast similarity scheme

- Fast similarity search working with non explicit kernels and with all fast knn search methods
- Good trade-off between Precision and Speed for R=5.2: 10 time faster and median precision 99%

But ...

TOPN not good enough for category retrieval
Is it RETIN compatible ?
Need adaptation for online category learning

Scalable active learning

Introduction to fast online retrieval

Can we decrease the complexity of Active Learning using similar strategy than ICPR08 ?

Scalable active learning

Introduction to fast online retrieval

Can we decrease the complexity of Active Learning using similar strategy than ICPR08 ?
Not straightforward to combined fast similarity schemes with online/Active Learning.

Scalable active learning

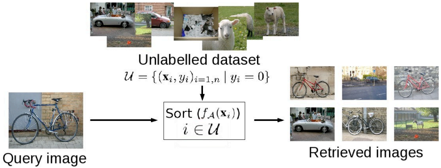
Introduction to fast online retrieval

Can we decrease the complexity of Active Learning using similar strategy than ICPR08 ?

Not straightforward to combined fast similarity schemes with online/Active Learning.

Active Learning schemes: at least a complexity linear regarding the size of the database. $\Rightarrow$ impracticable for large database.

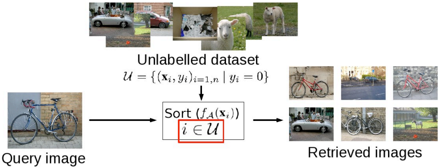
Scalability Problem



Active Learning have 2 problems of scalability. The database have to be sorted to extract :

- the relevant images for intermediate results.
- the most uncertain images for annotation strategy.

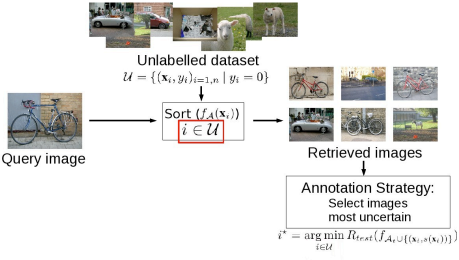
Scalability Problem



Active Learning have 2 problems of scalability. The database have to be sorted to extract :

- the relevant images for intermediate results.
- the most uncertain images for annotation strategy.

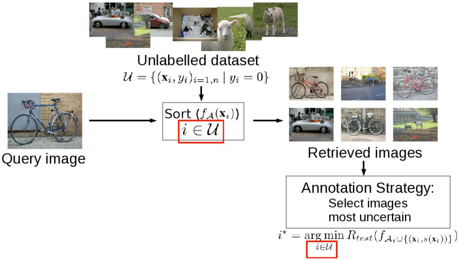
Scalability Problem



Active Learning have 2 problems of scalability. The database have to be sorted to extract :

- the relevant images for intermediate results.
- the most uncertain images for annotation strategy.

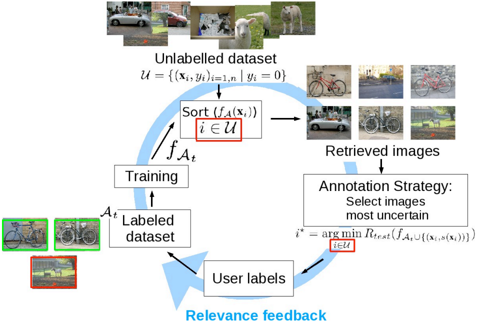
Scalability Problem



Active Learning have 2 problems of scalability. The database have to be sorted to extract :

- the relevant images for intermediate results.
- the most uncertain images for annotation strategy.

Scalability Problem

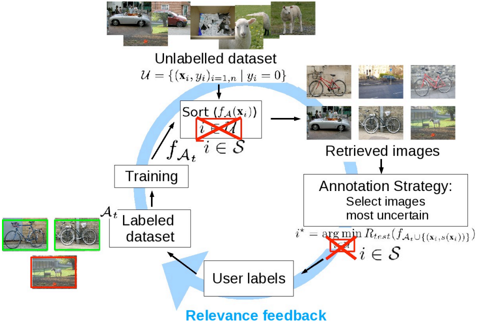


Active Learning have 2 problems of scalability. The database have to be sorted to extract :

- the relevant images for intermediate results.
- the most uncertain images for annotation strategy.

These scalability problems occure at each feedback loop

Scalability Problem

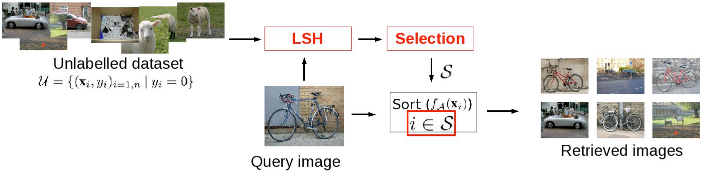


Active Learning have 2 problems of scalability. The database have to be sorted to extract :

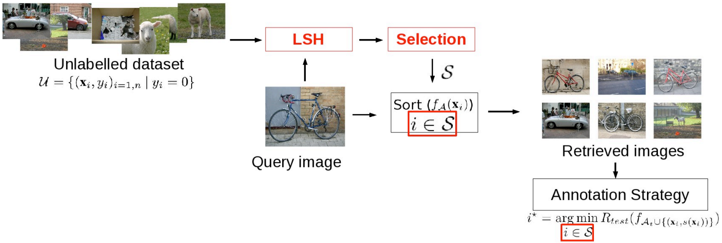
- the relevant images for intermediate results.
- the most uncertain images for annotation strategy.

We tackle these problems by considering only a relevant subset $\mathcal{S}$ instead of $\mathcal{U}$.

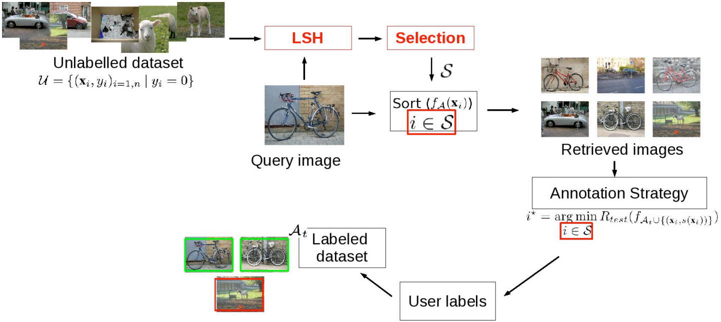
Approx. scheme [icip09]



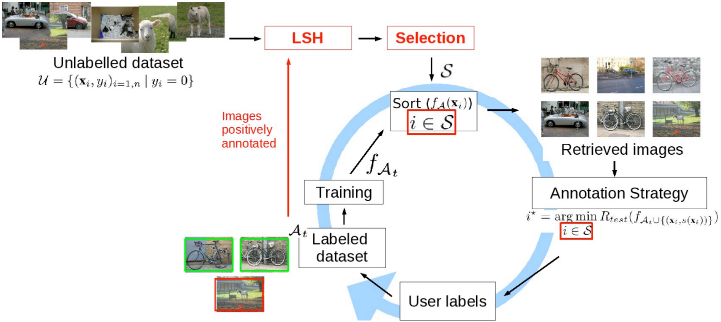
Approx. scheme [icip09]



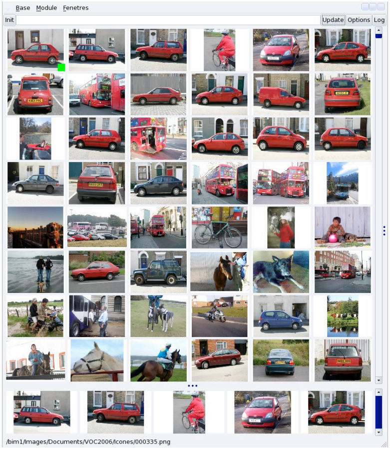
Approx. scheme [icip09]

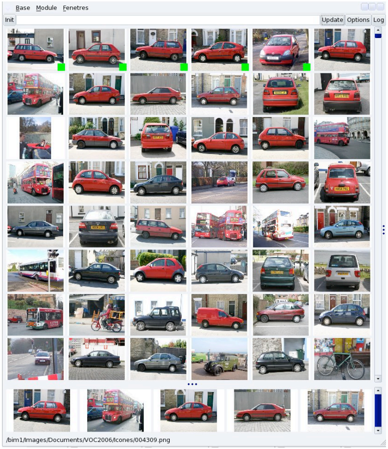


Approx. scheme [icip09]



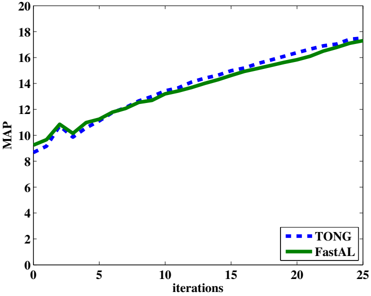
Example



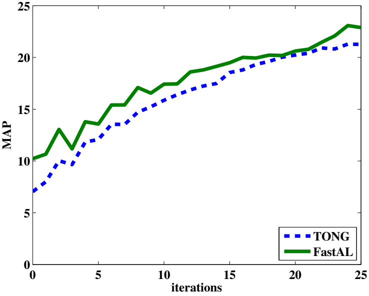


Each image is represented by a 192-dimension vector: 64 chrominances CIE Lab and 2 histograms of 64 textures from Gabor filters.

Experiments



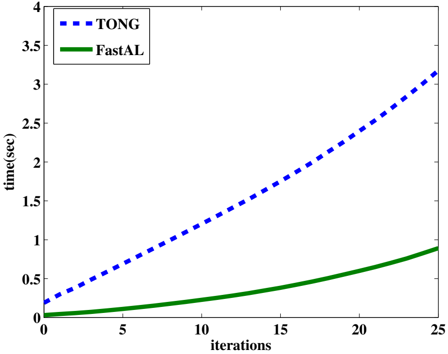
Mean Average Precision(%) of the 10 classes



Mean Average Precision(%) for an interactive search of the class CAR

- Performances are evaluated with Mean Average Precision of the TOP500, i.e., the sum of the Precision/Recall curve for the first 500 images retrieved.
- E2LSH parameters are R = 16.0 and L = 30 hash tables of K = 20 projections.

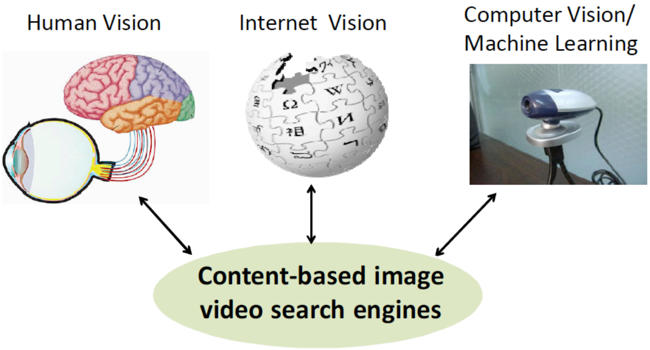
Experiments



Average time of an interactive search function of the number of iteration

Conclusion

Next-Generation Visual Search



Credit: Prof. Shih-Fu Chang

People

Matthieu Cord, Nicolas Thome
LIP6, Univ. UPMC-PARIS VI
matthieu.cord@lip6.fr

Frédéric Precioso
ETIS UCP
precioso@ensea.fr

- Shuji Zhao (PhD student),
- David Gorisse (PhD student),
- David Picard (PostDoc LIP6)

http://webia.lip6.fr/∼cord/
http://frederic.precioso.free.fr

Bibliography pieces

- Optimization on Active Learning Strategy for Object Category Retrieval ,D. Gorisse, M. Cord, F. Precioso, ICIP 2009
- Spatio-Temporal Tube Kernel for Actor Retrieval, S. Zhao, F. Precioso, M. Cord, ICIP, 2009
- Active learning methods for Interactive Image Retrieval, P.H. Gosselin, M. Cord, IEEE Transactions on Image Processing. Volume: 17, Issue: 7, Pages 1200-1211, ISSN: 1057-7149, 2008
- Fast Approximate Kernel-based Similarity Search for Image Retrieval Task, D. Gorisse, M. Cord, F. Precioso, S. Philipp-Foliguet, 19th ICPR International Conference on Pattern Recognition - dec 2008
- Combining visual dictionary, kernel-based similarity and learning strategy for image category retrieval, P.H. Gosselin, M. Cord, S. Philipp, In Computer Vision and Image Understanding, Volume 110, Issue 3, Pages 403-417, ISSN:1077-3142, 2008
- Machine Learning Techniques for Multimedia – Case Studies on Organization and Retrieval, Springer 2008 M. Cord, P. Cunningham (Eds.)