Incremental Indexing and Distributed Image Search using Shared Randomized Vocabularies

Raphaël Marée, Philippe Denis, Louis Wehenkel, Pierre Geurts

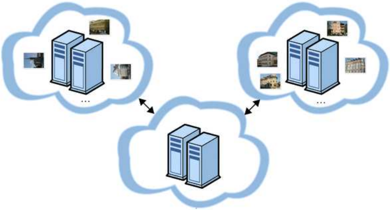
GIGA Bioinformatics
GIGA Research ; Dept. EE & CS (Montefiore Institute)
University of Liège, Belgium

MIR 2010
March 29–31, 2010
Philadelphia, Pennsylvania, USA

Université de Liège

Context: a realistic setting

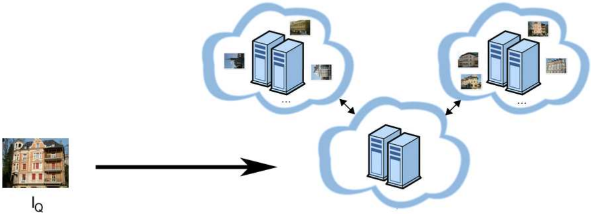
Content-based image indexing and retrieval when images are **distributed** and added in a **incremental** fashion.



e.g. networks of hospitals, institutional repositories, community websites, peer-to-peer networks, etc.

Context: a realistic setting

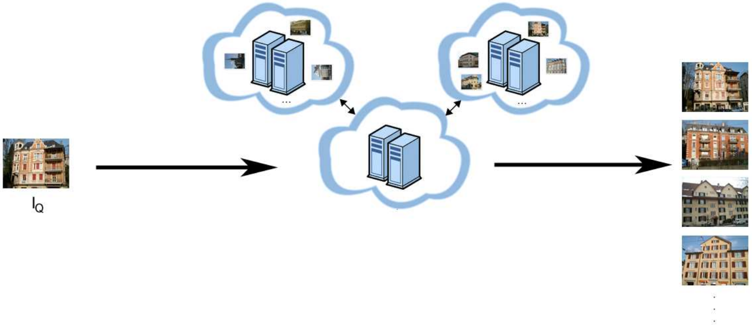
Content-based image indexing and retrieval when images are **distributed** and added in a **incremental** fashion.



e.g. networks of hospitals, institutional repositories, community websites, peer-to-peer networks, etc.

Context: a realistic setting

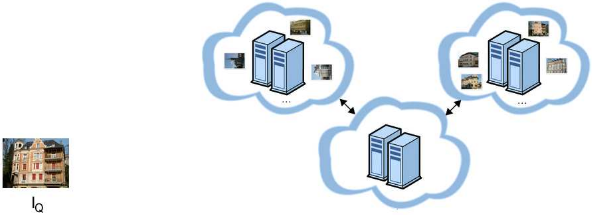
Content-based image indexing and retrieval when images are **distributed** and added in a **incremental** fashion.



e.g. networks of hospitals, institutional repositories, community websites, peer-to-peer networks, etc.

Context: a realistic setting

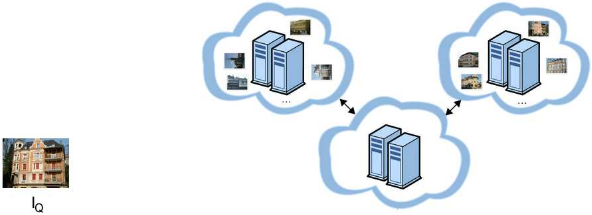
Content-based image indexing and retrieval when images are **distributed** and added in a **incremental** fashion.



e.g. networks of hospitals, institutional repositories, community websites, peer-to-peer networks, etc.

Context: a realistic setting

Content-based image indexing and retrieval when images are **distributed** and added in a **incremental** fashion.

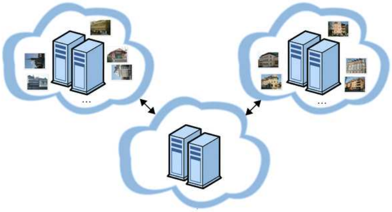


e.g. networks of hospitals, institutional repositories, community websites, peer-to-peer networks, etc.

Context: a realistic setting

Content-based image indexing and retrieval when images are **distributed** and added in a **incremental** fashion.

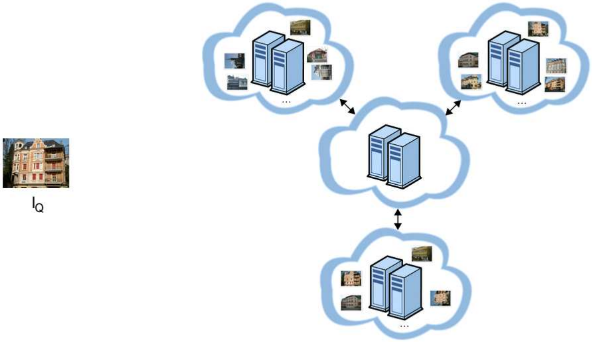
I_Q



e.g. networks of hospitals, institutional repositories, community websites, peer-to-peer networks, etc.

Context: a realistic setting

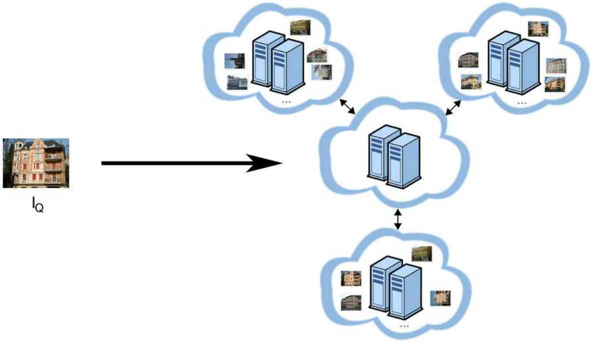
Content-based image indexing and retrieval when images are **distributed** and added in a **incremental** fashion.



e.g. networks of hospitals, institutional repositories, community websites, peer-to-peer networks, etc.

Context: a realistic setting

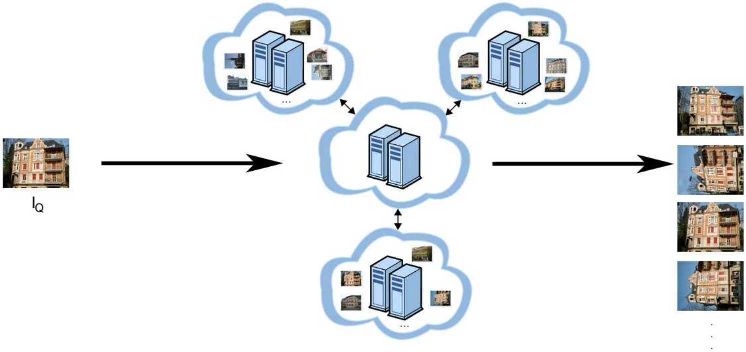
Content-based image indexing and retrieval when images are **distributed** and added in a **incremental** fashion.



e.g. networks of hospitals, institutional repositories, community websites, peer-to-peer networks, etc.

Context: a realistic setting

Content-based image indexing and retrieval when images are **distributed** and added in a **incremental** fashion.

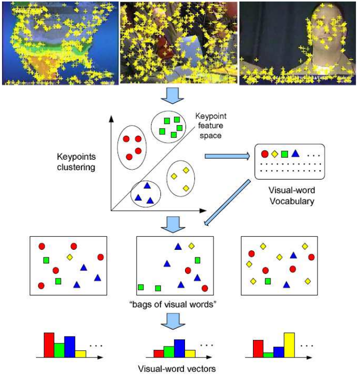


e.g. networks of hospitals, institutional repositories, community websites, peer-to-peer networks, etc.

Université de Liège

Bag-of-visual-words [Leung & Malik 2001 ; Sivic et.al 2003; Dance et al. 2004]

- Inspired by bag-of-words approaches in text retrieval



(figure taken from [Yang et al., MIR 2007])

- State-of-the-art results (often better than global methods), e.g. better than GIST in [Douze et al., CIVR 2009].

Université de Liège

Bag-of-visual-words problems in a realistic setting

- The **visual vocabulary** is usually built using **data-dependent** algorithms (K-Means, Vocabulary Tree, Randomized Trees, ...). It uses only available data so visual vocabularies built from different servers are neither "complete" nor "aligned". Therefore, **image similarities are not directly comparable**.

- The visual vocabulary structure (e.g. number of cluster centers, number of levels in a tree, ...) **can not be easily updated** when new images are becoming available.

... *How can we cast bag-of-visual-words into a distributed, incremental setting ?*

This work

- A **data-independent visual vocabulary** algorithm to map patches to visual words.
- **The same visual vocabulary structure is deployed** on all local servers and used by clients.
- Each local server populates its local inverted indexes with its own images, **locally and incrementally**.
- During retrieval, **image similarities are computed locally by each server** using the standardized visual vocabulary and its local inverted indexes.
- Similarities **are directly comparable**. The retrieval process only requires a **small amount of data transfers** between servers.

From Extra-Trees to Vectors of Random Tests (1/4)

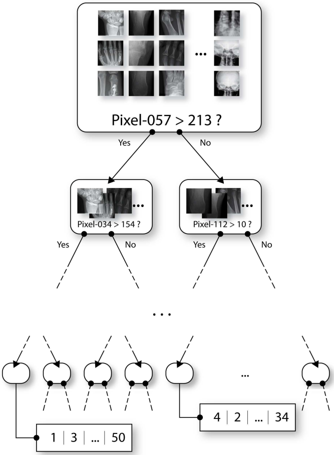
Related work

- Extremely/totally randomized trees [Geurts et al., 2006] for supervised image classification and image retrieval [Marée et al., 2003-2009]
- Random ferns or randomized lists for object tracking [Ozuysal et al. 2007; Williams et al. 2007]
- Random hyperplane hashing [Rajaram & Scholz 2008], Random Features [Rahimi & Recht 2007], ...
- Vector quantizing with a regular lattice [Tuytelaars & Schmid 2007]

Université de Liège

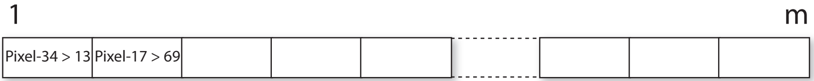
From Extra-Trees to Vectors of Random Tests (2/4)

Visual vocabulary using "totally" randomized trees [ACCV 2007]:



From Extra-Trees to Vectors of Random Tests (3/4)

A single vector of random tests (totally unsupervised, really):

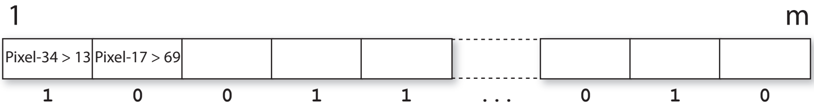


A vector V_t is composed of m binary tests $(test_1(t), ..., test_m(t))$ **randomly generated**, where each test $test_i(t) \equiv 1(x_{j_i} > th_i)$ compares a randomly chosen attribute x_{j_i} to a randomly chosen threshold th_i

Each patch is mapped to a binary code $B = b_1 b_2 ... b_m$ where each $b_i =$ equals to 1 if $test_i(t)$ is true, 0 otherwise.

From Extra-Trees to Vectors of Random Tests (3/4)

A single vector of random tests (totally unsupervised, really):

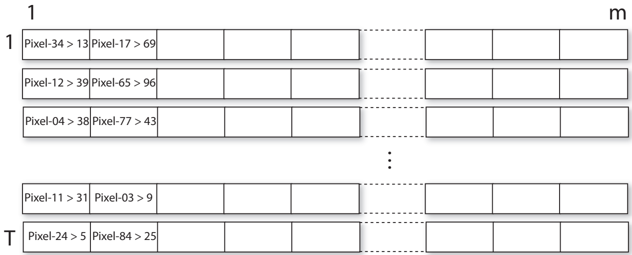


A vector V_t is composed of m binary tests $(test_1(t), ..., test_m(t))$ **randomly generated**, where each test $test_i(t) \equiv 1(x_{j_i} > th_i)$ compares a randomly chosen attribute x_{j_i} to a randomly chosen threshold th_i

Each patch is mapped to a binary code $B = b_1 b_2 ... b_m$ where each $b_i =$ equals to 1 if $test_i(t)$ is true, 0 other

From Extra-Trees to Vectors of Random Tests (4/4)

An ensemble of T random vectors:



- Parameters
  - m: the number of tests in each vector
  - T: the number of vectors
- $T2^m$ **possible visual words**

Similarity between two patches (one vector)

The similarity between two patches s_1 and s_2 is first defined for a given vector V_t by:

$$k_t(s_1, s_2) = \begin{cases} \frac{1}{N_{B,t}} & \text{if } s_1 \text{ and } s_2 \text{ are mapped to the same} \\ & \text{word } B \text{ by } V_t \\ 0 & \text{otherwise,} \end{cases}$$

where $N_{B,t}$ is the total count of indexed patches that were mapped to the visual word B by V_t.

Two patches are **very similar** *if they are mapped to a same visual word that has a* **very small** *number of patches.*

Similarity between two patches (T vectors)

The similarity induced by an *ensemble* of T vectors is defined by:

$$k_T(s_1, s_2) = \frac{1}{T}\sum_{t=1}^{T} k_t(s_1, s_2). \quad (1)$$

Two patches are more similar if they are considered similar by a larger proportion of the vectors.

Université de Liège

Similarity between two images

We derive a similarity between a query image I_Q and a reference image I_R by:

$$k(I_Q, I_R) = \frac{1}{|S(I_Q)||S(I_R)|} \sum_{s_Q \in S(I_Q), s_R \in S(I_R)} k_T(s_Q, s_R), \qquad (2)$$

where $S(I_Q)$ and $S(I_R)$ are the sets of all patches that can be extracted from I_Q and I_R respectively.

The similarity between two images is thus the average similarity between all pairs of their patches.

Université de Liège

Finite sample estimation by Monte-Carlo

The similarity (2) is actually estimated by sampling a finite number of patches from each image and may be rewritten as:

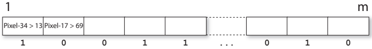
$$k(I_Q, I_R) = \sum_{t=1}^{T} \frac{1}{T} \sum_{B \in \mathcal{V}_{I_Q,t}} \frac{1}{N_{B,t}} \frac{N_{I_Q,B,t}}{N_{I_Q}} \frac{N_{I_R,B,t}}{N_{I_R}}, \qquad (3)$$

where the inner sum is over the set $\mathcal{V}_{I_Q,t}$ of non-empty visual words induced by the vector V_t for the query image I_Q, $N_{B,t}$ is the number of patches from all indexed images that are mapped to word B by V_t, and $N_{I_Q,B,t}$ (resp. $N_{I_R,B,t}$) is the number of patches from I_Q (resp. I_R) that are mapped to B by V_t.

Local image indexing by each server

- Server initialization (once)
  - Get random seed, T, and m
  - Generate the T vectors of m random tests
  - Create an empty inverted index for each vector

- For each new image I_R to index
  - Extract randomly N_{I_R} patches (of random sizes at random locations [Marée et al., CVPR 2005]) and describe them (16 $\times$ 16 raw pixel values)
  - Each patch is mapped by each vector V_t to a visual word B of m bits



  - Update inverted indexes for non-empty visual words with pairs $(I_R.N_{I_R,B,t})$
  - Indexing a new image is $O(TN_{I_R}m)$

Distributed retrieval (1/3)

- Client initialization (once)
  - Get random seed, T, and m
  - Generate the T vectors of m random tests
- Process the image query I_Q
  - Extract N_{I_R} patches (of random sizes at random locations) and describe them (16 $\times$ 16 raw pixel values)
  - Each patch is mapped to T visual words
  - The image is then described by a list $\mathcal{B}$ of triplets $(B, t, \frac{N_{I_Q,B,t}}{N_{I_Q}})$ ranging over the non-empty visual words of I_Q.
  - The list $\mathcal{B}$ is sent to the central server.

Distributed retrieval (2/3)

1. The central server receives the list $\mathcal{B}$ and sends to each cooperating image server the visual word identifiers (B, t) to request their number of patches $N_{Blocal,t}$;
2. Each cooperating server replies to the central server by sending its list of non-empty pairs $(B, t, N_{Blocal,t})$;
3. The central server adds these counts to compute $N_{B,t} = \sum_{local} N_{Blocal,t}$ and sends back to all the image servers the list of four-tuplets $(B, t, \frac{1}{N_{B,t}}, \frac{N_{I_Q,B,t}}{N_{I_Q}})$;

These data exchanges made each local server virtually aware of the complete, global, dataset of images to compute the similarities.

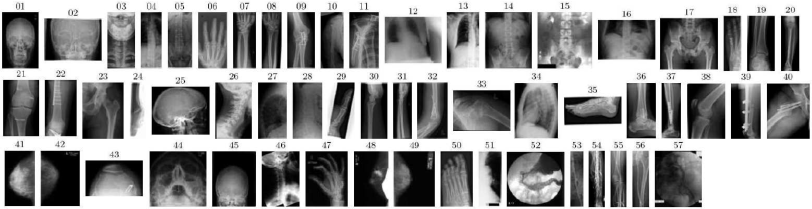
Distributed retrieval (3/3)

4. Each cooperating image server uses the received four-tuplets to compute the global similarity measure between the query image and its indexed images using Eq. (2), and sends back its top list of images with non-zero similarities to the central server as pairs $(I_R, k(I_Q, I_R))$;

5. The central server sends the top list of pairs $(I_R, k(I_Q, I_R))$ to the user, who can download the most similar images.

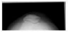
The procedure is strictly equivalent to using Eq. (2) in a non-distributed setting i.e. as if we were in a situation where all images were available at a single server.

IRMA (1/3): query $\longrightarrow$ top 10 retrieved images



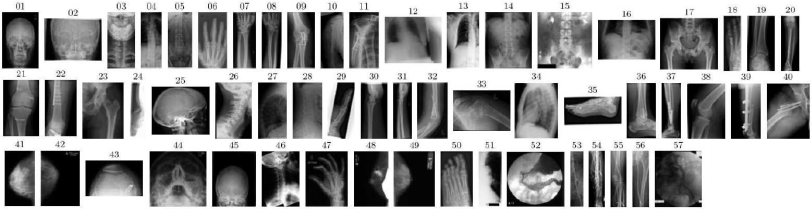
 $\longrightarrow$

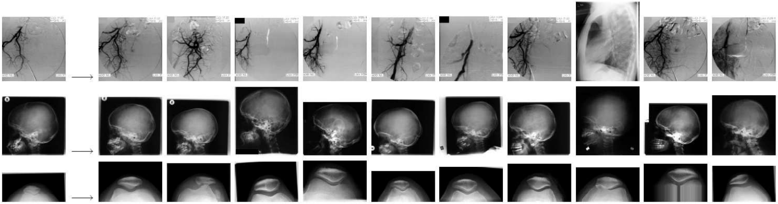
 $\longrightarrow$

 $\longrightarrow$

Université de Liège

IRMA (1/3): query ⟶ top 10 retrieved images





Université de Liège

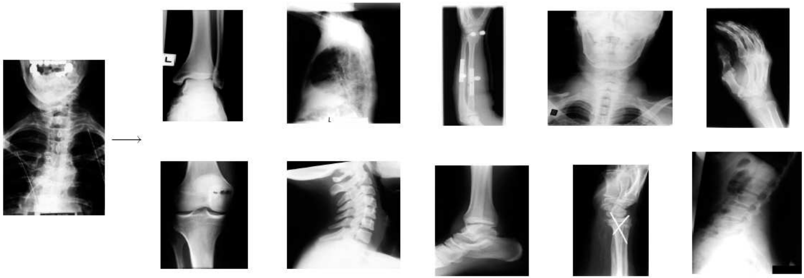
IRMA (2/3): query $\longrightarrow$ top 10 retrieved images

Not so nice results...

 $\longrightarrow$

IRMA (2/3): query $\longrightarrow$ top 10 retrieved images

Not so nice results...



IRMA (3/3): quantitative results

10000 images (approx. 512×512) in 57 classes

- Protocol [ImageCLEF 2005]
  - 9000 *unlabeled* reference images
  - 1000 *labeled* test images
  - **Recognition rate** of the first ranked image
- Results

MIR2010	naïve	NN	ACCV 2007	KDGN07
81.6%	29.7%	63.2%	85.4%	87.4%

(with 10 vectors, $m = 40$ tests, 1000 patches per image)

Université de Liège

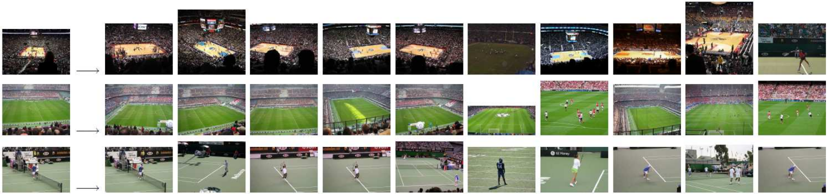
SPORTS (1/3): query ⟶ top 10 retrieved images

 ⟶

 ⟶

 ⟶

SPORTS (1/3): query $\longrightarrow$ top 10 retrieved images



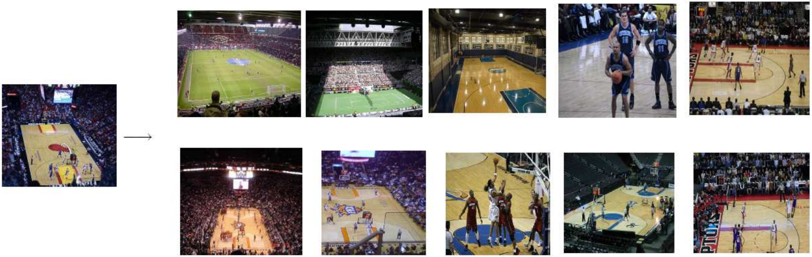
SPORTS (2/3): query $\longrightarrow$ top 10 retrieved images

Not so nice results...

 $\longrightarrow$

SPORTS (2/3): query $\longrightarrow$ top 10 retrieved images

Not so nice results...



SPORTS (3/3): quantitative results

2449 images in 5 classes (baseball, basketball, football, soccer, and tennis)

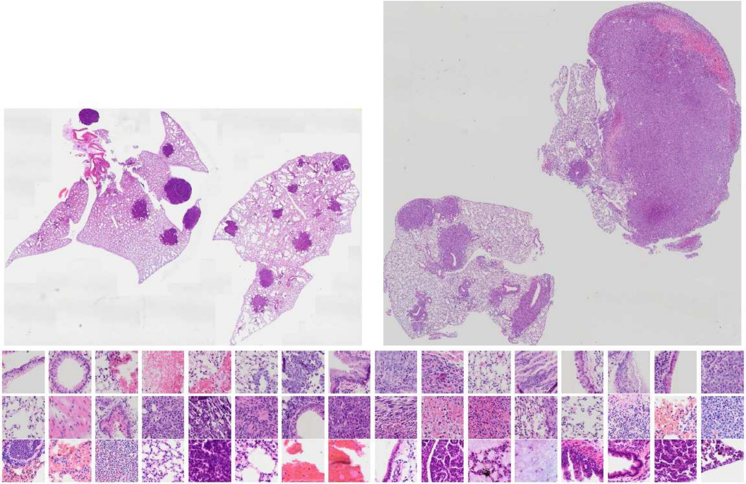
- Protocol [Jain et al., CVPR 2008]
  - 75% *unlabeled* reference images
  - 25% *labeled* test images
  - **Recognition rate** of the first ranked image
- Results

MIR2010	JSL08
71.02 %	41.56% to 65.28%

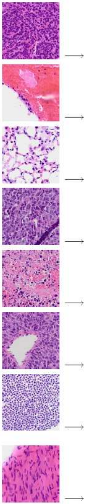
(with 10 vectors, $m = 40$ tests, 1000 patches per image)

PATHO (1/2): whole-slide histology images

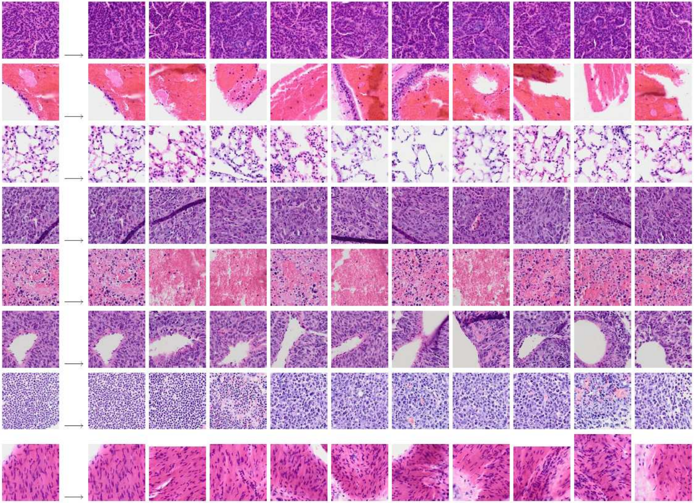
8 whole-slide images (approx. 20000×20000), 53000 tiles (256×256)



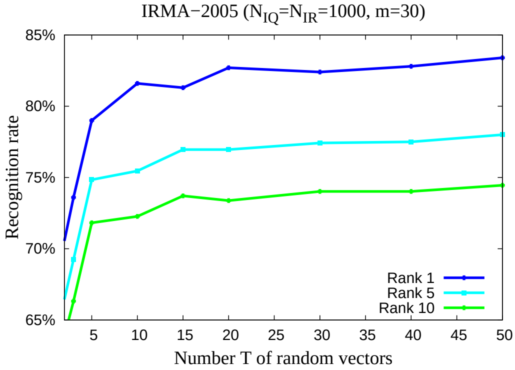
PATHO (2/2): query $\longrightarrow$ top 10 retrieved images



PATHO (2/2): query $\longrightarrow$ top 10 retrieved images

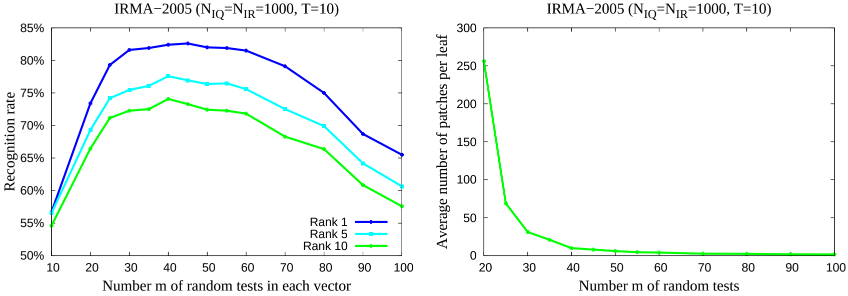


Influence of the number T of vectors



Recognition rate up to rank 10 on IRMA-2005.

Influence of the number *m* of random tests



Recognition rate and average number of patches per visual word.

Summary

- Bag-of-visual-words approaches were not originally designed for incremental image indexing and distributed search therefore limiting their practical usefulness.
- We propose to use a data-independent visual vocabulary algorithm based on multiple vectors of random tests to map patches to visual words.
- Results using the exact same parameters are promising on three diverse, real-world, image sets, with distributed and incremental capabilities.

Université de Liège

Perspectives

- The approach opens the door for large-scale, collaborative, studies.
- We seek to apply our approach on very large-scale and very high-resolution biomedical imaging datasets where images are naturally distributed and incrementally added.
- Optimization of parameters and/or combination with other techniques should improve results for specific applications.
- Extensions to other multimedia sources such as audio and video data might be investigated.
- We plan to release an optimized Java implementation mid-2010.

Acknowledgments

- IRMA-2005 dataset courtesy of T. Deselaers and T.M. Lehmann, Dept. of Medical Informatics, RWTH Aachen, Germany. SPORTS dataset courtesy of V. Jain, University of Massachusetts Amherst, USA. Histopathology images courtesy of D. Cataldo and N. Rocks, GIGA-Research, University of Liège, Belgium.
- Trees/Vectors figures by Vincent Botta
- Funding: Walloon Region, European Regional Development Fund, National Fund for Scientific Research, IAP BioMagnet