Incremental Indexing and Distributed Image Search using Shared Randomized Vocabularies

Raphaël Marée
GIGA Bioinformatics,
University of Liège, Belgium
raphael.maree@ulg.ac.be

Philippe Denis
University of Liège, Belgium
philippe.denis@student.ulg.ac.be

Louis Wehenkel
Systems and Modeling,
University of Liège, Belgium
l.wehenkel@.ulg.ac.be

Pierre Geurts
Systems and Modeling,
University of Liège, Belgium
p.geurts@ulg.ac.be

ABSTRACT

We present a cooperative framework for content-based image retrieval for the realistic setting where images are distributed across multiple cooperating servers. The proposed method is in line with bag-of-features approaches but uses fully data-independent, randomized structures, shared by the cooperating servers, to map image features to common visual words. A coherent, global image similarity measure (which is a kernel) is computed in a distributed fashion over visual words, by only requiring a small amount of data transfers between nodes. Our experiments on various image types show that this framework is a very promising step towards large-scale, distributed content-based image retrieval.

Categories and Subject Descriptors

H.3 [**Information Systems**]: Information Storage and Retrieval; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis

General Terms

Algorithms

Keywords

Content-based image retrieval, distributed, incremental, randomized algorithms, subwindows

1. INTRODUCTION

1.1 Context

Visual image search or content-based image retrieval (CBIR) [5] aims at retrieving a ranked list of similar images to a given query image, based on the visual content of these images.

In the Internet era, images and other multimedia data are de facto distributed across multiple servers (such as on photo-sharing websites or in institutional archives). Also, previous to these newly introduced web services, image repositories have always been distributed geographically, for example different hospitals host different image databases. Even within a single, large, institution or company, images can be distributed across different computing nodes to overcome storage capacity and memory limitations. Additionally, images are generated in an on-line manner, *e.g.* tens of thousands of radiographies are produced per day in medium-sized city hospitals, and terabytes of visual data are being uploaded to popular websites every day.[1]

In this context, incremental image indexing and distributed image search are needed at the web scale, as well as within networks of specialized image repositories. For example, it is expected that the medical diagnosis field would benefit from distributed CBIR across networks of biomedical centers [9, 21]. Generally speaking, for users, the ideal situation would be a single entry point to initiate an image search, without the need to use a specific search engine for every existing image databases. On the image repository side, the ideal situation would allow each administrator to make its images available for external image searches, but still keep control of the image indexing structure locally with the possibility to update it in real-time and distribute the data across computing nodes, and that external image search requests do not require heavy computations, because local computing resources might be limited (*e.g.* in small institutions, peer-to-peer personal networks, or sensor networks). Importantly, the amount of data that has to be transmitted across the network should be as limited as possible.

1.2 State-of-the-art approaches

In image classification and retrieval, bag-of-features approaches [4, 16, 28] have recently become popular due to their conceptual simplicity and robust recognition performances [38]. They are based on the extraction of local features (or

[1] According to [17], the universe of still (film and digital) photographs existing in 2002 was about 900 billion which would roughly translate to about 4.5 exabytes of data. According to various press releases late 2008, Flickr, Photobucket and Facebook hitted 3, 6 and 10 billion pictures respectively. Also, about 850 million photos are uploaded to Facebook each month, *i.e.* a new image every 3 ms.

patches) in images [31], their description by fixed-size feature vectors [20, 33], the construction of a “visual vocabulary” (composed of “visual words”), and image similarity computations based on the occurences of these visual words in the images, inspired by text retrieval approaches [1, 28].

In contrast to the text domain where bag-of-words approaches rely on fixed vocabularies (they consist in language-specific words), there is no pre-defined vocabulary for images. In practice, the so-called visual vocabularies are thus usually built using unsupervised clustering techniques (typically k-means) on training sets of local features and visual words are defined as the cluster centers. Recent works have tried to optimize this step because of its computational complexity that does not allow to handle millions of local features efficiently. Techniques such as approximate k-means (using randomized kd-trees at each k-means step) [25] and hierachical k-means [23] are different ways to address this computational issue. Other works in image classification (*e.g.* [15, 22]) use labeled images to learn a discriminative visual dictionary through supervised learning, but we want here to be able to exploit a distributed environment with unlabeled data, which is more plentiful [12].

While these methods yield state-of-the-art results on medium-sized datasets on a single computer (up to hundreds of thousands of images), they were not designed for incremental and distributed environments. A foreseeable problem when transposed to such settings is that the structures implemented by each local server, i.e. its visual dictionary and its similarity measures between images, strongly depend on the locally available data which offers only a limited view, at a given time, of the rapidly evolving “visual world” available at the global scale. Therefore, the image similarities induced by a local server that uses such data-dependent mapping structures are not directly comparable to the similarities of other servers, and not equivalent to the case where all data are available at a single point. Hence, an additional filtering or re-ranking step would be needed at a central location in order to get reliable results. For example, simple pathological cases include the situation where a local server does not have any relevant images to the query and/or an unbalanced total number of images compared to other servers, two scenarios that might downweight/upweight the similarity measure in an unwanted fashion. Furthermore, these local data-driven structures are not designed with the possibility to be easily updated as the size of their visual dictionary (the number of clusters) is fixed a priori. Adaptive schemes [19, 37, 36] have been suggested but these concretely require the update, rebuilding, and/or transfer of the structures across the network from time to time, and the permanent storage or re-extraction of the local features used for their construction.

More recently, several approaches have reconsidered the use of simple methods (such as nearest neighbors) and/or global “gist” descriptors on tiny (32×32) images [8, 29, 30] to reduce computational and memory requirements and therefore be able to exploit very large amounts of unlabeled data on a single computer (indexing up to 110 million images). Interesting results were there obtained for near-duplicate detection or recognition of the most represented classes (such as faces). However, bag-of-features approaches yield significantly better results compared to approaches using global descriptors for object/location recognition [8]. Still, these approaches currently require more ressources and they are not well suited for an incremental and distributed setting, as mentionned above.

1.3 Method rationale and related work

In this paper we propose a bag-of-features framework for large-scale content based image retrieval, in a context where the images are distributed across multiple servers. In this framework, new images as well as additional image servers may be seamlessly added within the distributed repository. Our approach is totally unsupervised and essentially free of any manual tuning. Similarities among images are computed by letting each image server exploit a common standardized catalog of visual words, which allows to align its similarity measure with those of the other servers.

Our framework is based on the following key components:

- A central server (that might be the user client) that is aware of and communicates with a network of cooperating image servers; each image server stores and indexes a part of the complete dataset of images;

- A mapping structure that uses multiple vectors of tests on the values of individual pixels of an image patch, so as to assign multiple visual words to each patch of a given image. The exact same mapping structure is deployed on all servers;

- A global ranking of images based on a similarity measure computed and averaged locally by each server over visual words, but as if we were in a situation where each local server would be aware of the complete dataset of images.

Instead of using data-dependent approaches to locally build bag-of-features image representations, our framework uses a common mapping that is built once and for all by selecting a very large number of visual words, each one being defined by a fixed number of binary tests on the pixel values and being chosen in a random, data-independent fashion.

Our work is inspired by the randomized algorithms that were proposed recently for diverse purposes. Extremely and totally randomized trees [10] were notably used for supervised image classification and unsupervised image retrieval [18, 22, 19]. Random ferns or randomized lists are vectors of random tests used to store class posterior probabilities for object tracking applications and image classification in a supervised setting [24, 34, 3]. Random projections in locality sensitive hashing [11] were used to map data points into buckets where linear search is performed to find approximate nearest points (so all local features need to be explicitely stored). Random hyperplane hashing was used in [27] to construct binary features, sent by local servers to a central server, and used as input of a linear classifier for text classification. Random features are used by [26] for large scale supervised learning problems.

Also of relevance, though not randomized and applied for image (pixelwise) classification, is the vector quantization method with a regular lattice [32]. It discretizes the feature space of image patches by performing 4 data-independent, fixed, subdivisions for every 128 SIFT-like descriptor dimensions, giving a potentially huge visual vocabulary of 4^{128} bins. The success of such a simple discretization scheme is probably due to their additional class-specific discriminative bin selection step, and certainly to their very dense, expensive, local feature sampling as about 1.5 million features were extracted from each 640 × 480 image.

Instead, we build many mapping structures, and for each one we randomly select a subset of dimensions and subdivision thresholds to map image patches to visual words. Because these structures are data-independent and identical across the cooperating servers, it is possible to index new images incrementally, and search images by computing in a distributed fashion a coherent, global similarity measure between distributed images ie. without the need of a reranking step. The work is achieved by each server by averaging patch frequencies over non-empty visual words found in its indexing structure for the query image, inspired by [19] that computes a similarity measure over leaves of ensembles of totally randomized trees, a measure reminiscent of tf-idf in information retrieval [1, 28].

The remainder of the paper proceeds as follows. We describe the method steps in Section 2, show experimental results in Section 3 and discuss computational requirements. We discuss general properties of the method and suggest future work in Section 4. Finally, we conclude.

2. METHOD

In this section, we describe our indexing structures, the image similarity measure exploiting these structures, and the distributed computation framework we propose.

2.1 From randomized trees to shared randomized visual vocabularies

Our approach is derived from [19], where an ensemble of totally randomized trees is grown to define the visual vocabulary, from a sample of image subwindows of random sizes extracted at random locations within images of the reference dataset. Subwindows are rescaled to a 16 × 16 patch and then represented by their vector of attributes derived from its pixel values (768 HSV color components, or 256 gray levels). In this method, the binary tests chosen to split tree nodes are randomized but they still depend on the original dataset, and also the depth of the leaves (i.e. the number of binary tests defining a particular visual word) is data-dependent.

To get rid of these data-dependencies, we made the two following adaptations to this method:

- In our method, the tree depth is uniform and fixed a priori, independently of the dataset; note that the tree depth is equal to the number of binary tests defining a visual word (or a leaf of the tree).
- To select a binary tree t of depth m, a vector V_t composed of m binary tests $(test_1(t), ..., test_m(t))$ is generated randomly, where each test $test_i(t) \equiv 1(x_{j_i} < th_i)$ compares a randomly chosen attribute x_{j_i} (among those describing the patches) to a randomly chosen threshold th_i (in the range of possible values of that attribute). These tests are then attached to the internal nodes of the tree, by replicating at each internal node at level i ($i = 1$ corresponding to the root) the $test_i$. This means that instead of 2^{m-1} different tests, the tree structure in our method is actually defined by only m tests, which allows us to exploit trees of much larger depth than the method of [19].

As in [19], we use an ensemble of T trees (but here all of the same depth), or equivalently an ensemble of T vectors V_t $(t = 1 \dots T)$ each one composed of m binary tests. According to these tests, each patch is mapped by each V_t 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. Each pair (B, t) identifies a "visual word" (or leaf) among the $T2^m$ ones induced by an ensemble of size T, a potentially huge vocabulary for large m (e.g. values of $T = 50$ and $m = 50$ are used in some of our experiments).

2.2 Local image indexing

All servers use the same mapping structure to index their local subset of images. To this end, each server populates the vectors V_t $(t = 1 \dots T)$ locally and incrementally with its own images in the following fashion. From each new reference image I_R stored in the local dataset, N_{I_R} subwindows of random sizes are extracted at random locations, then resized to a patch of fixed size of 16 × 16 pixels and encoded by its raw pixel values.[2] Each patch is then mapped by each test vector V_t to a visual word B of m bits, which is indexed through a hash table or inverted index file for future constant time access. Initially empty, the local index will be progressively populated, with non-empty visual words. For each t and each non-empty word B, we maintain a (sparse) list of pairs composed of local image identifiers I_R and a count of the number $N_{I_R,B,t}$ of patches of I_R mapped by V_t to that visual word, as well as the total count $N_{Blocal,t}$ of patches of the local image set mapped to this visual word. Notice that indexing a new image is $O(TN_{I_R}m)$, *i.e.* independent of the number of reference images stored in the local dataset.

2.3 Deriving image similarities

Our similarity measure between images is a straightforward adaptation of the similarity measure derived in [19] from ensembles of trees. We first define the notion of patch similarity and then derive an efficient computation of the image similarity as the average of the similarities of patches extracted from them.

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 patches from the global dataset that are mapped to the visual word B by V_t. For a list of T such vectors, the aggregated similarity between two patches is then obtained by:

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

Intuitively, this measure says that two patches are similar if they share many visual words and that they are more similar if they share visual words that are less frequently present in the reference images. Main theoretical properties of this kernel are given in Appendix A.[3]

[2] Any other extraction/description scheme could be used but we favor the former method because of its simplicity and excellent performances in a large range of conditions [18].

[3] As suggested by our notation, both the patch similarity measure and the induced image similarity measure are indeed (semi)positive-definite kernels [19].

Given a query image I_Q and a reference image I_R, their similarity is then defined, according to [19], as the average similarity between all pairs of their patches:

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

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 sets $S(I_Q)$ and $S(I_R)$ are in practice of very large size (on the order of $(w \times h)^2$ where w (resp. h) denotes the width (resp. height) of the original images). Hence, we estimate expression (2) by Monte-Carlo, by sampling a finite number of patches from each image. Denoting by N_{I_Q} and N_{I_R} the number of patches sampled respectively from I_Q and I_R, we show in Appendix B that the finite sample version of (2) may be rewritten as:

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

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 reference 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.

In our method, the sampling of patches from the reference images is carried out once and for all and locally, when they are incorporated into the local indexes. The patches from the query image are, on the other hand, sampled on the fly when the query is issued. In the next section, we show how the similarity may be computed in a distributed fashion over all images of the global dataset, while exploiting the locally maintained index structures and using minimal information exchange among severs.

2.4 Distributed image search

Assuming the user's computer is fast enough, it can process the image query locally ie. extract, describe, and map patches to visual words using the common mapping structures. We note here that each patch can be processed independently by each vector of random tests, making the method well suited for massively parallel architectures such as graphical processor units. It is therefore reasonable to think that a user computer is able to process the query image locally, and only sends the non-empty visual word identifiers and frequency counts to the central server. This process is expected to be faster than sending the whole image across the network (see Section 4). More precisely, the image query I_Q is thus 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. Then,

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 $(B, t, \frac{1}{N_{B,t}}, \frac{N_{I_Q,B,t}}{N_{I_Q}})$;
4. Each image server uses the received four-tuplets to compute the global similarity measure between the query image and its reference images using Eq. (3), 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.

To sum up, the procedure is strictly equivalent to using Eq. (3) in a non-distributed setting.

3. EXPERIMENTAL RESULTS

The long-term aim of image search is to be applicable to any type of images, a problem that remains largely unsolved. With this generic goal in mind, we perform experiments on three very different image types and we use the exact same parameter values: We build $T = 10$ vectors with $m = 30$ random tests, and we exract $N_{IQ} = N_{IR} = 1000$ patches in each image. In order to compare our results to other works, we compute the classification accuracy of the first retrieved images for each query image.

3.1 IRMA-2005

IRMA-2005 dataset [7] contains 10000 X-ray images grouped into 57 classes that depict human body regions under different orientations. We used 9000 reference images (hence a total of 9 millions reference patches), and 1000 query images, like other works. About 40 methods were evaluated on this dataset with results ranging from 26.7% to 87.4% [7]. We obtained 81.6% recognition rate. According to results reported in [19], our method with its distributed and incremental capabilities are thus inferior to the use of totally randomized trees (85.4%) but that approach requires that a single server holds the entire dataset of images and stores local features to be able to update its structures as new images comes in. Our results are also inferior to the best published result obtained by using a nearest-neighbor classifier based on a distance taking into account local image distortions (87.4%), but significantly better than using euclidian distance computed on downscaled 32 × 32 images (63.2%). We also compute recognition accuracy up to rank 5 (75.46%) and 10 (72.27%). It means on average 75.46% of the first 5 images retrieved for each query are of the correct class. Figure 3 illustrates successful retrieval results for several query images.

3.2 SPORTS

The Sports collection [14] contains 2449 photographs grouped into five classes (baseball, basketball, football, soccer, and tennis). We used 75% of the images as the reference set and the remaining 25% for the query test images, similarly to [14]. We obtain 71.02% (averaged accuracy per class). It is better than classification results reported in [14] that were obtained by variants of supervised approaches exploiting labeled training images, and that ranged from 41.56% (using a linear SVM built on top of a bag-of-words image representation generated using k-means from local features extracted by DoG, MSER, and affine-Harris, and described by SIFT), to 65.28% (using domain-specific features and Selective Hidden Random Fields). We also compute recognition accuracy up to rank 10 (62.25%) and rank 20 (59.89%). Figure 4 illustrates top-10 retrieval results for several query images.

3.3 HISTOPATHOLOGY

Our third dataset consists in whole-slide histopathology images. In the biomedical field, recent advances in digital scanning systems allows to digitize the slide of a tissue into a high resolution image within a few minutes. The potential number of such images, their sizes, their availability accross different hospitals, and frequent additions (as new tissues are scanned) makes efficient, distributed and incremental CBIR a strong need to help researchers and pathologists to explore their data and support their findings or diagnosis. For example, such an approach can help to interpret a new patient case by finding older cases which contain visual patterns similar to a user-selected region of interest. To illustrate the potential of our approach in this context, we used 8 whole-slide images of experimental lungs generated for a cancer project. Each whole-slide image has an average size of roughly 20000×20000 pixels. We divided off-line these large images into smaller 256×256 pixels tiles and indexed all these tiles (roughly about 53000 images) using the same parameter values than for other datasets except that we extract only $N_{IQ} = N_{IR} = 500$ patches in each tile. We picked several tiles representative of different tissue types as queries. Because ground-truth is not available for such a large amount of data, we only show qualitative results in Figure 5. This figure shows the method is able to retrieve similar images using color information, texture information (such as the repetition of elongated cells or dark round ones), and also the global shapes of image queries. We hypothesize this is a consequence of the randomization of the sizes of the patches: small patches captures texture information while large patches capture global shape information.

3.4 Influence of parameters

Figure 1 shows the influence of the number of vectors, and the number of random tests in each vector, on IRMA-2005 database. As expected, increasing the number of random vectors improves recognition results for the first retrieved image, from 67.5% with $T = 1$ up to 83.4% with $T = 50$. Similar trends are observed at rank 5 and 10. The number of random tests m in each vector should be neither too small, nor too large. The best value appears to be 45 random tests (among $16 \times 16 = 256$ dimensions for our patches encoded by gray values) with 82.6% recognition rate at rank 1. As illustrated by Figure 2, low values of m yield a small number of visual words which are highly populated ($m = 10$ yields on average 807 non-empty visual words per vector with 11151 patches each), so they are not distinctive enough. On the opposite, higher values of m yield a higher number of non-empty visual words but with only a few patches ($m = 100$ yields on average 5212007 non-empty visual words per vector with 1.76 patches each). With such detailed vectors, a high proportion of patches of query images fall into empty visual words so these patches do not contribute to image similarity computations (on average 55% of patches from all test images are mapped to empty visual words with $m = 100$ while it is about 0.01% with $m = 10$).

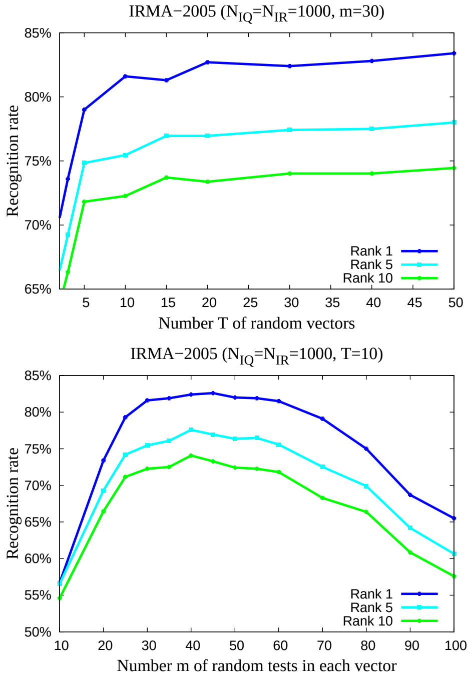

Figure 1: Influence of the number T of random vectors and m of random tests on the recognition rate up to rank 10 on IRMA-2005.

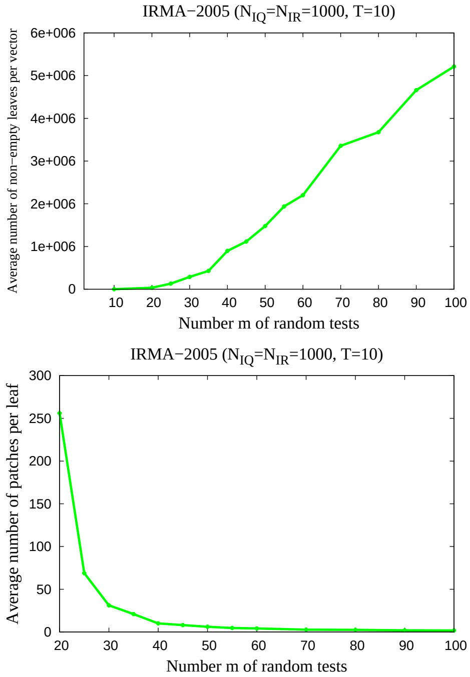

Figure 2: Influence of the number of random tests on the visual vocabulary on IRMA-2005.

3.5 Computational and memory requirements

The method provides various ways to meet specific computational requirements. Because of the sparsity of the bag-of-words representation, the list $\mathcal{B}$ sent by the user for a query image contains much less elements than the potential size of the visual vocabulary $T2^m$, a maximum of TN_{I_Q} identifiers and frequency counts being sent by the user over the network in the pessimistic case where all patches are mapped to distinct visual words. Assuming we use 8 bytes to identify a visual word (unsigned long integer), 4 bytes for a local image identifier (unsigned integer), 4 bytes for frequency counts and similarities (float), the maximum data transfered during the distributed search is $TN_{I_Q}(8+4) + TN_{I_Q}S(3 \times 8 + 3 \times 4) + SK(2 \times 4) + K \times 4$ bytes, where S is the number of local servers, and K the number of desired retrieved images. For example, if $T = 10$, $N_{I_Q} = 1000$ and the user wants to retrieve the top $K = 10$ images, the client emits 120 kbytes and the total network trafic between the central and local servers is $S \times 360$ kbytes which can be further reduced downto $S \times 200$ kbytes if servers cache visual words identifiers. We note that T and N_{I_Q} can be adjusted if speed and data transfers are a primary concern, but at the risk of less relevant results. Another way to reduce data transfers is to neglect the $N_{B,t}$ term of Eq. (3) so that the similary measure is equivalent to a simple counting measure that does not require data transfers to gather the global $N_{B,t}$. However, our recognition results were less good using this variant. We also note that following step 3 of the distributed search, the central server might implement stop lists [1] to avoid data transfers and computations based on widely populated visual words ($N_{B,t} >>$). Indeed, the influence of such visual words on the similarity measure should be limited due to the $\frac{1}{N_{B,t}}$ term.

Overall, the size of the mapping structures is small (with Tm tests), each patch can be processed in parallel, and the whole set of patches do not need to be stored in contrast to approaches that need to re-construct or adapt their structures incrementally. However, like with other bag-of-features approaches, the critical point is the size of the inverted indexes (implemented as hash tables) that can rapidly exceed the amount of RAM of a single server. The proportion of non-empty visual words and their average number of patches essentially depend on the number of reference images, the amount of spatial redundancy in images, the patch extraction scheme, the patch descriptor, and the number of random tests m (higher numbers of tests typically yield more visual words with less patches). As already mentionned, Figure 2 illustrates the influence of the number of random tests on the average number of non-empty visual words and the average number of patches in each of them. On IRMA-2005, we obtained with $m = 30$ on average for each vector V_t less than 0.03% non-empty visual words (288292 among 2^{30}) where each non-empty visual word indexes on average 31.22 patches. On SPORTS it was less than 0.04% (415975) with 4.41 patches each, and on HISTOPATHO it was less than 0.05% (535287) with 49.80 patches each. Assuming an average of 4 patches per visual word, $T = 10$, $N_{IR} = 1000$, and 10000 reference images, we roughly need 200 Mbytes to store visual word keys (a total of 25 million unsigned long integers), and 400 Mbytes to reference image identifiers (100 million unsigned integers). To reduce memory requirements, tailored compression mechanisms with efficient encoding/decoding operations could be investigated [35]. Our framework could be seen as a complementary way to address that problem by distributing images across multiple computing nodes and implementing locally populated indexes.

4. DISCUSSION

An essential property of the proposed framework is that the query image is neither propagated, nor processed by any server, since only visual word identifiers, image identifiers, and feature counts are sent over the network. Thus only the mapping structures (i.e., an initial random seed, the numbers T of vectors and m of tests per vector) need to be shared. Each cooperating server is autonomous and only effectively indexes its own images. Moreover, a cooperating server can be easily added/removed within this framework without any disruption. Once a new local server is added to the distributed framework, subsequent image search results will instantly be based on image similarities that take into account new images brought by the new server.

In our experiments, our purpose was to show the general performances of the approach, rather than trying to optimize its results on each dataset. Our experiments show that such a generic and simple bag-of-features approach based on raw pixel values of randomly extracted patches and data-independent mapping structures yields interesting results on real-world images while being straightforward to implement, so it should be of great interest for practitioners. In particular, we believe the incremental and distributed capabilities of our method makes it a good candidate for very large image retrieval studies. Future research work should then regard more extensive experiments to face the huge "visual world" reality and take advantage of large, recently available image datasets [6, 29, 2]. Somehow similarly, other recent works [13, 29] have shown that simple methods can work reasonably well for challenging computer vision tasks given that very large collections of images are available.

In practice, if the objective is to obtain the best results within a network of specialized images, one could try to optimize parameters (such as the numbers and sizes of patches, the number of random tests, the number of vectors, etc.). The approach might also benefit from other ideas that could be combined to refine retrieval results: computations of diverse image descriptors (such as local color invariant features [33] or application-specific features), matching strategies that encode spatial information or perform verification of spatial consistency (if global image geometry is relevant for the problem at hand), integration of other types of data if available (textual tags, gps coordinates, ...), and relevance feedback mechanisms (one can imagine to update image similarities based on an adaptation of Eq. (3) where more weights are given to visual words corresponding to user-selected relevant images).

5. CONCLUSIONS

In recent years, approaches based on bag-of-features image representations yielded state-of-the-art results for content-based image retrieval but they were not originally designed for incremental image indexing and distributed search therefore limiting their practical usefulness.

In this paper, we have shown that this family of methods can be adapted for real-world settings through lightweight, fast, randomized mapping structures, and simple exchange mechanisms between cooperative servers. We hope these technical ideas and our promising results on real-world images will foster research in large-scale image search.

Finally, we seek to apply our approach on very large-scale and very high-resolution biomedical imaging datasets. Applications with other multimedia sources such as audio and video data are also possible.

6. ACKNOWLEDGMENTS

IRMA-2005 data 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. RM is supported by the GIGA (University of Liège) with the help of the Walloon Region and the European Regional Development Fund. PG is Research Associate of the F.R.S.-FNRS. This paper presents research results of the Belgian Network BIOMAGNET (Bioinformatics and Modeling: from Genomes to Networks), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. This work is supported by the European Network of Excellence, PASCAL2. The scientific responsibility rests with the authors.

APPENDIX

A. PATCH KERNEL PROPERTIES

Forgetting the normalization by $N_{B,t}$ in (1) and assuming, without loss of generality, that all attributes are scaled in $[0, 1]$, we show here that:

$$k_\infty(s_1, s_2) = \lim_{T\to\infty} k_T(s_1, s_2) = (1 - \frac{1}{n}||x(s_1) - x(s_2)||_1)^m, \tag{4}$$

where n is the number of attributes describing the patches, $x(s_1)$ and $x(s_2)$ are the attribute vectors (in our case gray levels or HSV values) corresponding to patches s_1 and s_2, m is the number of tests in each vector, and $||.||_1$ is the L^1 norm. Indeed, the quantity

$$k_\infty(s_1, s_2) = \lim_{T\to\infty} k_T(s_1, s_2) \tag{5}$$

represents the probability that the two subwindows are mapped into the same word by a random vector of m tests, each of the form $x_i < th$ that compares the value of an attribute x_i (randomly selected among n) to a threshold th (randomly choosen in the interval of variation of that attribute). Denoting by $x_i(s), i = 1 \dots n$, the value of the ith attribute for the patch s and assuming without loss of generality that the range of variation of these attributes is $[0, 1]$, the probability that a single random test on the ith attribute will not separate two patches s_1 and s_2 is $1 - |x_i(s_1) - x_i(s_2)|$, i.e. the probability to select a threshold outside the interval $]\min(x_i(s_1), x_i(s_2)), \max(x_i(s_1), x_i(s_2))]$. When $m = 1$, we thus have:

$$\begin{aligned} k_\infty(s_1, s_2) &= 1 - \sum_{i=1}^{n} \frac{1}{n}|x_i(s_1) - x_i(s_2)| \\ &= 1 - \frac{1}{n}||x(s_1) - x(s_2)||_1, \end{aligned}$$

since the probability to select a particular attribute is $1/n$.

All tests in a vector being selected independantly of the others, we have for m tests:

$$k_\infty(s_1, s_2) = \prod_{i=1}^{m}(1 - \frac{1}{n}||x(s_1) - x(s_2)||_1),$$

which proves (4).

Our indexing structure thus provides a finite sample approximation of (4) and the number of tests per vector, m, determines the spreading of the similarity. The introduction of the factor $N_{B,t}$ in (1) has the effect of deflating (inflating) the similarity in regions of the attribute spaces which are very much (very little) populated by patches from the reference image set. We noticed that this scaling significantly improved the results in our experiments.

B. IMAGE KERNEL COMPUTATION

We show here how to derive (3) from (2) in the finite sample case. Denoting by $\hat{S}(I_Q)$ ($\hat{S}(I_R)$) the set of N_{I_Q} (N_{I_R}) subwindows extracted from the image I_Q (I_R) to estimate (2), one gets:

$$k(I_Q, I_R) = \frac{1}{N_{I_Q} N_{I_R}} \sum_{s_Q \in \hat{S}(I_Q), s_R \in \hat{S}(I_R)} k_T(s_Q, s_R). \tag{6}$$

Writing $k_T(s_Q, s_R)$ as:

$$k_T(s_Q, s_R) = \frac{1}{T}\sum_{t=1}^{T}\sum_{B\in\mathcal{V}_t}\frac{1}{N_{B,t}}1(s_Q \in B)1(s_R \in B), \tag{7}$$

with $\mathcal{V}_t$ the set of visual words induced by the vector V_t, and plugging this expression in (6), one gets:

$$\begin{aligned} k(I_Q, I_R) &= \frac{1}{N_{I_Q} N_{I_R}} \sum_{s_Q, s_R} \frac{1}{T}\sum_{t=1}^{T}\sum_{B\in\mathcal{V}_t}\frac{1}{N_{B,t}}1(s_Q \in B)1(s_R \in B) \\ &= \frac{1}{T}\sum_{t=1}^{T}\sum_{B\in\mathcal{V}_t}\frac{1}{N_{B,t}}\frac{\sum_{s_Q} 1(s_Q \in B)}{N_{I_Q}}\frac{\sum_{s_R} 1(s_R \in B)}{N_{I_R}} \\ &= \sum_{t=1}^{T}\frac{1}{T}\sum_{B\in\mathcal{V}_{I_Q,t}}\frac{1}{N_{B,t}}\frac{N_{I_Q,B,t}}{N_{I_Q}}\frac{N_{I_R,B,t}}{N_{I_R}}, \end{aligned}$$

given the definitions of $\mathcal{V}_{I_Q,t}$, $N_{I_Q,B,t}$, and $N_{I_R,B,t}$ in (3).

C. REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, May 1999.

[2] P. Bolettieri, A. Esuli, F. Falchi, C. Lucchese, R. Perego, T. Piccioli, and F. Rabitti. CoPhIR: a test collection for content-based image retrieval. *CoRR*, abs/0905.4627v2, 2009.

[3] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *Proc. ICCV*, 2007.

[4] C. Dance, J. Willamowski, L. Fan, C. Bray, and G. Csurka. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[5] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 39(65), 2007.

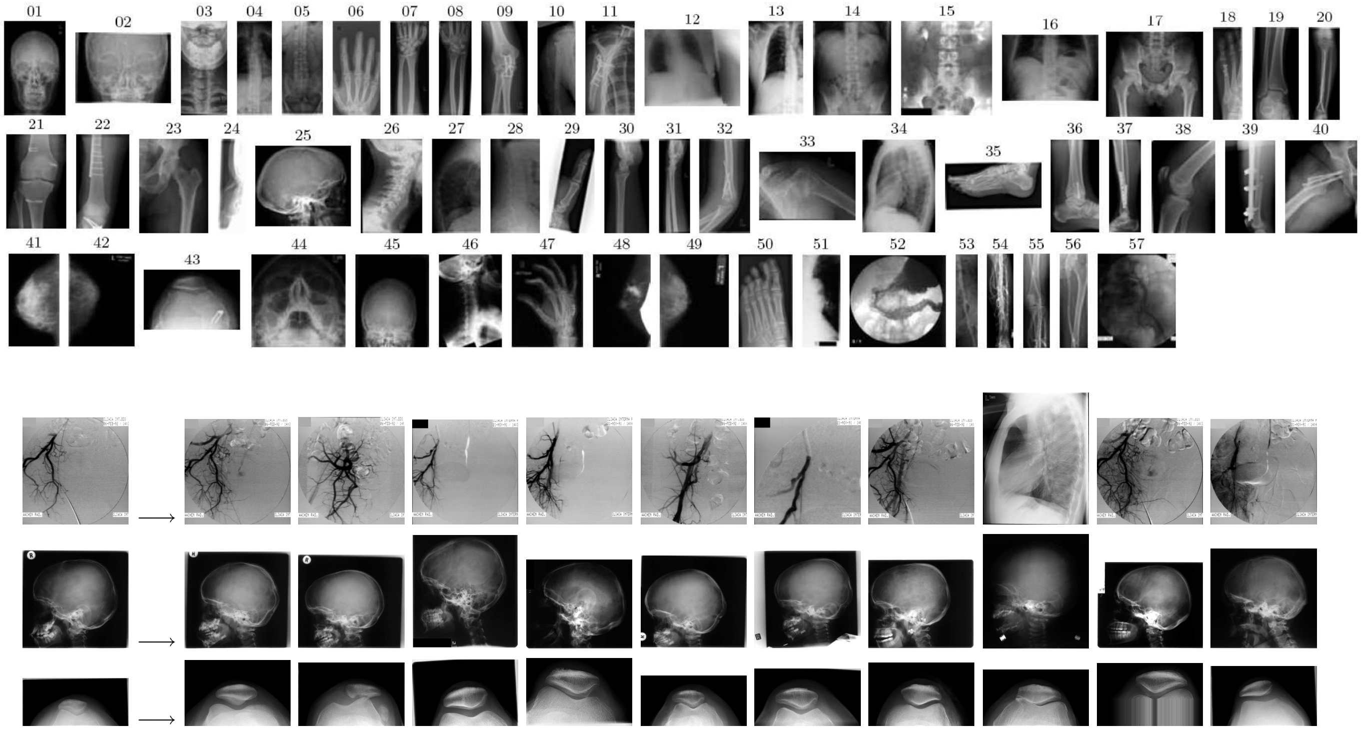

Figure 3: IRMA-2005. Top: Illustration of the database content with one image for each of the 57 classes. Bottom: Illustration of several retrieval results ranked according to their similarity to three query images (left).

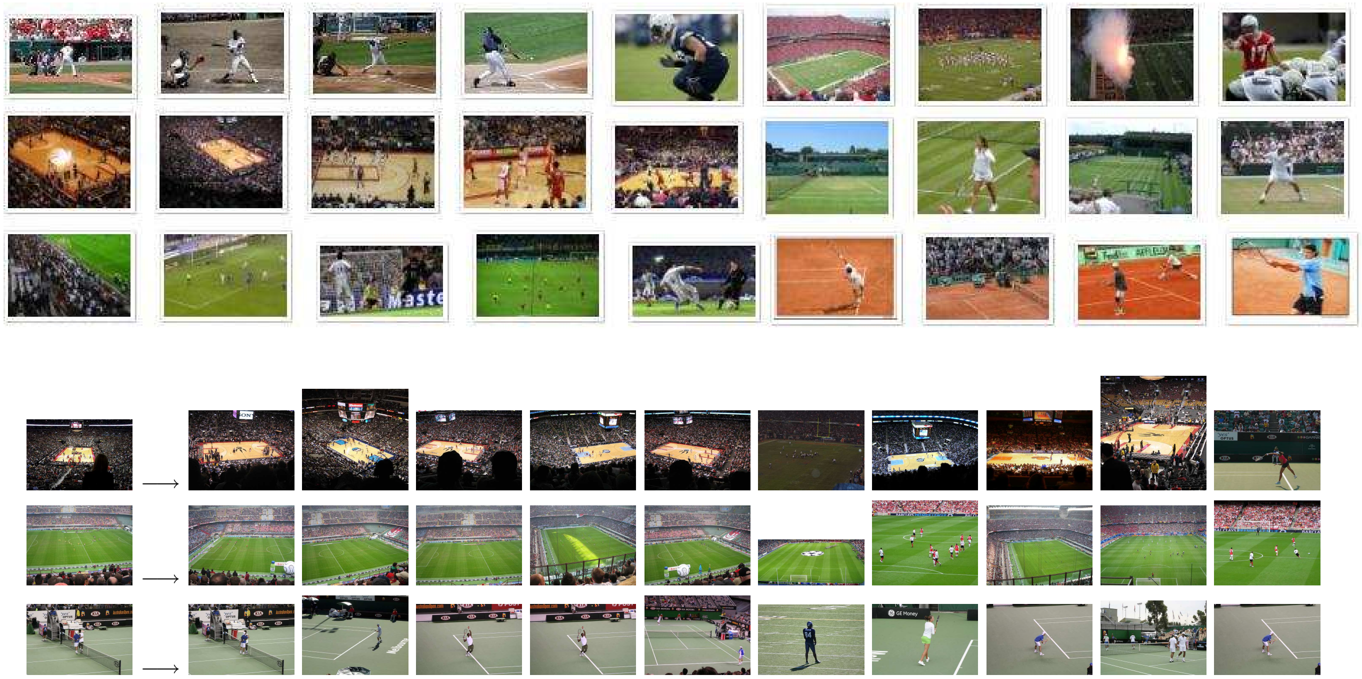

Figure 4: SPORTS. Top: Illustration of the database content with several images for each of the 5 classes. Bottom: Illustration of several retrieval results ranked according to their similarity to three query images (left).

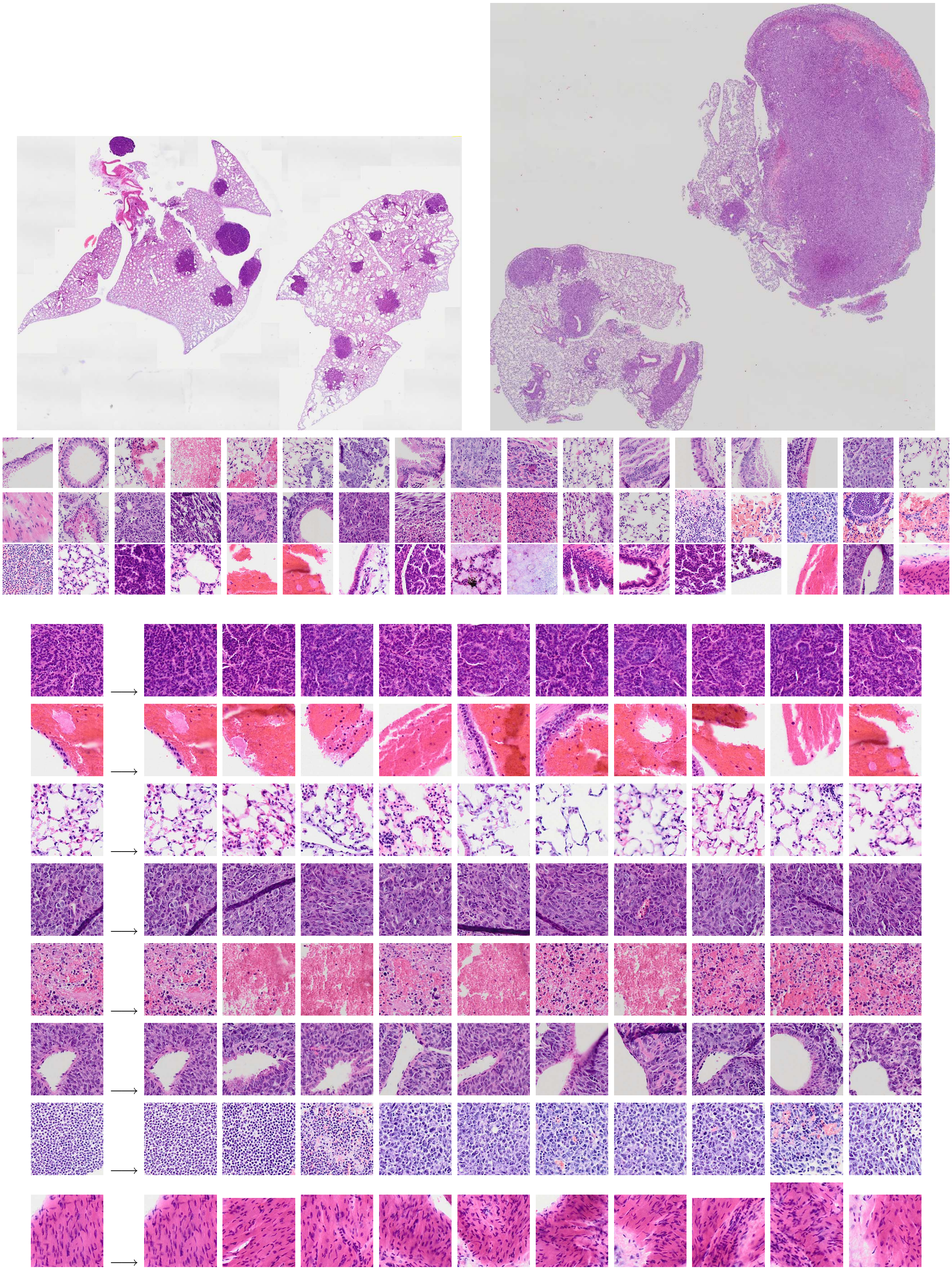

Figure 5: HISTOPATHO. Top: Illustration of two whole-slide images and several 256×256 tiles actually indexed. Bottom: Illustration of several retrieval results ranked according to their similarity to eight query image tiles (left).

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proc. CVPR*, 2009.

[7] T. Deselaers, H. Müller, P. Clogh, H. Ney, and T. M. Lehmann. The CLEF 2005 automatic medical image annotation task. *IJCV*, 74(1):51–58, 2007.

[8] M. Douze, H. Jégou, H. Singh, L. Amsaleg, and C. Schmid. Evaluation of gist descriptors for web-scale image search. In *International Conference on Image and Video Retrieval (CIVR)*. ACM, july 2009.

[9] W. G. Finn. Diagnostic pathology and laboratory medicine in the age of ”omics”. *Journal of Molecular Diagnostics*, 9(4), 2007.

[10] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 36(1):3–42, 2006.

[11] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proc. VLDB*, pages 518–529, 1999.

[12] A. Halevy, P. Norvig, and F. Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009.

[13] J. Hays and A. A. Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (SIGGRAPH 2007)*, 26(3), 2007.

[14] V. Jain, A. Singhal, and J. Luo. Selective hidden random fields: Exploiting domain specific saliency for event classification. In *Proc. CVPR*, 2008.

[15] S. Lazebnik and M. Raginsky. Supervised learning of quantizer codebooks by information loss minimization. *IEEE Transactions on PAMI*, 31(7):1294–1309, 2009.

[16] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 43(1):29–44, 2001.

[17] P. Lyman and H. R. Varian. How much information. Technical report, University of California at Berkeley, 2003. Retrieved from http://www.sims.berkeley.edu/how-much-info-2003 on 8th September 2009.

[18] R. Marée, P. Geurts, J. Piater, and L. Wehenkel. Random subwindows for robust image classification. In *Proc. IEEE CVPR*, volume 1, pages 34–40. IEEE, 2005.

[19] R. Marée, P. Geurts, and L. Wehenkel. Content-based image retrieval by indexing random subwindows with randomized trees. *IPSJ Transactions on Computer Vision and Applications*, 1(1):46–57, jan 2009.

[20] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on PAMI*, 27(10):1615–1630, 2005.

[21] H. Müller, A. Rosset, A. Garcia, J.-P. Vallée, and A. Geissbuhler. Benefits of content-based visual data access in radiology. *RadioGraphics*, 25:849–858, 2005.

[22] F. Moosmann, E. Nowak, and F. Jurie. Randomized clustering forests for image classification. *IEEE Transactions on PAMI*, 30(9):1632–1646, 2008.

[23] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *Proc. IEEE CVPR*, volume 2, pages 2161–2168, June 2006.

[24] M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *Proc. CVPR*, June 2007.

[25] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. CVPR*, 2007.

[26] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Neural Information Processing Systems (NIPS)*, 2007.

[27] S. Rajaram and M. Scholz. Client-friendly classification over random hyperplane hashes. In *Proc. ECML/PKDD (2)*, pages 250–265, 2008.

[28] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, volume 2, pages 1470–1477, Oct. 2003.

[29] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on PAMI*, 30(11):1958–1970, 2008.

[30] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. *Proc. CVPR*, 2008.

[31] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2008.

[32] T. Tuytelaars and C. Schmid. Vector quantizing feature space with a regular lattice. In *Proc. ICCV*, oct 2007.

[33] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluation of color descriptors for object and scene recognition. In *Proc. CVPR*, 2008.

[34] B. Williams, G. Klein, and I. Reid. Real-time SLAM relocalisation. In *Proc. ICCV*, 2007.

[35] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishers, 1999.

[36] T. Yan, D. Ganesan, and R. Manmatha. Distributed image search in camera sensor networks. In *Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems*, 2008.

[37] T. Yeh, J. Lee, and T. Darrell. Adaptive vocabulary forests br dynamic indexing and category learning. In *Proc. ICCV*, 2007.

[38] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *IJCV*, 73(2):213–238, jun 2007.