



**HAL**  
open science

# Large-scale Content-based Visual Information Retrieval

Alexis Joly

► **To cite this version:**

Alexis Joly. Large-scale Content-based Visual Information Retrieval. Computer Science [cs]. Université de Montpellier, 2015. hal-01182797

**HAL Id: hal-01182797**

**<https://inria.hal.science/hal-01182797>**

Submitted on 3 Aug 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ MONTPELLIER 2  
Faculté des sciences et techniques de Montpellier

# H D R   T H E S I S

## Large-scale Content-based Visual Information Retrieval

*Submitted for the degree of “Habilitation a Diriger des Recherches”  
of the University Montpellier 2  
Speciality: Computer Science*

By

Alexis JOLY

March 2015

LIRMM & INRIA Sophia-Antipolis, ZENITH Team

**Thesis committee:**

<i>Reviewers:</i>	Jenny BENOIS-PINEAU	-	Professor at University of Bordeaux 1 (France)
	Ioannis Kompatsiaris	-	Research Director at CERTH-ITI (Greece)
	Bernard Merialdo	-	Professor at EUROCOM (France)
<i>Examinators:</i>	Nozha Boujemaa	-	Research Director at Inria Saclay (France)
	Olivier Buisson	-	Research scientist at INA (France)
	Guillaume Gravier	-	Research Director at IRISA (France)
	Patrick Valduriez	-	Research Director at Inria Sophia-Antipolis (France)
<i>President:</i>	William Puech	-	Professor at University of Montpellier

Copyright ©2015 Alexis JOLY  
All rights reserved.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>High-dimensional features indexing and search via hashing</b>	<b>5</b>
2.1	Dimensionality curse and approximate similarity search methods . . .	6
2.2	Probabilistic multi-probe queries in hash tables . . . . .	10
2.3	Partitioning and compressing high-dimensional data with RMMH . .	14
2.4	Hash-based Linear Classifiers Approximation . . . . .	20
2.5	Distributed KNN-graph approximation via hashing . . . . .	21
<b>3</b>	<b>User-centric content-based retrieval methods and systems</b>	<b>25</b>
3.1	Interactive Object Retrieval using Efficient Boosting . . . . .	25
3.2	Interactive Object Retrieval using Interpretable Visual Models . . .	28
3.3	Object-based Visual Query suggestion . . . . .	30
3.4	Event-centric media search and content suggestion . . . . .	32
3.5	Interactive plant identification based on social image data . . . . .	35



# Chapter 1

## Introduction

Rather than restricting search to the use of metadata, content-based information retrieval methods attempt to index, search and browse digital objects by means of *signatures* or *features* describing their actual content. Such methods have been intensively studied in the multimedia community [120, 102, 60, 66, 72, 24, 76] to allow managing the massive amount of raw multimedia documents created every day (e.g., video will account for 84% of U.S. internet traffic by 2018 [47]). Recent years have consequently witnessed a consistent growth of content-aware and multi-modal search engines deployed on massive multimedia data. Popular multimedia search applications such as Google images, Youtube, Shazam, TinEye or MusicID clearly demonstrated that the first generation of large-scale audio-visual search technologies is now mature enough to be deployed on real-world big data. Google images enables content-based similarity search and near-duplicates retrieval on more than 50 billion images. Youtube content-based video identification system analyses the equivalent of 100 years of videos each day to be compared with 8 million reference video files in the database. Shazam songs identification service works on about 10 million of songs whereas Music-ID exceeded the number of 38 millions of songs indexed in their alternative content-based indexing technology.

All these successful applications did greatly benefit from 15 years of research on multimedia analysis and efficient content-based indexing techniques. Figure 1.1 presents some of the most influential academic works of the last decades as a function of the number of items used in their experiments. It is interesting to notice that the only scientific publication reporting an experiment on more than 1 billion multimedia documents is a work of Google Research Lab (in 2007) which required the use of 2000 CPUs, far away from the hardware resources available for most other research players in the field. It shows that bridging the last orders of magnitude between algorithmic research and real-world applications is clearly a matter of large-scale infrastructures and distributed architectures. On the other side, fundamental research on breaking algorithm complexity has been a crucial prerequisite and continue attracting many research works in the field. Whatever the efficiency of the implementation and the use of powerful hardware and distributed architectures, the ability of an algorithm to scale-up is actually strongly related to its time complexity and space complexity.

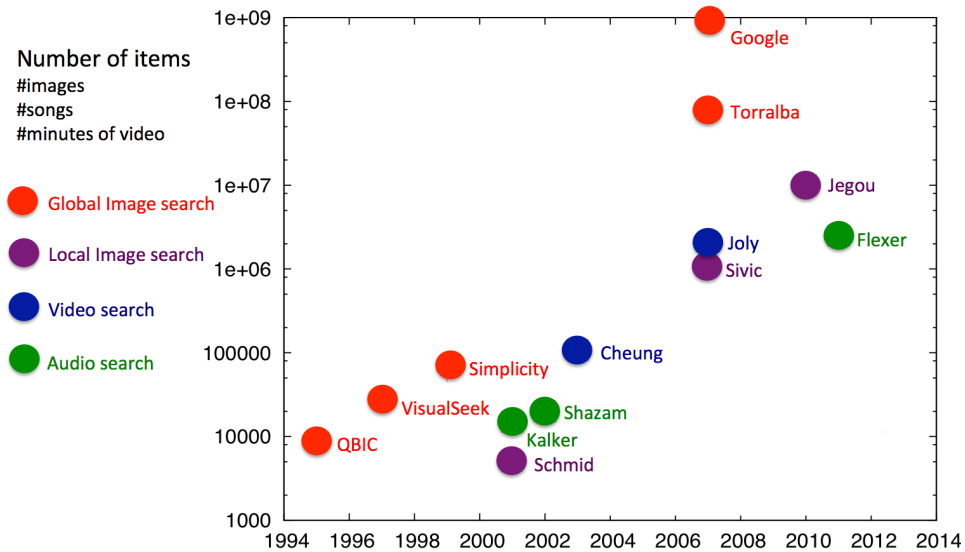


Figure 1.1: Some of the most influential works related to large-scale multimedia search as a function of the number of items used in the reported experiments. References of the works: QBIC [32], VisualSeek [103], Simplicity [113], Kalker [44], Cheung [19], Shazam [112], Torralba [106], Google [71], Joly [148], Flexer [98], Schmid [77], Sivic [89], Jegou [55]

Yet the maturity reached by the first generation of content-based search engines does not preclude an intensive research activity in the field. There is actually still a lot of hard problems to be solved before we can retrieve any information in images or sounds as easily as we do in text documents. Content-based search methods actually have to reach a finer understanding of the content. This requires modeling the raw signals by more and more complex and numerous features, so that the algorithms for analyzing, indexing and searching such features have to evolve accordingly. The bad news is that the searchable space created by the massive amounts of existing multimedia files greatly exceeds the area searched by today's major engines. And unfortunately, this will become more and more critical in the next decade. The amount of raw data is indeed still growing exponentially, boosted by the emergence of new technologies and related usage, such as mobile search [39, 18] and social search [83, 15]. Consistent breakthroughs are therefore still needed if we don't want to be lost in data space in ten years.

This thesis describes several of my works related to large-scale content-based information retrieval. The different contributions are presented in a bottom-up fashion reflecting a typical three-tier software architecture of an end-to-end multimedia information retrieval system. As illustrated by Figure 1.2, the lowest layer is only concerned with managing, indexing and searching large sets of high-dimensional feature vectors, whatever their origin or role in the upper levels (visual or audio features, global or part-based descriptions, low or high semantic

level, etc. ). The middle layer rather works at the document level and is in charge of analyzing, indexing and searching collections of documents. It typically extracts and embeds the low-level features, implements the querying mechanisms and post-processes the results returned by the lower layer. The upper layer works at the applicative level and is in charge of providing useful and interactive functionalities to the end-user. It typically implements the front-end of the search application, the crawler and the orchestration of the different indexing and search services. The core chapters 2, 3 and 4 of the thesis respectively correspond to the lower, middle and upper layer as the works they describe fit in the corresponding tier. The last chapter of the thesis (chapter 5) is rather concerned with long-term perspectives and big future challenges of the domain.

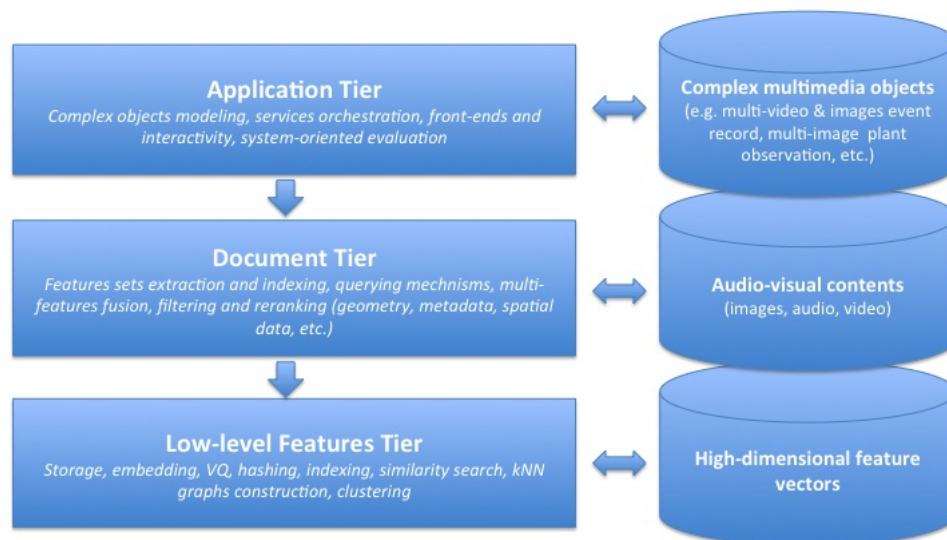


Figure 1.2: Three-tier architecture of a content-based multimedia information retrieval system

Please note that this document does not exhaustively cover all my past research works. Some of them were not included, not because they are of lower interest but only to keep this thesis as focused and coherent as possible. Among the most significant ones, I would like to cite the brilliant PhD work of Amel Hamzaoui [45], achieved under my supervision and the scientific direction of Nozha Boujemaa. She deeply revisited the theory of shared-nearest neighbors clustering methods and generalized them to the case of multiple information sources and bi-partite K-Nearest Neighbors (KNN) graphs [137, 138]. This work is not disconnected from the ones



described in this thesis as it typically enters the lowest layer of our architecture (low-level features clustering). It could typically be used on top of the last work of chapter 2 about the efficient construction of KNN graphs. But as the proposed clustering methods do not rely on hashing and do not scale well because of their intrinsic algorithmic complexity, I did not integrate them in chapter 2 (that is focused on high-dimensional features indexing and search via hashing).

## Chapter 2

# High-dimensional features indexing and search via hashing

The core functionality of a content-based search engine is to manage, index and search high-dimensional feature vectors extracted from the content of the multimedia documents. Let  $\mathbf{X}$  be a dataset of  $N$  feature vectors  $\mathbf{x}$  lying in  $\mathbb{R}^D$ ,  $D$  being referred as the *dimensionality* of the feature space. For any two points  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$ , we denote as  $\mathbf{x} \cdot \mathbf{y}$  their inner product and as  $d(\mathbf{x}, \mathbf{y})$  their distance according to a metric  $d()$  on  $\mathbb{R}^D$ . Now, the content-based search of similar documents or objects usually relies on one or several nearest neighbors queries in the features space. More precisely, for a query feature vector  $\mathbf{q} \in \mathbb{R}^D$ , we generally define two kinds of nearest-neighbors queries:

**Definition - Range Query:** Given a query  $\mathbf{q} \in \mathbb{R}^D$  and a radius  $r \in \mathbb{R}$ , find all  $\mathbf{x}_k \in \mathbf{X}$  such as  $d(\mathbf{q}, \mathbf{x}_k) \leq r$

**Definition - K-Nearest Neighbors (KNN) Query:** Given a query  $\mathbf{q} \in \mathbb{R}^D$  and an integer  $K \in \mathbb{R}^*$ , find the  $K$  items  $\mathbf{x}_k \in \mathbf{X}$  that are the closest to  $\mathbf{q}$  according to the metric  $d(\mathbf{q}, \mathbf{x})$

The straightforward algorithm to solve both kind of queries is a brute-force *exhaustive scan* of all feature vectors in  $\mathbf{X}$ . It iteratively computes the distances  $d(\mathbf{q}, \mathbf{x}_i)$  one by one and updates a heap or a priority queue to filter the nearest neighbors. It therefore has a linear time complexity in the dataset size  $N$  as well as in the feature space dimensionality  $D$  (typically  $O(N \cdot D)$ ,  $O(N \cdot D \cdot K)$  or  $O(N \cdot D \cdot \log K)$  depending on the Top-K filtering algorithm and the structure used to manage the nearest neighbors). Although it may appear reasonable at a first glance, this linear complexity prevents using the brute-force algorithm in most content-based applications because of the huge number of features in the dataset, the potentially large number of features in the query itself, and the low response time requirements (typically few seconds in online search scenarios). Ideally, to support efficiently massive

growths of data and increasing complexity of the extracted features, a scalable search engine should allow sub-linear search time and near-linear indexing time and memory usage.

## 2.1 Dimensionality curse and approximate similarity search methods

Many indexing structures were designed in the 80's and the 90's for managing, indexing and searching multi-dimensional feature vectors with relatively small numbers of attributes, typically from 2 to 10 [35, 12]. And in fact, popular structures such as the R-tree [43], the SR-tree [58], the M-tree [22] or more recently the cover-tree [8] are still at the core of many information systems (in particular geographic and environmental information systems). Unfortunately, these early proposed techniques are not time efficient for data with very high dimensionalities due to a well known phenomenon referred as the *dimensionality curse* [61]. Whatever the type of partitioning technique and the nearest neighbor search algorithm implemented, it actually has been proved that the number of chunks to be visited increases exponentially with the dimension of the feature space [117]. A first intuition of the phenomenon is that if you split each axis of a  $D$ -dimensional space in two parts, you will get an exponential number of chunks (bounded by  $2^D$ ) so that techniques based on grids or rectangular bounding regions are not adapted. The major other problem is that when the dimensionality  $D$  tends to infinity, all data points tend to be at a similar distance between each others comparatively to their norm. So that whatever the partition used, the fraction of chunks you need to visit to find the closest points of a given query also tend to 1 whatever the search algorithm (branch-and-bound, best-bin-first, etc.). You consequently have to scan all data points anyway and using a brute force exhaustive scan all points becomes more efficient (it actually doesn't involve any complex pruning algorithm to select the relevant chunks). It has been shown in the seminal work of Weber et al.[116] that when the dimensionality exceeds about 10, all existing data structures based on space partitioning are slower than the brute-force, linear-scan approach.

The vast majority of the following works on high-dimensional data search as well as the techniques used in modern large-scale applications are therefore based on *approximate* similarity search methods. As any partition of a high-dimensional space actually suffers from the dimensionality curse, the idea of solving exact similarity search queries more efficiently than a brute-force scan was progressively abandoned. Approximate similarity search methods rather try to return as much as possible of the exact nearest neighbors but do never reach a 100% rate. The quality of a given search algorithm can be measured in many ways (percentage of the exact NN retrieved, controlled distance from the real NN's, user-oriented ground-truth, etc.) but in any case there is a trade-off between the quality of the results and the efficiency gains. Trading quality for time can be done in many ways and it has been the key objective of a large literature [6, 38, 70, 3, 85, 79, 54].

Early proposed query approximation methods were typically aimed at implementing the approximate search paradigm within classical multidimensional indexing structures such as R-trees, KD-trees, SR-trees or M-trees. Most of them were simply extensions of exact search algorithms to the search of  $\epsilon$ -NN [5, 6, 122, 21]; a  $\epsilon$ -NN being an object whose distance to the query is lower than  $(1 + \epsilon)$  times the distance of the true nearest neighbor. One of the most influential work on the topic is the one of Arya et al. [6] who introduced an optimal approximate NN algorithm based on a *Balanced Box Decomposition tree* (BBD-tree). They did show that, in such indexing structure, it is possible to solve  $(1 + \epsilon)$ -approximate nearest neighbor queries in  $O(1/\epsilon)^D O(\log N)$  time with a  $O(1/\epsilon)^D O(N)$  preprocessing. Another interesting work was the one of Zezula et al. [122] who introduced a new structure, the M-tree, and several  $\epsilon$ -NN approximation mechanisms to efficiently process range and KNN queries (a M-tree is a pivot-based hierarchical structure whose main advantage is to generalize similarity search to any metric whereas previous tree-based structures were restricted to the use of the Euclidean distance). In their experiments, the performance gain was around 20 times faster compared to exact queries but for moderate recall values around 50% of the true nearest neighbors and moderately high dimensions. This first generation of approximate similarity search techniques was pioneering but the performance gain over a brute-force scan of the data still remained very low when the dimensionality of the feature space increases.

Modern approximate similarity search schemes allowing to have consistent speed-ups even for very high-dimensional spaces are based on three main principles. The first one is *embedding* or *dimension reduction* [33], meaning that the original feature space is *transformed* or *embedded* in a new one of lower dimension, yet preserving some of its topological properties. Most of them rely on the well known Johnson-Lindenstrauss lemma [56, 34], which states that a given set of points in a high-dimensional space can be embedded through a map into a space of much lower dimension in such a way that distances between the points are nearly preserved. This covers a large range of methods such as principal components analysis or independent component analysis, features extraction [69], random projections [48, 25] or kernel methods [100]. The second main principle is *Vector Quantization* (VQ), meaning that the real-valued feature vectors are associated with quantized indices thanks to more or less complex quantization functions (e.g. a simple grid or a cluster-based partition). The quantized indices can be used to partition the data points and construct efficient data structures such as inverted lists, hash tables or even trees. The third main principle is *lossy compression*, i.e. data encoding methods that uses inexact approximations for representing the original feature vectors. The main objective of such compression is to drastically reduce the amount of storage in order to fit the whole dataset in main memory and reduce as much as possible the number of memory cache outputs that occur when exhaustively scanning a large set of features. Beyond, most methods making use of lossy compression also make use of efficient similarity metrics computed directly in the compressed

space rather than decompressing the features and computing the exact metric. This provides additional speed-ups and contributes to the success of these approaches.

One of the pioneering works that inspired many of the modern techniques is the VA-file of Weber et al. [117]. After demonstrating that existing partitioning and clustering techniques exhibit linear complexity at high dimensionality, they rather proposed to speed-up the unavoidable sequential scan by an alternative organization based on approximations. The vector-approximation file (VA-file) simply divides the space with a  $2^b$ -cells grid but instead of hierarchically organizing these cells like in grid-files or R-trees, the VA-file allocates a unique bit-string of length  $b$  for each cell and approximates the data points that fall into a cell by that bit-string. Nearest neighbor queries are performed by *scanning* the entire approximation file, and by excluding the vast majority of vectors from search (filtering step) based only on these approximations. This raw principle is still in use in modern content-based retrieval methods such as the ones based on hashing [52, 118, 63, 41, 46] – including our own work described in this thesis, e.g. [147] – or the popular product quantization method of Jegou et al. [54]. The main difference is that the grid-based lossy compression is now usually computed after a dimension reduction step and that the approximated vectors are stored in main memory rather than in a file on disk. Also, the *fine quantizer* used to build the vector approximations is now usually combined with another coarser quantizer allowing to index them in inverted lists or hash tables, and avoid scanning the whole dataset. In the case of binarized hashing methods, the *coarse quantizer* can be simply a prefix of the full bit-string of the fine quantizer and this is actually what we usually do in our own work. Another very popular coarse quantizer is rather based on the k-means clustering algorithm thanks to the influential work of Sivic et al. [101] who introduced the bag-of-visual-words model.

Another seminal work regarding approximate similarity search is the one of Indyk et al. [48] who introduced the principle of Locality Sensitive Hashing (LSH) [25, 3]. This was the first work suggesting the use of random projections embedding as a way to index very high-dimensional features for which classical statistical dimension reduction techniques such as PCA are not computable because of their quadratic complexity in dimensionality. The basic principle of LSH is to use a family of randomized hash functions that map similar objects into the same hash bucket with a probability higher than non-similar objects. More formally, given two points  $\mathbf{q}, \mathbf{v} \in \mathbb{R}^d$  and a family  $\mathcal{H}$  of hash functions of the form  $h : \mathbb{R}^d \rightarrow \mathbb{N}$ , the key property of locality sensitive hashing functions is that:

$$\Pr[h(\mathbf{q}) = h(\mathbf{v})] = f(\kappa(\mathbf{q}, \mathbf{v})) \quad (2.1)$$

where  $\kappa$  is the similarity function associated with the original feature space and  $f()$  is a monotonically increasing function. A popular LSH function is the one sensitive to the inner product. It is defined as:

$$h(\mathbf{x}) = \text{sgn}(\mathbf{w} \cdot \mathbf{x}) \quad (2.2)$$

where  $\mathbf{w} \in \mathbb{R}^d$  is a random variable distributed according to  $\mathcal{N}(0, \mathbf{I})$ . In that case, equation 2.1 becomes:

$$Pr[h(\mathbf{q}) = h(\mathbf{v})] = 1 - \frac{1}{\pi} \cos^{-1} \left( \frac{\mathbf{q} \cdot \mathbf{v}}{\|\mathbf{q}\| \|\mathbf{v}\|} \right) \quad (2.3)$$

Following the success of LSH, a new research thread on high-dimensional hashing methods progressively emerged in the 2000s and is still very active today [52, 118, 63, 41, 46][145, 147]. One advantage of hashing methods over trees or other structures is that they allow simultaneously efficient indexing and data compression. It actually appeared that the binary hash codes produced by popular hashing functions such as LSH could be used directly as quantized versions of the feature vectors and not only as keys to index them in hash tables. The expensive exact distance computations in the original high-dimensional feature space could then be replaced by efficient Hamming distances computations in the compressed space. This *Hamming Embedding* (HE) [52] principle is not an alternative to using indexing structures but a complementary way to reduce both the memory usage and the distance computation cost. Indeed, as the produced hash codes are typically 8 to 32 times more compact than the original feature vectors, the required memory space is reduced by the same factor as well as the number of memory cache misses (i.e. the number of data blocks that need to be transferred from the RAM memory to the CPU cache memory). Furthermore, there exists several easy ways to drastically speed-up Hamming distance computations, either by using look-up tables on 8 or 16-bits sub-words or by using specific assembler instructions such as population count instructions. Hash-based methods can be classified across three main categories:

**Data independent hashing functions:** in these methods, the hashing function family is defined independently from the data to be processed. We can distinguish the one based on randomized process, to which Locality Sensitive Hashing (LSH) functions belong ( $L_p$  stable [25], min-hash [20], random Fourier features [91], Shift-Invariant Kernel [91], random orthogonal projections [52]), and the one based on a deterministic structuring, including grids [116], space filling curves [150][90] or more recently, lattices [86, 99]. The randomized ones are usually considered as more adaptive to heterogeneous data distributions and are thus usually more efficient than deterministic hash functions. Some recent works did show that using more complex lattices may be more effective [86, 99] in some cases, but their higher complexity and computational costs make them less successful.

**Data dependent hashing functions:** In that case, the hashing function family is defined uniquely only for a given training dataset and the hash functions usually involve similarity comparisons with some features of the training dataset. The objective of these methods is to closely fit the data distribution in the feature space in order to achieve a better selectivity while preserving locality as much as possible. Among the most popular methods, we can cite Spectral Hashing (SH) [118], KLSH [63], k-means based hashing [86], PQ-code [54], ITQ [41] or RMMH

[147] (our own work described in section 2.3).

**(Semi-)supervised data dependent hashing functions:** In this last category, the training dataset contains additional supervised information, e.g. class labels [115, 68] or pairwise constraints [78]. These methods usually attempt to minimize a cost function on the hash functions set, combining an error term (to fit training data) and a regularization term (to avoid over-fitting).

## 2.2 Probabilistic multi-probe queries in hash tables

*This section describes one of my first contribution to approximate similarity search that was initiated during my PhD work (e.g. in [150]) and further generalized in [145]. This work was done in close collaboration with Olivier Buisson (computer scientist at the French National Institute of Audio-visual (INA)).*

This work concerns the definition of an approximate search algorithm for indexing methods making use of hash functions as coarse quantizer to build one or multiple hash tables. Whereas the choice of the quantization function is crucial for fine quantization purposes (i.e. for the compression of the features), it is somehow less important when the quantization function is used as a coarse quantizer to build an index. The search algorithm used to filter the visited chunks is actually as much important for the efficiency and the quality of the results. It is for instance noticeable that many of the most popular visual search methods using k-means as the coarse quantizer [89, 88, 111, 57], do not allow any control of the quality of the retrieved NN's. A one-size-fit-all search parameter (e.g. the number of the closest chunks to be visited) is usually estimated once and search is far from being optimal for each query. On the other side, the original LSH indexing and search framework offered some nice quality guarantees. It works as follows:

1. Choose  $L$  hash functions  $\mathbf{g}_1, \dots, \mathbf{g}_L$ , independently and uniformly at random, each hash function  $\mathbf{g}_j = (h_{j,1}(\mathbf{v}), \dots, h_{j,k}(\mathbf{v}))$  being the concatenation of  $k$  unitary LSH functions randomly generated from a family  $\mathcal{H}$ .
2. Use each of the  $L$  hash functions to construct one hash table (resulting in  $L$  hash tables).
3. Insert all points  $\mathbf{v} \in \mathcal{V}$  in all hash tables by computing the corresponding  $L$  hash codes.

At query time, the  $L$  hash codes of a given query vector  $\mathbf{q}$  are computed in order to select a set of  $L$  hash buckets containing potential nearest neighbors. The candidate feature vectors belonging to the selected buckets might then be refined by computing their exact distance to the query. It is then possible to show that the probability to retrieve a nearest neighbor  $v$  belonging to a range query of radius  $\theta$  around  $q$  is equal to:

$$p_\theta(q, v) = 1 - (1 - f^k(\theta))^L \quad (2.4)$$

where  $f$  is the sensitivity function as defined in Equation 2.1. For a given query radius  $\theta$ , it is consequently possible to control the expected quality of any range query by choosing the appropriated number of hash tables. For a required quality

$p_\theta(q, v) = \alpha$ , we get:

$$L(\alpha) = \frac{\log(1 - \alpha)}{\log(1 - f^k(\theta))} \quad (2.5)$$

However, although it is theoretically very attractive, this indexing and search strategy has two main drawbacks in practice. The first one is that the required number of hash tables  $L$  becomes too big for large datasets so that the whole index cannot fit in memory anymore. Reducing the hash code length  $k$  could resolve the problem but then the filtering efficiency becomes too bad limiting the efficiency gain. A second drawback is that the quality control is restricted to the case of range queries whereas in practice KNN queries are much more convenient. Tuning the radius  $\theta$  of range queries might for instance be very tricky in high-dimensional space as this parameter becomes unstable with no results at all for a given value and almost all the database for a slightly larger value  $\theta + \epsilon$ .

Probabilistic multi-probe queries is the solution we proposed to solve both issues. Its basic principle is to select multiple buckets in each hash table and not only the single bucket in which the query falls. Intuitively, this allows increasing the probability to retrieve nearest neighbors in a single hash table and consequently the number of required hash tables can be strongly reduced. The selection of the multiple buckets is done according to (i) a probabilistic model estimated offline according to the targeted query type (range query, KNN, etc.) and (ii) a greedy algorithm allowing to select the most probable buckets during the online search. Using a probabilistic search model allows modeling more accurately the distribution of the real nearest neighbors and thus to focus search on the right buckets. We did prove in [140] that probabilistic queries, even based on a simple isotropic normal distribution, are much more effective than classical query approximation methods based on distance approximations and  $\epsilon$ -NN's (as illustrated in Figure 2.1). The last ones are equivalent to consider a uniform spherical distribution within the range of the query and we did show that such distribution do not model well the features distortion encountered in real-world image retrieval applications.

The first version of probabilistic multi-probe queries we developed in [150, 151, 148] was built on top of a hierarchical grid-like partition of the original feature space (i.e. a Hilbert space-filling curve) and a single hash table. It allowed very high speed up over the sequential scan (about two order of magnitude faster) but the absence of a reduction dimension embedding restricted its use to moderate dimensions up to 32. As mentioned earlier, the probabilistic model itself was a simple isotropic normal distribution with a single parameter  $\sigma$  that was estimated thanks to synthetic attacks of real images. The simplicity of the probabilistic model allowed us to derive an analytical cost model of the whole indexing and search scheme. We actually did prove the sub-linear complexity of the search algorithm in database size for trivial theoretical data distributions. The cost model was not directly applicable to estimate the search time of the method on real-world data but at least it explained why it was still efficient on such data and why it scales very well.

We further improved and generalized our probabilistic multi-probe model in [145]. We first generalized it to the use of multiple hash tables and to any hashing scheme in particular locality sensitive hashing as it was increasingly recognized for its good embedding properties of very-high-dimensional data. We also introduced a richer formalism and a more accurate probabilistic model of the distribution of the real nearest neighbors. We also positioned our work over alternative multi-probe LSH schemes that did appear meanwhile [73, 82]. We notably show that the distance-based buckets selection strategy used in that schemes was theoretically



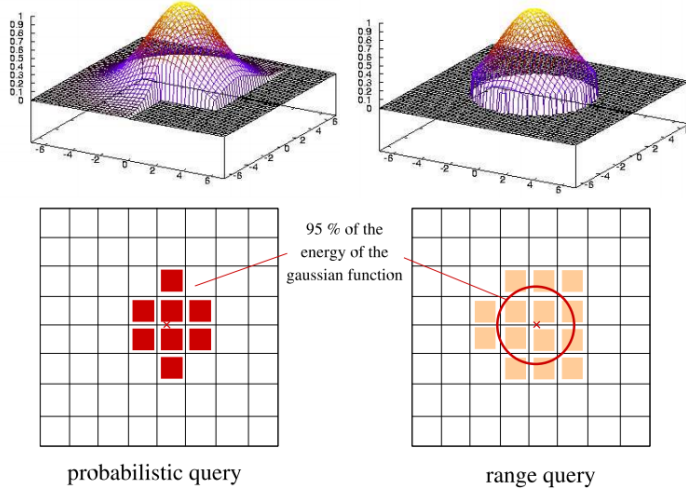


Figure 2.1: Illustrative comparison between a probabilistic query and a range query for a normally distributed search model. Less chunks are selected by the probabilistic query whereas the energy of the gaussian function covered by both methods is the same. In real-world high-dimensional data partitions, the number of selected chunks can be several orders of magnitude lower.

less justified than our probabilistic model and practically less efficient.

More formally, locality sensitive hashing theory is based on the probability distribution of the hash values of two given points  $\mathbf{q}$  and  $\mathbf{v}$ , over the random choices of the hash functions, e.g. over random choices of a parameter set  $\mathbf{w}$ . In other words,  $\mathbf{w}$  is considered as a random variable whereas  $\mathbf{v}$  and  $\mathbf{q}$  are considered as constants. Based on this formalism, it is possible to derive the probability density function  $p_{\delta_{\mathbf{q},\mathbf{v}}(\theta)|(q,v)}$  that  $\mathbf{q}$  and  $\mathbf{v}$  belongs to adjacent buckets over the randomly picked hash functions. The principle of the method of Lv et al. [73] is to use this probability as the likelihood  $l_{\delta_{\mathbf{w}}(\mathbf{q},\mathbf{v})|\mathbf{w}}$  that a given bucket contains a neighbor  $\mathbf{v}$  of  $\mathbf{q}$  when the partition  $\mathbf{w}$  is fixed. Considering this likelihood as the probability that the neighboring bucket contains a real neighbor is unfortunately a case of prosecutor’s fallacy since the real density depends on the prior distribution of  $\mathbf{v} \in n(\mathbf{q})$ . Our method rather estimates the success probability of a given hash bucket in a given hash table a posteriori, i.e. for an observed partition parameterized by  $\mathbf{w}$ . For a given query  $\mathbf{q}$ , in the absence of evidence, a point  $v \in n(\mathbf{q})$  is indeed a random variable to which we associate a prior probability distribution  $p_{v|q}(\mathbf{x})$ ,  $\mathbf{x} \in \mathbb{R}^d$ . The prior distribution itself is estimated beforehand on training data such as real visual matches or exact nearest neighbors computed through an offline exhaustive scan of training samples.

Based on this theoretical new framework, we then introduced a greedy algorithm aimed at efficiently selecting the most probable buckets to be visited in each hash table. A naive way to do so would be to compute the success probability of all possible keys and sort them, but it is of course practically impossible. A more efficient but approximative way would be to restrict the computation of the success probability to the neighboring buckets up to a given radius  $s$  from the hash code in which the query falls. This method has the advantage to be generic but is still not very efficient since the number of hash buckets probabilities to estimate

remains  $\sum_{n=1}^s 2^n \binom{k}{n}$ . If we tolerate an independence hypothesis on the probabilistic search model, it is however possible to use a drastically more efficient algorithm, generalizing the Query-Directed Probing Sequence algorithm defined in [73] to probabilistic queries. Also to speed-up the computation of the buckets probabilities, we did implement our algorithm through the use of look-up-tables containing pre-computed values of the discrete probabilities characterizing a switch from a bucket to another one according to a given hash function.

Table 2.1 illustrates the efficiency gain of our probabilistic multi-probe queries over the more classical distance-based criterion of Lv et al. [73]. It shows that our method requires substantially fewer number of probes (i.e. fewer number of buckets and data points to be visited) to achieve the similar or slightly better recall. Since the main initial objective of multi-probe methods is to reduce the large

dataset	method	L	recall	nb of probes
HSV	<b>a posteriori</b>	5	<b>0.94</b>	<b>31,813</b>
	likelihood	5	0.92	196,500
SIFT	<b>a posteriori</b>	4	0.92	<b>2,689</b>
	likelihood	4	0.92	6,400
DIPOLE	<b>a posteriori</b>	2	<b>0.96</b>	<b>5,752</b>
	likelihood	2	0.95	51,200

Table 2.1: Search performance comparison of a posteriori probabilistic probing vs. likelihood probing

space requirements of LSH, it is also interesting to compare the time efficiency of our method and LSH according to their space requirements. To do that, we vary the amount of memory allocated to LSH and re-ran the same benchmark for each setting. Comparative time efficiency is measured by the ratio between *LSH* query time and the query time of our method. Figure 2.2 plots this time ratio as a function of the space requirement of LSH which is measured by the ratio between the index size and the data size. Note that the space ratio of our technique for this dataset was equal to 0.125, which means that the index was almost 10 times smaller than the data itself. The figure show that our method is always faster than LSH since the time ratio is always larger than 1. For a reasonable space requirement of 1 (index size equal to data size), our method is about 15 times faster than LSH. Since the curve is converging for large space ratio, we can also estimate that our method is about 2 times faster for unlimited memory space.

Overall, the advantages of our probabilistic multi-probe model are the following:

1. More efficient filtering: taking account the prior distribution of the searched objects allows to reduce significantly the required number of probes to achieve similar recall than likelihood or distance-based probing methods. High recall can even be obtained efficiently with a single hash table allowing to save a lot of memory usage.
2. Search quality control and parameters estimation: the relevance criterion of a hash bucket being a probability and not a likelihood or distance-based criterion, it allows to have a coarse estimation of the probability to find relevant objects without tuning. Having an estimation of the probability also allows to estimate automatically the required number of hash tables  $L$  without tuning.
3. Genericity and Query adaptivity: our probabilistic filtering algorithm is fully independent of the query type. It just requires query samples and correspond-

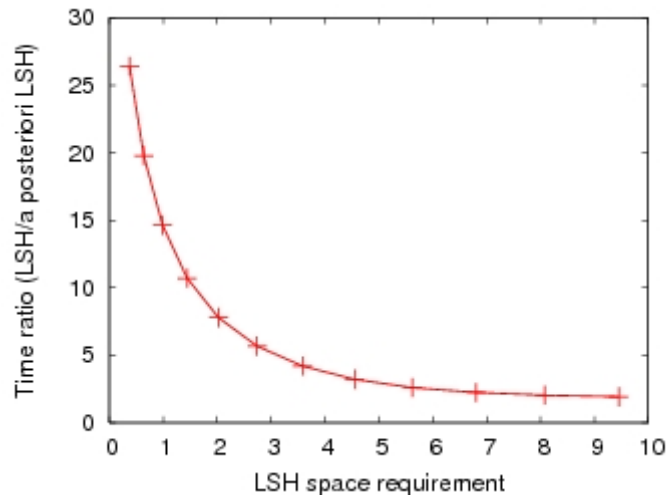


Figure 2.2: Search time ratio (LSH / our method) according to LSH space requirement (normalized by dataset space)

ing relevant objects sets, not necessarily nearest neighbors. Examples of other relevant objects are distorted features obtained after transformation of a multimedia content or nearest neighbors of the query in an other dataset (e.g. a category specific dataset or a training dataset). Search can also be easily adapted to different objectives by pre-computing different prior models and corresponding probabilities Look-up tables for the same index structure. A typical application is to achieve class dependent queries.

## 2.3 Partitioning and compressing high-dimensional data with RMMH

*This work [147] was done in close collaboration with Olivier Buisson (INA).*

Random Maximum Margin Hashing (RMMH [147]) is a new hashing method that we introduced in 2011 to answer several limitations of previous data dependent methods. Efficiency improvements of data dependent methods over independent ones (such as LSH) were shown in several studies [97, 118, 53]. But this acted only for limited hash code sizes, up to 64 bits. Indeed, the drawback of data dependent hash functions is that their benefit degrades when increasing the number of hash functions, due to a lack of independence between the hash functions. This is illustrated by Figure 2.3 showing the performance of a standard LSH function compared to the popular Spectral Hashing method [118], known to outperform several other data dependent methods. This conclusion was confirmed by [91] who did show that their Shift-Invariant Kernel hashing function (data independent) dramatically outperformed Spectral Hashing for all hash code sizes above 64 bits. Our main claim in [147] was that the lack of independence between unitary hash functions is the main issue affecting the performance of data dependent hashing methods compared to data independent ones. Indeed, the basic requirement of any hashing method is that the hash function provide a uniform distribution of hash values, or at least as uniform as possible. Non-uniform distributions do increase the overall expected number of collisions and therefore the cost of resolving them. For high-dimensional data hashing methods, we argue that this uniformity

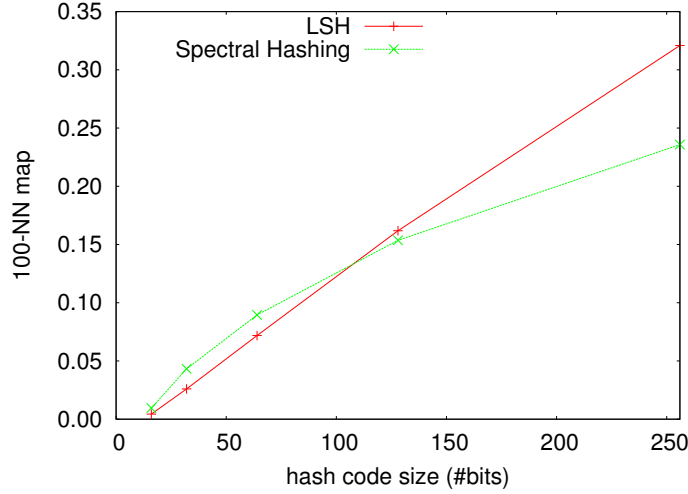


Figure 2.3: LSH vs Spectral Hashing for increasing hash code sizes

constraint should not be relaxed too much even if we aim at maximizing the collision probability of close points.

More formally, if we denote as  $\mathbf{h}_p = [h_1, \dots, h_p]$  a binary hash code of length  $p$ , lying in  $\mathbb{B}^p = \{-1, 1\}^p$ , where the hash functions  $h_i$  are built from a *data independent* hash function family  $\mathcal{H}$ , the collision probability follows:

$$Pr_p(\mathbf{q}, \mathbf{v}) = Pr[\mathbf{h}_p(\mathbf{q}) = \mathbf{h}_p(\mathbf{v})] = [f(d(\mathbf{q}, \mathbf{v}))]^p$$

where  $f(\cdot)$  is the sensitivity function of the family  $\mathcal{H}$  for a given metric  $d(\cdot)$ , i.e the collision probability function of a single hash function.

Data dependent hash functions usually aim at providing a better sensitivity function than data independent ones. They are indeed built to boost the collision probability of close points while reducing the collision probability of irrelevant point pairs. But when the hash functions are dependent from each other, we have:

$$\frac{Pr_p(\mathbf{q}, \mathbf{v})}{Pr_{p-1}(\mathbf{q}, \mathbf{v})} = Pr[h_p(\mathbf{q}) = h_p(\mathbf{v}) | \mathbf{h}_{p-1}(\mathbf{q}) = \mathbf{h}_{p-1}(\mathbf{v})]$$

and as this ratio is usually tending to one when  $p$  increases, new bits are less and less useful. At a certain point, the number of irrelevant collisions might even be not reduced anymore.

Following these remarks, we considered the uniformity of the produced hash codes as a primary constraint for building an efficient data dependent hash function family. For a dataset drawn from a probability density function  $p_x$  defined on  $\mathbb{X}$ , an ideal hash function should respect:

$$\forall p \in \mathbb{N}^*, \quad \forall \mathbf{h}_i \in \mathbb{B}^p \quad \int_{\mathbf{h}(x)=\mathbf{h}_i} p_x(x) dx = c \quad (2.6)$$

where  $c$  is a constant (equal to  $\frac{1}{2^p}$ ). From this follows that (i) each individual hash function should be balanced (when  $p = 1$ ):

$$\int_{h(x)=1} p_x(x) dx = \int_{h(x)=0} p_x(x) dx = \frac{1}{2} \quad (2.7)$$

and (ii) all hash functions must be independent from each others.

The principle of RMMH is to approximate this ideal objective by training balanced

and independent binary partitions of the feature space. For each hash function, we pick up  $M$  training points selected at random from the dataset  $\mathbf{X}$  and we randomly label half of the points with  $-1$  and the other half with  $1$ . We denote as  $\mathbf{x}_j^+$  the resulting  $\frac{M}{2}$  positive training samples and as  $\mathbf{x}_j^-$  the  $\frac{M}{2}$  negative training samples. The hash function is then computed by training a binary classifier  $h_\theta(\mathbf{x})$  such as:

$$h(\mathbf{x}) = \operatorname{argmax}_{h_\theta} \sum_{j=1}^{\frac{M}{2}} h_\theta(\mathbf{x}_j^+) - h_\theta(\mathbf{x}_j^-) \quad (2.8)$$

Now, the remaining question is how to choose the best type of classifier. Obviously, this choice may be guided by the nature of the targeted similarity measure. For non-metric or non-vectorial similarity measures for instance, the choice may be very limited. In such context, a KNN classifier might be very attractive in the sense that it is applicable in all cases. Using a 1NN classifier for kernalized feature spaces would for exemple define the following hash function family:

$$h(\mathbf{x}) = \operatorname{sgn} \left( \max_j \kappa(\mathbf{x}, \mathbf{x}_j^+) - \max_j \kappa(\mathbf{x}, \mathbf{x}_j^-) \right) \quad (2.9)$$

Interestingly, it is easy to show that such family is indeed sensitive to the expected number of shared neighbors. Shared neighbors information has already been proved to overcome several shortcomings of traditional metrics. It is notably less sensitive to the dimensionality curse, more robust to noisy data and more stable over unusual features distribution [29, 50][138]. Better classifiers may however be found for linear and kernel spaces. In this way, let us now consider the second main requirement of an ideal Locality Sensitive Hashing function family, that is preserving locality. Maximizing the collision probability of *close points* is indeed the primary principle of classical LSH methods. Within our balanced training strategy, we should thus minimize the probability that a point close to one of the training sample *spill over* the boundary between the two classes. In this context, maximizing the margin between positive and negative samples appear to be very well appropriated. This will indeed maximize the distance of all training samples to the boundary and guaranty that neighbors with a distance lower than the half margin do not spill over. This remark is closely related to Vapnik & Chervonenkis theory which states that large margin classifiers have low capacities and thus provide better generalization [9]. We therefore proposed to define our hash function family by the set of hyperplanes maximizing the margin between random balanced samples:

$$h(\mathbf{x}) = \operatorname{sgn}(\mathbf{w}_m \cdot \mathbf{x} + b_m) \quad (2.10)$$

$$(\mathbf{w}_m, b_m) = \operatorname{argmax}_{\mathbf{w}, b} \frac{1}{\|\mathbf{w}\|} \min \left[ \min_j (\mathbf{w} \cdot \mathbf{x}_j^+ + b), \min_j (-\mathbf{w} \cdot \mathbf{x}_j^- - b) \right] \quad (2.11)$$

We referred to the proposed method as RMMH, for **R**andom **M**aximum **M**argin **H**ashing. To the best of our knowledge, this is the first work relying on the supervised learning of randomly labeled data points, which makes it conceptually very original. In practice, optimal hyperplanes  $\mathbf{w}_m$  can be computed easily by a Support Vector Machine (SVM). For kernel spaces,  $\mathbf{w}_m$ 's can only be expressed as a weighted sum over support vectors, so that the hash function becomes more costly:

$$h(\mathbf{x}) = \operatorname{sgn} \left( \sum_{i=1}^m \alpha_i^* \kappa(\mathbf{x}_i^*, \mathbf{x}) + b_m \right) \quad (2.12)$$

where  $\mathbf{x}_i^*$  are the  $m$  support vectors selected by the SVM ( $\mathbf{x}_i^* \in \{\mathbf{x}_j^+, \mathbf{x}_j^-\}$ ).

The number  $M$  of samples selected for each hash function is the only parameter of RMMH. Deriving a theoretical optimal value for  $M$  unfortunately appears to be a tricky task. It would require to formally model the distribution  $p_w$  of  $\mathbf{w}_m$  which is still an open problem. Some interesting logical guidelines can however be discussed according to three constraints: hashing effectiveness, hashing efficiency and training efficiency. Let us first discuss efficiency concerns. SVM training being based on quadratic programming, an acceptable training cost implies that  $M \ll N$  (even if it is an offline process). But hashing efficiency is even more critical: hash functions usually need to be computed online and the resulting cost is part of the overall search cost. In the linear case, this is obviously not a problem since a single projection on  $w_m$  needs to be computed, making our method as efficient as normal projections. In kernel spaces however, the hashing cost is higher since we need to compute as much kernel values as the number of support vectors, for each of the  $p$  hash functions. Worst case hashing cost complexity is therefore  $O(pM)$ . So that an important requirement is that:

$$M \ll \frac{N}{p} \quad (2.13)$$

Let us now discuss effectiveness concerns related to the two ideal objectives discussed above: *uniformity* and *locality preservation*. The larger the training size  $M$  and the better the uniformity. For an extreme value  $M = N$  (supposing that the capacity of the classifier is large enough to separate any training set of size  $N$ ), we would get a perfect uniformity and the probability of irrelevant collisions would be minimal. In other words, the data would be perfectly *shattered*, to re-use Vapnik-Chervonenkis terminology. But this would also lead to overfitting, since close pairs would be shattered as well. On the other extreme, too small training data would increase the error expectation of the classifier and thus degrade the expected uniformity. Data would be not shattered enough. The optimal value for  $M$  is thus guided by the a tradeoff between uniformity (data shattering) and locality preservation (generalization). In [147], we conducted an empirical study of  $M$  parameter confirming that there always exist an empirical maximum. To estimate an approximate max bound on  $M$ , we can at least control the risk that close points might be split in the training data itself. Let us consider KNN's as relevant matches and any other pair of point as irrelevant. In that case, the expected number of relevant pairs in a random training set of  $M$  points is equal to  $\frac{M^2 k}{N}$ . If we want to have this expected number lower than 1, we get:

$$M < \sqrt{\frac{N}{k}} \quad (2.14)$$

Interestingly, this value is sub-linear in dataset size  $N$ . With this max bound value, the hashing cost complexity  $O(pM)$  becomes  $O(p\sqrt{\frac{N}{k}})$  which guaranties that it does not become preeminent for very large datasets. Experiments conducted in [147] did show that  $M$  is rather stable around its maximum and that it evolves only slightly for varying data sizes (from 10K to 1M) and varying number of neighbors (from 10 to 1000). The max bound of Equation 2.14 is not always respected by the empirical optimum, but the order of magnitude is correct.

### 2.3.1 Comparison to state-of-the-art

We first evaluated RMMH in  $\mathbb{R}^d$  to allow comparisons to state-of-the-art methods. We used a dataset of 1 M SIFT features (**SIFT-1M**) normalized according to  $L_2$ -norm, so that the exact KNN according to  $L_2$  are equivalent to the exact top  $k$

items according to the inner product, the triangular L2 kernel or the RBF kernel. This allowed us to compare a quite large range of methods on this dataset: RMMH was experimented with 3 different kernels: linear, triangular L2 and RBF. For the RBF kernel, we estimated  $\gamma$  on real KNN samples. We did compare RMMH to two data dependent methods (KLSH [63] and spectral hashing [118]) and two data independent methods (LSH and Random Fourier Features (RFF), the RBF-sensitive method of Raginsky et al. [91]). For Spectral Hashing and KLSH, we used the same number of training samples than the one required by RMMH ( $p \times M$ ). For KLSH we used the  $L_2$  triangular kernel, since we got the best performance with it. For LSH, we used the family sensitive to the inner product (see Equation 2.2). For Raginsky’s method, we used the same *RBF* kernel parameter  $\gamma$  than for RMMH. Results are provided in Figure 2.4. They show that RMMH clearly outperforms the two other data dependent methods, whatever the used kernel, even the linear one. Thanks to the better independence of RMMH hash functions, the performance are indeed less degrading when increasing the hash code size. Comparisons to data independent methods show that RMMH performs better for a wide range of useful hash code sizes from 1 to about 800 bits which covers many hashing applications. Beyond the quite slight effectiveness gain, the most important point is that RMMH succeed in producing independent enough hash functions. Further experiments conducted in [147] did notably confirm that RMMH performs better than KLSH [63] on two kernelized testbeds (including one on bag-of-words features with a Chi Square kernel) .

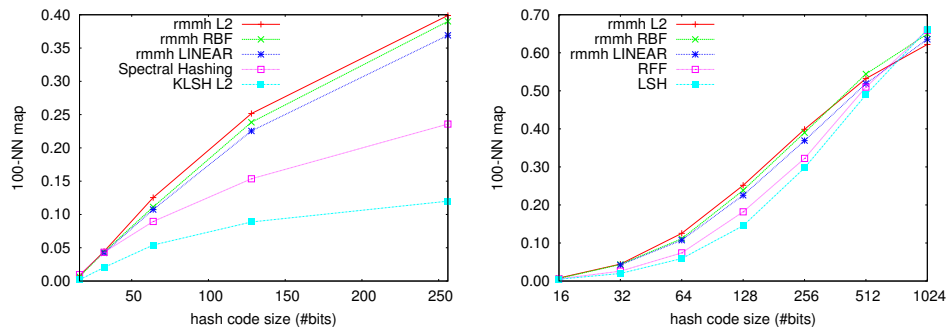


Figure 2.4: Comparison of RMMH to state-of-the-art methods (**left**) comparison to data dependent methods (**right**) comparison to data independent methods

### 2.3.2 Image retrieval performance

We then evaluated the performance of RMMH for image retrieval. The aim here is not to retrieve the  $k$ -nearest neighbors in the original feature space but the most relevant images. We therefore used a dataset of 1.2 M Bags-Of-SIFT Features ( $D=1000$ ) provided within ImageNet/PASCAL VOC Large Scale Visual Recognition Challenge (ILSVRC) [96]. As suggested within this challenge, we relaxed the classification tolerance to the five best retrieved classes (recognition rate@5). Figure 2.5 presents the classification rate of RMMH with the a linear kernel and the one of LSH for varying hash code sizes. The horizontal line corresponds to the classification rate obtained with the exact inner product in the original feature space. We can first remark that the gain of RMMH over LSH is much larger than previous experiments (when searching approximate  $k$ -nearest neighbors). That means that RMMH provides a better embedding of the underlying data structure, whereas LSH only converges to the original metric. RMMH is even better than the exact distances for hash code sizes larger than 600 bits, meaning that it also works as a

metric learning method. Finally, with only 512 bits (64 bytes), RMMH hash codes provide equivalent performance than the original 1000 dimensional bag-of-features.

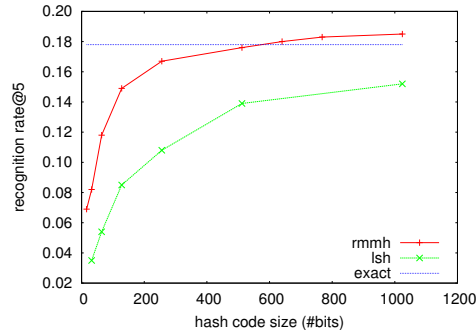


Figure 2.5: Classification performance on **ImageNet-BOF**

### 2.3.3 Indexing performance

We finally evaluated RMMH in terms of indexing performance, using our a posteriori multi-probe method as described in section 2.2 or in [145]. Results on the **ImageNet-BOF-1M** mentioned above are reported in Figure 2.6 and table 2.2. The plot shows that both LSH and RMMH achieve sub-linear search time in data size, providing consistent efficiency gains over the linear scan (which is not trivial with a dimension equal to 1000). But RMMH clearly outperforms LSH (as much as LSH outperforms the exhaustive scan). The sub-linearity coefficient of RMMH is indeed higher, leading to increasing efficiency gains when the size increases. That confirms again that RMMH closely fit the data distribution while keeping a good independence between the hash functions. For the full dataset of 1M BoF (see table 2.2), RMMH is finally 37 times faster than exhaustive scan and 5 times faster than LSH. If we use RMMH for compression in addition to indexing (using 1024-bits hash-codes), the search time can be further divided by a factor 5 and the memory usage by 13.

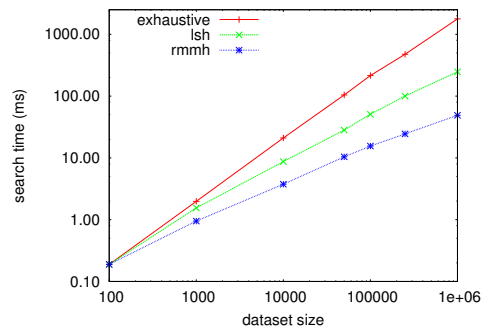


Figure 2.6: search time vs data size - comparison of RMMH to LSH and exhaustive scan



method	time (ms)	NN recall	Mem (Gb)
Exhaustive	1777	1.0	5.05
LSH index	247	0.67	5.11
RMMH index	<b>49</b>	0.69	5.11
RMMH index + sketch	<b>10</b>	0.62	<b>0.39</b>

Table 2.2: Indexing and Search statistics on **ImageNet-BOF**

## 2.4 Hash-based Linear Classifiers Approximation

*This work [165] was achieved in collaboration with Saloua Litayem who implemented and experimented the proposed contributions under my supervision.*

In this work, we were specifically interested in speeding-up the prediction phase of Linear Support Vector Machines through hashing. Previous research works on scalable SVMs had mainly been dedicated to the training phase with the objective of reducing both the processing time and the memory requirement of the solver [30, 121]. Several studies have also proposed methods for improving multi-class SVM efficiency for a large number of categories [40, 17, 119, 36, 67]. Closer to our work, an efficient search method using LSH was proposed in [49] with the objective of efficiently solving hyperplane queries (i.e finding the closest feature points to the hyperplane) in the context of active learning. Our own work rather uses LSH to build efficient hash-based classifiers approximating any linear SVM and converging to the exact classification performance. The core idea is that any binary linear classifier defined as  $h(\mathbf{x})$ :

$$h(\mathbf{x}) = \text{sgn}(\omega \cdot \mathbf{x} + b) \quad (2.15)$$

can be approximated by a Hash-based classifier defined as :

$$\begin{cases} \hat{h}(\mathbf{x}) = \text{sgn}(r_{\omega,b} - d_H(\mathbf{F}_D(\mathbf{x}), \mathbf{F}_D(\omega))) \\ r_{\omega,b} = \frac{D}{\pi} \cos^{-1} \left( \frac{-b}{\|\omega\|} \right) \end{cases} \quad (2.16)$$

where  $d_H$  is the Hamming distance and  $\mathbf{F}_D$  is a  $D$ -length binary LSH function composed of the concatenation of  $D$  unitary LSH function  $f : \mathbb{R}^d \rightarrow \{-1, 1\}$  such that:

$$f(\mathbf{x}) = \text{sgn}(\mathbf{w} \cdot \mathbf{x})$$

where  $\mathbf{w} \in \mathbb{R}^d$  is a random variable distributed according to  $p_w = \mathcal{N}(0, \mathbf{I})$  in the original feature space. Intuitively, that means that the initial linear classifier  $h(\mathbf{x})$  requiring the computation of high-dimensional and costly inner products  $(\omega \cdot \mathbf{x})$  can be approximated by efficient Hamming distances computations  $d_H(\mathbf{F}_D(\mathbf{x}), \mathbf{F}_D(\omega))$  on compact binary hash codes. In the paper [165], we did theoretically prove that the classification results of the hash based classifier  $\hat{h}(\mathbf{x})$  converges to the results of the exact SVM classifier  $h(\mathbf{x})$  as the number  $D$  of binary hash functions tends to infinity.

Applying a Hash-based SVM classifier  $\hat{h}(\mathbf{x})$  with a brute-force scan of the features to be classified does not change the prediction complexity, which is still  $O(N)$  in the number of images to classify. Performance gains are more related to memory usage and the overall speed when a very large number of classifiers have to be applied simultaneously (which is often the case when dealing with a large number of classes). Within our implementation, the space requirement of a single

exact classifier was actually  $S_{SVM} = 12 + 8d$  bytes when using a double precision for  $\omega$  and  $b$ . On the other hand, the space requirement of a single Hash based SVM classifier was  $S_{HBMS} = 6 + \frac{D}{8}$  bytes. With the typical values  $d = 1000$  and  $D = 256$  bits used in our experiments, the memory usage required by Hash based SVM classifiers is about 200 times lower than the exact classifiers. The second main advantage is to speed up the computation of the classification function. A Hamming distance on typically  $D = 256$  bits can be much faster than an inner product on high-dimensional data with a double precision (particularly when benefiting from *pop-count* assembler instructions).

Our experiments consisted in approximating a one-against-one linear multi-class SVM with a large number of categories and a large dataset to be classified (i.e. ImageNet-BOF dataset [26] provided within the ImageNet PASCAL VOC Large Scale Visual Recognition 2010 Challenge). Figure 2.7 illustrates the convergence of the Hash-based multi-class SVM classifier for  $C = 300$  categories. About  $D = 4096$  bits would be required to approximate well the original results. With this setting, the average prediction time of the hash-based SVM is about 3.5 times faster than the exact linear SVM. Smaller hash codes and better speed-ups can however be used through a filter-and-refine strategy that first selects a set of  $c$  candidate classes with a hash-based one-against-one multi-class classifiers and then refine the results by computing the real uncompressed classifiers on the remaining classes. Note that  $c$  is supposed to be relatively small compared to the whole number of classes  $C$  so that the cost of the refinement step ( $O(c(c-1))$ ) is negligible compared to the cost of the filtering step ( $O(C(C-1))$ ). In our experiments, using this strategy allowed the hash-based classifier to be more than two orders of magnitude faster than the exact classifier with minor losses in quality.

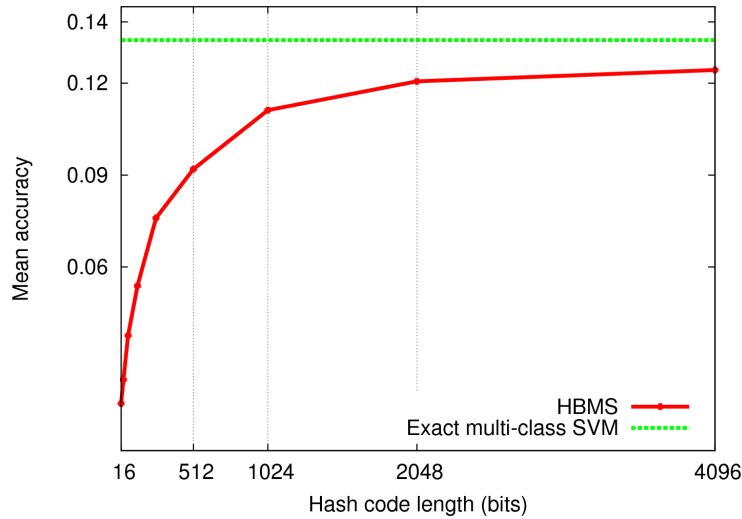


Figure 2.7: Exact Multi-class SVM vs Hash-based Multi-class SVM (HBMS)

## 2.5 Distributed KNN-graph approximation via hashing

*This work [170, 169] was mainly achieved in the scope of the PhD of Riadh Mohamed Trad under my supervision and the scientific direction of Nozha Boujemaa.*

Given a set  $\mathcal{X}$  of  $N$  objects, the KNN graph (KNNG) consists of the vertex set  $\mathcal{X}$  and the set of edges connecting each object from  $\mathcal{X}$  to its  $K$  most similar objects in  $\mathcal{X}$  under a given metric or similarity measure. Efficiently constructing the KNNG of large and high dimensional datasets is crucial for many applications with feature-rich objects, such as image clustering or event mining in multimedia collections. At first, the KNNG problem can be seen as a nearest neighbors search problem where each data point itself is issued as a query. The brute-force approach, consisting in  $N$  exhaustive scans of the whole dataset, has the cost  $O(N^2)$ . Its practical usage is therefore limited to very small datasets. Building a high-dimensional index and iteratively processing the  $N$  items in the dataset with approximate Nearest Neighbors search techniques is an alternative option that is more efficient but that is still not optimal. Close query features are indeed processed independently whereas they could share some replicated processing's (e.g. for selecting neighboring buckets of the one containing two similar query features). Some recent studies therefore focus more specifically on the KNNG construction problem as a whole, i.e. not by processing iteratively and independently the  $N$  top- $K$  queries, but trying to exploit shared operations across all queries. In the text retrieval community, recent studies [7, 123] focused on the  $\epsilon$ -NNG construction in which one is only interested in finding pairs whose similarity exceeds a predefined threshold. In [123], the authors present a permutation based approach both to filter candidate pairs and to estimate the similarity between vectors. However, their approach is only applicable on sparse vectors. Recently, Dong et al. [27], proposed the NN-*Descent* algorithm, an approximate KNNG construction method purely based on query expansion operations and applicable to any similarity measure. Their experiments show that their approach is more efficient than other state-of-the-art approaches. However, designing an efficient distributed version of this method is not trivial, limiting its practical scalability as it requires the entire dataset to be loaded into a centralized memory.

In this work, we investigated the use of high dimensional hashing methods for efficiently approximating the KNNG, notably in distributed environments. The raw principle is simply to construct  $L$  hash tables in parallel and to count the number of collisions of any pair of items in all hash tables. When using a LSH function, we actually did show that using the theoretical *expected* number of collisions as a metric preserves the topology of the original feature space. More formally we have:

$$\mathbf{q} \cdot \mathbf{v}_1 < \mathbf{q} \cdot \mathbf{v}_2 \Leftrightarrow E[\hat{n}_{\mathbf{q}, \mathbf{v}_1}] < E[\hat{n}_{\mathbf{q}, \mathbf{v}_2}] \quad (2.17)$$

where  $\mathbf{q}, \mathbf{v}_1$  and  $\mathbf{v}_2$  denote any feature vectors in  $\mathcal{X}$  and  $\hat{n}_{\mathbf{q}, \mathbf{v}_i}$  denotes the *empirical* number of collisions between  $\mathbf{q}$  and  $\mathbf{v}_i$  in the  $L$  hash tables.

Unfortunately using this scheme with a classical LSH function does not provide as good result as the NN-descent method of Dong et al. [27]. On the other side, we did show in [170] that using RMMH [147] instead does consistent speed-ups, the main reason being that the resulting hash tables are much more balanced and that the number of resulting collisions can be greatly reduced without degrading quality. Our experiments show that our hash-based method when using RMMH slightly outperforms the state-of-the-art method of Dong et al. [27] in centralized settings while being more efficiently scalable given its inherently distributed design. We further improved the load balancing of our method in distributed settings by designing a parallelized local join algorithm, implemented within the MapReduce framework to allow an easy deployment in the cloud. Our largest experiment was achieved on a dataset of 828,902 793-dimensional visual features extracted from a

collection of Flickr images and took about 10 minutes with 16 CPU's. To the best of our knowledge, no other work has reported full KNN graphs results on such large datasets.



## Chapter 3

# Matching-based visual information retrieval in large multimedia collections

One of the guideline of my research work on visual information retrieval has been to stick on matching-based approaches involving rich image representation although this imposes challenging issues regarding their scalability. Year after year, I addressed different problems ranging from content-based copy detection to image classification but I always tried to benefit from my research on high-dimensional features indexing to design fine-grained yet scalable methods.

### 3.1 Robust video copy detection in huge archives

*This work was mainly carried on within my PhD thesis (2003 -2005) performed in the context of an industrial contract with the French National Audiovisual Institute (INA) under the supervision of Olivier Buisson and the scientific direction of Carl Frélicot (Professor at the University of La Rochelle)*

Content-based retrieval methods dedicated to *(near-)copies* detection have emerged at the beginning of the 2000s for copyright protection issues [?, ?, ?, ?, 149, ?]. Contrary to the *watermarking* approach, the identification of a document is not based on previously inserted marks but on low-level visual features extracted from the content itself. As illustrated by Figure ??, the main advantage of the content-based approach is that copies of already existing materials can be detected even if the original document was not marked or was strongly altered by successive transformations [148]. Beyond copyright issues, the key objective is to try reconstructing the whole diffusion context of a given visual document in order to derive new informative content from the resulting graph [?]. Google image search engine, for instance, makes use of such automatic linking principle to identify the most popular images and rank them accordingly [?, ?].

Now, the main challenge to be solved is the *robustness* to the potential transformations that can alter the original document. This includes light alterations such as encoding artifacts or photometric corrections, but also more severe ones such as resizing, cropping or external data insertions (texts, logos, pictures-in-pictures, etc.). Inspired by the seminal work of Schmid et al. [?] on the use of local visual features for image retrieval, our introductory work on content-based video copy de-

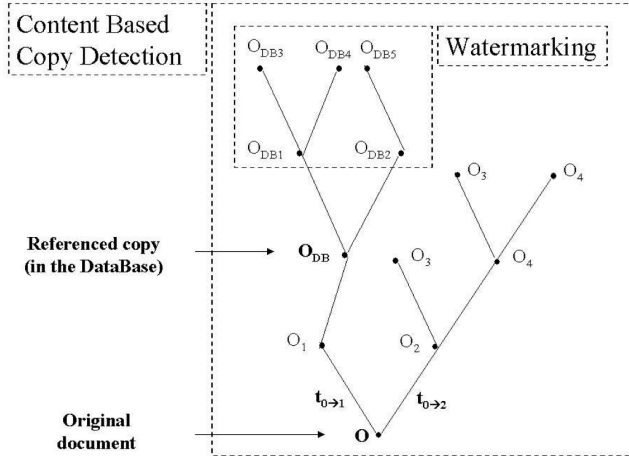


Figure 3.1: Tree of all copies of an original document  $O$ .

tection [149] was the first one making use of *local* visual features extracted around stable interest points in videos. Previous methods such as [?, ?, ?, ?] rather used block-based correlation approaches and sequence matching techniques. The *ordinal signature* [?], a global visual feature based on the ranking of blocks descriptions, did notably get a certain success [?] thanks to its simplicity and compactness. We however did prove the higher robustness of local features based approaches in several challenges and evaluations [157, 158, 156, 155], particularly for the transformations altering the global geometric structure of the video frames (cropping, scaling, picture-in-picture, external data insertion, etc.). Figure ?? displays two real detections achieved by our method that could not have been detected with global approaches.

More precisely, our video copy detection framework (introduced in [149, 150] and finalized in [141]), was based on *spatio-temporally shifted* local features. Each local feature  $\mathbf{X}_t(\mathbf{P})$  extracted at time  $t$  around an interest point  $\mathbf{P}$  (typically Harris points [?]), is actually the concatenation of  $m$  sub-features extracted at  $m$  shifted spatio-temporal positions around  $(t, \mathbf{P})$ :

$$\mathbf{X}_t(\mathbf{P}) = \{\mathbf{x}_{t+\delta_i}(\mathbf{P} + \Delta_i)\}_{1 < i < m}$$

where the spatial offsets  $\Delta_i$ 's are chosen so as to reduce the correlation between the sub-features for the most common camera motions (static shot, tracking shot, panning). In further work [143, 142], we did show that using such *non local* differential operators can also improve the robustness of classical differential operators in the case of static images. By introducing a spatial separation between the excitatory and the inhibitory lobe of differential operators, it is actually possible to build new differential local features more robust to small imprecisions in the position or the orientation of the described patches. We notably achieved comparable copy detection accuracies than the popular SIFT features [?] (based on gradient orientation histograms) while using 6 times less components for each local feature. Note that the principle of extracting dissociated dipoles at the local level was successfully re-investigated later on within the popular BRIEF features [?]. They however used randomized position shifts to improve the independence of the components and an additional binarization step to compress the features.

Although using local visual features instead of global ones provides a better robustness, it has the disadvantage to produce much more feature vectors to be



Figure 3.2: Example of two content-based copy detections involving severe alterations between the original video (right) and the broadcasted ones (left)

managed and searched. Scaling such methods to very large video datasets and real-time monitoring contexts was therefore one of the main key issue of our work on video copy detection. As already discussed in section 2.2, our main contribution in this regard did concern (i) the introduction of a hashing scheme to efficiently index the local features and (ii), the introduction of a probabilistic multi-probe nearest neighbors search algorithm to speed-up the matching. We did confirm in [141] that the search time of the proposed method was sub-linear in the size of the dataset resulting in up to three order of magnitudes speed-ups over a naive sequential scan of the dataset. This allowed us to monitor real-world TV channels continuously with more than 30,000 hours of video archives in the referenced dataset (actually two orders of magnitude larger than the other experiments in the literature at that time). Furthermore, a nice trick in the context of content-based copy detection, is that the search model used by the probabilistic multi-probe algorithm can be trained from real transformations of video samples randomly selected from the dataset. The estimated probability used to select the most relevant buckets to be visited consequently reflects the probability that they contain some distorted versions of the query one according to some targeted transformations. We referred to this principle as *distortion-based similarity search* [141]. It allows to more accurately control the quality of the generated raw visual matches while optimizing the fraction of the dataset to be scanned.

Simply voting on the raw matches produced by the distortion-based similarity search is however still insufficient to achieve high precision rates at the video level. Individual local features are actually not discriminant enough and the resulting pair-wise matches contain a large fraction of false alarms, notably because of the burstiness phenomenon (i.e. a given visual element appears more times in an image than a statistically independent model would predict [?]). To eliminate many of such false matches, we suggested in [149] to rely on the temporal arrangement of the local features within a sliding window of a few seconds length. Thanks to the temporal positions of the raw visual matches belonging to the window, we first robustly estimate the most likely temporal shift between the query sequence and each retrieved one. We therefore used the following robust M-estimate of the



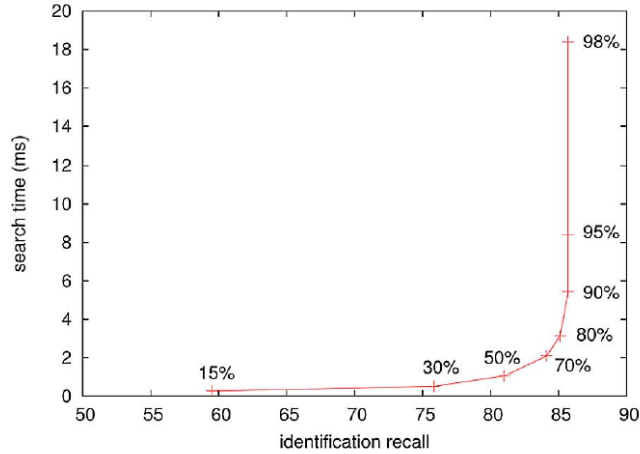


Figure 3.3: Time vs. quality of our content-based copy detection framework for varying values of the nearest neighbors search quality control parameter  $\alpha$

temporal offset  $\hat{\delta}t$ :

$$\hat{\delta}t = \operatorname{argmin}_{\delta t} \sum_j \rho_{\theta}(\delta t - \delta t_j) \quad (3.1)$$

where  $\rho_{\theta}$  is the  $\rho$ -function of the Tukey's biweight M-estimator [?] parameterized with a deviation  $\theta$  (allowing to penalize the contribution of the outlier matches, i.e. the ones with too much inconsistent temporal shifts compared to the evaluated solution). Once the best temporal shift  $\hat{\delta}t$  is estimated for each targeted video  $V$ , we then compute the temporally consistent matching score of  $V$  as:

$$S(V) = \sum_j \mathbb{1} \left\{ \left| \hat{\delta}t - \delta t_j \right| < \theta \right\} \quad (3.2)$$

where  $\mathbb{1} \{.\}$  is an indicator function equals to 1 if the assumption in the braces is true and zero otherwise.

To illustrate the rationale of our complete video retrieval scheme, Figure ?? represents the average search time with respect to the recall of the method (for a set of 5 transformations with various randomized parameters as described in [141]). Each measurement on the graph corresponds to a specific value of the search quality control parameter  $\alpha$  of our *distortion-based similarity search* scheme (ranging from 15% to 98%). The recall values were determined at constant precision (i.e., a ROC curve has been built for each point). The curve illustrates the power of the approximate search paradigm: the overall recall remains almost constant when the probability to find individual relevant neighbors decreases from  $\alpha = 98\%$  to 70%, whereas search is more than three times faster. For smaller values of the probability, the recall starts to degrade more significantly. It is, however, interesting to see that the recall is still 60% when only 15% of the signatures are expected to be retrieved. It is also important to emphasize that the distortion-based probabilistic search paradigm is always more efficient than reducing the number of queries. We actually did prove in [141] that searching a fixed rate of the candidate local features is always slower than searching all the features with the appropriate approximation (for the same quality). Overall, our method and its variants [?, 141, ?] did obtain the best performance in several content-based video copy detection benchmarks [157, 158, 156, 155]. In [141] we

notably extended our temporal verification step to a spatio-temporal verification and did show that it can significantly improve the performance in the case of strong attacks such as *picture-in-picture* (a small video inserted in a larger one).

In [144], we proposed another specific improvement with the objective to reduce the surprisingly much higher number of false alarms that appeared in Japanese TV content. The omnipresence of big textual characters and logos in these contents actually attracts many local features and generates a lot of irrelevant matches. To remove such indiscriminant features, we introduced an efficient approximate kernel density estimation technique based on hashing and probabilistic multi-probe queries (inspired by our probabilistic multi-probe search algorithm [150]). We did show that very accurate density estimates can be computed through that method while it is several orders of magnitude faster than an exhaustive scan of the dataset. It enabled the real-time density estimation of each local feature of a query video clip so that only the top-k less frequent in the dataset (i.e. the top-k most discriminant) can be selected for the temporally consistent vote (cf. equation ??). Copy detection experiments did show the qualitative and quantitative benefits of the pruned features.

## 3.2 Matching-based objects retrieval in large image collections

*This work [146, 159] was achieved in collaboration with Olivier Buisson (INA) & the Belgium press agency Belga in the context of the EU project VITALAS. It was further improved and experimented in the context of the PhD of Pierre Letessier [?].*

The predominant model for content-based image retrieval, as well as for image classification, is based on the pooling of local visual features into global image representations such as the popular Bag-of-Words representation (BoW) [101]. Its principle is to first train a so called visual vocabulary thanks to an unsupervised clustering algorithm computed on a given training set of local features. The produced partition is then used to quantize the visual features of a given new image into *visual words* that are aggregated within a single high-dimensional histogram. This method remains a key concept in many recent methods although the raw initial scheme of [101] is now outperformed by several alternative new schemes [64, ?, ?, 111, 51, 51]. As it relies on vector quantization, the raw BoW representation is actually affected by quantization errors. Very similar visual features might be split across distinct clusters whereas more dissimilar ones might be affected to the same visual word. This results in both mismatches and potentially irrelevant matches. To alleviate this problem, several improvements have been proposed in the literature. The first one consists in expanding the assignment of a given local feature to its nearest visual words [?, 88, 111, 57]. This allows reducing the number of mismatches without degrading much the encoding time. Other researchers have investigated alternative ways to avoid the vector quantization step, using sparse coding [?] or locality-constrained linear coding [114]. Such methods optimize the affectation of a given local feature to a few number of visual words thanks to sparsity or locality constraints on the global representation. A third alternative is to use aggregation-based models such as the improved Fisher Vector of [?] or the VLAD encoding scheme [51]. Such methods do not only encode the number of occurrences of each visual word but also encode additional information about the distribution of the descriptors by aggregating the component-wise differences.

As powerful as global image representations are for capturing generative visual

patterns and efficiently indexing them, they also have some drawbacks. First of all, they are agnostic to the localization and the spatial arrangement of the local features so that they cannot well characterize the geometric structure of the objects contained in the images. Secondly, they are very sensitive to highly cluttered background since the objects of interest are not described independently from the background. Thirdly, the generative aggregation step also results in a loss of discrimination. The less frequent visual patterns are highly penalized whereas they are often expected to be more discriminant than the most repeated ones in the dataset (such as textures, texts, etc.).

An alternative line of research therefore consists in sticking on the matching of the individual low level local features [59, ?, 89, 52, ?, 94][143, 146, 159] and to aggregate the results afterwards. Such *matching-based* image retrieval schemes are primarily aimed at retrieving instances of a given query object such as a building, a manufactured object or a logo. They are notably often associated with the *query-by-window* search paradigm, in which the user can select an object of interest in a query image typically by drawing a bounding box. As the number of local features to be managed and searched is usually huge, efficient *matching-based* image retrieval schemes usually rely on local descriptors embedding and compression methods (e.g. Hamming Embedding [52], data-dependent hashing [118, 147] or product quantizer [54]) and scalable indexing structures (such as an inverted index on visual words [89], hash tables [59, 146] or randomized trees [79]). This allows matching online all local descriptors one by one in the full index and favors a more precise matching of small objects in highly cluttered pictures. The next subsections present some of our contributions within this line of research.

### 3.2.1 Spatial filtering of approximate KNN

Using geometry can significantly improve the precision of retrieval systems, particularly in the case rigid or slightly deformable objects. As two views of a rigid object are actually related by epipolar geometry, the raw noisy visual correspondances between individual local features can be filtered by geometric rules [89, 52, 88, ?, 4][146]. Such *spatial verification* procedure typically estimates a transformation between the query region and each target image, based on how well its feature locations are predicted by the estimated transformation [89]. Images are then re-ranked based on the discriminability of the spatially verified correspondances. Note that this principle was first introduced in the context of near-duplicate and near-copies retrieval [59][143] for which the class of geometric transformations can be even more restricted. As discussed in [148], it is also somehow a transposition / extension of the *temporal verification* method that we introduced in [149] beforehand. Our equations ?? and ?? of section ?? can actually be easily extended to the spatial case [143, 146] (or spatio-temporal case as in [148]) by replacing the single temporal shift parameter  $\delta t$  by a more complex spatial or spatio-temporal transformation model. The difference however is that the model is much more complex to estimate because of the larger number of parameters (from 4 to 8 depending on the used class of transformation) and the resulting polynomial number of candidate solutions to be evaluated.

The standard solution to perform such estimation is to use the RANSAC algorithm [?]; it consists in generating transformation hypotheses using a minimal number of visual correspondances and then evaluating each hypothesis based on the number of *inliers* among all features under that hypothesis. The main advantage of the RANSAC algorithm is that it is robust to the presence of a high number of *outliers* which makes it suitable to deal with the large numbers of false

alarms that are generated by the raw visual matching of the local features. As the RANSAC algorithm can be rather slow, an efficient variant, LO-RANSAC, was proposed by Chum et al. [?] and has been proved to provide consistent speed-ups in many image retrieval frameworks [89, 88, ?, 4]. It involves generating hypotheses of an approximate model thanks to the shape information provided with the affine-invariant image regions from which the visual features were extracted. With this method, an hypothesis can be generated with only a single pair of corresponding features whereas two or three are required when using only the feature positions. This greatly reduces the number of possible hypotheses which need to be considered by the RANSAC algorithm and significantly speeds up the spatial verification procedure. An even faster strategy [?], consists in considering only the shape information of the image regions, without exploiting the positions of the features at all. A rough approximation of the best transformation can then actually be estimated by a Hough-like voting strategy on the quantized differences of the characteristic orientation and scale of each visual correspondance. Using this so-called *weak geometry* method allows trading quality for time and is the only acceptable solution when dealing with huge image sets and real-time contexts (e.g. a search engine working on billions of images).

Similarly to the LO-RANSAC algorithm, the spatial verification we use in our own work [146, 159, 161, 139, 163] is a variant of the RANSAC algorithm making use of weak geometry rules generated from the region shape characteristics. We however do not use the weak geometry to directly generate an hypothesis from a single visual correspondance. We rather use it to filter the exact hypothesis generated by the classical RANSAC algorithm. Concretely, if we restrict our class of transformations to rotation and scaling, the RANSAC algorithm can generate an hypothesis from any pair of visual correspondances. To quickly decide whether this hypothesis is relevant or not, we check its consistency with regard to the two approximate hypothesis generated from the shape characteristics of each visual correspondance. If any of the two approximate models does not fit the RANSAC hypothesis, we reject that solution without computing the costly consensus phase. In practice, up to 99% of the RANSAC hypothesis can be rejected in that way.

Another major difference between our method and the ones in [89, 88, ?, 4] is that we use the ranking of the visual correspondances to further improve the matching. Our retrieval framework does actually not rely on the popular bag-of-words model to generate the raw visual correspondances but on a more accurate approximate KNN search algorithm using our hash-based methods described in Chapter 2 (RMMH [147] + multi-probe search [145]). The main benefit is that the precision of our raw visual matches is already much better than the ones produced by the bag-of-words model (based on vector quantization). The RANSAC algorithm therefore works on less correspondances and less false alarms. Another benefit is that each raw visual correspondance  $\{\mathbf{x}_q, \mathbf{x}_i\}$  is associated with a rank  $r_q(\mathbf{x}_i)$ . This allows two things: (i) to restrict the generation of the hypothesis of the RANSAC algorithm to the best match of each query feature  $x_q$  in the targeted image. The number of evaluated hypothesis is consequently reduced, particularly in the presence of numerous repeated visual patterns (the burstiness phenomenon [?]) (ii) the ranking can be used in the computation of the final score by weighing the contribution of each inlier according to its rank in the whole dataset. Closest points are then favored to the detriment of the farthest ones, independently from the feature space density in the neighborhood of  $x_q$ . Finally, the geometrically consistent score of a

retrieved image  $I$  is computed as:

$$S_Q(I) = \sum_q \mathbb{1} (\| \mathbf{P}_q - (\mathbf{A}\mathbf{P}_q^I + \mathbf{B}) \| < \theta) \cdot f(r_q(\mathbf{x}_q^I)) \quad (3.3)$$

where  $(\mathbf{A}, \mathbf{B})$  are the parameters of the best transformation estimated by the RANSAC algorithm for the image  $I$ ,  $\mathbf{P}_q$  and  $\mathbf{P}_q^I$  are the spatial positions of respectively the query feature  $\mathbf{x}_q$  and its best match  $\mathbf{x}_q^I$  in  $I$ ,  $f(\cdot)$  is a decreasing weighting function on the rank  $r_q(\mathbf{x}_q^I)$  (typically the inverse or a linearly decreasing function),  $\mathbb{1}\{\cdot\}$  is an indicator function equals to 1 if the assumption in the braces is true and zero otherwise.

### 3.2.2 Query expansion with a contrario adaptative thresholding

In [?], Chum et al. introduced a new effective retrieval paradigm, referred to as visual *query expansion* by analogy to the so-called text retrieval paradigm [?]. The principle is that a number of highly ranked documents from the original query are reissued as a new query to improve performance. However, as mentioned by Chum et al. [?], improvements can be achieved only if no false positives (or very few) are included in the expanded query. To achieve this, Chum et al. suggested the use of a spatial verification criterion derived by thresholding a geometric consistency score. However, they do not provide a way to estimate this crucial threshold and simply suggest a fixed hand tuned threshold equal to 20 inliers. This threshold however depends on many factors including global factors, e.g. average number of features per image, redundancy of the features in the dataset, parameters of the retrieval algorithm (e.g spatial threshold  $t$  of accepted inliers), etc. It also depends on factors varying for each query, such as the size of the query, the redundancy of the features in the query, the spatial distribution of the query features, etc. In [146], we proposed to solve this issue by an *a contrario adaptive thresholding* method.

The a contrario framework was initially proposed by Desolneux et al. [?] in order to group low-level visual features. The basic principle is to detect events in images a contrario to a random situation modeled by a *background model*. Usually, the unlikeliness of a given event is ensured by controlling the expected number of false detections. This generic approach has been applied with success to, among other things, the detection of alignments [?], contrasted edges, vanishing points, and grouping [?], or shape matching [?]. Closer to our work, in [?], Rabin et al. proposed an a contrario matching of SIFT like features. However they aimed at thresholding directly the SIFT feature distances and not a global geometric consistency score as in our case. The rationale of our a contrario adaptative thresholding method is actually to transform the original geometrically consistent scores  $S_Q(I)$  of equation ?? in a new a contrario normalized score  $\hat{S}_Q(I)$ . The normalization is based on an estimation of the false alarms distribution  $\hat{N}_{fa}(S)$  with respect to the random variable  $S = S_Q(I)$ . According to Equation ??,  $S_Q$  mostly depends on the spatial coordinates of the visual correspondances. High  $S_Q$  scores are thus directly related to the statistical dependence between the spatial positions of the query and the matched features. We thus define our a contrario background model by the probability function  $\hat{p}_{fa}(S)$  of the variable  $S$  under the hypothesis  $\mathcal{H}_0^Q$  that  $\mathbf{P}_q$  and  $\mathbf{P}_q^I$  are mutually independent random variables for all  $q$ :

$$\hat{p}_{fa}(S) = Pr \left[ S_Q(I) = S \mid \mathcal{H}_0^Q \right]$$

The cumulative distribution function  $\hat{P}_{fa}(S)$  can be obtained by:

$$\hat{P}_{fa}(S) = \int_{s=0}^S \hat{p}_{fa}(s)$$

We finally keep as normalized score  $\hat{S}_Q(I)$  an estimation of the results precision according to  $\hat{P}_{fa}(S)$ :

$$\hat{S}_Q(I) = \frac{\#\{I' \mid S_Q(I') > S_Q(I)\} - N \cdot \hat{P}_{fa}(S_Q(I))}{\#\{I' \mid S_Q(I') > S_Q(I)\}}$$

where  $N \cdot \hat{P}_{fa}(S_Q(I))$  is the expected number of false alarms having a score higher than  $S_Q(I)$  and  $\#\{I' \mid S_Q(I') > S_Q(I)\}$  is the actual number of retrieved images having a score higher than  $S_Q(I)$ . The difference between that two quantities is thus the expected number of correct results.

In practice, we estimate the cumulative probability function  $\hat{P}_{fa}(S)$  for each query  $Q$  by a Monte Carlo simulation. To generate independent spatial matches according to the hypothesis  $\mathcal{H}_0^Q$ , we simply randomize the spatial positions of the query features  $\mathbf{P}_q$  and we keep the matched positions  $\mathbf{P}_q^I$  unchanged. More precisely, we affect to a given query feature  $\mathbf{x}_q$  a new spatial position randomly selected among the other points positions of the query. Compared to a purely uniform random generation of points position this method has the advantage to preserve some prior knowledge about the points distribution, such as bounds and principal orientations. We then recompute our spatial verification algorithm as described in the previous section and we estimate  $\hat{P}_{fa}(S)$  by counting the number of results having a score  $S_Q$  greater than  $S$ . To limit the estimation bias due to the presence of correct images in the random results list, we keep only in the count the results having a score higher than the one they obtained with the normal query. To reduce the noise of the estimated distribution, the Monte Carlo simulation can be ran several times for each query. But in practice a single iteration already provides a good estimation of the false alarms scores. As an illustration of the relevance of the method, Figure ?? shows the real false alarm distribution of two queries of the OxfordBuilding dataset compared to the estimated distributions  $\hat{P}_{fa}(S)$ . The figure shows that the accuracy of the estimated distribution is very good although the distribution varies significantly between both queries.

The query expansion method we implemented in on top of our a contrario normalized score  $\hat{S}_Q(I)$  is then very similar to the transitive closure method described in [?]. The main difference is that that geometrically verified results are selected by a threshold  $\hat{S}_t$  on  $\hat{S}_Q(I)$  instead of a threshold on  $S_Q(I)$ . As  $\hat{S}_Q(I)$  is directly an estimation of the false alarm rate, the thresholding is much more intuitive and fully adaptive to the query. It guaranties that the expected percentage of false alarms included in the expanded query will be lower than  $1 - \hat{S}_Q(I)$ . Now the query expansion itself works as follows: all retrieved images having a score  $\hat{S}_Q$  score higher than  $\hat{S}_t$  are inserted in a priority queue keyed by  $\hat{S}_Q$ . Then, an image is taken from the top of the queue and the region corresponding to the original query region is used to issue a new query. Verified results of the expanded query that have not been inserted to the queue before are inserted (again in the order of  $\hat{S}_Q$ ).

### 3.2.3 Comparison to state-of-the-art

The table ?? presents a comparison of the performance of the best performing retrieval systems on the popular *OxfordBuildings* dataset introduced by [89]) and

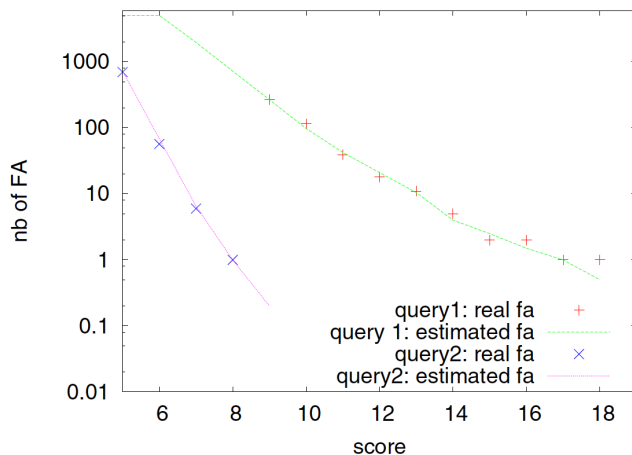


Figure 3.4: Real vs. estimated false alarms distribution for two queries of the OxfordBuilding dataset.

on a more challenging one, *BelgaLogos*, that we introduced in [146]. Note that the performance of our method did progress a lot from the initial version in [146] to the more recent one evaluated in [?]. The performance gain is due to several improvements: (i) the extraction of SIFT features at a finer resolution ( $O_{min} = -1$  instead 0) (ii) the use of RMMH instead of LSH for the partitioning of the feature space (iii) the use of the Hamming distance on RMMH hash codes (1024 bits) instead of the  $L_2$  distance on the original SIFT features (iv) the use of the RANSAC algorithm with weak geometry filtering rules as discussed in subsection ?? (v) the cross-validation of some of the parameters including the number  $k$  of nearest neighbors of each local feature and the RANSAC parameter  $\theta$ . Without query expansion and without using an external dataset for the training of the partitioning, our method achieves better results than the best ones reported in the literature on both datasets. Using query expansion, the performance of our method (0,896) is slightly lower than the one of [4] but it is important to notice that we use significantly less numerous and less effective SIFT features (computed on DoG points [?] and not on Hessian affine regions [?]).

### 3.3 Scalable mining of small visual objects

*This work [159, 160] was achieved in the scope of the PhD of Pierre Letessier [?] under my supervision, the one of Olivier Buisson (INA), and the scientific direction of Nozha Boujemaa (Inria).*

Automatically linking multimedia documents that contain one or several instances of the same visual object has many applications including: salient media events detection, content-based filtering recommendation, web browsing, etc. Whereas efficient methods such as the ones discussed above now exist for searching rigid objects in large collections, discovering them from scratch is more challenging in terms of scalability, particularly when the targeted objects are rather small compared to the whole visual content. It is for instance noticeable that most previous work on object discovery [?, ?, ?, ?, ?] were evaluated on the popular Oxford buildings dataset [89], where the targeted objects (buildings) occupy a very large fraction of the images themselves. One of the claim of this work is that the complexity of

System	Visual features	Partitioning	Indexing- & Search	Spatial checking	Query Expansion	mAP Oxford Buildings		mAP Belgalogos	
						$\overline{QE}$	$QE$	$\overline{QE}$	$QE$
Philbin et al. [88] (2008)	Hessian affine [?] + SIFT [?]	AKM [?]	Bow + tf-idf + SA	LO-RANSAC	Average expansion [?]	0,731	0,825		
Our method (2009) [146]	DoG + SIFT [?] ( $O_{min} = 0$ )	LSH [?]	APMPLSH [145]	RANSAC	a contrario adaptive thresholding	0,608	0,807	0,208	0,341
Perdoch et al. [?] (2009)	Perdoch [?]	AKM [?]	Bow + tf-idf + SA	LO-RANSAC	Average expansion [?]	0,846	0,916		
Jegou et al. [?] (2010)	Hessian affine [?] + SIFT [?]	K-means	Inverted List + HE [52]	weak-geometry	Transitive Closure [?]	0,66	0,74		
Arandjelovic et al. [4] (2012)	Perdoch [?] + RootSIFT [4]	AKM [?]	Bow + tf-idf	LO-RANSAC	Discriminative QE		<b>0,929</b>		
Revaud et al. [?] (2012)	Hessian affine [?] + SIFT [?]	K-Means	Inverted List + HE [52]	LO-RANSAC + Correlation-based burstiness	none			0,414	
Our method (2012) [?]	DoG + SIFT [?] ( $O_{min} = -1$ )	RMMH [147]	APMPLSH [145] + RMMH-HE	LO-RANSAC	a contrario adaptive thresholding	<b>0,851</b>	0,896	<b>0,419</b>	<b>0,51</b>

Table 3.1: Comparison of content-based retrieval systems on Oxford Buildings et Belgalogos datasets

Bow = Bag of visual Words, tf-idf = term frequency - inverse document frequency, SA = soft assignment, HE = Hamming Embedding,  $QE$  = Query Expansion,  $\overline{QE}$  = Without Query Expansion, mAP = mean Average Precision, P. ad hoc/indep = Partitioning trained on the test data (ad hoc.) or not (indep)



mining repeated visual objects is closely related to the relative size of the targeted objects and to their frequency of occurrence. Therefore, the complexity of classical mining algorithms for discovering repeated item sets is known to be highly related to their frequency. To illustrate the variety of problems, let us compare the objects considered in the Oxford buildings dataset to those considered in the BelgaLogos dataset [146]. Figure ?? shows the repartition of the instance sizes for these two datasets. The measure used here to analyze the sizes of instances is the number of SIFT features falling into the bounding boxes of the objects in the ground truth. We found 29,968,910 features in Oxford buildings, with 6,056,353 belonging to objects in the ground truth, and 38,093,296 descriptors in BelgaLogos, with 184,698 belonging to the annotated logos. We observe that the coverage of objects in Oxford buildings is around 20% while it is only 0.5% in BelgaLogos.

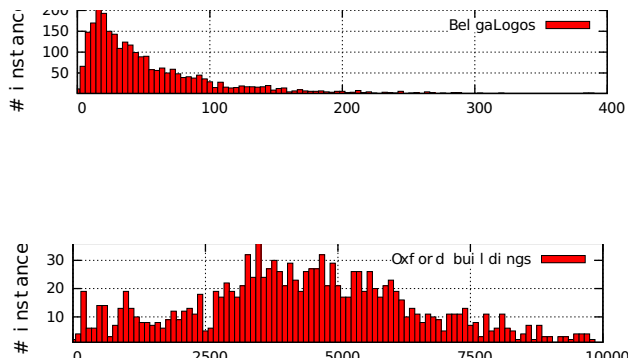


Figure 3.5: Comparison of instances size in BelgaLogos and Oxford buildings

The first contribution of our work was to formally revisit the problems of *mining* or *discovering* rigid objects in an image collection. Let us consider a dataset  $\mathbf{I}$  of  $N_I$  images described by a set  $\mathbf{X}$  of  $N$  local feature vectors  $\mathbf{x}_i$ , each being extracted at position  $\mathbf{p}_i = (I_i, \chi_i, \psi_i)$  where  $I_i$  is the identifier of the image and  $(\chi_i, \psi_i)$  the coordinates of the local feature in the image. Now, we consider a set  $\mathbf{O}$  of objects  $O^m$ , each being represented by  $S_m$  instances  $O_s^m$ . An instance  $O_s^m$  is associated with a unique area  $A_s^m$  (in a single image) and contains a set of local features:

$$\mathbf{X}_s^m = \{\mathbf{x}_i \mid \mathbf{p}_i \in A_s^m\}_{1 \leq i \leq N}$$

We can then introduce some basic definitions:

**Definition - Global cover**

The global cover  $c_{\mathbf{X}}(O^m)$  of an object  $O^m$  is defined by:

$$c_{\mathbf{X}}(O^m) = \frac{1}{N} \sum_{s=1}^{S_m} |\mathbf{X}_s^m| \tag{3.4}$$

It measures the fraction of features in the dataset covered by all instances of a given object.

**Definition - Average cover**

The average cover  $c(O^m)$  of an object  $O^m$  is defined by:

$$c(O^m) = \frac{1}{S_m} \sum_{s=1}^{S_m} \frac{1}{N_{I_s}} |\mathbf{X}_s^m| \quad (3.5)$$

with  $N_{I_s}$  the number of features in the image including  $O_s^m$ . It measures the average fraction that an instance of the object occupies in an image.

**Definition - Frequency**

The *frequency*  $f(O^m)$  of an object  $O^m$  is defined by:

$$f(O^m) = \frac{S_m}{N_I} \quad (3.6)$$

**Definition -  $\{c, f\}$ -frequent object**

An object  $O^m$  is said  $\{c, f\}$ -frequent if:

$$\begin{cases} c(O^m) = c \\ f(O^m) = f \end{cases}$$

Based on this last concept, we defined the two following problems as the main objectives to be solved by objects mining and discovery methods:

**Objects Discovery:** find at least one instance  $O_q^m$  of all  $\{c, f\}$ -frequent objects  $O^m$  such that:

$$\begin{cases} c \geq c_0 \\ f \geq f_0 \end{cases}$$

**Objects Mining:** find all instances  $O_s^m$  of all  $\{c, f\}$ -frequent objects  $O^m$  such that:

$$\begin{cases} c \geq c_0 \\ f \geq f_0 \end{cases}$$

Now the second main contribution of this work was an object mining framework allowing to answer such problems. Its principle is to use a *weighted* and *adaptive* sampling strategy to select candidate image regions to be issued afterwards to a matching-based object search algorithm such as the one discussed in section ?? (e.g. based on large scale approximate KNN search and RANSAC registration). To avoid querying all possible regions of interest while keeping a good coverage of the content, *Sampling* is indeed a simple yet efficient statistical paradigm allowing to yield some knowledge about a population without surveying it entirely. Adaptive weighted sampling is a more advanced paradigm allowing to iteratively update the sampling distribution according to the results obtained during previous iterations. This allows the mining process to progressively focus on unvisited image regions and consequently reduce the number of required probes for achieving a good completeness of the mining. More precisely, the algorithm iteratively samples a candidate feature  $\mathbf{x}_t$  ( $0 \leq t < T$ ) according to a probability mass function  $p_t$  on  $\mathbf{X}$ . A query window centered around  $\mathbf{x}_t$  is then issued to a *precise search* algorithm allowing it to find other instances of the object captured by the query window (if any exist in the dataset). The probability mass of the features belonging to the retrieved instances are then decreased, resulting in a new probability mass function  $p_{t+1}$  to be used in the next sampling iteration.

As the prior distribution  $p_0$  is the initial condition of the Adaptive Weighted Sampling algorithm, the way it is built has a strong impact on the whole performance of the mining process, i.e. on the number  $T$  of iterations required to discover

instances of  $(c, f)$ -frequent objects. We did show in [160] that building an accurate prior distribution can divide by up to 32 the number of required probes compared to a uniform sampling. Admittedly, the speed and the effectiveness of the matching-based search algorithm also have an influence on the performance of the method, but in a more limited way. The overall complexity of the approach actually remains  $O(T)$  whatever the speed of the search algorithm used, e.g. [?, 52, ?][146] (because one single search remains an expensive process). We did prove in the paper that the expected number of probes  $\hat{T}$  is equal to

$$\hat{T} = \frac{1 - c_0(O^m)}{c_0(O^m)} = \frac{1}{c_0(O^m)} - 1 \quad (3.7)$$

in the theoretical case of a perfect search algorithm and approximately equal to

$$\hat{T} = \frac{\log(\epsilon)}{\log(1 - r)} \left( \frac{1}{c_0(O^m)} - 1 \right) \quad (3.8)$$

for a real search algorithm that would return on average only a fraction  $r$  of the instances of an object  $O^m$ . In both cases, the complexity of our mining algorithm is  $O(\frac{1}{c_0(O^m)})$  where  $c_0(O^m)$  is the *prior statistical cover* of the object  $O^m$  defined as

$$c_0(O^m) = \sum_{\mathbf{x}_i \in \{O_s^m\}} p_0(\mathbf{x}_i)$$

This show why the prior distribution  $p_0$  has a strong impact on the whole cost of our mining framework.

The last contribution of this work was therefore to efficiently build a prior probability mass function  $p_0$  on  $\mathbf{X}$  to be passed to the random sample and search algorithm. As stated before, the objective is that  $p_0(\mathbf{x}_i)$  reflects as much as possible the likelihood that a given local feature  $\mathbf{x}_i$  belongs to a  $(c, f)$ -frequent object. In other words, we attempt to maximize  $c_0(O^m)$  for all  $(c, f)$ -frequent objects in the dataset. Rather than using visual saliency measures computed at the image level as in our first paper [159], we proposed in [160] to build much more effective prior distributions based on a two-stage hashing scheme working first at the visual level, and then at the geometric level. The developed algorithm mainly relies on collisions frequency in hash tables making it scalable and easily distributable if needed. Its cost is yet much higher than simple image-based priors but the complexity reduction of the sampling-and-search phase still makes it widely profitable. More precisely, the algorithm includes the four following steps:

1. Construction of a set of visual hash tables based on SIFT features [?] and RMMH [147] hash function.
2. Visual collisions filtering based on a max-bound of the intra-image collision frequency (to favor unicity of the features), a min-bound on the inter-tables collision frequency (equivalent to a range in the original feature space), a KNN filtering to reduce the impact of ambiguous visual features that are present many times in the dataset (typically texts).
3. Weak Geometry (WG) hashing of the remaining pairs of features. For each candidate visual match  $(\mathbf{x}_q, \mathbf{x}_m) \in \mathbf{Z}_Q$  we compute the following WG attributes vector:

$$\Delta_{q,m} = (\Delta\theta_{q,m}, \Delta\sigma_{q,m}, \chi_q, \psi_q) \quad (3.9)$$

where  $(\chi_q, \psi_q)$  is the position of  $\mathbf{x}_q$  in image  $I_Q$ ,  $\Delta\sigma_{q,m} = \sigma_m - \sigma_q$  is the weak scaling factor estimation, and  $\Delta\theta_{q,m}$  is the weak rotation angle estimation. To

create a sparse multi-dimensional voting space from the initial WG space in which lies the vectors  $\Delta_{q,m}$ , we proposed building a LSH family inspired by the classical Euclidean LSH family [?] but slightly modified to have an adaptive quantization step for each random projection. An  $L_2$ -sensitive hashing is actually not especially adapted for creating our voting space. We rather would like to guaranty a good and normalized dynamic on each of the projection axis.

4. The prior distribution is finally computed by voting in the WG multi-dimensional voting space. The resulting WG score  $z_0(\mathbf{x}_q)$  measures the number of visual matches in  $\mathbf{Z}_Q$  that are geometrically consistent with the direct visual matches  $(\mathbf{x}_q, \mathbf{x}_m)$ . The higher  $z_0(\mathbf{x}_q)$ , the more likely  $\mathbf{x}_q$  belongs to a frequent object.

Evaluating the accuracy of object discovery and mining algorithms is more challenging than evaluating object retrieval with a pre-fixed set of queries. We actually need a complete ground truth with all  $(c, f)$ -frequent objects of the dataset and with the precise location of all their instances. As no previous evaluation dataset met these objectives, we created a new one by manually localizing all instances of the 37 logos of BelgaLogo dataset and then cutting and pasting the cropped logos into a dataset of 10K distractor images crawled from Flickr. To reduce the probability of finding  $(c, f)$ -frequent objects in the distractors, all images come from distinct users and distinct geographic areas (1 degree of longitude and latitude). The BelgaLogos instances were then pasted without any modifications (rotation or scaling, ...) at random positions in the distractors. The resulting dataset, called *FlickrBelgaLogos* is publicly available on the Web\*.

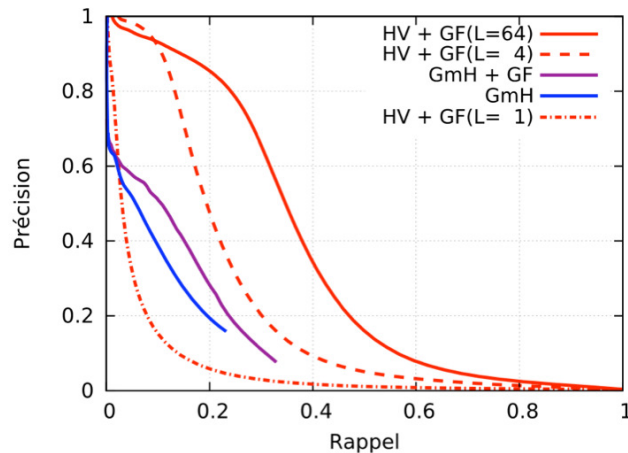


Figure 3.6: Overall comparison between our method and GmH [?]

Figure ?? displays the precision-recall curves of our method compared to the previous baseline of Chum et al. (Geometric min-Hash, GmH[?]). The red curves correspond to our method with an increasing number of visual hash tables, up to a typical and reasonable value of 64 tables. The blue curve corresponds to GmH with a consistent visual vocabulary of 1M words and a huge number of 100K sketches per image (much higher than the 60 sketches per image recommended in their original paper). The purple curve corresponds to an augmented version of the GmH algorithm using our weak geometry hashing algorithm as a second step. It

\*<http://www-sop.inria.fr/members/Alexis.Joly/BelgaLogos/FlickrBelgaLogos.html>

shows that it also improves the raw performance of GmH but in a more limited way due to the fact that GmH already uses neighborhood constraints. Overall, the retrieval performance of both GmH and augmented GmH are still far from what we are able to achieve with our full scheme. Using only 4 visual hash tables in our method is sufficient to clearly outperform GmH with 1M k-means clusters and 100K sketches per image.

### 3.4 Instance-based Visual Classification of Legal Entities

*This work [163] was achieved in the scope of the ongoing PhD of Valentin Leveau under my supervision, the one of Olivier Buisson (INA), and the scientific direction of Patrick Valduriez (Inria).*

Legal entities (such as firms, government bodies, political parties, societies, associations, etc.) are entities other than natural persons (human being) created by law and recognized as having duties and rights. It does not exist any estimation of the number of such legal entities but they are omnipresent in our all day life as well as in all media contents. Beyond their legal identity, most of them also have a corporate visual identity, that is a set of graphical rules and elements providing an organisation with visibility and recognizability (graphic charter, logotype, insignia, colors, polices, fonts, etc.). As for natural persons, it is therefore possible to recognize them automatically in visual content in order to provide automatic annotations. This is of high interest for many applications involving huge amounts of weakly annotated image or video content (YouTube, social media, TV archives, etc.) [?].

As the number of legal entities to be recognized is potentially very large, the problem is primarily related to large-scale image classification. Existing methods and techniques for this problem are typically based on local descriptors pooling techniques (BoW [101], Fisher vectors [?], SPM [64]) and the use of efficient classifiers in the high-dimensional embedded space such as linear support vector machines [87]. An alternative is *deep convolutional neural networks* that have been recently proved to achieve similar results on large-scale image datasets such as ImageNet [?]. The problem of recognizing legal entities is however slightly different. Because the visual identity of a legal entity is actually aimed at guarantying its recognizability, it relies on visual objects with small intra-class variations (such as logos, landmarks, insignias, etc.) but in highly cluttered contexts (very small objects & weak image-level annotations). This problem has been referred as *instance classification* in a recent paper of Krapac et al. [?] and is at the crossroad between *object recognition* and *instance-level image retrieval*. The method they propose is based on a feature-wise prototype selection approach: local descriptors are all kept in their original form (without quantization) and a distance-adaptive prototype is trained for each of them in a supervised way. They report some consistent performance improvements over several state-of-the-art classification methods (including Fisher Vectors [87]).

The other family of techniques related to legal entities recognition is *instance-based image retrieval* techniques [4] and in particular the ones focused on logo retrieval [94] including ours [146]. These techniques are primarily aimed at retrieving instances of a given query object in an unsupervised way but any of them can be used for classification purposes when search is performed on a labelled

set of pictures (typically by voting on the top-K retrieved images or through any other instance-based classifier). The method we developed in [163] improves such techniques in order to construct precise class-specific saliency maps and build a strong image classifier highly robust to noise and clutter. The whole classification scheme can be summarized as follows: local descriptors are extracted from the query image  $I_q$  and searched independently in the reference set using an efficient KNN search scheme. A local geometry consistency checking is then performed at every potential region of interest using a newly introduced sliding RANSAC procedure. The resulting lists of checked patches are then back-propagated in the query image and merged in order to produce pixel-wise saliency maps for each of the retrieved label. A strong classifier is finally derived from the class-specific saliency maps through a max-pooling strategy.

**Hash-based KNN search.** The approximate KNN of each local feature  $\mathbf{x}_j^Q$  belonging to a query image  $I_Q$  are computed efficiently thanks to the hash-based multi-probe search method we introduced in [145] using RMMH [147] as hash function (cf. Chapter 2).

**Sliding RANSAC.** As discussed before, post-checking the geometric consistency of the raw visual matches is an efficient strategy to filter false positives and consolidate good matches. The RANSAC algorithm and its variants have notably been successful in rigid objects retrieval [?] in particular logos [146]. A global RANSAC algorithm applied at the image-level is however not adapted to the detection of very small objects in highly cluttered images for which the percentage of inlier pairs of matches can be typically lower than 0.1% of the whole set of possible pairs. Furthermore, as it is computed on the retrieved images one by one, it does not allow consolidating locally the matches from different training images. To address these issues, we introduced a *sliding* RANSAC strategy aimed at checking the geometric consistency locally for each of the  $N_Q$  query features of the query image  $I_Q$ . More precisely, for a given local feature  $\mathbf{x}_j^Q \in I_Q$ , its  $m$  spatial nearest neighbors are computed so as to define a candidate region of interest to be geometrically checked in all the retrieved pictures (i.e. in the ones having some visual matches within the  $m + 1$  lists of KNN). For a given candidate region of interest and a given retrieved image, the RANSAC algorithm then works in the same way as described in section ?? . Note that the support of both the *random sampling* and the *consensus* phases is bounded by the set of local features belonging to the current region of interest. This allows improving the recall and the precision of the inliers compared to a classical global RANSAC algorithm. The parameter  $m$  controls the locality constraint of the geometry consistency analysis. Ideally, it should fit the size of the targeted objects of interest. Too large values of  $m$  would lead to the same problem than a global RANSAC. Too small values of  $m$  would degrade the dynamic of the number of inliers and possibly miss some consistent matches. In our experiments,  $m$  was trained by cross-validation.

**Class-specific geometry consistency maps.** The output of the sliding RANSAC algorithm is a set of  $N_Q$  lists of consolidated results (i.e. one list per query feature  $\mathbf{x}_j^Q$ ). Each consolidated result  $R_{j,t}^Q$  is itself defined as a set of individual matches of the form  $(\mathbf{x}_{j'}^Q, \mathbf{x}_{t'})$  where the  $\mathbf{x}_{j'}^Q$  belong to the  $m$  spatial neighbors of  $\mathbf{x}_j^Q$  and the  $\mathbf{x}_{t'}$  belong to an image  $I_t$  of the training set. In order to construct saliency maps, we first associate each consolidated result  $R_{j,t}^Q$  with an individual geometry consistency score  $f_{j,t}^Q$  and a bounding box  $B_{j,t}^Q$  in the query image. Rather than simply counting the number of inlier matches, the score  $f_{j,t}^Q$

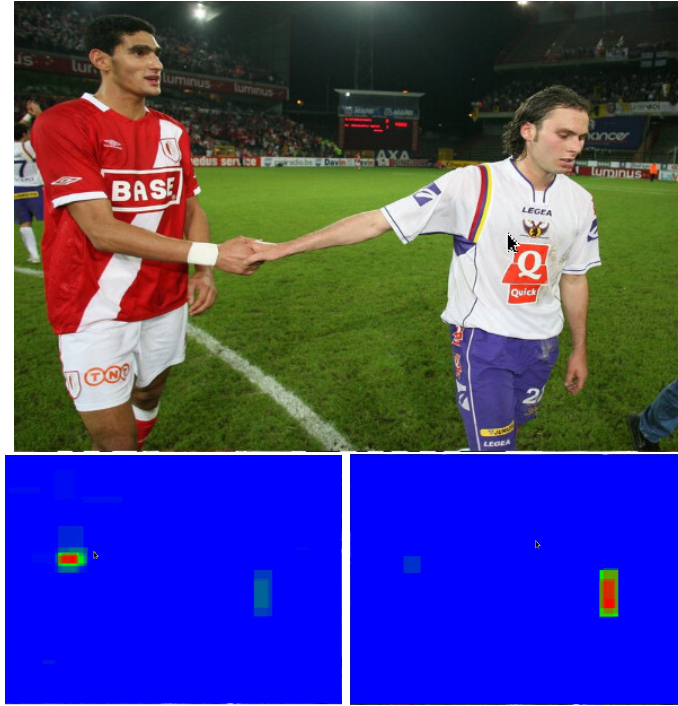


Figure 3.7: Class-specific geometry consistency maps.

is computed as the sum of the inverse rank of the matched features  $\mathbf{x}_{t'}$  (rank in the KNN of  $\mathbf{x}_{j'}^Q$ ). This allows giving more importance to the most confident visual matches. The bounding box  $B_{j,t}^Q$  is defined as the minimum bounding rectangle containing all the individually matched features  $\mathbf{x}_{j'}^Q \in R_{j,t}^Q$ .

The pixel-wise consistency score  $g_c^Q(w, h)$  of a pixel  $(w, h)$  according to class label  $c$  is then computed by (i) selecting the consolidated results  $R_{j,t}^Q$  whose bounding box  $B_{j,t}^Q$  intercepts  $(w, h)$  (ii) grouping them according to the provenance image  $I_t$  and averaging the scores  $f_{j,t}^Q$  for each group (iii) summing the averaged scores of the groups whose provenance images  $I_t$  are labeled with  $c$ . This allows voting on the number of pictures retrieved for label  $c$  and weighting each vote by an average geometry consistency in each image. Figure ?? displays two saliency maps  $g_c^Q(w, h)$  computed for two distinct class labels in a single query image.

**Multi-label Scoring.** As illustrated by Figure ??, the saliency maps produced by the previous step could be easily used for a precise localization of the visual patterns recognized for each of the retrieved legal entity. The scope of the work presented in [163] was however only on classification so that we only use the maps to build a strong classifier at the image level. This is done by simply taking the value of the most salient pixel in each map (i.e for each returned label):

$$s^Q(c) = \max_{(\mathbf{w}, \mathbf{h})} g_c^Q(w, h) \quad (3.10)$$

where  $s^Q(c)$  is the detection score of the label  $c$ .

As shown in Figure ??, this last step also acts as a disambiguation procedure where different classes can co-occur in different images and bring geometry consistency in other class-specific saliency maps.

To decide whether a given legal entity is detected or not, a threshold  $\tau_s$  is applied on the  $s^Q(c)$  scores (several annotations can thus be produced for each image). To better model the density distribution over the classes, a normalization is then applied according to:

$$p^Q(c) = \frac{s^Q(c)}{\sum_{c'} s^Q(c')} \quad (3.11)$$

where  $p^Q(c)$  is the probability estimation of the presence of the label  $c$ .

**Experiments.** The proposed method was evaluated on 3 challenging datasets of the literature (FlickrLogos32 [94], BelgaLogo [146], Vehicles29 [?]) and a new one we created specifically for the large-scale recognition of legal entities. It consists of 371,924 images noisy labelled with 5,824 legal entities. This dataset was automatically created by querying Google Image search engine with the entities names. The list of the entities is the union of several thesaurus found on the web and contains world-wide companies, associations, organizations and sport teams. Table ?? reproduces the results of [?] and reports our own results using the same evaluation protocol. It can be seen that our method outperforms the previous baseline of [?] (and de facto the other state-of-the-art classification methods) on the two experimented datasets, whereas the training stage of our method is much more scalable. It took respectively 13 minutes and 22 minutes to index and to compute the a posteriori multi-probe search model of the 51,054,054 descriptors of the Vehicles29 training set and the 91,800,540 descriptors of the FlickrLogos32 training set (including distractors images). The good results achieved by our method on the *Vehicles29* dataset shows that it is well suited for such fine-grained image classification tasks.

Method	FlickrL32	Vehicles29
Fisher Vectors (128x4,096)	0.866	0.497
Prototype voting [?]	0.914	0.557
<b>Our method (S-Ransac)</b>	<b>0.928</b>	<b>0.597</b>

Table 3.2: Classification performance.

Table ?? then reports the results achieved when using the large LegalEntities5K dataset as training set. It took 1 hour and 55 minutes to index the 500,957,407 SIFT features it contains. The results show that the effectiveness of our method is still very satisfactory considering that (i) the number of classes in the training set is two orders of magnitude higher (ii) the training set was built automatically without any human validation and therefore contains a high level of noise.

Benchmark	mAP	Avg Search time
2.5K images / LegalEntities5K	<b>0.686</b>	7.4 sec
FlickrLogos / LegalEntities5K	<b>0.648</b>	6.3 sec

Table 3.3: Classification results and computation time on the *LegalEntities5K* dataset (Intel(R) Xeon(R) E5-2650 CPU 2.00GHz).

**Discussion.** The main line of this work was to use online geometry consistency checking to disambiguate instance-based matches rather than training discriminative models offline. This is justified in several ways. First, our training phase is



---

reduced to a simple indexing process with a linear time and space complexity  $O(N)$ . The prototype selection technique of [?] requires computing the 20,000NN of each of the  $N$  features of the training set, leading to a much more important training time (over-linear in  $N$ ). Concerning the memory storage, their method requires at least 8 times more RAM to store the original SIFT features. Besides, the complexity of other state-of-the-art methods making use of pooling and SVM's is typically  $O(N + |C| \cdot |S|^2)$  so that they are less scalable in both the number of classes and the number of images. Beyond scalability, our method has several other advantages including the easy management of multi-labeled images, the precise localisation of the recognized patterns making them highly interpretable and the possibility of dynamically inserting additional training images in an incremental way.

## Chapter 4

# User-centric content-based retrieval methods and systems

This chapter is concerned with the upper level of our three-tier architecture of a content-based multimedia information retrieval system (cf. Introduction). This layer works at the applicative level and is in charge of providing useful and interactive functionalities to the end-user. In the following sections, I present several of my work that can be classified in this category because of their user-centric nature and the interactive mechanisms they introduce.

### 4.1 Interactive Object Retrieval using Efficient Boosting

*This work [164] was achieved in the scope of the MASTER internship of Saloua Litayem under my supervision and the scientific direction of Nozha Boujema.*

Contrary to usual supervised object recognition schemes, *interactive* object retrieval consists in learning visual models as an *on-line* process so as to retrieve related content in a large dataset. It is of high interest for personalizing the targeted concepts and avoiding mis-understandings of the user when he is faced to irrelevant results. A popular interactive retrieval scheme is *relevance feedback* [95, 125] in which the user is asked to positively or negatively label the results returned by the system in an iterative way. Another scenario is to interactively crawl some illustrations of the visual concept targeted by the user by asking him to formulate a text query to be issued to a text-based image search engine [62]. In such contexts, the efficiency and the scalability of the *retrieval phase* are critical. The training phase is also important but it is usually less critical in as the number of training samples remains rather low. In [164], we built an interactive object recognition method as an extension of the supervised object recognition method of Opelt et al. [81] which was the first work suggesting the use of *matching-based* weak learners for generic object recognition. Given a training set  $X$  of  $M$  weakly labeled local features  $\mathbf{x}$ , their learning model is actually based on the AdaBoost algorithm, such as:

$$H(I) = \sum_{t=1}^T w_t h_t(I)$$

but with a specific family of weak learners  $h_t(I)$  based on the minimum distance between the local features of the positive training samples and the ones of the test images:

$$h_t(I) = \text{sign}(\theta_t - \min_{\mathbf{x}_i \in I} d(\mathbf{x}_t, \mathbf{x}_i))$$

where  $\mathbf{x}_t \in X^+$  is a positive sample of the training set  $X$  that was selected by the  $t$ -th iteration of the Adaboost training algorithm. In other words, the image  $I$  is classified by Adaboost according to a  $M+$ -dimensional global representation composed of the  $M+$  matching scores of the positive samples of the training set  $X$  in the test image. This method was shown to effectively capture small and complementary details of the targeted objects (notably when using different feature types) and it inspired many following works on objects recognition (e.g. [2, 92, 75]).

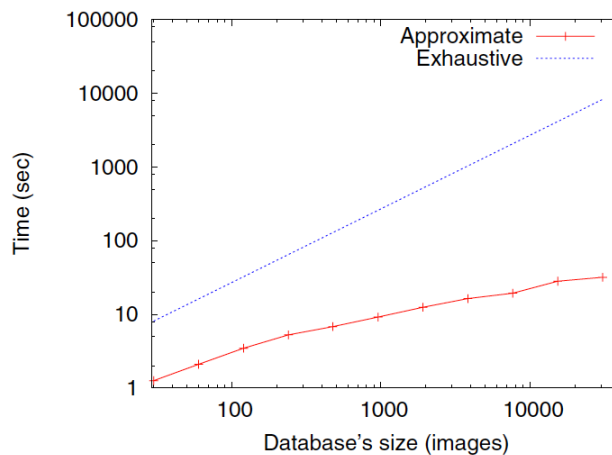


Figure 4.1: Retrieval time of our index-based boosting classifier compared to the brute-force approach

Using this scheme for interactive object retrieval is however not trivial. The complexity of computing the classification score  $H(I)$  for all images  $I$  of a large repository is actually  $O(T.N)$  where  $N$  is the total number of local features in all images. Even for moderately large datasets involving thousands of pictures and thousands of local features per picture, the brute-force approach predicting the scores of the images one by one is already not affordable as an online process. Our contribution rather consisted in replacing the original weak learners  $h_t(I)$  of Opelt et al. by approximate versions  $h'_t(I)$  making use of high-dimensional hashing indexing and approximate range queries. Practically, an index  $Y$  containing all the local features  $\mathbf{y}$  of the images in the repository is constructed offline. Then, instead of predicting a given online trained model by computing the score  $H(I)$  of all the images in the repository one by one, our new classifier directly search the most positive images of the whole dataset in a two steps process:

**STEP 1 - Approximate range queries:** we first perform  $T$  (approximate) range queries using the  $T$  features  $\mathbf{x}_t$  of the weak learners as queries issued to the index  $Y$ . In the case of an exact search, each range query would return a set of features  $R_Y(\mathbf{x}_t)$  such as:

$$R_Y(\mathbf{x}_t) = \{\mathbf{y} \in Y \mid d(\mathbf{x}_t, \mathbf{y}) < \theta_t\}$$

class	exact boosting [81] (mAP)	index-based boosting (mAP)
airplanes	0.2037	<b>0.3881</b>
american-flag	0.2922	<b>0.3903</b>
chess-board	0.7156	<b>0.7446</b>
golf-ball	0.1156	<b>0.2361</b>
mars	<b>0.1603</b>	0.0909
motorbikes	0.2863	<b>0.4516</b>
sunflower	0.5797	<b>0.6214</b>
swiss-army-knife	0.0201	<b>0.1196</b>
tennis-racket	0.2266	<b>0.2715</b>
tower-pisa	0.2683	<b>0.5512</b>
All	0.2868	<b>0.3865</b>

Table 4.1: Retrieval quality of our index-based boosting classifier compared to the brute-force approach of Opelt *et al.*[81]

In the approximate search case, each query returns only an approximated set  $R'_Y(\mathbf{x}_t)$ :

$$R'_Y(\mathbf{x}_t, \alpha) = \{\mathbf{y} \in Y_\alpha(\mathbf{x}_t) \mid d(\mathbf{x}_t, \mathbf{y}) < \theta_t\}$$

where  $Y_\alpha(\mathbf{x}_t)$  is the subset of features visited in the index for the query  $\mathbf{x}_t$  according to a search quality control parameter  $\alpha \in [0, 1]$  (if  $\alpha = 100\%$ ,  $R'_Y(\mathbf{x}_t, \alpha) = R_Y(\mathbf{x}_t)$ ).

**STEP 2 - Prediction:** for each image  $I$  of the repository having at least one feature in the results of the approximate range queries, we can then construct the following approximate classifier:

$$H'(I) = \sum_{t=1}^T w_t h'_t(I)$$

where

$$\begin{aligned} h'_t(I) &= 1 \text{ if } \exists \mathbf{y} \in I \cap R'_Y(\mathbf{x}_t, \alpha) \\ h'_t(I) &= -1 \text{ if } \nexists \mathbf{y} \in I \cap R'_Y(\mathbf{x}_t, \alpha) \end{aligned} \quad (4.1)$$

Note that  $H'(I)$  converges to  $H(I)$  when the search quality control  $\alpha$  tends to 100% (exact search).

In [164], the local features indexing and approximate search of STEP 1 was based on our a posteriori Multi-Probe LSH scheme introduced in section 2.2 and published in [145]. The prior search model of the method was trained for different radius  $\theta_j$  on 10,000 randomly sampled features that were searched with an exhaustive scan of  $Y$ . The default value of the search quality parameter  $\alpha$  was set to 0.9 meaning that on average only 90% of the features that should be labeled positively are retrieved by the approximate weak learner. Fig 3.1 presents the search time achieved by our method compared to the brute-force approach for varying sizes of the image repository. It shows that thanks to the sub-linearity of our approach in the size of the dataset, very high gains can be achieved (several orders of magnitude). More surprisingly, our approximate classifier was also shown to provide better search quality than the exact one, as proved by the Mean Average Precision scores obtained on the Caltech-256 dataset and presented in Table 3.1.

This is mainly due to the fact that the approximate weak classifiers significantly reduce the number of false positives as they only visit the most relevant regions of the feature space according to the search model. The search model somehow acts as a generative model that compensates overfitting issues of the initial classifier.

In [164], we further evaluated our method in the context of active learning through user relevance feedbacks. The simulated scenario was that the user provides a small set of training images that is used to retrieve a first list of results on which he can put new positive or negative annotations. The trained model can then be upgraded with the new annotated content and re-issued as a new query on the index. This process is iterated  $m$  times which is made possible only because of the efficiency of our approximate search algorithm. Experiments did show that this active learning strategy provides consistent gains over the batch learning process. Much greater Mean Average Precision scores can be achieved with lower number of annotated samples.

## 4.2 Interactive Object Retrieval using Interpretable Visual Models

*This work [166, 167] was achieved in the scope of PhD of Ahmed Rebai under my supervision and the scientific direction of Nozha Boujemaa*

Rather than integrating user feedbacks on the content (as in the work of the previous section), we suggested in [166, 167] that the user could interact directly on the trained model. The results returned by usual image classification or object recognition methods are actually often difficult to interpret from a users point of view. The trained visual models are indeed highly dependent on the volume and the quality of the training data and might convey a different semantic than the originally targeted concept. This often makes users uncomfortable with these technologies since they do not get what they expected from their interpretation of the trained concept. It is of interest, therefore, to try learning *interpretable* visual models on which the user can interact according to its own perception. The solution we proposed in [166, 167] lies in constructing an interactive system that allows users to define their own visual concept from a concise set of visual patches given as input. These patches—which represent the most informative clues of a given visual category—are trained beforehand with a supervised learning algorithm in a discriminative manner. By analogy with text information retrieval, we refer to these automatically selected patches as *visual keywords*. Textual keywords carry information about the type of matter and the subject the document deals with. However, they don't express the writers point of view. Similarly, our claim is that it suffices to determine a few visual keywords (of a given category) that allow the content of an image to be interpreted in order to correctly classify it. Then, and in order to specialize their models, users have the possibility to send their feedback on the model itself by choosing and weighting the visual keywords they are confident of.

The real challenge thus consists in how to generate concise and visually interpretable models. We therefore introduce three requirements guiding the design of our method:

**Readability** - each visual keyword must be displayable, i.e. it has a uniquely defined visual representation that can be displayed in a GUI (typically as a thumbnail). The analogy to textual keywords would be that keywords must be readable regardless of whether they are understandable or not.

**Conciseness** - the set of the selected visual keywords must be as concise as possible. Conciseness is important for two reasons: first, it helps users to get a global overview from the very first glance and second, it increases the systems efficiency.

**Disambiguation:** each visual keyword must be as unambiguous as possible. Clearly, having a unique semantic meaning for each keyword is not realistic. Textual words themselves are known to be ambiguous (the same word having different meanings). Nonetheless, reducing the ambiguity of the visual keywords produced should remain a crucial objective towards interpretability.

Now, our contribution relied on two points. First, in contrast to common image classification approaches that rely on the bag-of-words model [80, 64], we proposed embedding local visual features without any quantization or aggregation of the local features, which means that each component of the high-dimensional feature vectors used to describe an image is associated to a unique and precisely localized image patch. Using a classical bag-of-words or even more advanced aggregation-based models such as [111, 114, 87, 51] may actually not satisfy our readability and disambiguation requirements because it discards the spatial positions of the features being learned. The information of many patches is actually pooled within each component and it is difficult to know if that component pertains to tangible parts of the targeted objects (i.e eye, tooth, finger, etc.) or if they are just a statistical combination of some of these parts. Instead of using vector quantization or generative models, we rather choose keeping all local features as visual word candidates. More formally, any image  $I$ , represented by a set of local features  $X$ , is embedded into a  $M$ -dimensional feature vector  $\Phi(X)$  according to:

$$\Phi(X) = \sum_{i=1}^M \min_{\mathbf{x} \in X} d(\mathbf{z}_i, \mathbf{x}) \cdot \vec{e}_i$$

where  $M$  is the total number of local features  $\mathbf{z}_i$  in the training set  $Z$ . Note that this *matching-based* representation, when associated with the standard inner product, can also be interpreted as a *match kernel* of the form:

$$K(X, Y) = \Phi(X)^T \Phi(Y) = \sum_{i=1}^M \min_{\mathbf{x} \in X} d(\mathbf{z}_i, \mathbf{x}) \cdot \min_{\mathbf{y} \in Y} d(\mathbf{z}_i, \mathbf{y})$$

Contrary to the *normalized sum match kernel* proposed by [74], the local features in  $X$  and  $Y$  are however not compared by a direct matching but rather as the degree of correlation of their matches in the training set  $Z$ . The principle of this indirect matching was conceptually already introduced in the *intermediate matching kernel* of [13] but using cluster centers for the pivots rather than the whole set of local features as in our representation. In the same spirit, it was shown in [10] that the popular bag-of-words model can be viewed as a special match kernel, which counts 1 if two local features fall into the same regions partitioned by visual words and 0 otherwise. Finally, the more recent NBN kernel of [110] also makes use of such matching-based embedding but at the category level rather than at the local feature level.

Now, the second main contribution of our work was to use regularization constraints in the loss function of the classifier trained on top of our *matching-based* with the objective to favor sparsity in the produced models. Sparsity is indeed essential for the matter of interpretability as the number  $M$  of patches in the

visual model need to be strongly reduced before being presented to the user. From a machine learning perspective, it also reduces the risk of over-fitting (that is particularly high when using such over-complete representations). To meet these objectives, we used a modified version of the BLasso algorithm (or *stagewise Lasso*) [124]. BLasso is a boosting-like algorithm that regularizes the  $L_2$  loss function with an additive  $L_1$  penalization by alternating between forward and backward steps at each iteration. In the cases of a finite number of base learners and a bounded Hessian of the loss function, the BLasso path is shown to converge to the Lasso path when the step size goes to zero.

Quantitatively, our experiments in [167] did show that our method achieved similar performance as current state-of-the-art systems but outperformed them when training very small objects in highly cluttered images. From a user-centric information retrieval perspective, we also did show that the interpretability allows users to construct their own model from the original set of learned patches, thus allowing for more compound semantic queries. We therefore developed a GUI implementing several possible user interactions including the elimination of ambiguous visual keywords and the emphasizes of some object parts and/or appearance. User-centric experiments did quantitatively and qualitatively show how specializing the models improves the retrieval effectiveness. In [93], the initial version was extended to a geometrically consistent version using spatially consistent neighboring feature sets as patch descriptors and a rigid affine transformation consistency checking in the assignment phase. Experiments did show that this further improves the effectiveness of the method in the case of rigid object categories such as buildings or logos.

### 4.3 Object-based Visual Query suggestion

*This work [162, 139] was achieved in the context of a collaboration between two of the PhD students I supervised, i.e. Pierre Letessier and Amel Hamzaoui (under the scientific direction of Nozha Boujemaa)*

Large-scale object retrieval schemes such as the one discussed in section ?? or others in the literature [59, 89, 94, 88, 4] are often associated with the *query-by-window* search paradigm: the user can freely select a region of interest in any image, and the system returns a ranked list of images that are the most likely to contain an instance of the targeted object of interest. This paradigm has however several limitations related to users perception: (i) When no (or very few) other instances of the query object exist in the dataset, the system mostly returns false positives making the user uncomfortable with the results. Indeed, he does not know if there are actually no other instances of the query object or if the system did not work correctly. (ii) When the user selects a deformable or complex object that the system is actually not able to retrieve, the system mostly returns false positives as well. As the user can freely select any object, this appears very frequently leaving the user with a bad impression of the effectiveness of the tool. The second remark is even more critical if the user believes that the system can retrieve any semantically similar objects (e.g. object categories or visual concepts such as cats or cars). We do not argue here that such queries will never be solved effectively in the future. We just emphasize that bridging the gap between the users understanding of the system and the actual capabilities of the underlying tools is essential to make it successful in a real world search engine. A first possible solution to address these limitations would be to use some adaptive thresholding

method, allowing only relevant results to be filtered, and possibly returning no results if none are found. Our a contrario method introduced in section ?? ([146]), for instance, allows the actual false alarm rate of rigid object instances retrieval to be controlled very accurately. But still, as the user can select any region of interest, the system might return no results in many cases and leave the user disappointed.

We proposed in [162, 139] to solve these user perception issues by a new visual query suggestion paradigm. Rather than letting the user select any region of interest, the system will suggest only visual query regions that actually contain relevant matches in the dataset. By mining offline object instances in the dataset, it is indeed possible to suggest to the user only query objects having at least a predetermined number of instances in the collection. When a user clicks on a highlighted region, the system returns only the images containing other object instances of the same discovered cluster. From a user perception point of view, the proposed paradigm is very different from the window query paradigm. Indeed, since all suggested objects mostly return correct results, the user might rather perceive them as visual links (or hyper-visual links by analogy to hypertext links). It is important to notice, that unlike existing approaches, the links produced by our method are not links between images, but links between automatically localized image regions containing instances of the same rigid object. An image can thus contain several suggestions belonging to different objects clusters and the user can navigate in collection by moving from an object to another, step-by-step. These object-based visual links can be used in many different retrieval paradigms. In [139], we introduced two visual query suggestion scenarios:

**Mouseover visual objects suggestion:** when the user *moves* or *hovers* the mouse cursor over a particular image, the system suggests object queries by highlighting the object instances present in the image. The suggested objects do not depend on the preliminary textual query but are guaranteed to match some other instances in the collection (if the user click on one of them).

**Text-aware visual objects suggestion:** After a user submits a text query, the most frequent visual items discovered in the result list are suggested as new object-based visual queries (typically displayed as few clickable thumbnails on top of the result GUI). Images containing other instances of the suggested object are returned if the user clicks one.

In [139], the offline process allowing the efficient discovery of the suggested objects was based on the objects mining framework described in section ?? of this thesis ([160]). Once a matching graph between frequent object's instances has been constructed with this method (nodes representing the discovered frequent objects instances and edges weights the number of shared matched images), object clusters were extracted thanks to a bi-partite graph-based clustering algorithm derived from the PhD work of A. Hamzaoui [45][137, 138] on shared-neighbors clustering. Alternatively, to handle very large datasets, one might use a more scalable graph-based clustering algorithm such as MCL [28].

We built a real demonstrator of the *mouseover visual objects suggestion* scenario that was presented in the context of ACM Multimedia conference 2013 [162]. It was based on a web corpus of 110K images constructed so as to contain many potentially interesting instances of small objects such as sports or international organizations logos, famous buildings and places, etc. For that purpose, we queried a popular web image search engine with a list of 170 ad hoc keywords. Around 600 millions SIFT features were extracted from this corpus and the mining with our algorithm was



completed within 48 hours on a single computer (two hexa-cores CPU Intel X5660). Then the interactive GUI of the system allowed three main things: (i) visualize all discovered objects (as thumbnails) and their instances in the dataset by clicking on any of them (ii) visualize all the discovered object's instances in a given image by clicking on its thumbnail (thanks to colored bounding boxes) (iii) iteratively move from an object to another one by simply clicking on the bounding box and visualize the images of the new cluster in a pop-up window. Due to the small size of the discovered objects, their multiplicity and the precision of the clusters, such navigation actually offers a very nice and unusual user experience in exploring an image collection. Many of the users who experimented the system reported that they really had the impression to follow hyper-links as in classical web browsers but on the visual content.

## 4.4 Event-centric media search and content suggestion

*This work [168] was achieved in the scope of the PhD of Mohamed Riadh Trad under my supervision and the scientific direction of Nozha Boujema*

An event can be described as an action that occurs at a specific time in a specific place. This notion is potentially useful for connecting individual facts and discovering complex relationships. It is worth noting that photos in User Generated Content (UGC) websites, as well as in personal collections, are often organized into events. Indeed, users are usually more likely to upload or gather pictures related to the same event, such as a given holiday trip, a music concert, a wedding, etc. This applies as well to professional content such as journalism or historical data that are even more systematically organized according to hierarchies of events. Defining new methods for organizing, searching and browsing media according to real-life events therefore attracted many works in the multimedia community [109, 84, 83, 14, 23]. In this work, we primarily addressed the problem of automatically matching distinct records of the same event in large picture datasets, typically in User Generated Content's photo collections. Given a query event record represented by a set of photos, the objective is to retrieve other records of the same event, notably those generated by other actors or witnesses of the same real-world event. An illustration of two matching event records is presented in Figure 3.2.



Figure 4.2: Two events records of an Alanis Morissette concert

It shows how a small subset of visually similar and temporally coherent pictures might be used to match the two records, even if they include other distinct pictures covering different aspects of the event. We therefore introduced a visual-based event matching algorithm highly robust to the presence of outliers [168]. It is in essence similar to object retrieval methods based on local features indexing models and a spatial verification re-ranking stage to improve query performance (see

section ??) but at a different level. We might give the following analogy: images are replaced by event records (picture sets), local visual features are replaced by global visual features describing each picture of a record globally, spatial positions of the local features are replaced by the geo-coordinates and/or time stamps of the pictures. Matching spatially and temporally coherent event records is finally equivalent to retrieving geometrically consistent visual objects.

More formally, given a set of  $N$  event records  $E_i$ , each record being composed of  $N_i$  pictures  $I_j^i$  captured from the same real-world event and each picture being associated with a time stamp  $t_j^i$ , our retrieval method works as follows:

**STEP 1 - Visual Matching:** The query image is visually matched to the full features dataset thanks to global visual features and a hash-based approximate KNN search using RMMH as hashing function and probabilistic multi-probe queries (see Chapter 2). It typically returns the  $K$  most similar pictures in the dataset. When multiple matches occurred for a given query image feature and a given retrieved record, we only keep the best match according to the feature distance. The visual matching step finally returns a set of candidate event records  $E_i$ , each being associated with  $M_i^q$  picture matches of the form  $(I_m^q, I_m^i)$ . Only the retrieved records with at least two image matches are kept for the next steps.

**STEP 2 - Temporal consistency:** For each remaining record, we compute a temporal consistency score by estimating a translation model between the query record and the retrieved ones. The resulting scores  $S_q(E_i)$  are used to produce the final records ranking returned for query  $E_q$ . The translation model estimation is based on a robust regression and can be expressed as:

$$\hat{\delta}(E_q, E_i) = \underset{\delta}{\operatorname{argmin}} \sum_{m=1}^{M_i^q} \rho_{\theta}(t_m^q - (t_m^i + \delta)) \quad (4.2)$$

where  $\hat{\delta}$  represents the estimated temporal offset between  $E_q$  and  $E_i$ . The cost function  $\rho_{\theta}$  is typically a robust  $M$ -estimator allowing to reject outliers with a tolerance  $\theta$  (in our experiments we used Tukey’s robust estimator). The estimated translation parameter  $\hat{\delta}$  should be understood as the temporal offset required to register the query event record  $E_q$  with the retrieved event record  $E_i$ . Once this parameter has been estimated, the final score of an event  $E_i$  is finally computed by counting the number of inliers, i.e the number of visual matches that respect the estimated translation model:

$$S_q(E_i) = \sum_{m=1}^{M_i^q} \mathbb{1} \left( \left| t_m^q - (t_m^i + \hat{\delta}) \right| \leq \theta \right) \quad (4.3)$$

where  $\theta$  is a tolerance error parameter, typically the same than the one used during the estimation phase. Depending on the application context, further improvements can be obtained by additional constraints on the tolerated values for  $\hat{\delta}$ . Rejecting events with a too large temporal offset from the query record is indeed a good way to reduce the probability of false alarms. In [168], we did conduct large-scale experiments on a set of about 1M Flickr images annotated with LastFM music events tags. We did show that our method allows to alleviate most of the issues related to the use of metadata in particular the imprecision of the spatio-temporal metadata. Distinct records of the same event are not necessarily located at the same place or can be recorded at different times. Some events might, for example, have wide spatial and temporal coverage such as a volcano eruption or an eclipse, so that geo-coordinates and time stamps might be not sufficiently discriminant.

This lack of discrimination can be problematic even for precisely located events, typically in crowded environments such as train stations, malls or tourist locations. In such environments, many records might be produced at the same time and place while being related to very distinct real-world events. Furthermore, location and time information is not always available or might be noisy. The Flickr dataset used in the experiments did for instance not contain any geographic information and it contained noisy time EXIF data because of the different reference times of the used devices.

Application scenarios related to such a retrieval paradigm are numerous. By simply uploading their own record of an event users might, for example, access to the community of other participants. They can then *revive* the event by browsing or collecting new data complementary to their own view of the event. If some previous events records had already been uploaded and annotated, the system might also automatically annotate a new record or suggest some relevant tags. The proposed method might also have nice applications in the context of citizen journalism. Automatically detecting the fact that a large number of amateur users did indeed record data about the same event would be very helpful for professional journalists in order to cover breaking news. Finally, tracking events across different media has a big potential for historians, sociologists, politicians, etc.

To further answer such scenarios, we investigated new content suggestion and summarization methods making use of the full event records matching graph of a given UGC images collection (such as the one displayed on Figure 3.3). Such matching graph can be easily obtained by querying all event records of the collection one by one thanks to our visual-based event matching method. Also, we assume that we are given an event and a corresponding set of records. Such identified record clusters can be obtained either by automatically clustering the matching graph or by using metadata associated to the media such as time, location or tags when available. Event clusters may, however, be noisy and contain records associated with some other events. This is particularly true in the case of co-located events where a set of people may be interested in the same event but also share images of other local events. Hence, records from different events are likely to share a subset of visually similar images and thus, appear within the same cluster. Figure 3.3 illustrates the situation described above. The event cluster C1 contains 4 records related mostly to the social event E1, it also includes an occurrence (Record 3) of the social event E3. Conversely, distinct records may reflect different aspects of the event and, thus, be scattered between clusters.

Our content selection approach relies on the observation that widely covered moments are likely to reflect key aspects of the event as they reflect a common interest. Should a sufficient number of users take a large number of shots at a particular moment, then we might consider this to be an objective evaluation of interest at that moment. Given a cluster  $C$  of  $n$  identified event records and their associated set of media documents  $I_c$ , our method counts, for each image  $I \in I_c$ , the number  $S(I)$  of temporally consistent visual matches with another image within a different record of the same cluster (i.e. the number of times that  $I$  contributed to a link with a record within the cluster). More formally, if we denote as  $G$  the graph having elements from  $I_c$  and whose edges link pairs of temporally and visually consistent images from  $I_c$ , the  $S(I)$  score represents the in-degree centrality of  $I$ . This relevance score can then be used in distinct scenarios. For event summarization, we can simply return the top-K images (according to  $S(I)$ ) of the most representative cluster of that event. In a more interactive and personalized way, we may present a given user only documents that

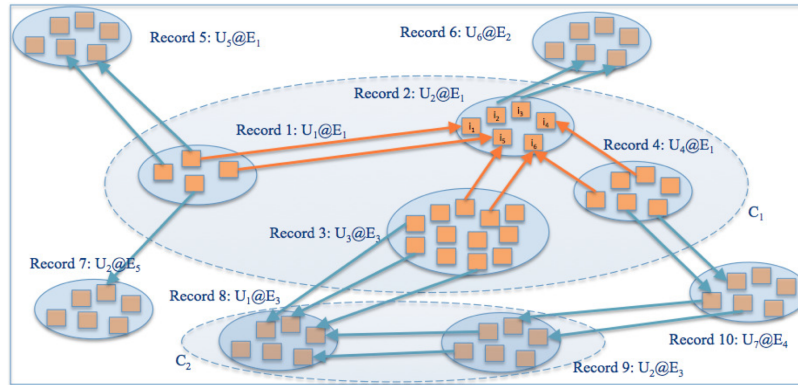


Figure 4.3: Matching graph of 10 event records.  $C_1$  = an event cluster of 4 event records related mostly to the social event  $E_1$  ( $U_1@E_1$ ,  $U_2@E_1$ ,  $U_4@E_1$  and  $U_3@E_3$ ).  $C_2$  = an event cluster of 2 event records related to the social event  $E_3$  ( $U_1@E_3$  and  $U_2@E_3$ ).

provide additional information about the event than its own content. Given a set of  $N_q$  images (i.e. a record of a user), the recommendation system first identifies the corresponding event and then returns the images with the highest score  $S(I)$  among the ones that were not directly matched by the visual matching step.

Our content selection technique was evaluated through a user-centric evaluation involving 10 users and a dataset of 828,902 Flickr images related to music events. Each user was asked to evaluate a set of 20 event summaries chosen at random from a set of 168 events, each having at least 5 associated event records. A 1 to 5 scale was used to rate the overall quality of the summary, where a score of 5 signifies strong relevance and clear usefulness, and a score of 1 signifies no relevance and no usefulness. Similarly, a 1 to 5 rating was used to score the images of the summary individually. The results did show that 39% of the suggested images were rated with the highest score while only 5% had the lowest. Overall, 68% of the rated images were judged good enough to represent the event they belonged to. Looking at the results in more detail, we concluded that, without much surprise, the worst rated images are generally those displaying only a few people not directly participating in the main event itself (friends of the photographer, a lunch break, etc.) or images of very poor quality. On the other hand, the top-rated images are usually good quality images where the artist(s) is(are) clearly visible and/or where the scene presents a specific interest. The comparison of the event-centric rating to the image-centric rating did show that although they are globally correlated, they also exhibit some variations. A higher image-centric rating, for instance, reflects a limited event coverage despite the quality of the suggested images. Conversely, a higher event-centric score reflects a good coverage of the event even if the individual images are of lower qualities.

## 4.5 Interactive plant identification based on social image data

*This work [135, 129, 131, 152, 126, 136, 130] was carried on within the Pl@ntNet project funded by Agropolis Foundation and involving 5 key partners (Inria, Inra, CIRAD, IRD, Tela Botanica). As the responsible of the research track of the project, I supervised the collaborative and trans-disciplinary work reported in this section. It is without doubt*

*the most advanced integration and real-world experimentation of my research work.*

Building accurate knowledge of the identity, geographic distribution and uses of plants is essential for a sustainable development of agriculture as well as for biodiversity conservation. Unfortunately, such basic information is often only partially available for professional stake-holders, teachers, scientists and citizens, and often incomplete for ecosystems that possess the highest plant diversity. A noticeable cause and consequence of this sparse knowledge, expressed as the *taxonomic gap*, is that identifying plant species is usually impossible for the general public, and often a difficult task for professionals, such as farmers or foresters and even for the botanists themselves. In this context, using multimedia identification and collaborative data management tools is considered as one of the most promising solution to help bridging the taxonomic gap [65, 37, 16, 108, 104, 1, 107][132, 154]. With the recent advances in digital devices/equipment, network bandwidth and information storage capacities, the production of multimedia data has indeed become an easy task. In parallel, the emergence of citizen science and social networking tools has fostered the creation of large and structured communities of nature observers (e.g. e-bird\*, iNaturalist†, TelaBotanica‡, Xeno-Canto§, etc.) who already started to produce outstanding collections of multimedia records. Building effective and sustainable ecological surveillance systems based on such collaborative approaches is however still challenging. Modeling the evolution of species distribution at a large scale would require much more substantial data streams, producing typically two or three orders of magnitude more observations than current streams [16][152]. Current data creation and validation workflows are too much dependent on the labor of a small number of expert naturalists, thus could not scale to the required millions of observations. The PI@ntNet experience [152] is an attempt to solve this issue through an innovative participatory sensing platform that relies on image-based plant identification as a mean to enlist non-expert contributors and facilitate the production of botanical observation data.

The platform has evolved since 2010 with iterative developments based on research advances, data aggregation and integration by a growing community of volunteers, and infrastructure evolution based on users feedback's and human perception evaluation. The following outcomes illustrate this evolution:

**2010, PI@ntScan:** this on-line application was the first visual-based plant species identification system based on crowdsourced data. This prototype, that allowed in its first version to identify 27 Mediterranean tree species based on leaf scans [135], was a first step toward a large scale crowd-sourcing application promoting collaborative enrichment of botanical visual knowledge. From the technical side, contrary to state-of-the-art methods that were mostly based on leaf segmentation and shape boundary features, the visual search engine was based on local features and large-scale matching techniques. Indeed, we realized that matching-based object retrieval methods (see section ??), usually aimed at retrieving more rigid objects (buildings, logos, etc.), do work surprisingly well on leaves. This can be explained by the fact that even if a small fraction of the leaf remains affine invariant, this is sufficient to discriminate it from other species. Conversely, segmentation-based approaches have several strong limitations in a crowdsourcing environment where acquisition protocol (presence of clutter and background information, shadows, leaflets occlusion, holes, cropping, etc.) cannot

---

\*<http://www.e-bird.org/>

†<http://www.inaturalist.org/>

‡<http://www.tela-botanica.org/>

§<http://www.xeno-canto.org/>

be accurately controlled.

**2011, Pl@ntNet-ID:** less than one year later, Pl@ntScan was replaced by a more user-friendly version and a more advanced visual search engine relaxing geometrical constraints and focusing more on multi-features and saliency concerns [127, 129]. Pl@ntNet-ID was dedicated to the identification of 54 Mediterranean tree and shrub species from photographs of leaves or flowers. It offered the possibility to combine several pictures of the same organ (i.e. up to 3 leaf or flower images of the same species) in order to improve the identification performance. Still, users had the possibility to enrich the dataset by submitting their own pictures. The validation or correction of these contributions was done manually by a botanist of the project.

**2012, Pl@ntNet-Identify:** an important milestone was marked in 2012 with the launch of Pl@ntNet-Identify web application<sup>¶</sup> and the development of an end-to-end innovative workflow involving the members of TelaBotanica social network [152]. Beyond expert data integration efforts and purely crowdsourced approaches, we actually argued that thematic social networks have the advantage to connect experts, enlightened amateurs and novices around the same topic so that all of them can play complementary roles in a real-world ecological surveillance workflow. Experts can typically drive projects, define observation protocols and teach, enlightened amateurs can collaboratively validate data according to their level of expertise, enthusiast novices can provide massive sets of observations according to well defined and documented protocols. Technically speaking, Pl@ntNet-Identify can be considered as one of the first *collaborative active learning system* in which the learning algorithm is able to interactively query a network of users, rather than a single user, to annotate new data points. Two complementary web applications were developed, allowing collaborative revision, validation, qualification and enrichment of the data before their integration in the training dataset. These two applications, called *IdentiPlante* (for collaborative identification validation) and *PictoFlora* (for collaborative picture evaluation and tagging process) are intensely used by the community, resulting in a nightly update of Pl@ntNet-Identify's training dataset with new records. Besides, Pl@ntNet-Identify was also the first botanical identification system based on the potential combination of several habit, leaf, flower, fruit and bark pictures, thanks to the fusion mechanisms introduced in [129] and refined in [152]. This allows identification all year round, including when leaves and flowers are not available. The resulting multimedia system has been extensively experimented through massive leave-one-out tests as well as participations to system-oriented benchmarks and human-centered evaluations [129, 127, 126, 152]. This has shown the great potential of the approach and the good acceptance of such a new way of identifying plants by the users.

Figure 3.4 provides a global picture of the query processing chain of the visual search engine. Pictures belonging to a given plant view category are indexed and searched in separate visual indexes. This allows reducing confusion between pictures of different parts of the plant and therefore increases identification performance. At query stage, the  $N^Q$  query pictures belonging to a query plant  $Q$  are searched separately in their respective visual index and the top-K most similar images are returned for each of them (K was learned by cross-validation on the training data and finally fixed to K=20). Identification is then performed thanks to an instance-based classifier computed across all retrieved pictures [152]. Depending on the taxonomic level selected by the user, this process is applied either on species

<sup>¶</sup><http://identify.plantnet-project.org/>

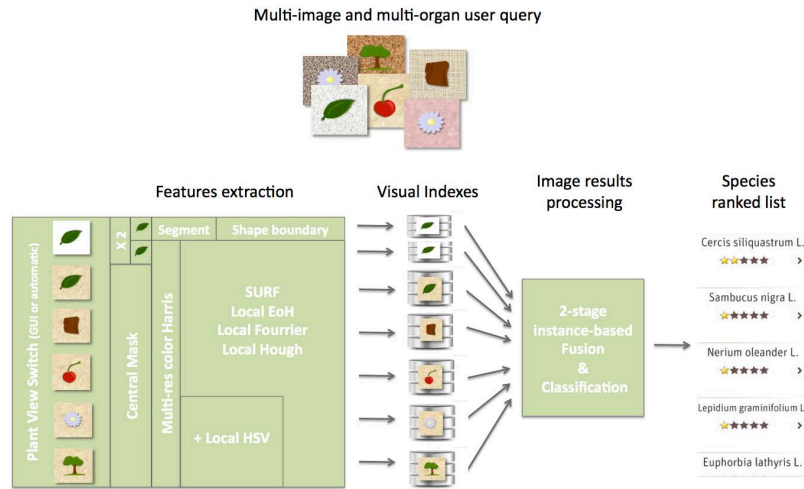


Figure 4.4: Multi-organ query processing chain of the visual search engine

(selected by default), or on genus or family. Thanks to the proposed late fusion approach, each visual search index can be specialized for the targeted organ with specific visual features and similarity metrics. On the other side, it is also important that the whole application generalizes well to other floras with different organs. A good compromise is achieved by using the same generalist content-based image retrieval method (CBIR) for all plant organs but with specialized local features for each of them. Only the leaf scans category involves additional processings that are detailed in [152]. The generalist CBIR method we are using is built from the research work reported in this thesis [146, 145, 147] and is based on the following steps: (i) Local features extraction around multi-resolution color Harris points ([42] and [143]), (ii) Local features compression via RMMH [147], (iii) Local features matching with probabilistic multi-probe queries in a single hash table [145], (iv) voting based on the number of matched features weighted by their distance to the query.

**2013, Pl@ntNet-mobile iOS:** the Pl@ntNet workflow was extended to mobile devices at the beginning of 2013, a first iOS application being launched in March 2013 [131]. At that time, the training dataset included 22,574 images of 957 common European plants species. Less than one year later, thanks to the success of Pl@ntNet-mobile iOS among Tela Botanica members, these figures had increased to 66 000 and 3 600. Pl@ntNet-mobile has 4 main functionalities : (i) an image feeds reader to explore the last contributions; (ii) a taxonomic browser including common names in several European languages, with a full-text search function; (iii) a user profile and personal content management screen; and of course, (iv) the image-based identification tool. The visual search engine is focused on 4 simple view types (flower, leaf, fruit and bark) that can be combined in a single visual request composed of up to 5 images. Current response times range between 3 and 15 seconds depending on the number of pictures, the types of views and the connection conditions. Matched species are displayed by decreasing confidence scores and users can refine the result by visualizing available images in the training dataset as well as species description sheets from eFlore (Tela Botanica’s collaborative encyclopedia for the French flora) and Wikipedia. As soon as the user is willing to share his observation, he can do so whether he succeeded in identifying his observation or not. The picture(s),


<b>Leave-one-plant-out experiments</b>			
multi-view & multi-image id@1	15%		
multi-view & multi-image id@5	39%		
multi-view & multi-image id@10	49%		
<b>User trial evaluation criterion</b>	<b>Best</b>	<b>Worst</b>	<b>Avg</b>
User Identification rate	100%	64%	85%
Confidence score (/10)	9	5.57	7.49
Number of trials (/5)	1.21	2.21	1.62
Utility (/10)			
- Overall	10	4	7.93
- Crop	10	3	7.9
- Taxonomic filtering	10	4	7
- "More details"	10	3	7.67
- "More of this species"	10	8	8.67
Usability/ergonomics (/10)	9	5	7.6
<b>Global appreciation (/5)</b>			<b>3.7</b>

Table 4.2: Human-centered evaluation results

date, positioning and author name are then sent to the collaborative apps (under a Creative Commons licence) as well as to its personal collection, which is accessible from both the Tela Botanica platform and the mobile application itself.

**2014, Pl@ntNet-mobile Android:** finally, an android version of the PlantNet mobile application [130] was distributed one year after the iOS version with some innovations such as (i) the use of metadata additionally to the visual content in the identification process [126] (ii) a new multi-organ, multi-image and multi-feature fusion strategy using separated indexes for each visual feature [126] (iii) the port to new languages [130]. The android version is working on the same visual search engine than the iOS version. At the time of writing, the repository contains about 120K images from 5,500 species of the French flora. This makes it far away the largest plant identification tool ever built.

Beyond the extensive system-oriented experiments reported in [152, 127, 126] to evaluate the identification performance of the system, we also achieved a human-centered evaluation to assess the utility and the ergonomics of the visualization and interactive functionalities. This user trial was carried out in February 2012 so that the amount of data was much smaller than today. The dataset actually contained about 10K images and 127 tree species living in France area. 10 non-expert users were asked to identify 14 plants randomly selected from two pools of about 200 multi-organ queries that were built outside the application. For each of the 14 queries, the user could do anything he/she wanted with the application to perform the identification. We limited to 5 the number of times he/she can click the identify button and to 5 minutes the identification of each of the 14 queries. At the end of each query, the user filled a form with the name of the species he/she chose and a confidence score from 1 to 10. At the end of the whole session, each user was asked to give a note on several aspects including: utility of the application for identifying plants, utility of optional functionalities (more details, crop, etc.), ergonomics and global appreciation. Results are summarized in Table 3.2. The average identification rate achieved by users was 85%, which is positively higher than the leave-one-plant-out identification rates (39% for the top-5 species and 49% for the top-10). This shows the benefit of using the interactive functionalities of the



Table 4.3: Pl@ntNet Users Loyalty

	iOS	Android
Total users	84,437	6018
1 session	9,563 (11.3%)	2,050 (34.1%)
2-5 sessions	44,549 (52.2%)	2,115 (35.1%)
5-10 sessions	22,304 (26.4%)	1,281 (21.3%)
> 10 sessions	8,021 (9.5%)	565 (9.4%)
> 25 sessions	456 (0.5%)	64 (1.03%)
> 100 sessions	47 (0.05%)	7 (0.12%)

application on top of the raw returned results. The best user identified correctly all query plants and the worst one 64% of them. The number of times users press the identify button is on average 1.62 showing that they quickly understood how the application can give the best results. Most functionalities have been considered very useful to complete an accurate identification. Interestingly, the confidence scores show that some users still have some doubts even when they provide the correct identification. Any botanist would confirm that an identification is rarely 100% sure. Other evaluated criteria show the good acceptance and usability of the application.

Now the main question is whether Pl@ntNet participatory sensing platform could be used as a sustainable and effective ecological surveillance tool. In [134], we carried out a self-critical evaluation of the experience to answer that question (one year after the public launch of the first mobile application). We first demonstrated the attractiveness of the developed multimedia system and the nice self-improving capacities of the whole collaborative workflow. We then point out the current limitations of the approach towards producing timely and accurate distribution maps of plants at a very large scale. We discuss in particular two main issues: the bias and the incompleteness of the produced data. We finally open some perspectives and describe upcoming realizations towards bridging these gaps. The assessment of the attractiveness, effectiveness and sustainability of Pl@ntNet as a participatory sensing platform was mainly achieved through the analytics of usage data compiled in April 2014. The PlantNet-mobile iOS application, which was launched in March 2013, had already been downloaded by 84,437 iPhone users at that time. The Android port that was publicly announced in February 2014 had only been downloaded by 6,018 users. As the application is primarily focused on the French flora, 68.05% of the users were located in France. However, the number of users living in other countries is not negligible, with 12K users in the US, 8.7K users in European countries (other than France), 1.8K in Canada, and 3.5K in the rest of the world.

As for any application, the degree of involvement and loyalty is highly variable among users. Table 3.3 shows the relationship between the number of users and the number of sessions. The percentage of users who tested the iOS application only once is quite low (11.3%). It is larger for the Android application but this is mainly due to the shorter runtime at that time (2 months). Then, there is about half of the users who experimented the application just few times, either because they were not convinced, or because this corresponds to their usage of the application (*I am curious about a plant few times in the year*). Note that this category also

includes new entrants who might use the application again later. Anyway, we can roughly consider that about one third of the users who downloaded the application became real active users which is still a pretty good acceptance rate and form a community of several tens of thousands of users. Finally, there is a long tail of few hundreds very active users. These last ones should definitely not be neglected. As in any social network, they are actually likely to be the most influencers and the ones who produce the more data and knowledge.

At the time of writing, the two Pl@ntNet mobile applications (iOS and android) have been downloaded by more than 300,000 users in about 150 countries and they are used by thousands of users per day. The amount of raw plant observations collected by logging the queries of the users is reaching several millions, i.e. the same order of magnitude than the number of specimens in the world's largest herbariums. The quality of that crowdsourced data is of course very far from the one collected by the botanists during centuries but it still has a great potential in terms of data analytics thanks to its volume and velocity. One of the main issue is the bias related to the usage of the application. Observations are indeed more numerous in the most populated regions (cities, national parks, etc.) and more focused on some species than others (because of their attractiveness, visibility, frequency, etc.). As such bias is occurring in any participatory sensing initiative, some convincing solutions have already been proposed in the literature to compensate such bias [11, 31]. Among few others, the e-bird initiative [105] has notably demonstrated that crowdsourced naturalistic observations can be used as a new source of information for biodiversity and ecology studies.

Now, the main bottleneck of the Pl@ntNet data stream is that less than 2% of the raw observations are validated by at least one people so that the vast majority of them are only tagged with the most probable species automatically determined by the visual search engine. This does not prevent trying to exploit them at a macroscopic level (e.g. by regions or taxonomic groups), but the degree of noise is nowadays still too high for building an accurate and trust-full monitoring system. A good news, however, is that the quality of the pictures themselves is pretty good. We estimated in [134] that about 94% of them do correspond to *in-scope* plant observations (i.e. with an entry in the official repository of the French flora) and about 75% of the observations shared with the network are judged as having the minimal quality of 3 stars to be exploited for identification. With the upcoming progress in fine-grained image classification, we can therefore hope building a fully automatized participatory sensing system within the next decade.



# Bibliography - First part

- [1] *MAED '12: Proceedings of the 1st ACM International Workshop on Multimedia Analysis for Ecological Data*, New York, NY, USA, 2012. ACM. 433127.
- [2] Jaume Amores, Nicu Sebe, and Petia Radeva. Context-based object-class recognition and retrieval by generalized correlograms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(10):1818–1833, 2007.
- [3] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 459–468. IEEE, 2006.
- [4] Relja Arandjelovic and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2911–2918. IEEE, 2012.
- [5] Sunil Arya and David M Mount. Approximate nearest neighbor queries in fixed dimensions. In *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, pages 271–280. Society for Industrial and Applied Mathematics, 1993.
- [6] Sunil Arya, David M Mount, Nathan S Netanyahu, Ruth Silverman, and Angela Y Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, 45(6):891–923, 1998.
- [7] Roberto J. Bayardo, Yiming Ma, and Ramakrishnan Srikant. Scaling up all pairs similarity search. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 131–140, 2007.
- [8] Alina Beygelzimer, Sham Kakade, and John Langford. Cover trees for nearest neighbor. In *Proc. of conf. on Machine learning*, pages 97–104, New York, NY, USA, 2006.
- [9] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.
- [10] Liefeng Bo and Cristian Sminchisescu. Efficient match kernel between sets of features for visual recognition. In *Advances in neural information processing systems*, pages 135–143, 2009.
- [11] Elizabeth H Boakes, Philip JK McGowan, Richard A Fuller, Ding Chang-qing, Natalie E Clark, Kim O'Connor, and Georgina M Mace. Distorted views of biodiversity: spatial and temporal bias in species occurrence data. *PLoS biology*, 8(6):e1000385, 2010.
- [12] Christian Böhm, Stefan Berchtold, and Daniel A Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys (CSUR)*, 33(3):322–373, 2001.
- [13] Sabri Boughorbel, Jean Philippe Tarel, and Nozha Boujemaa. The intermediate matching kernel for image local features. In *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, volume 2, pages 889–894. IEEE, 2005.
- [14] Markus Brenner and Ebroul Izquierdo. Social event detection and retrieval in collaborative photo collections. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, page 21. ACM, 2012.

- 
- [15] Jiajun Bu, Shulong Tan, Chun Chen, Can Wang, Hao Wu, Lijun Zhang, and Xiaofei He. Music recommendation by unified hypergraph: combining social media information and music content. In *Proceedings of the international conference on Multimedia*, pages 391–400. ACM, 2010.
- [16] Jinhai Cai, D. Ee, Binh Pham, P. Roe, and Jinglan Zhang. Sensor network for the monitoring of ecosystem: Bird species recognition. In *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, pages 293–298, Dec 2007.
- [17] Jair Cervantes, Xiaou Li, Wen Yu, and Javier Bejarano. Multi-class support vector machines for large data sets via minimum enclosing ball clustering. *Electrical and Electronics Engineering ICEEE 2007 4th International Conference on*, 2007.
- [18] Elisavet Chatzilari, Georgios Liaros, Spiros Nikolopoulos, and Yiannis Kompatsiaris. A comparative study on mobile visual recognition. In *Machine Learning and Data Mining in Pattern Recognition*, pages 442–457. Springer, 2013.
- [19] Sen-ching Samson Cheung and Avideh Zakhor. Efficient video similarity measurement with video signature. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(1):59–74, 2003.
- [20] Ondřej Chum, James Philbin, Michael Isard, and Andrew Zisserman. Scalable near identical image and shot detection. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 549–556. ACM, 2007.
- [21] P. Ciaccia and M. Patella. Pac nearest neighbor queries: Approximate and controlled search in high-dimensional and metric spaces. In *Proc. of Int. Conf. on Data Engineering*, pages 244–255, 2000.
- [22] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *Proc. of Int. Conf. on Very Large Data Bases*, pages 426–435, 1997.
- [23] Minh-Son Dao, Giulia Boato, Francesco GB De Natale, and Truc-Vien Nguyen. Jointly exploiting visual and non-visual information for event-related social media retrieval. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pages 159–166. ACM, 2013.
- [24] Stamatia Dasiopoulou, Eirini Giannakidou, Georgios Litos, Polyxeni Malasioti, and Yiannis Kompatsiaris. A survey of semantic image and video annotation tools. In *Knowledge-driven multimedia information extraction and ontology evolution*, pages 196–239. Springer, 2011.
- [25] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262. ACM, 2004.
- [26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
- [27] Wei Dong, Charikar Moses, and Kai Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In *Proceedings of the 20th international conference on World wide web, WWW '11*, pages 577–586, New York, NY, USA, 2011. ACM.
- [28] Anton J Enright, Stijn Van Dongen, and Christos A Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic acids research*, 30(7):1575–1584, 2002.
- [29] Levent Ertoz, Michael Steinbach, and Vipin Kumar. A new shared nearest neighbor clustering algorithm and its applications. In *Workshop on Clustering High Dimensional Data and its Applications at 2nd SIAM International Conference on Data Mining*, pages 105–115, 2002.
- [30] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 2008.

- [31] Daniel Fink, Wesley M Hochachka, Benjamin Zuckerberg, David W Winkler, Ben Shaby, M Arthur Munson, Giles Hooker, Mirek Riedewald, Daniel Sheldon, and Steve Kelling. Spatiotemporal exploratory models for broad-scale survey data. *Ecological Applications*, 20(8):2131–2147, 2010.
- [32] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, et al. Query by image and video content: The qbic system. *Computer*, 28(9):23–32, 1995.
- [33] Imola K Fodor. A survey of dimension reduction techniques, 2002.
- [34] Peter Frankl and Hiroshi Maehara. The johnson-lindenstrauss lemma and the sphericity of some graphs. *Journal of Combinatorial Theory, Series B*, 44(3):355–362, 1988.
- [35] Volker Gaede and Oliver Günther. Multidimensional access methods. *ACM Computing Surveys (CSUR)*, 30(2):170–231, 1998.
- [36] Tianshi Gao and Daphne Koller. Discriminative learning of relaxed hierarchy for large-scale visual recognition. In *ICCV*, 2011.
- [37] Kevin J. Gaston and Mark A. O’Neill. Automated species identification: why not? 359(1444):655–667, 2004.
- [38] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, volume 99, pages 518–529, 1999.
- [39] Bernd Girod, Vijay Chandrasekhar, David M Chen, Ngai-Man Cheung, Radek Grzeszczuk, Yuriy Reznik, Gabriel Takacs, Sam S Tsai, and Ramakrishna Vedantham. Mobile visual search. *Signal Processing Magazine, IEEE*, 28(4):61–76, 2011.
- [40] Shantanu Godbole, Sunita Sarawagi, and Soumen Chakrabarti. Scaling multi-class support vector machines using inter-class confusion. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002.
- [41] Yunchao Gong and Svetlana Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 817–824. IEEE, 2011.
- [42] V. Gouet and N. Boujemaa. Object-based queries using color points of interest. In *IEEE Workshop on Content-based Access of Image and Video Libraries*, pages 30–38, 2001.
- [43] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proc. of ACM SIGMOD Conf. of Management of Data*, pages 47–57, 1984.
- [44] Jaap Haitsma and Ton Kalker. A highly robust audio fingerprinting system. In *ISMIR*, volume 2002, pages 107–115, 2002.
- [45] Amel Hamzaoui. *Shared-Neighbours methods for visual content structuring and mining*. Theses, Université Paris Sud - Paris XI, May 2012.
- [46] Jae-Pil Heo, Youngwoon Lee, Junfeng He, Shih-Fu Chang, and Sung-Eui Yoon. Spherical hashing. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2957–2964. IEEE, 2012.
- [47] Cisco Visual Networking Index. Forecast and methodology, 20132018. *White Paper*, 2011.
- [48] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM, 1998.
- [49] Prateek Jain, Sudheendra Vijayanarasimhan, and Kristen Grauman. Hashing hyperplane queries to near points with applications to large-scale active learning. In *NIPS*, 2010.
- [50] Raymond A Jarvis and Edward A Patrick. Clustering using a similarity measure based on shared near neighbors. *Computers, IEEE Transactions on*, 100(11):1025–1034, 1973.

- [51] H. Jégou, F. Perronnin, M. Douze, C. Schmid, et al. Aggregating local image descriptors into compact codes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(9):1704–1716, 2012.
- [52] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Computer Vision–ECCV 2008*, pages 304–317. Springer, 2008.
- [53] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2010. to appear.
- [54] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(1):117–128, 2011.
- [55] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE, 2010.
- [56] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mapping into Hilbert space. In *Conf. in modern analysis and probability*, volume 26 of *Contemporary Mathematics*, pages 189–206. American Mathematical Society, 1984.
- [57] Herv Jgou, Matthijs Douze, and Cordelia Schmid. Improving bag-of-features for large scale image search. *International Journal of Computer Vision*, 87:316–336, 2010.
- [58] N. Katayama and S. Satoh. The sr-tree: An index structure for high-dimensional nearest neighbor queries. In *Proc. of ACM SIGMOD Int. Conf. on Management of Data*, pages 369–380, 1997.
- [59] Yan Ke, Rahul Sukthankar, and Larry Huston. Efficient near-duplicate detection and sub-image retrieval. In *ACM Multimedia*, volume 4, page 5, 2004.
- [60] Manesh Kokare, BN Chatterji, and PK Biswas. A survey on current content based image retrieval methods. *IETE Journal of Research*, 48(3-4):261–271, 2002.
- [61] Flip Korn, B-U Pagel, and Christos Faloutsos. On the dimensionality curse and the self-similarity blessing. *Knowledge and Data Engineering, IEEE Transactions on*, 13(1):96–111, 2001.
- [62] Josip Krapac, Moray Allan, Jakob Verbeek, and Frédéric Jurie. Improving web image search results using query-relative classifiers. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1094–1101. IEEE, 2010.
- [63] Brian Kulis and Kristen Grauman. Kernelized locality-sensitive hashing for scalable image search. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2130–2137. IEEE, 2009.
- [64] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.
- [65] Dah-Jye Lee, Robert B Schoenberger, Dennis Shiozawa, Xiaoqian Xu, and Pengcheng Zhan. Contour matching for a fish recognition and migration-monitoring system. In *Optics East*, pages 37–48. International Society for Optics and Photonics, 2004.
- [66] Michael S Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 2(1):1–19, 2006.
- [67] Ping Li, Anshumali Shrivastava, Joshua L. Moore, and Arnd C. König. Hashing algorithms for large-scale learning. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*. 2011.

- [68] Ruei-Sung Lin, D.A. Ross, and J. Yagnik. Spec hashing: Similarity preserving algorithm for entropy-based coding. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 848–854, june 2010.
- [69] Huan Liu and Hiroshi Motoda. *Feature extraction, construction and selection: A data mining perspective*. Springer, 1998.
- [70] Ting Liu, Andrew W Moore, Ke Yang, and Alexander G Gray. An investigation of practical approximate nearest neighbor algorithms. In *Advances in neural information processing systems*, pages 825–832, 2004.
- [71] Ting Liu, Charles Rosenberg, and Henry A Rowley. Clustering billions of images with large scale nearest neighbor search. In *Applications of Computer Vision, 2007. WACV'07. IEEE Workshop on*, pages 28–28. IEEE, 2007.
- [72] Ying Liu, Dengsheng Zhang, Guojun Lu, and Wei-Ying Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262–282, 2007.
- [73] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li. Multi-probe lsh: efficient indexing for high-dimensional similarity search. In *Proc. of Conf. on Very Large Data Bases*, pages 253–262, 2007.
- [74] Siwei Lyu. Mercer kernels for object recognition with local features. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 223–229. IEEE, 2005.
- [75] Marcin Marszałek and Cordelia Schmid. Accurate object recognition with shape masks. *International journal of computer vision*, 97(2):191–209, 2012.
- [76] Tao Mei, Yong Rui, Shipeng Li, and Qi Tian. Multimedia search reranking: A literature survey. *ACM Computing Surveys (CSUR)*, 46(3):38, 2014.
- [77] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 525–531. IEEE, 2001.
- [78] Yadong Mu, Jialie Shen, and Shuicheng Yan. Weakly-supervised hashing in kernel space. In *Computer Vision and Pattern Recognition*, pages 3344–3351, june 2010.
- [79] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP (1)*, pages 331–340, 2009.
- [80] Eric Nowak, Frédéric Jurie, and Bill Triggs. Sampling strategies for bag-of-features image classification. In *Computer Vision–ECCV 2006*, pages 490–503. Springer, 2006.
- [81] Andreas Opelt, Axel Pinz, Michael Fussenegger, and Peter Auer. Generic object recognition with boosting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(3):416–431, 2006.
- [82] Rina Panigrahy. Entropy based nearest neighbor search in high dimensions. In *Proc. of annual ACM-SIAM symposium on Discrete algorithm*, pages 1186–1195, 2006.
- [83] Symeon Papadopoulos, Raphael Troncy, Vasileios Mezaris, Benoit Huet, and Ioannis Kompatsiaris. Social event detection at mediaeval 2011: Challenges, dataset and evaluation. In *MediaEval*, 2011.
- [84] Symeon Papadopoulos, Christos Zigkolis, Yiannis Kompatsiaris, and Athena Vakali. Cluster-based landmark and event detection on tagged photo collections. *IEEE Multimedia*, 2010.
- [85] Marco Patella and Paolo Ciaccia. Approximate similarity search: A multi-faceted problem. *Journal of Discrete Algorithms*, 7(1):36–48, 2009.
- [86] Loïc Paulevé, Hervé Jégou, and Laurent Amsaleg. Locality sensitive hashing: A comparison of hash function types and querying mechanisms. *Pattern Recognition Letters*, 31(11):1348–1358, 2010.
- [87] F. Perronnin, J. Sánchez, and T. Mensik. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010.



- [88] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.
- [89] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.
- [90] Sébastien Poullot, Olivier Buisson, and Michel Crucianu. Z-grid-based probabilistic retrieval for scaling up content-based copy detection. In *CIVR '07: Proceedings of the 6th ACM Int. Conf. on Image and video retrieval*, pages 348–355, 2007.
- [91] Maxim Raginsky and Svetlana Lazebnik. Locality-sensitive binary codes from shift-invariant kernels. In *Advances in neural information processing systems*, pages 1509–1517, 2009.
- [92] Ignacio Ramirez, Pablo Sprechmann, and Guillermo Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3501–3508. IEEE, 2010.
- [93] Ahmed Rebai. *Interactive Object Retrieval using Interpretable Visual Models*. Thesis, Université Paris Sud - Paris XI, May 2011.
- [94] Stefan Romberg, Lluís Garcia Pueyo, Rainer Lienhart, and Roelof van Zwol. Scalable logo recognition in real-world images. In *ICMR*, 2011.
- [95] Yong Rui, Thomas S Huang, Michael Ortega, and Sharad Mehrotra. Relevance feedback: a power tool for interactive content-based image retrieval. *Circuits and Systems for Video Technology, IEEE Transactions on*, 8(5):644–655, 1998.
- [96] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*, 2014.
- [97] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *ICML '07: Proceedings of the 24th Int. Conf. on Machine learning*, pages 791–798, New York, NY, USA, 2007. ACM.
- [98] Dominik Schnitzer, Arthur Flexer, and Gerhard Widmer. A fast audio similarity retrieval method for millions of music tracks. *Multimedia Tools and Applications*, 58(1):23–40, 2012.
- [99] Gregory Shakhnarovich, Trevor Darrell, and Piotr Indyk. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. MIT Press, 2006.
- [100] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.
- [101] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE, 2003.
- [102] Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(12):1349–1380, 2000.
- [103] John R Smith and Shih-Fu Chang. Visualeek: a fully automated content-based image query system. In *Proceedings of the fourth ACM international conference on Multimedia*, pages 87–98. ACM, 1997.
- [104] Concetto Spampinato, Yun-Heh Chen-Burger, Gayathri Nadarajan, and Robert B Fisher. Detecting, tracking and counting fish in low quality unconstrained underwater videos. In *VISAPP (2)*, pages 514–519. Citeseer, 2008.
- [105] Brian L Sullivan, Christopher L Wood, Marshall J Iliff, Rick E Bonney, Daniel Fink, and Steve Kelling. ebird: A citizen-based bird observation network in the biological sciences. *Biological Conservation*, 142(10):2282–2292, 2009.
- [106] Antonio Torralba, Robert Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):1958–1970, 2008.

- [107] Michael Towsey, Birgit Planitz, Alfredo Nantes, Jason Wimmer, and Paul Roe. A toolbox for animal call recognition. *Bioacoustics*, 21(2):107–125, 2012.
- [108] Vlad M Trifa, Alexander NG Kirschel, Charles E Taylor, and Edgar E Vallejo. Automated species recognition of antbirds in a mexican rainforest using hidden markov models. *The Journal of the Acoustical Society of America*, 123:2424, 2008.
- [109] Raphaël Troncy, Bartosz Malocha, and André TS Fialho. Linking events with media. In *Proceedings of the 6th International Conference on Semantic Systems*, page 42. ACM, 2010.
- [110] Tinne Tuytelaars, Mario Fritz, Kate Saenko, and Trevor Darrell. The nbn kernel. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1824–1831. IEEE, 2011.
- [111] J.C. van Gemert, C.J. Veenman, A.W.M. Smeulders, and J.M. Geusebroek. Visual word ambiguity. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(7):1271–1283, 2010.
- [112] Avery Wang et al. An industrial strength audio search algorithm. In *ISMIR*, pages 7–13, 2003.
- [113] James Ze Wang, Jia Li, and Gio Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(9):947–963, 2001.
- [114] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367. IEEE, 2010.
- [115] Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Semi-supervised hashing for scalable image retrieval. *Computer Vision and Pattern Recognition*, 0:3424–3431, 2010.
- [116] R. Weber, H. J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proc. of Int. Conf. on Very Large Data Bases*, pages 194–205, 1998.
- [117] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB*, volume 98, pages 194–205, 1998.
- [118] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In *Advances in neural information processing systems*, pages 1753–1760, 2009.
- [119] Jian xiong Dong, Ching Y. Suen, and Adam Krzyzak. Effective shrinkage of large multi-class linear svm models for text categorization. In *ICPR*, 2008.
- [120] Atsuo Yoshitaka and Tadao Ichikawa. A survey on content-based retrieval for multimedia databases. *Knowledge and Data Engineering, IEEE Transactions on*, 11(1):81–93, 1999.
- [121] Hsiang-Fu Yu, Cho-Jui Hsieh, Kai-Wei Chang, and Chih-Jen Lin. Large linear classification when data cannot fit in memory. *ACM Trans. Knowl. Discov. Data*, 2012.
- [122] P. Zezula, P. Savino, G. Amato, and F. Rabitti. Approximate similarity retrieval with m-trees. *Very Large Data Bases Journal*, 7(4):275–293, 1998.
- [123] Jiaqi Zhai, Yin Lou, and Johannes Gehrke. Atlas: a probabilistic algorithm for high dimensional similarity search. In *Proceedings of the 2011 international conference on Management of data, SIGMOD '11*, pages 997–1008, New York, NY, USA, 2011. ACM.
- [124] Peng Zhao and Bin Yu. Stagewise lasso. *The Journal of Machine Learning Research*, 8:2701–2726, 2007.
- [125] Xiang Sean Zhou and Thomas S Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia systems*, 8(6):536–544, 2003.



# Bibliography - Second part

## (author's references)

- [126] Vera Bakic, Sofiène Mouine, Saloua Ouertani-Litayem, Anne Verroust-Blondet, Itheri Yahiaoui, Hervé Goëau, Alexis Joly, et al. Inria's participation at imageclef 2013 plant identification task. In *CLEF (Online Working Notes/Labs/Workshop) 2013*, 2013.
- [127] Vera Bakic, Itheri Yahiaoui, Sofiene Mouine, Saloua Litayem Ouertani, Wajih Ouertani, Anne Verroust-Blondet, Hervé Goëau, Alexis Joly, et al. Inria imedia2's participation at imageclef 2012 plant identification task. In *CLEF (Online Working Notes/Labs/Workshop) 2012*, 2012.
- [128] Barbara Caputo, Henning Muller, Bart Thomee, Mauricio Villegas, Roberto Paredes, David Zellhofer, Herve Goeau, Alexis Joly, Pierre Bonnet, Jesus Martinez Gomez, et al. Imageclef 2013: the vision, the data and the open challenges. In *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, pages 250–268. Springer Berlin Heidelberg, 2013.
- [129] Hervé Goëau, Pierre Bonnet, Julien Barbe, Vera Bakic, Alexis Joly, Jean-François Molino, Daniel Barthelemy, and Nozha Boujemaa. Multi-organ plant identification. In *Proceedings of the 1st ACM international workshop on Multimedia analysis for ecological data*, pages 41–44. ACM, 2012.
- [130] Hervé Goëau, Pierre Bonnet, Alexis Joly, Antoine Affouard, Vera Bakic, Julien Barbe, Samuel Dufour, Souheil Selmi, Itheri Yahiaoui, Christel Vignau, et al. Pl@ntnet mobile 2014: Android port and new features. In *Proceedings of International Conference on Multimedia Retrieval*, page 527. ACM, 2014.
- [131] Hervé Goëau, Pierre Bonnet, Alexis Joly, Vera Bakić, Julien Barbe, Itheri Yahiaoui, Souheil Selmi, Jennifer Carré, Daniel Barthélémy, Nozha Boujemaa, et al. Pl@ntnet mobile app. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 423–424. ACM, 2013.
- [132] Hervé Goëau, Pierre Bonnet, Alexis Joly, Nozha Boujemaa, Daniel Barthelemy, Jean-François Molino, Philippe Birnbaum, Elise Mouysset, Marie Picard, et al. The imageclef 2011 plant images classification task. In *ImageCLEF 2011*, 2011.
- [133] Hervé Goëau, Pierre Bonnet, Alexis Joly, Nozha Boujemaa, Daniel Barthelemy, Jean-François Molino, Philippe Birnbaum, Elise Mouysset, Marie Picard, et al. The imageclef 2012 plant identification task. In *CLEF working notes 2012*, 2012.
- [134] Hervé Goëau, Alexis Joly, Pierre Bonnet, Julien Barbe, Souheil Selmi, Julien Champ, Jean-François Molino, Daniel Barthélémy, and Nozha Boujemaa. A look inside the pl@ntnet experience: the good, the bias and the hope. *Multimedia Systems (accepted)*, pages 0–0, 2015.
- [135] Hervé Goëau, Alexis Joly, Souheil Selmi, Pierre Bonnet, Elise Mouysset, Laurent Joyeux, Jean-François Molino, Philippe Birnbaum, Daniel Bathelemy, and Nozha Boujemaa. Visual-based plant species identification from crowdsourced data. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 813–814. ACM, 2011.

- [136] Hervé Goëau, Alexis Joly, Itheri Yahiaoui, Vera Bakic, Anne Verroust-Blondet, Pierre Bonnet, Daniel Barthélémy, Nozha Boujemaa, Jean-François Molino, et al. Plantnet participation at lifeclef2014 plant identification task. *CLEF2014 Working Notes. Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014*, pages 724–737, 2014.
- [137] Amel Hamzaoui, Alexis Joly, and Nozha Boujemaa. Multi-source rsc model for multiple search result clustering. In *Image Analysis for Multimedia Interactive Services (WIAMIS), 2010 11th International Workshop on*, pages 1–4. IEEE, 2010.
- [138] Amel Hamzaoui, Alexis Joly, and Nozha Boujemaa. Multi-source shared nearest neighbours for multi-modal image clustering. *Multimedia Tools and Applications*, 51(2):479–503, 2011.
- [139] Amel Hamzaoui, Pierre Letessier, Alexis Joly, Olivier Buisson, and Nozha Boujemaa. Object-based visual query suggestion. *Multimedia Tools and Applications*, 68(2):429–454, 2014.
- [140] A. Joly. *Recherche par similarité statistique dans une grande base de signatures locales pour l'identification rapide d'extraits vidéo*. 2005.
- [141] A Joly, O Buisson, and C Frélicot. Content-based copy retrieval using distortion-based probabilistic similarity search. *IEEE Transactions on Multimedia*, 9(2):293–306, 2007.
- [142] Alexis Joly. Imageval task 1: Retrieving transformed images with new local descriptors based on dissociated dipoles. 2007.
- [143] Alexis Joly. New local descriptors based on dissociated dipoles. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 573–580. ACM, 2007.
- [144] Alexis Joly and Olivier Buisson. Discriminant local features selection using efficient density estimation in a large database. In *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pages 201–208. ACM, 2005.
- [145] Alexis Joly and Olivier Buisson. A posteriori multi-probe locality sensitive hashing. In *Proceedings of the 16th ACM international conference on Multimedia*, pages 209–218. ACM, 2008.
- [146] Alexis Joly and Olivier Buisson. Logo retrieval with a contrario visual query expansion. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 581–584. ACM, 2009.
- [147] Alexis Joly and Olivier Buisson. Random maximum margin hashing. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 873–880. IEEE, 2011.
- [148] Alexis Joly, Olivier Buisson, and Carl Frélicot. Content-based copy retrieval using distortion-based probabilistic similarity search. *Multimedia, IEEE Transactions on*, 9(2):293–306, 2007.
- [149] Alexis Joly, Carl Frélicot, and Olivier Buisson. Robust content-based video copy identification in a large reference database. In *Image and Video Retrieval*, pages 414–424. Springer Berlin Heidelberg, 2003.
- [150] Alexis Joly, Carl Frélicot, and Olivier Buisson. Feature statistical retrieval applied to content based copy identification. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 1, pages 681–684. IEEE, 2004.
- [151] Alexis Joly, Carl Frélicot, and Olivier Buisson. Content-based video copy detection in large databases: A local fingerprints statistical similarity search approach. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 1, pages I–505. IEEE, 2005.
- [152] Alexis Joly, Hervé Goëau, Pierre Bonnet, Vera Bakić, Julien Barbe, Souheil Selmi, Itheri Yahiaoui, Jennifer Carré, Elise Mouysset, Jean-François Molino, et al. Interactive plant identification based on social image data. *Ecological Informatics*, 2013.

- [153] Alexis Joly, Hervé Goëau, Pierre Bonnet, Vera Bakic, Jean-François Molino, Daniel Barthélémy, Nozha Boujemaa, et al. The imageclef plant identification task 2013. In *International workshop on Multimedia analysis for ecological data*, 2013.
- [154] Alexis Joly, Hervé Goëau, Hervé Glotin, Concetto Spampinato, Pierre Bonnet, Willem-Pier Vellinga, Robert Planque, Andreas Rauber, Robert Fisher, and Henning Müller. Lifeclef 2014: multimedia life species identification challenges. In *Information Access Evaluation. Multilinguality, Multimodality, and Interaction*, pages 229–249. Springer International Publishing, 2014.
- [155] Alexis Joly, Julien Law-To, and Nozha Boujemaa. Inria-imedia trecvid 2008: Video copy detection. In *TRECVID*, 2008.
- [156] J Law-To, A Joly, and N Boujemaa. Muscle-vcd-2007: a live benchmark for video copy detection, 2007.
- [157] Julien Law-To, Li Chen, Alexis Joly, Ivan Laptev, Olivier Buisson, Valerie Gouet-Brunet, Nozha Boujemaa, and Fred Stentiford. Video copy detection: a comparative study. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 371–378. ACM, 2007.
- [158] Julien Law-To, Alexis Joly, Laurent Joyeux, Nozha Boujemaa, Olivier Buisson, and Valerie Gouet-Brunet. Video and image copy detection demo. In *Proceedings of the 6th ACM International conference on Image and Video Retrieval*, pages 97–100. ACM, 2007.
- [159] Pierre Letessier, Olivier Buisson, and Alexis Joly. Consistent visual words mining with adaptive sampling. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, page 49. ACM, 2011.
- [160] Pierre Letessier, Olivier Buisson, and Alexis Joly. Scalable mining of small visual objects. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 599–608. ACM, 2012.
- [161] Pierre Letessier, Olivier Buisson, Alexis Joly, et al. Scalable mining of small visual objects (with new experiments). 2013.
- [162] Pierre Letessier, Nicolas Hervé, Julien Champ, Alexis Joly, Buisson Olivier, and Amel Hamzaoui. Small objects query suggestion in a large web-image collection. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 417–418. ACM, 2013.
- [163] Valentin Leveau, Alexis Joly, Olivier Buisson, Pierre Letessier, and Patrick Valduriez. Recognizing thousands of legal entities through instance-based visual classification. In *Proceedings of the ACM International Conference on Multimedia*, pages 1029–1032. ACM, 2014.
- [164] Saloua Litayem, Alexis Joly, and Nozha Boujemaa. Interactive objects retrieval with efficient boosting. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 545–548. ACM, 2009.
- [165] Saloua Litayem Ouertani, Alexis Joly, Nozha Boujemaa, et al. Hash-based support vector machines approximation for large scale prediction. *British Machine Vision Conference*, 2012.
- [166] Ahmed Rebai, Alexis Joly, and Nozha Boujemaa. Interpretable visual models for human perception-based object retrieval. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, page 21. ACM, 2011.
- [167] Ahmed Rebai, Alexis Joly, and Nozha Boujemaa. Blasso for object categorization and retrieval: Towards interpretable visual models. *Pattern Recognition*, 45(6):2377–2389, 2012.
- [168] Mohamed Riadh Trad, Alexis Joly, and Nozha Boujemaa. Large scale visual-based event matching. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, page 53. ACM, 2011.
- [169] Mohamed Riadh Trad, Alexis Joly, and Nozha Boujemaa. Distributed knn-graph approximation via hashing. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, page 43. ACM, 2012.

- [170] Mohamed Riadh Trad, Alexis Joly, and Nozha Boujemaa. Large scale knn-graph approximation. In *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on*, pages 439–448. IEEE, 2012.