Advanced Techniques in CBIR
Local Descriptors, Visual Dictionaries and Bags of Features

Eduardo Valle
Computing Institute
State University of Campinas
Campinas — SP, Brazil
mail@eduardovalle.com

Matthieu Cord
Department of Databases and Machine Learning
LIP6, University of Paris 6
Paris, France
Matthieu.Cord@lip6.fr

***Abstract*— Local descriptors have been extensively used in CBIR systems, where their robustness to intense geometric and photometric transformations allows the identification of a target object/image with great reliability. However, due to their excessive discriminating power, their application to the retrieval of complex categories is challenging. The introduction of the technique of visual dictionaries (also known as dictionary of visual terms) is an important step towards the conciliation between the robustness of local descriptors and the flexibility of generalization needed by complex queries. As a bonus, we become able to employ advanced retrieval techniques which were so far available only for textual data.**

***Keywords*-CBIR; information retrieval; local descriptors; visual dictionaries; bags of features.**

I. INTRODUCTION

In this tutorial we will address the conceptual components of a powerful framework, which allows performing the semantic classification and retrieval of images and videos, using a technique of visual dictionaries and visual words.

Multimedia retrieval and classification is often solved by associating keywords or other textual annotation to the data, either by hand, either by using related text (e.g., captions, alternate descriptions, subtitles, text near to an image in a page, etc.). However, this solution is far from ideal: manual annotation is expensive, slow and, often, inconsistent; the related text, besides not always being available, is imprecise and scarce. In addition, there are the difficulties inherent to the use of keywords: synonymy, polysemy, inconsistent generalization / specialization, etc.

An interesting alternative is to use *content-based information retrieval* (CBIR), which exploits information automatically extracted from the multimedia content. Content-based techniques have the enormous advantage of bypassing the need for keywords or other annotation metadata explicitly associated to the documents.

We consider two applicative contexts: semantic classification and interactive information retrieval. In the former, there is a previous learning of the semantic categories, using a set of training images, and then a classification step, where the images on the database are classified. In the latter, learning, classification and presentation of the results are performed interactively, with the participation of the user, and often involving the so called *relevance feedback*, like we show on Figure 1.

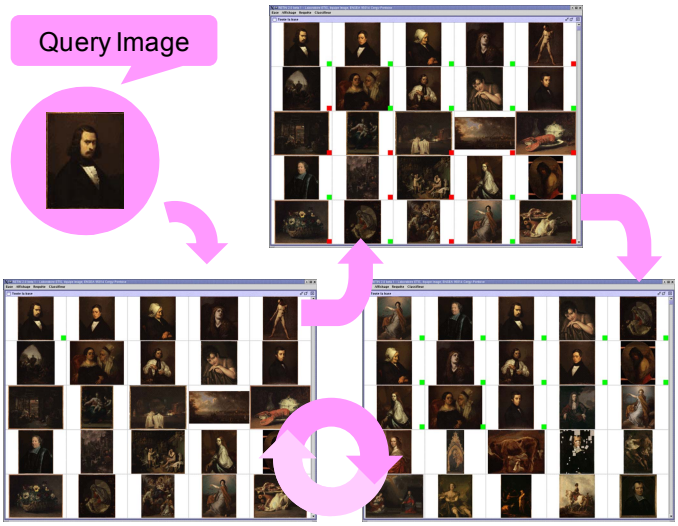

Figure 1. Interactive retrieval for semantic categories — from a sample image, the user looks for the category "portraits"; the concept is learned interactively, through positive and negative annotations.

Content-based systems face many challenges, among which the first is the sheer volume of data to process, since multimedia information is so much more massive than textual data. But the biggest hindrance is created by multimedia information not being composed by easy to identify semantic units (like words in text), creating the so called *semantic gap*, i.e., the large divergence between how the data is coded (using pixels, frames, samples, etc.) and how the users want to retrieve and classify it (using complex concepts like "person", "tropical vegetation" or "vacation with Auntie Rita").

To try to overcome those difficulties, indexing, retrieval and classification of multimedia is mediated by *descriptors*,

which are a more compact and (hopefully) semantically richer representation of the data. Descriptors appear in a large variety of forms: color and texture histograms, invariant moments, Fourier coefficients, local jets, gradient maps, etc. They often appear as vectors, which are frequently high-dimensional. To establish the similarity / dissimilarity between documents, their descriptors are compared using a distance or dissimilarity function [1].

A CBIR system has an *offline phase*, when the documents will have their descriptions computed and the descriptors will be stored in a database and, very probably, indexed. When description and indexing are ready, the user can present a query to the system, in what is called the *online phase*.

The separation between the online and offline phases is more conceptual than concrete, all depending on the system. If it is static, then the separation is perceived concretely by the user, because in order to add new images, some kind of "rebuilding" of the descriptor database/index has to be issued. However, on a dynamic system, images can be inserted or removed during or between queries (and the database/index updated), without further ado.

The entire process is illustrated on Figure 2.

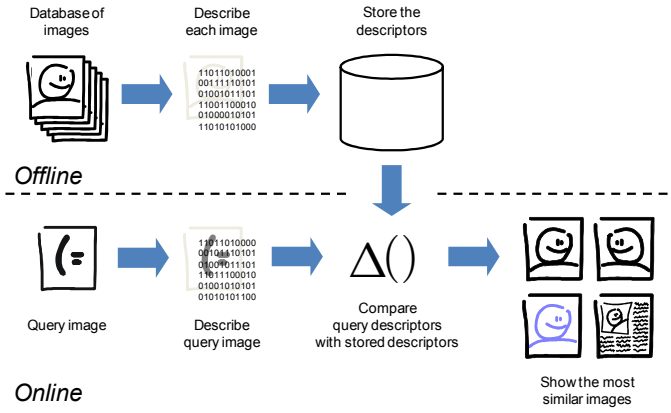

Figure 2. The basic workings of a CBIR system — descriptors are used to mediate the retrieval operation. A dissimilarity function between descriptors is used to establish the dissimilarity between images.

The two main difficulties faced by every content-based system are:

- How to synthesize the information contained in the image in a descriptor (or set of descriptors) which can be used to faithfully represent the desired visual, aesthetical or semantical characteristics which the system will be later looking for;
- How to create a dissimilarity criterion between descriptors (or sets of descriptors), which correctly emulates the user mental model of (visual, aesthetical or semantical) similarity.

The same desiderata apply for the descriptor and the dissimilarity: as little cost as possible; and a delicate balance between analytic and synthetic powers, in order to be able to set apart the images that we (using the metal model) consider extraneous, but put together those that we consider fitting. Incidentally, it is interesting to remark that the burthen of obtaining this balance may be shared by the descriptor and the dissimilarity in radically different proportions, accordingly to the design choices of the system. To mention two extremes, some systems choose to have very simple descriptors, and use complex dissimilarity functions capable to recognize the patterns [2]–[3]; others put most of the intelligence on the descriptors themselves, and use very simple dissimilarity criterions [4].

In spite of the difficulties, the tasks of visual recognition are becoming more and more daring. While, the first works on semantic search focused on stereotypical categories (e.g., sunset scenes, zebras, savanna landscapes…), in the last editions of the international campaign TRECVID [5] the categories to classify had huge visual variability (e.g., flowers), and even included the representation of abstract concepts (e.g., entertainment, sadness, art…). The ultimate challenge is answering very high-level user queries (general semantic categories such as "images from Europe", fuzzy aesthetic criterions such as "paintings I like").

A. *Local Descriptors*

In the context of retrieval and classification of *images*, the documents may be described either by a single descriptor or a set of descriptors. In the former case, when a single descriptor must capture the entire information of the image, we say it is a *global descriptor*. In the latter case, the descriptors are associated to different features of the image (regions, edges or small patches around points of interest) and are called *local descriptors*. Local descriptors have been initially proposed to solve problems in computer vision, from point matching in stereovision, to object detection [6]. They are very robust to occlusions, cropping and geometric transformations.

Local descriptors are computed over local features: regions, borders or Points of Interest. Repeatability is the most important quality for a local feature technique: even if the image suffers geometric or photometric deformations, or if the scene is observed from another viewpoint, the "same" features must be found. In practical terms, that means that the regions, borders and points extracted must have suffered the same geometric transformation than the image, in order to fall over the same objects.

Points of Interest (PoI), which are points of the image that can be uniquely located, are the most popular features, due to their robustness. Once a point is detected, a small patch around the point is used to compute the descriptor. It is usual to associate a scale and an orientation to the point, covariant to any deformation the image might suffer, in order to obtain descriptor invariance.

Some PoI detectors are based on locating corners, like the Harris [7] and Hessian [8] techniques, and their scale-invariant [9] and affine-invariant versions [10]. Other PoI

detectors are based on locating "blobs", i.e., relatively homogeneous regions, like the ones based on Difference-of-Gaussians [11]. A feature detector not based on PoI which is worth noting for its robustness is MSER [12], which is based on the concept of regions which remain "stable" over large ranges of binarization thresholds.

Some techniques propose both the feature detector and the descriptor. One of the most used is SIFT (*Scale Invariant Feature Transforms*, [11][13]), which finds the points through local extrema in a Difference-of-Gaussians function and describe them using histograms of gradients. The descriptor used in SIFT is considered one of the most robust in the literature [14], but it is very high-dimensional. A lesser dimensionality alternative is SURF (*Speeded Up Robust Features*, [15]) — which also proposes a PoI detector.

A comparative survey for PoI detectors has been presented in [16] and a comparative survey of local descriptors has been presented in [14].

In order to establish the similarity between images, a local-descriptor based system has to compare sets of descriptors. This is a potentially complex operation, but most systems adopt a criterion of plain vote count, which has the merit of being simple, and of avoiding the expensive pairwise comparison between the query image and each image on the database. The technique works like that: each individual descriptor in the query is matched with its most similar descriptors in the entire database. Each matched descriptor votes for the image to which it belongs. Then we count the votes each image received and use this number as a criterion of similarity (Figure 3).

The method is robust because the descriptors are many: if some get too distorted or are completely lost, enough will remain to guarantee good results. Even if some are matched incorrectly, giving votes for the wrong images, only a correctly identified image will receive a significant amount of votes. The process can be made even more robust by adding geometric consistency constraints which eliminate most spurious matches (more about that on § III).

The earliest trace of the use of a vote-counting algorithm for recognition in computer vision appears in Hough's method for line identification [17], later generalized for arbitrary shapes [18]. The essential elements of the architecture described above appeared in Schmid et al. [19]: the use of PoI, local descriptors computed around those points, a dissimilarity criterion based on a vote-counting algorithm, and a step of consistency checking on the matches before the final vote-count and ranking of the results. Developing on those ideas, Lowe proposed an efficient scheme to find points of interest using Differences-of-Gaussians of the image, and then a very robust descriptor computed on a patch around those points. He then uses a voting algorithm to locate objects on scenes [11].

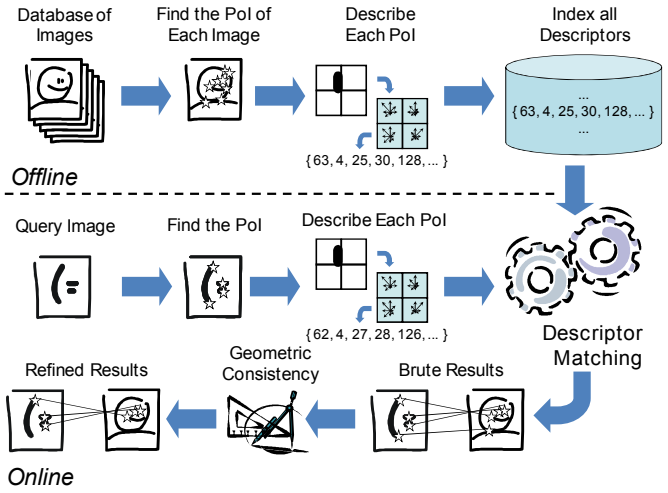

Figure 3. The workings of a CBIR system based of voting over local descriptors. This system architecture is very effective for target search, but cannot cope with semantic category search.

II. Scalability Issues

The robustness of local-descriptor based systems comes with a price, for the multiplicity of descriptors brings a performance penalty, now that hundreds, even thousands of matches must be found in order to process a single query.

To answer the user queries, the descriptors must be matched, and this is done with some form of similarity search: like kNN search (*k Nearest Neighbors search*) or range search. For very small databases, those operations can be performed by sequential comparison, where we compare each element of the database to the query, and keep the most similar. Unfortunately, this brute-force solution is not feasible in our context, where the databases are very large.

The alternative is using an indexing scheme, in order to accelerate the search. It is very difficult, however, to effectively index multimedia descriptors, since the well-known *curse of dimensionality* hinders the working of indexing schemes as dimensionality grows [20]–[22]. Typically, for over 10 dimensions, it will be impossible to perform *exact* similarity search faster than the sequential method, and one is forced to resort to approximate methods, where one trades-off precision for speed.

Similarity search is an important topic of research, which finds many applications in addition to descriptor matching, and not surprisingly there is a vast literature about the subject. For a introduction and state-of-the-art on the subject the reader is referred to [23]. Other comprehensive references are [24] (which explores the applications of similarity search in computer vision) and [25] (which specializes on the family of metric methods).

Here, however, we are only interested on methods of practical interest in our specific context. Those methods should:

- Perform well for high-dimensional data, presenting a good compromise between precision and speed;
- Adapt well to secondary-memory storage, which in practice means that few random accesses should be performed;
- Ideally, the index generated should by dynamic, i.e., allow data insertion and deletion without much cost or performance degradation.

Surprisingly few methods are able to accomplish those requirements: many methods assume implementation in main memory (and thus, cheap random access throughout the index), other methods have prohibitively high index building times (with a forced rebuilding if the index changes too much), and so on.

LSH, or locality-sensitive hashing, uses locality-preserving hashing functions to index the data. The method uses several hash tables at once, to improve reliability [24].

Another interesting method is MEDRANK, which projects the data into several random straight lines. The one-dimensional position in the line is used to index the data [26]. The NV-Tree (formerly known as PvS) is an improvement on MEDRANK, and uses segmentation and re-projection on straight lines. Points are grouped together only if they are simultaneously near in multiple straight lines [27].

Multicurves is a scheme especially conceived for indexing high-dimensional descriptors. The technique, based on the simultaneous use of moderate-dimensional space-filling curves, has as main advantages the ability to handle high-dimensional data, to allow the easy maintenance of the indexes, and to adapt well to secondary storage, thus providing scalability to huge databases [28].

A common pattern of those methods is the use of multiple subindexes. Conceptually what happens is that a sub-query is performed on each of those subindexes, and the method then aggregates the results. Intuitively, the idea is to increase the chances of a well-succeeded search by using several parallel structures.

III. Consistency Constraints

Descriptors may be matched incorrectly, either because of invariance failures of the description algorithm, or because of approximation errors of the similarity search. On either case, the mismatched descriptors will vote for the wrong document.

If the number of query descriptors is large and the fraction of incorrect matches is small, the incorrect documents will receive significantly less votes than the correct one. Otherwise, the reliability of the voting mechanism is compromised.

If a significant fraction of matches is known to be incorrect (e.g., if the image distortions are known to be very strong), the reliability of the system can be improved by enforcing geometric consistency constraints. The idea is to scrutinize the list of images obtained by the vote counting algorithm and check, for each image, if the matches are compatible with the expected geometric transformation. The matches which do not follow the general trend are removed, the votes are recounted and the list is re-ranked.

There are several different ways to perform the geometric consistency, but they all fall on two general strategies: (1) estimating the geometric transformation by applying an (usually robust) estimator, and eliminating the matches incompatible with that transformation; (2) computing a statistical distribution of some geometrical transformation parameter (rotation, scale) of each match, and eliminating the matches which deviate too much from the mode of that distribution.

The first strategy is more precise, but also more complex to implement. In order to apply it, one has to choose the kind of transformation model whose parameters will be estimated (e.g., scale change, similarity transformation, affine transformation, etc.) and the estimator used. Since many outliers are expected, the estimator has to be robust.

A usual combination is to use the RANSAC estimator [29]. The RANSAC (RANdom Sample Consensus) is a Maximum Likelihood technique that is robust to the presence of a large fraction of outliers. It works by selecting (at random) a small set of samples and estimating the model parameters from them. The model so estimated is then used to count the inliers and the outliers. The process is iterated several times, selecting potentially different samples at each iteration. The model which generates the largest fraction of inliers is kept. The idea behind RANSAC is that, if it happens to select a sample exclusively composed of inliers, then there is a good chance that the estimated model will be compatible with all the other inliers.

The second strategy is less precise, but much simpler. In order to apply it, we first choose which parameters of the transformation we will inspect — rotation and scale being the most common —and then study the statistical behavior of that parameter. In [30], a histogram is created with the observed rotations between matched descriptors, and only the matches corresponding to the highest peak (the mode) on that histogram are kept. In [13], the generalized Hough transform [18] is used to find if there is a consistent accumulation of matches in the 4-dimensional parameter space composed by scale, rotation and 2D position.

A comparative study between the two strategies, for contexts where a large number of mismatches is expected, is presented in [31].

IV. Visual Dictionaries

That local descriptors are much more precise and discriminating than global descriptors is both an advantage and a drawback. When looking for a specific image or target

object, this discrimination power is extremely welcome, but when looking for complex categories it becomes an obstacle, since the ability to generalize becomes then essential.

A possible solution to this problem is the technique of visual dictionaries. The visual dictionary is nothing more than a representation which considers the (high-dimensional) descriptor space and split it into multiple regions, usually by employing non-supervised learning techniques, like clustering. Each region becomes then a visual "word" of the dictionary. The idea is that different regions of the description space will become associated with different semantic concepts, for example, vegetation, rocks, clear sky, clouds, corners of buildings, etc. It is important to emphasize that the association between the visual word and its semantics is latent; there is no need to explicitly attribute meanings to the words (Figure 4).

Figure 4. Image patches associated to the decriptors belonging to a typical visual word. This "word" seems to be strongly related to the concept "window in a building". There is no explicit association, however, between the words and the concepts in the visual dictionary technique: the semantics of the words remain latent. (Image from [32], reproduced with permission).

Once the dictionary is obtained, the image description is simplified, since it is no longer based on the individual descriptors, but on the words it contains. The condensed description used in this higher level may be, for example, a histogram or simply a set of the words the image contains. The advantage is twofold: the rougher description is better adapted to operations involving complex semantics; and the computational burthen of the online retrieval is alleviated, since it now operates on a summarized description, instead of a myriad of local descriptors considered individually.

Apparently, the dictionaries of features were first devised by Ma and Manjunath in their NeTra image retrieval system [33], which uses codebooks to efficiently index color, texture and shape descriptors. The idea was revisited by Fournier et al. in the RETIN system [34]. Those systems were based on global descriptors. The visual dictionaries of local descriptors, as we describe here, appear in Sivic and Zisserman in their Video Google approach [35].

Currently, building a good dictionary is the biggest challenge when employing the technique. The creation of the dictionary requires the quantization of the description space, which can be obtained by a clustering algorithm. The commonest choice found in the literature is a combination of a dimensionality reduction step using PCA (*Principal Component Analysis*) and then a clustering using the *k-means* algorithm with Euclidean distance. This typical choice (PCA + *k-means*), choice, however has also been criticized [36].

In fact, this solution is far from adequate, not only because of the incapacity of *k-means* to deal with high dimensional spaces, but also because the optimality criterion of PCA does not match the needs of clusterization, since it chooses the components which maximize the global variance of the data, not the ones which better preserve the local clusters.

Most clusterization methods pertaining to the state-of-the-art have not been conceived taking the needs of visual dictionary construction into account. Few, if any, methods are adapted for high-dimensional data, large databases and a large number of clusters to be found.

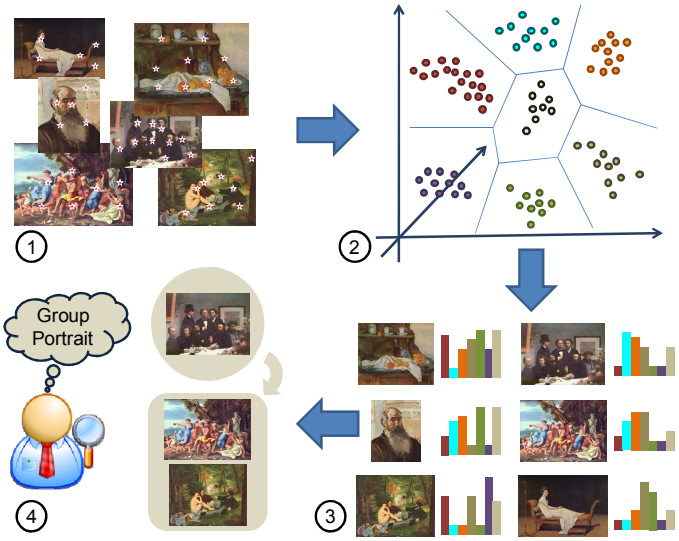

Figure 5. CBIR using bags of features — The processing steps are (1) extraction of the local descriptors; (2) quantification of the descriptor space to create the visual dictionary; (3) summarized description of the visual documents as histograms of visual words; (4) use of the summarized description to answer the user queries.

Methods based on *subspace clustering* may help to ease the problem of clustering high-dimensional data, but they are not adapted at obtaining a large number of clusters [37]–[38]. A possible solution to this issue, is to cluster hierarchically (obtain a small number of clusters and then cluster again each of the clusters obtained).

V. Bags of Features and Beyond

In addition to moderating the discriminating power of the local descriptors, the visual dictionaries introduce the possibility of bringing to CBIR techniques heretofore restricted to the universe of textual retrieval. In fact, the adaptation of those techniques becomes straightforward, by substituting the text words by the visual "words" metaphor.

Thus, classical text-retrieval methods like the pLSA [39] or the LDA [40] have been adapted to image databases [41]–[42].

One of the most successful borrowings from the text-retrieval universe has been the technique of *bags of words* (which considers textual documents simply as sets of words, ignoring any inherent structure). The equivalent in the CBIR universe has been called *bags of features*, *bags of visual features*, *bags of visual words* or simply *bags of words*, and equivalently, considers visual documents simply as sets of visual words. This technique greatly simplifies the description of the documents, which becomes a histogram of the (visual) words it contains (Figure 5).

In the same fashion the textual bags of words have been extended to take into consideration some of the structure of the document (like pairs of words, phrases, etc.) the visual bags of features has also been extended in order to capture some of the spatial structure which is lost when the document is represented simply as a "bag" of visual of words. Proposed methods have taken into account the spatial distribution of features [43]; considered pairs of spatially adjacent features (a metaphor of visual "phrases"), and considered multiscale descriptions of the images [44].

VI. Applications

Local descriptors have been extremely successful for problems involving retrieval of a target image or object. Related applications are object matching and tracking [11][13] and near-duplicate detection on image and video databases [27][28]. Both applications have benefited from the performance gains brought by the use of the visual dictionaries.

More recently, local descriptors have been employed in tasks involving complex categories. By using the visual dictionary technique, the possible applications have been greatly broadened and encompass category retrieval on image databases, category classification / detection on image databases and recognition tasks on video databases [32][44].

References

[1] R. Torres and A. Falcão. Content-based image retrieval: Theory and applications. *Revista de Informática Teórica e Aplicada*, vol. 13, no. 2, pp. 161–185. 2006.

[2] E. Chang, B. Li, G. Wu and K. Goh. "Statistical Learning for Effective Visual Information Retrieval" *in Proc. of the 2003 Int. Conf. on Image Processing (ICIP)*, vol. 3, pp. III.609–612. Barcelona, Spain, September 14–17, 2003. IEEE, 2003.

[3] M. Cord, P-H. Gosselin and S. Philipp-Foliguet. Stochastic exploration and active learning for image retrieval. *Image and Vision Computing*, vol. 25, n. 1, pp. 14–23. Elsevier, January, 2007.

[4] F. Mindru, T. Tuytelaars, L. van Gool and T. Moons. Moment invariants for recognition under changing viewpoint and illumination. *Computer Vision and Image Understanding*, vol. 94, n. 1–3, pp. 3–27. Elsevier, April–June, 2004.

[5] TRECVID — TREC Video Retrieval Evaluation. National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA. http://www-nlpir.nist.gov/projects/trecvid/

[6] N. Boujemaa, H. Houissa and H. Bischof (eds.). *WP5: State of the Art Report: Image and Video Processing for Multimedia Understanding*. Technical Report. INRIA — Imedia, France, September, 2004.

[7] C. Harris, and M. Stephens. "A combined corner and edge detector", in *Alvey Vision Conference*, pp. 147–151. 1988.

[8] K. Mikolajczyk. *Interest point detection invariant to affine transformations*. PhD thesis. Institut National Polytechnique de Grenoble. 2002.

[9] K. Mikolajczyk and C. Schmid. "Indexing based on scale invariant interest points", in *Proc. of the 8th Int. Conf. on Computer Vision*. Vancouver, Canada, pp. 525–531. 2001.

[10] K. Mikolajczyk and C. Schmid. "An affine invariant interest point detector" in *Proc. of the 7th European Conf. on Computer Vision*. Copenhagen, Denmark, vol. I, pp. 128–142. 2002.

[11] D. Lowe. "Object Recognition from Local Scale-Invariant Features", in *Proc. of the 7th Int. Conf. on Computer Vision*. vol. 2, p. 1150. September 20–25, 1999. IEEE, 1999.

[12] J. Matas, O. Chum, M. Urban, and T. Pajdla. "Robust wide baseline stereo from maximally stable extremal regions", in *Proc. of British Machine Vision Conference*, pp. 384–396. 2002.

[13] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. Journal of Computer Vision*, vol. 60, N. 2, pp. 91–110. 2004.

[14] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, n. 10, pp. 1615–1630. IEEE, October 2005

[15] H. Bay, A. Ess, T. Tuytelaars, Luc Van Gool. "SURF: Speeded Up Robust Features". *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359. 2008.

[16] T. Tuytelaars, K. Mikolajczyk, Local Invariant Feature Detectors: A Survey. *Foundations and Trends in Computer Graphics and Vision*. vol. 3, no. 3, pp.177–280. 2008.

[17] P. Hough. *Method and Means for Recognizing Complex Patterns*. US Patent n. 3,069,654. December 18, 1962.

[18] D. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, vol. 13, n. 2, pp. 111–122. Elsevier, 1981.

[19] C. Schmid and R. Mohr. Local Greyvalue Invariants for Image Retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, n. 5, pp. 530–535. IEEE, May 1997.

[20] R. Bellman. *Adaptive Control Processes: a guided tour*. Princeton University Press, 1961.

[21] C. Böhm, S. Berchtold, and D. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. ACM Computing Surveys (CSUR), vol. 33, no. 3, pp. 322–373. September, 2001.

[22] 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 the 24rd Int. Conf. on Very Large Data Bases*, New York, NY, USA, August 24–27, 1998. Morgan Kaufmann, 1998.

[23] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, 2006

[24] G. Shakhnarovich, T. Darrell and P. Indyk (eds.). *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. The MIT Press, 2006.

[25] P. Zezula, G. Amato, V. Dohnal and M. Batko. *Similarity search: the metric space approach*. Springer, 2006.

[26] R. Fagin, R. Kumar and D. Sivakumar. "Efficient similarity search and clas-sification via rank aggregation", in *Proc. of the 2003 ACM SIGMOD Int. Conf. on Management of Data*, pp. 301–312, San Diego, CA, USA, June 09–12, 2003. ACM, 2003.

[27] H. Lejsek, F. Ásmundsson, B. Jónsson and L. Amsaleg. NV-tree: An Efficient Disk-Based Index for Approximate Search in Very Large High-Dimensional Collections. *IEEE Trans. on Pattern Analysis and Machine Intelligence.* preprint. IEEE, 2008.

[28] E. Valle, M. Cord, and S. Philipp-Foliguet. “High-dimensional descriptor indexing for large multimedia databases”, in *Proc. 17th ACM Conf. on Information and Knowledge Management*. pp. 739–748, 2008.

[29] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395. 1981.

[30] H. Jegou, M. Douze, and C. Schmid. “Hamming embedding and weak geometric consistency for large scale image search”, in *Proc. 10th European Conf. on Computer Vision,* part I. pp. 304–317. 2008.

[31] E. Valle, D. Picard and M. Cord. “Geometric Consistency Checking for Local-Descriptor Based Document Retrieval” in *ACM DocEng 2009*. In Press.

[32] N. Batista, A. Lopes, A. Araújo. “Detecting Buildings in Historical Photographs Using Bag-of-Keypoints” in *SIBGRAPI 2009*. In press.

[33] W-Y. Ma, B. Manjunath. NeTra: A toolbox for navigating large databases. *Multimedia systems*, vol. 7, n. 3, pp. 184–198, 1999.

[34] J. Fournier, M. Cord, S. Philipp-Foliguet. RETIN: A content-based image indexing and retrieval system. *Pattern Analysis and Applications Journal*, vol. 4 (2/3), pp. 153–173, 2001.

[35] J. Sivic, A. Zisserman. “Video Google: a text retrieval approach to object matching in video” in Proc. 9th IEEE Int. Conf. on Computer Vision, pp. 1470–1477, 2003.

[36] F. Jurie, B. Triggs. “Creating efficient codebooks for visual recognition”, in *IEEE Int. Conf. on Computer Vision*, vol. 1, pp. 604-610. 2005.

[37] L. Parsons, E. Haque and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explor. Newslett*. 6 (1), pp. 90–105. 2004.

[38] C. Bouveyron, S. Girard and C. Schmid. High-dimensional data clustering. *Computational Statistics & Data Analysis*, vol. 52, pp. 502–519. 2007.

[39] T. Hofmann. “Probabilistic Latent Semantic Indexing”, *Proc. of the 22nd Annual Int. SIGIR Conf. on Research and Development in Information Retrieval (SIGIR-99)*. 1999

[40] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Advances in Neural Information Processing Systems*. 2002

[41] J. Sivic, A. Zisserman. “Video Google: A Text Retrieval Approach to Object Matching in Videos”, in Proc. of the Ninth IEEE Int. Conf. on Computer Vision (ICCV), October 13-16, 2003. IEEE, 2003.

[42] D. Larlus and F. Jurie. “Latent mixture vocabularies for object categorization” in *British Machine Vision Conference*. 2006.

[43] M. Marszalek, C. Schmid. “Spatial weighting for bag-of-features” in *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 2118–2125. 2006.

[44] S. Lazebnik, C. Schmid and J. Ponce. “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories”, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 2169–217., 2006.