Large Scale Image Retrieval Using Vector of Locally Aggregated Descriptors

Giuseppe Amato, Paolo Bolettieri, Fabrizio Falchi, Claudio Gennaro
{giuseppe.amato, paolo.bolettieri, fabrizio.falchi claudio.gennaro}@isti.cnr.it

ISTI - CNR, Pisa, Italy

Abstract. Vector of locally aggregated descriptors (VLAD) is a promising approach for addressing the problem of image search on a very large scale. This representation is proposed to overcome the quantization error problem faced in Bag-of-Words (BoW) representation. However, text search engines have not be used yet for indexing VLAD given that it is not a sparse vector of occurrence counts. For this reason BoW approach is still the most widely adopted method for finding images that represent the same object or location given an image as a query and a large set of images as dataset.
In this paper, we propose to enable inverted files of standard text search engines to exploit VLAD representation to deal with large-scale image search scenarios. We show that the use of inverted files with VLAD significantly outperforms BoW in terms of efficiency and effectiveness on the same hardware and software infrastructure.

Keywords: bag of features, bag of words, local features, compact codes, image retrieval.

1 Introduction

In the last few years, local features [16] extracted from selected regions [22] have emerged as a promising method of representing image content in such a way that tasks of object recognition, and other similar (e.g. landmark recognition, copy detection, etc.) can be effectively executed. A drawback of the use of local features is that a single image is represented by a large set (typically thousands) of (local) descriptors that should be individually matched and processed in order to compare the visual content of two images. In principle, a query image should be compared with each dataset object independently. In fact, each local feature of the query should be compared with all the local features of any dataset image in order to find a possible match. Moreover, candidate matches should be validated evaluating a geometric transformation (typically an Homography) able to map a region of the query to a region of the dataset image. Even though data structures as kd-tree [9] are used to efficiently search candidate matching pairs in any two images, still the approach is not scalable.

A very popular method to achieve scalability is the Bag-of-Words (BoW) [21] (or bag-of-features) approach that consists in replacing original local descriptors with the id of the most similar descriptor in a predefined vocabulary. Following the BoW approach, an image is described as a histogram of occurrence of visual words over the global vocabulary. Thus, the BoW approach used in computer vision is very similar to the traditional BoW approach in natural language processing and information retrieval [5]. However, as mentioned in [24], "a fundamental difference between an image query (e.g. 1500 visual terms) and a text query (e.g. 3 terms) is largely ignored in existing index design". From the very beginning [21] a words reduction technique was used (e.g. removing 10% of the more frequent images). In [2], removing query words with small *tf*idf* [20] revealed very good performance in improving efficiency of the BoW approach with a reduced lost in effectiveness. In this work, we make use of the parametric *tf*idf* approach for facilitating trade-offs between efficiency and effectiveness in the BoW approach.

Efficiency and memory constraints have been recently addressed by aggregating local descriptors into a fixed-size vector representation that describe the whole image. In particular, Fisher Vector (FV) [18] and VLAD [12] have shown better performance than BoW [15]. In this work we will focus on VLAD which has been proved to be a simplified non-probabilistic version of FV. Despite its simplicity, VLAD performance is comparable to that of FV [15].

Euclidean Locality-Sensitive Hashing [7] is, as far as we know, the only indexing technique tested with VLAD. While may other similarity search indexing techniques [23] could be applied to VLAD, in this work we decide to investigate the use of inverted files for allowing comparison of the VLAD and BoW approach on the same index. Permutation-Based Indexing [6, 4, 8] allows using inverted files to perform similarity search with an arbitrary similarity function. Moreover, in [10, 1] a Surrogate Text Representation (STR) derivated from the MI-File has been proposed. The conversion of the image description in a textual form allows us to employ the search engine off-the-shelf indexing and searching abilities with a little implementation effort.

In this paper, we applied the STR technique to the VLAD method comparing both effectiveness and efficiency with the state-of-the-art BoW approach on the very same hardware and software infrastructure using the publicly available and widely adopted 1M photos dataset. Given that the STR combination gives approximate results with respect to a complete sequential scan, we also compare the effectiveness of VLAD-STR with the one of standard VLAD. Moreover, we considered balancing efficiency and effectiveness with both BoW and VLAD-STR approaches. For the VLAD-STR, a similar trade-off is obtained varying the number of results used for re-ordering. Thus, we do not only compare VLAD-STR and BoW on specific settings but we show efficiency vs effectiveness graphs for both. For the VLAD-STR, a trade-off is obtained varying the number of results used for re-ordering.

Results confirm the higher performance obtained by VLAD with respect to BoW already showed in [12, 15] even when VLAD is combined with STR a off-

the-shelf text search engine (i.e., Lucene) is used. Thus, our main contribution is proving that the proposed VLAD-STR approach, can be used, in place of BoW, in combination with traditional text search engines achieving good scalability and preserving the improvement in effectiveness already showed in [15]

The paper is organized as follows. Section 2 presents relevant previous works. In Section 3 we present the STR approach that is used for indexing VLAD with a text search engine. Results are presented in Section 4. Finally, in Section 5 we present our conclusions and describe future work.

2 Related Work

2.1 Local Features

Local features [16] describe the visual content of local interest regions computed for local interest regions [22]. Good local features should be distinctive and at the same time robust to changes in viewing conditions as well as to errors of the detector. Developed mainly in Computer Vision, their typical applications include finding locations and particular objects, detecting image near duplicates and deformed copies. A drawback of the use of local features is that a single image is represented by a large set (typically thousands) of descriptors that should be individually matched and processed in order to compare the visual content of two images.

2.2 Bag of Words (BoW)

State-of-the art techniques for performing large scale content based image retrieval using local features typically involve the BoW approach. BoW was initially proposed in [21] and has been studied in many other papers. The goal of the BoW approach is to substitute each local descriptor of an images with visual words obtained from a predefined vocabulary in order to apply traditional text retrieval techniques to CBIR.

The first step is selecting some visual words creating a vocabulary. The visual vocabulary is typically built clustering, using *k-means*, local descriptors of the dataset ad selecting the centroids. The second step assigns each local descriptor to the identifier of the first nearest word in the vocabulary. For speeding-up this second phase approximate *kd-tree* is often used at a small effectiveness price. At the end of the process, each image is described as a set of visual words. The retrieval phase is then performed using text retrieval techniques considering a query image as disjunctive text-query. Typically, the *cosine* similarity measure in conjunction with a term weighting scheme is adopted for evaluating the similarity between any two images.

Even though inverted files offer a significant improvement in efficiency, in many cases efficiency is not yet satisfactory. In fact, a query image is associated with thousands of visual words. Therefore, the search algorithm on inverted files

has to access thousands of different posting lists. From the very beginning [21] words reduction techniques were used (e.g. removing 10% of the more frequent images). However, as far as we know, no experiments have been reported on the impact of the reduction on both efficiency and efficacy.

In [2], various techniques to reduce the number of words describing an image obtained with the BoW approach were evaluated. *tf*idf* [20] revealed very good performance in improving efficiency with a reduced lost in effectiveness. In this work, we make use of the parametric *tf*idf* approach to allow trade-offs between efficiency and effectiveness.

2.3 Fisher Vector

Fisher kernels [11] describe how the set of descriptors deviates from an average distribution, modeled by a parametric generative model. Fisher kernels have been applied in the context of image classification [17] and large scale image search [18]. In [15] it has been proved that Fisher vectors (FVs) extend the BoW. While the BoW approach counts the number of descriptors assigned to each region in the space, FV also encodes the proximate location of the descriptors in each region and has a normalization that can be interpreted as an IDF term. The FV image representation proposed by [17] assumes that the samples are distributed according to a Gaussian Mixture Model (GMM) estimated on a training set. Results reported in [15] reveal that FV indexed using LSH outperforms BoW.

2.4 VLAD

The VLAD representation was proposed in [12]. As for the BoW, a codebook $\{\mu_1, \ldots, \mu_K\}$ is first learned using a cluster algorithm (e.g. k-means). Each local descriptor x_t in each image is then associated to its nearest visual word $NN(x_t)$ in the codebook. For each codeword the differences between the vectors x_t assigned to μ_i are accumulated:

$$v_i = \sum_{x_t : NN(x_t) = i} x_t - \mu_i$$

VLAD is the concatenation of the accumulated vectors, i.e. $V = [v_1^T \ldots v_K^T]$. Please note that all v_i $(i = 1, ...K)$ have the same size which is equal to the size of the used local feature (e.g. 128 for SIFT). Given a codebook $\{\mu_1, \ldots, \mu_K\}$, K is fixed (typically $16 \leq K \leq 128$. Thus the dimensionality of the whole vector V describing any image is fixed too. In other words, VLAD evaluates a global descriptor statistically describing a set of local features with respect to a predefined codebook.

In order to improve the effectiveness of the VLAD approach, two normalization are performed: first, a power normalization with power 0.5; second, a L2 normalization. After this process two global descriptor V_1 and V_2 related to any two images can be compared using the inner product.

The observation that VLAD descriptor has high dimensionality but is relatively sparse and very structured suggests a principal component analysis (PCA) that is usually performed to reduce the size of the K-dimensional VLAD vectors. In this work, we decide not to use dimensionality reduction techniques because we will show that our space transformation approach is independent from the original dimensionality of the description. In fact, the STR approach that we propose, transforms the VLAD description in a set of words from a vocabulary that is independent from the original VLAD dimensionality. In our proposal, PCA could be used to increase efficiency of the STR trasformation.

In [15], it has been shown that VLAD is a simplified non-probabilistic version of FV: VLAD is to FV what k-means is to GMM clustering. The k-means clustering can be viewed as a non-probabilistic limit case of GMM clustering.

In [15] Euclidean Locality-Sensitive Hashing and its variant have been proposed to efficiently search VLAD descriptors.

3 Perspective Transformation and Surrogate Text Representation

In this paper, we propose to index the VLAD descriptors using a surrogate text representation. This allows using any text retrieval engine to perform image similarity search. As discussed later, for the experiments, we implemented these ideas on top of the Lucene text retrieval engine.

The approach to encode global features (as VLAD) used in this paper leverages on the perspective based space transformation developed in [4, 10]. The idea at the basis of this technique is that when two descriptors are very similar, with respect to a given similarity function, they 'see' the 'world around' them in the same way. In the following, we will see that the 'world around' can be encoded as a *surrogate text representation* (STR), which can be managed with an inverted index by means of a standard text-based search. The conversion of the visual descriptor in a textual form allows us to employ the search engine off-the-shelf indexing and searching abilities with a little implementation effort.

3.1 STR Generation

Let $\mathcal{D}$ be the domain of the global descriptors o, and $d : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}$ a distance function able to assess the dissimilarity between any two $o_1, o_2 \in \mathcal{D}$. Let $R \in \mathcal{D}^m$, be a vector of m distinct *reference descriptors* (or *pivots*) r_i, i.e., $R = (r_1, \ldots, r_m)$. We denote the vector of positions of the reference objects in R ranked by increasing distance with respect to an object $o \in \mathcal{D}$ as $P(o) = (p_1(o), \ldots, p_m(o))$. As an example, if $p_3(o) = 2$ then r_3 is the 2nd nearest object to o among those in R.

The objective is to define a function that transforms a global descriptor into a sequence of terms (ie, a textual document) that can be fed into a text search engine as for instance Lucene. Of course, the ultimate goal is to obtain that the distance between the documents and the query is an approximation

of the original distance function of the global descriptors. To achieve this, we associate each element $r_i \in R$ with a unique alphanumeric keyword τ_i, and define a function $t^k(o)$ that returns a space-separated concatenation of zero or more repetitions of τ_i keywords, as follows:

$$t^k(o) = \bigcup_{i=1}^{m} \left(\bigcup_{j=1}^{(k+1)-p_i^k(o)} \tau_i \right)$$

where $p_i^k(o) = p_i(o)$ if $p_i(o) < k$ and $p_i^k(o) = k$ otherwise. By abuse of notation, we denote the space-separated concatenation of keywords with the union operator $\cup$. The inner $\cup$ simply repeat $(k+1)-p_i^k(o)$ times the alphanumeric keyword τ_i used for indicating the reference object $r_i \in R$. The outer $\cup$ concatenates the repeated occurrences, if any, of keywords τ_i for $i = 1...m$. The function $t^k(o)$ is used to generate the STR to be used for both indexing and querying purposes. k is used to consider only the k nearest reference object in R to o, and typically assumes two distinct values for the query q and for the objects in the dataset (k_x for indexing and k_q for querying). For instance, consider the case exemplified in Figure 1, and let us assume τ_1 =A, τ_2 =B, etc. The function t^k will generate the following outputs

$t^{k_x}(o_1)$ = "E E E B B A"
$t^{k_x}(o_2)$ = "D D D C C E"
$t^{k_q}(q)$ = "E E A"

As can be seen intuitively, strings corresponding to o_1 and q are more similar to those corresponding to o_2 e q, this approximate the original distance d. Without going to the mathematical details, we leverage on the fact that a text based search engine will generate a vector representation of STRs generated with $t^{k_x}(o)$ and $t^{k_q}(q)$ containing the number of occurrences of words in texts. This is the case of the simple term-frequency weighting scheme. This means that, if for instance keyword τ_i corresponding to the reference object $r_i \in R$ appears n times, the i-th element of the vector will contain the number n, and whenever τ_i does not appear it will contain 0. With simple mathematical manipulations, it is easy to see how applying the cosine similarity on the query vector and a vector in the database corresponding to $t^{k_x}(o)$ and $t^{k_q}(q)$ respectively, we get a degree of similarity that reflects the similarity order of reference descriptors (pivots) around descriptors in the original space.

For more information on how the technique works from the mathematical point of view, we remind the reader to [10, 1]. The impact of k_x on the effectiveness of the search has been studied in [3].

3.2 Reordering Search Results

The idea described so far uses a textual representation of the descriptors and a matching measure based on a similarity offered by standard text search engines to order the descriptors in the dataset in decreasing similarity with respect to the query. The result set is more precise if we order it using the original distance function d.

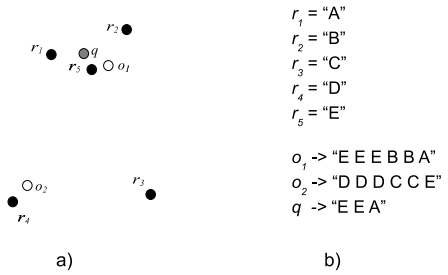

Fig. 1. Example of perspective based space transformation and Surrogate Text Representation. a) Black points are reference objects; white points are data objects; the gray point is a query. b) Encoding of the data objects in the STR.

Suppose we are searching for the k most similar (nearest neighbors) descriptors to the query. We can improve the quality of the approximation by re-ranking, using the original distance function d, the first c ($c \geq k$) descriptors from the approximate result set at the cost of more c distance computations. We will show that this technique significantly improves the accuracy, though only requiring a very low search cost. In fact, when c is much smaller than the size of the dataset, this extra cost can be considered negligible with respect to the cost of accessing the inverted file. For instance, when k is 10 and c =1,000, with a dataset size of 1,000,000 it means that we have to reorder a number of descriptors equivalent to just 0.1% of the entire dataset. Usually, this is not true for other access methods, for instance tree-based access methods, where the efficiency of the search algorithms strongly depends on the amount of descriptors retrieved.

4 Experiments

4.1 Setup

INRIA Holidays [14, 15] is a collection of 1,491 holiday images. The authors selected 500 queries and, for each of them, a list of positive results. To evaluate the approaches on a large scale, we merged the Holidays dataset with the Flickr1M[1] collection as in [13, 12, 15]. The ground–truth is the one built on the INRIA Holidays dataset alone, but it is largely accepted that no relevant images can be found between the Flickr1M images. SIFT descriptors and various vocabulary were made publicly available by Jegou et al. for both the Holidays and the Flickr 1M datasets[2]. For the BoW approach we used the 20K vocabulary.

[1] `http://press.liacs.nl/mirflickr/`
[2] `http://lear.inrialpes.fr/~jegou/data.php`

BoW		
avg#Words	**mAP**	**avg mSec**
7	0.03	525
16	0.07	555
37	0.11	932
90	0.14	1463
233	0.15	2343

Table 1. Effectiveness (mAP) and efficiency (mSec) with respect to the average number of distinct words per query obtained with the BoW approach varying the query size.

VLAD		
#reordered	**mAP**	**avg mSec**
0	0.13	139
100	0.24	205
1000	0.29	800
2000	0.30	1461
4000	0.31	2784

Table 2. Effectiveness (mAP) and efficiency (mSec) obtained with the VLAD approach in combination with STR, with respect to the number of results used for reordering.

For representing the images using the VLAD approach, we selected 64 reference descriptors using *k-means* over a subset of the Flickr1M dataset. As explained Section 3, a drawback of the perspective based space transformation used for indexing the VLAD with a text search engine is that it is an approximate technique. However, to alleviate this problem, we reorder the best results using the actual distance between the VLAD descriptors. For the STR we used 4,000 references (i.e., $m = 4,000$) randomly selected from the Flickr1M dataset.

During the experimentation also 256 references for VLAD and up to 10,000 references for the STR were selected but the results were only slightly better than the ones presented while efficiency significantly reduced.

All experiments were conducted on a Intel Core i7 CPU, 2.67 GHz with 12.0 GB of RAM a 2TB 7200 RPM HD for the Lucene index and a 250 GB SSD for the VLAD reordering. We used Lucene v3.6 running on Java 6 64 bit over Windows 7 Professional.

The quality of the retrieved images is typically evaluated by means of precision and recall measures. As in many other papers [19, 13, 18, 15], we combined this information by means of the mean Average Precision (mAP), which represents the area below the precision and recall curve.

4.2 Results

In Table 1, we report the mAP obtained with the BoW approach varying the size of the query in terms of average number of distinct words. The query words have been filtered using the *tf*idf* approach as mentioned in Section 2.2. The average number of words per image, as extracted by the INRIA group, is 1,471 and they were all inserted in the index without any filtering. The filtering was used only for the queries and results are reported for average number of distinct words up to 250. In fact, bigger queries result in heavy load of the system. It is worth to mention that we were able to obtain 0.23 mAP performing a sequential scan of the dataset with the unfiltered queries.

The results show that while the BoW approach is in principle very effective (i.e. performing a sequential scan), the high number of query visual words needed for achieve good results significantly reduces his usability. As mentioned in [24], "a fundamental difference between an image query (e.g. 1,500 visual terms) and a text query (e.g. 3 terms) is largely ignored in existing index design. This difference makes the inverted list inappropriate to index images".

In Table 2, we report the results obtained using the VLAD approach in combination with the use of the STR illustrated in Section 3. As explained in 4.1, given that for indexing the images we used a STR, it is useful to reorder the better results obtained from the text search engine using the actual VLAD distance. Thus, we report mAP and avg mSec per query for the non–reordering case and for various values of results used for reordering. The reordering phase dominates the average query time but it significantly improves effectiveness especially if only 100 or 1,000 objects are considered for reordering. As mentioned before, we make use of SSD for speed-up reordering phase but even higher efficiency could be obtained using PCA as proposed in [15]. Please note that even though the reordering phase cost for VLAD can be reduced, the reported results already show that VLAD outperform BoW.

It is worth to mention that we also performed a sequential scan of the entire dataset obtaining a mAP of 0.34 for VLAD. In fact, as depicted in 3, the results obtained with the VLAD-STR approach are an approximation of the results obtained with a complete pair wise comparison between the query and the dataset object. The same is true when LSH indexing is used as in [15]. Results show that the approximation introduced byt STR does not impact significantly the effectiveness of the system when at least 1,000 objects are considered for reordering.

In Figures 2 and 3 we report the precision and recall curves for BoW and VLAD. The results essentially confirm the ones reported in Table 1 and 2. In fact, no significant differences can be found in the distribution of the precision with respect to the recall.

In Figure 4 we plot mAP with respect to the average query execution time for both BoW and VLAD as reported in Table 1 and Table 2. The graph underlines both the efficiency and effectiveness advantages of the VLAD with respect to the BoW approach.

5 Conclusions and Future Work

In this work, we proposed the usage of STR in combination with VLAD descriptions in order to index VLAD with off-the-shelf text search engines. Using the very same hardware and text search engine (i.e., Lucene), we were able to compare with the state-of-the-art BoW. Results obtained for BoW confirm that the high number of visual terms in the query significantly reduces efficiency of inverted lists. Even though results showed that this can be mitigated reducing the number of visual terms in the query with a *tf*idf* weighting scheme, the VLAD-STR significantly outperforms BoW in terms of both efficiency and effectiveness.

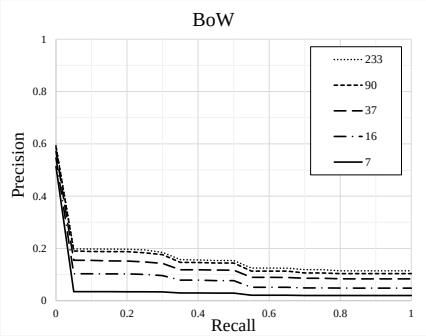

Fig. 2. Precision and recall curves obtained with the BoW approach in combination with STR for various query size.

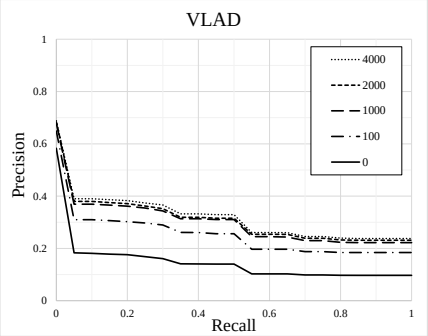

Fig. 3. Precision and recall curves obtained with the VLAD-STR for various number of results used for reordering.

The efficiency vs effectiveness graph reveals that VLAD-STR is able to obtain the same values of mAP obtained with BoW for an order of magnitude less in response time. Moreover, for the same response time, VLAD-STR is able to obtain twice the mAP of BoW.

Future work includes VLAD-STR improving the reordering phase. With regards to efficiency, PCA could be used on VLAD as suggested in [15]. Moreover, in recognition scenarios (e.g., landmark recognition) the reordering phase typically involves geometric consistency checks performed using RANSAC. This could be also done with the VLAD description.

As mentioned in the paper, VLAD is essentially a non probabilistic version of the Fisher Kernels that typically results in almost the same performance. It would be interesting to test the STR approach also with Fisher Kernels comparing with both VLAD-STR and BoW.

References

1. G. Amato, P. Bolettieri, F. Falchi, C. Gennaro, and F. Rabitti. Combining local and global visual feature similarity using a text search engine. In *Content-Based Multimedia Indexing (CBMI), 2011 9th International Workshop on*, pages 49 –54, june 2011.
2. G. Amato, F. Falchi, and C. Gennaro. On reducing the number of visualwords in the bag-of-features representation. In *VISAPP 2012 - Proceedings of the International Conference on Computer Vision Theory and Applications*, to appear.
3. G. Amato, C. Gennaro, and P. Savino. Mi-file: using inverted files for scalable approximate similarity search. *Multimedia Tools and Applications*, pages 1–30, 2012.
4. G. Amato and P. Savino. Approximate similarity search in metric spaces using inverted files. In *Proceedings of the 3rd international conference on Scalable information systems*, InfoScale '08, pages 28:1–28:10, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

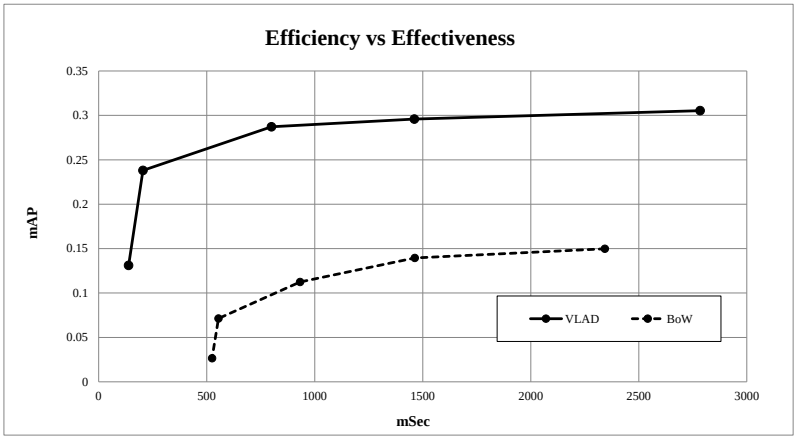

Fig. 4. Effectiveness (mAP) with respect to efficiency (mSec per query) obtained by VLAD and BoW for various settings.

5. R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval - the concepts and technology behind search, Second edition.* Pearson Education Ltd., Harlow, England, 2011.
6. E. Chávez, K. Figueroa, and G. Navarro. Effective proximity retrieval by ordering permutations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(9):1647–1658, 2008.
7. M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, SCG '04, pages 253–262, New York, NY, USA, 2004. ACM.
8. A. Esuli. Mipai: Using the pp-index to build an efficient and scalable similarity search system. In *Proceedings of the 2009 Second International Workshop on Similarity Search and Applications*, SISAP '09, pages 146–148, Washington, DC, USA, 2009. IEEE Computer Society.
9. J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, 3(3):209–226, 1977.
10. C. Gennaro, G. Amato, P. Bolettieri, and P. Savino. An approach to content-based image retrieval based on the lucene search engine library. In *Proceeding of the 14th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2010)*, LNCS.
11. T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *In Advances in Neural Information Processing Systems 11*, pages 487–493. MIT Press, 1998.
12. H. Jégou, M. Douze, J. Sánchez, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304 –3311, june 2010.
13. H. Jegou, M. Douze, and C. Schmid. Packing bag-of-features. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2357 –2364, 29 2009-oct. 2 2009.

14. H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 3304–3311, jun 2010.
15. H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Sept. 2012. QUAERO.
16. K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615 –1630, oct. 2005.
17. F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1 –8, june 2007.
18. F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3384 –3391, june 2010.
19. J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
20. G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.
21. J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, pages 1470–, Washington, DC, USA, 2003. IEEE Computer Society.
22. T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Found. Trends. Comput. Graph. Vis.*, 3(3):177–280, 2008.
23. P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity Search - The Metric Space Approach*, volume 32 of *Advances in Database Systems*. Kluwer, 2006.
24. X. Zhang, Z. Li, L. Zhang, W.-Y. Ma, and H.-Y. Shum. Efficient indexing for large scale visual search. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1103 –1110, 29 2009-oct. 2 2009.