Region-of-Interest Retrieval in Large Image Datasets with Voronoi VLAD

Aaron Chadha and Yiannis Andreopoulos(✉)

Department of Electrical and Electronic Engineering,
University College London (UCL), London, UK
i.andreopoulos@ucl.ac.uk

Abstract. We investigate the problem of visual-query based retrieval from large image datasets when the visual queries comprise arbitrary regions of interest (ROI) rather than entire images. Our proposal is a compact image descriptor that combines the vector of locally aggregated descriptors (VLAD) of Jegou *et. al.* with a multi-level, Voronoi-based, spatial partitioning of each dataset image, and it is termed as the Voronoi VLAD (VVLAD). The proposed multi-level Voronoi partitioning uses a spatial hierarchical K-means over interest-point locations, and computes a VLAD over each cell. In order to reduce the matching complexity when handling very large datasets, we propose the following modifications. First, we utilize the tree structure of the spatial hierarchical K-means to perform a top-to-bottom pruning for local similarity maxima, rather than exhaustively matching against all cells (Fast-VVLAD). Second, we propose to aggregate VLADs of adjacent Voronoi cells in order to reduce the overall VVLAD storage requirement per image. Finally, we propose a new image similarity score for Fast-VVLAD that combines relevant information from all partition levels into a single measure for similarity. For a range of ROI queries in two standard datasets, Fast-VVLAD achieves comparable or higher mean Average Precision against the state-of-the-art Multi-VLAD framework while offering more than two-fold acceleration.

1 Introduction

Image retrieval based on visual queries is now a topic of intensive research interest since it finds many applications in visual search, detection of copyright violation, recommendation services, object or person identification, etc. [1]. The state-of-the-art for image retrieval is to first describe salient points in images using a locally-invariant feature descriptor, such as SIFT [2]. A visual vocabulary is then learned using K-means or a mixture of Gaussians (MoG), which quantize the feature space into cells (visual words). The SIFT cell assignments are then aggregated over the image to obtain a compact image representation [3]. Notable contributions in this domain have relied on the bag-of-words (BoW) image representation, or its derivatives [1,4,5], where the number of SIFTs assigned to each visual word is aggregated into a histogram used for retrieval purposes.

This work was funded in part by Innovate UK, project REVQUAL (101855), and EPSRC (Industrial PhD CASE award, co-sponsored by BAFTA).

L. Nalpantidis et al. (Eds.): ICVS 2015, LNCS 9163, pp. 218–227, 2015.
DOI: 10.1007/978-3-319-20904-3_21

Despite the success of BoW approaches, their large storage and memory access requirements make them unsuitable for image retrieval within very large image datasets (e.g., tens of millions of images). For such problems, the current state-of-the-art is the vector of locally aggregated descriptors (VLAD) [6], which is a non-probabilistic variant of the Fisher vector image descriptor [7] that encodes the distribution of SIFT assignments according to cluster centers. VLAD has been shown to achieve very competitive retrieval performance to BoW methods with substantially smaller complexity and memory footprint, i.e., requiring 16–256 bytes per image instead of tens of kilobytes, as in BoW methods [6].

We are interested in the problem of designing a visual-query based retrieval system that is capable of handling both small and large-size "object", or, more broadly, region-of-interest (ROI) queries over very large image datasets. Given a ROI representing a visual query, the proposed system should return all images from the database containing this query, with matching complexity and storage requirements that remain of the order of VLAD. This is considerably more challenging than whole-image retrieval systems, as the query object may be occluded or distorted, or be seen from different viewpoints and distances in relevant images [8]. This is also the reason why the original VLAD proposal does not perform well for this problem [9]. We therefore propose a new compact image descriptor based on VLAD, termed as Voronoi VLAD (VVLAD), in which we spatially partition the image, using a hierarchical K-means, into Voronoi cells and thus compute multiple VLADs over cells. We couple this with an adaptive search algorithm via which we minimize the overall computation for similarity identification by first finding the cells most representative to the query and then computing a single score for the image over these cells. Our system design for object retrieval adheres to the following principles:

1. The system should improve on the VLAD mean Average Precision (mAP) when ROI queries are small relative to the image size.
2. The system should maintain competitive mAP to VLAD under ROI queries occupying a sizeable proportion (or the entirety) of images.
3. The system should be amenable to big-data processing, i.e., its image descriptors' size and matching complexity should be closer to that of VLAD rather than BoW descriptors.

In the following section we discuss the background and related work. In Sect. 3 we present the offline and online components of our proposed system, including the VVLAD descriptor upon which our system is derived. Section 4 presents experimental results on the Holidays [10] and Caltech Cars (Rear) datasets [11], and Sect. 5 draws concluding remarks.

2 Background and Related Work

2.1 VLAD

VLAD is a fixed-size compact image representation that stores first order information associated with clusters of image salient points. In essence, VLAD is intrinsically related to the Fisher vector image descriptor [12].

In the offline part of the VLAD encoding, a visual word vocabulary is first learned using K-means and comprises K clusters with cluster centers $\mathbf{M} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K\}$. For each image I, N interest points are detected using an affine invariant detector and described using d-dimensional SIFT descriptors, thus forming a descriptor ensemble $\mathbf{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$. The descriptors $\boldsymbol{x}_n$, $1 \leq n \leq N$, are assigned to the nearest cluster in the vocabulary via a cluster assignment function $f(\boldsymbol{x}_i)$. VLAD then stores the residuals of the SIFT assignments from their associated cluster centers. The VLAD d-dimensional encoding $\boldsymbol{v}_k$ for the k-th cluster is given by $(1 \leq k \leq K)$ $\boldsymbol{v}_k = \sum_{\forall x_i : f(x_i)=k} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)$. The VLAD encodings for each cluster are concatenated into a single descriptor $\boldsymbol{\phi}(I) = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K]^{\mathrm{T}}$ with fixed dimension Kd, which is independent of the number of the SIFT descriptors found in the image. The VLAD vectors are then sign square rooted and L_2-normalized [13] and the vectors across all W images of a dataset are thus aggregated into a single $Kd \times W$ matrix $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_W]$.

In a practical system, the SIFT descriptor length d is typically 128; if the feature space is coarsely quantized with K set to 64, then the VLAD image descriptor has 8192 dimensions. Further dimensionality reduction is achieved with principal component analysis (PCA) and whitening, thus further minimizing the memory footprint per image descriptor [14,15]. The $D \times Kd$ projection matrix used by VLAD comprises only the D largest eigenvectors of the covariance matrix [14,15]. The projected VLAD $\tilde{\boldsymbol{\phi}}_{\text{test}}$ of each test image is then L_2-normalized, thereby completing the offline part of the VLAD generation.

During online ROI-query based retrieval, after the VLAD encoding of the ROI query has been generated, the similarity between that and the VLAD of a test image, $\tilde{\boldsymbol{\phi}}_{\text{ROI}}$ and $\tilde{\boldsymbol{\phi}}_{\text{test}}$, is simply measured by:

$$S_{\text{ROI,test}} = \left\langle \tilde{\boldsymbol{\phi}}_{\text{ROI}}, \tilde{\boldsymbol{\phi}}_{\text{test}} \right\rangle . \tag{1}$$

With L_2 normalized vectors, $S_{\text{ROI,test}}$ ranges between -1 (completely dissimilar) to 1 (perfect match).

2.2 Multi-VLAD

For ROI-based retrieval, VLAD and the similarity measure of (1) will produce suboptimal results because information encoded from the remaining parts of the dataset image will distort the similarity scoring [9]. The recently-proposed Multi-VLAD descriptor [9] attempts to resolve this issue by spatially partitioning the dataset images into rectangular blocks over three scales and computing a VLAD descriptor per block. At the finest scale (level 2), nine VLADs are encoded over a 3×3 rectangular grid. At medium scale (level 1), four VLADs are encoded over a 2×2 grid, where each block is composed of 2×2 blocks from the finest scale. Finally, a single VLAD is encoded over the whole dataset image (level 0). At each scale, Multi-VLAD excludes featureless regions near image borders by adjusting the grid boundary. Moreover, each VLAD is PCA projected and truncated to a 128-dimensional vector. The similarity is thus computed between the VLAD encoded over the query ROI and each of the 14 VLAD descriptors via

(1) and the dataset image is assigned a similarity score to the ROI equal to the maximum similarity over its constituent VLADs.

For ROI queries occupying about 11 % of image real estate, the Multi-VLAD descriptor has been shown to outperform the single (128 × 14)-D VLAD (computed over the whole image) in terms of mAP. However, Multi-VLAD achieves 20 % lower mAP than the (128 × 14)-D VLAD when queries occupy a sizeable proportion of the image [9]. In addition, it incurs a 14-fold penalty in storage and matching complexity in comparison to the baseline 128-D VLAD.

3 Proposed Voronoi-Based VLAD and its Fast Variant

The proposed VVLAD encoding, described in Subsect. 3.1, constitutes the offline component of our system. The remaining two subsections describe of the proposed acceleration for online VVLAD-based ROI query search and the memory compaction to reduce storage requirements for very large image datasets.

3.1 VVLAD Encoding

Instead of spatially partitioning the images into rectangular blocks, we propose to partition the image into Voronoi cells over L levels (scales), using hierarchical spatial K-means clustering. The key intuition is that objects that may constitute ROI queries tend to appear as clusters of salient points, potentially interspersed with featureless regions in the image. Therefore, a ROI-oriented partitioning must attempt to adaptively isolate these spatial clusters at multiple levels.

Initially, the entire image is VLAD-encoded; this comprises level 0 of VVLAD. For level 1, a spatial K-means is computed over the interest point locations in the whole image, which effectively partitions the image into V_1 Voronoi cells. Next, for level 2, a spatial K-means is computed over the interest point locations within each level-1 Voronoi cell, thus partitioning each cell into V_2 constituent cells. In general, for level l, $1 \leq l < L$, each of the V_{l-1} cells of the previous level is partitioned into V_l cells, with $V_0 \triangleq 1$. A VLAD descriptor is encoded over each cell following the description of Sect. 2.1, giving a total of $V_{\text{tot}} = 1 + \sum_{l=1}^{L-1} \prod_{m=1}^{l} V_m$ VLADs per image. As such, we construct a single PCA projection matrix for the utilized image training set. When PCA projecting the VLAD of each cell, we aggregate each level l into a single matrix $\mathbf{\Phi}_l$.

A three-level Voronoi partitioning for an image from the Caltech Cars image dataset with $V_1 = V_2 = 3$ is illustrated in Fig. 1. The detected points are shown in color in the left image of Fig. 1, and the level-1 and level-2 Voronoi cells are superimposed with dashed lines on the middle and right image (resp.), with their corresponding descriptors appearing with different colors. Evidently, there is an intrinsic dependency on the characteristics of the feature point detector. For ROI-based retrieval, we require a detector robust to scale and viewpoint changes, which detects enough points in salient regions to allow for reliable partitioning. Therefore, we use the Hessian Affine detector [16,17], which is based on the multi-scale determinant of the Hessian matrix (computed locally), and detects affine

Fig. 1. Three-level Voronoi partitioning for an image from Caltech Cars dataset. For illustration purposes, SIFT descriptors are color-differentiated for each cell (Color figure online).

covariant regions. SIFT descriptors are then produced based on the detected points. Importantly, unlike Multi-VLAD, there is no need to preprocess the image and exclude featureless regions: as shown in Fig. 1, smaller Voronoi cells are adaptively formed around regions of tight clusters of detected points.

3.2 Adaptive Search for Fast-VVLAD and Image Score

For the standard VVLAD encoding we assign an image score as the global maximum similarity over cells, using (1) for each cell. However, the proposed Voronoi partitioning essentially gives us a tree of spatial Voronoi cells where, for L levels, $\prod_{l=1}^{L-1} V_l$ "leaf" Voronoi cells exist at the bottom of the tree. Given that there is inherent mutual information between a cell and its constituent cells, rather than accessing data for all levels and measuring VLAD similarity over all V_{tot} cells of the tree indiscriminately, we can design an adaptive search with top-to-bottom tree pruning to find the most relevant Voronoi cells to the query. This reduces the overall execution time and memory accesses when a query is processed, which makes our proposal applicable to very large image databases that would contain millions of images. The top-to-bottom search is carried out in two phases.

Phase-1: Considering the cell of level $l-1$ with maximum similarity to the query [measured via (1)], in Phase-1 of the search, we assume that either this cell or a constituent cell within it (at level l) will attain high similarity to the query. If the cell of level $l-1$ is found to attain the highest similarity to the query, we terminate the search for that image at level $l-1$ and proceed to Phase-2. On the other hand, if we find that a constituent cell of level l attains the maximum similarity, we repeat Phase-1 for that cell and its constituent cells at the next level $(l+1)$, until we reach the bottom of the tree, in which case we move to Phase-2.

Phase-2: Let us denote the maximum similarity found by Phase-1 for each level l as S_l^* and assume that Phase-1 exited at level l_{ph1}, $0 \leq l_{\text{ph1}} \leq L-1$. Rather than assigning $S_{l_{\text{ph1}}}^*$ as the similarity score between the ROI query and the test image I in the dataset, we perform a weighted sum over all $S_0^*, \ldots, S_{l_{\text{ph1}}}^*$. To this end, we first compute the difference d_l, $0 \leq l \leq l_{\text{ph1}}$, between the number of interest points in the query and the number of interest points in the image dataset cell corresponding to S_l^*. This difference is subsequently used within a scaled

Gaussian function, which serves as a smoothing function and also handles cases where $d_l = 0$. The weight for S_l^* $(0 \leq l \leq l_{\text{ph1}})$ is thus defined as $w_l = \exp\left(\frac{-d_l^2}{\sigma^2}\right)$ and the level of smoothing is controlled by σ. The weight vector over all levels is L_1-normalized so that the image score can be ranked independently of the level l_{ph1} at which Phase-1 terminated. Denoting the L_1-normalized weight as $\hat{w}_l$, the proposed similarity score between a ROI query and dataset image I after Phase-2 is:

$$S_{\text{ROI},I} = \sum_{l=0}^{l_{\text{ph1}}} \hat{w}_l S_l^*. \tag{2}$$

For example, for a three-level partition, if a query object is small relative to the image size, we expect that the total number of interest points over the query would be more comparable to that of a level-2 cell. Hence, the level-2 maximum dot product S_2^* should receive the largest weighting $\hat{w}_2$ when computing the similarity score. This is expected to be a more robust similarity scoring than just taking a global maximum over all $S_0^*, \ldots, S_{l_{\text{ph1}}}^*$ (as in Multi-VLAD) as the similarity score, since we account for relevant information from all levels.

Summary: We term this two-phase search for VVLAD as the Fast-VVLAD, because it reduces the expected number of cell VLADs that are accessed at runtime. The upper bound for the Fast-VVLAD's matching complexity is $1 + \sum_{l=1}^{\min\{l_{\text{ph1}}+1, L-1\}} V_l$ inner products per image instead of the V_{tot} inner products required by VVLAD. Finally, due to the weights of (2), per image I, along with the VVLAD vector we also store the number of interest points per cell, comprising V_{tot} additional values. These values can be quantized to four bits each in order to reduce their storage requirement.

3.3 Level Projection for VVLAD Storage Compaction

Beyond the online matching complexity reduction offered by Fast-VVLAD, we can adhere to memory constraints of a practical deployment for very-large image datasets by only storing offline the PCA projected VLADs for the last level, $L-1$ and computing the VLADs for levels $0, ..., L-2$ at runtime by aggregating smaller-cell VLADs. Specifically, in the storage-efficient VVLAD, the VLAD descriptor over two constituent cells x and y (i.e., spatially-neighboring cells belonging to the same cell of the upper level), is simply found as the sum of their VLADs:

$$\tilde{\boldsymbol{\phi}}_{x \cup y} = \tilde{\boldsymbol{\phi}}_x + \tilde{\boldsymbol{\phi}}_y. \tag{3}$$

This holds because both PCA and whitening are linear mappings, therefore, if we do not consider the L_2 normalization of the individual VLAD vectors, the additivity property holds in the projected domain as well. Given that directionality is preserved under normalization, (3) provides a close approximation to the normalized VLAD computed directly over the two cells. Therefore, we can trade-off computation for memory by solely storing the last-level PCA-projected VLADs (level $L-1$) and computing all other VLADs for all lower levels at runtime via repetitive application of (3) amongst constituent cells and renormalizing

before carrying out the similarity measurement of (1). We remark that vectorized addition and scaling for normalization is extremely inexpensive in modern SIMD-based architectures. As such, this approach requires offline storage for only $\prod_{l=1}^{L-1} V_l$, instead of V_{tot} VLAD vectors.

4 Experimental Evaluation

4.1 Datasets and Evaluation Procedure

We measure performance on the Holidays and Caltech Cars (Rear) image datasets. For both datasets, a set of predefined queries and hand-annotated ground truth is used. The Hessian Affine rotation-and-affine-invariant detector [17] is used for feature detection and SIFT is used for feature description.

Caltech Cars (Rear) [11]: This dataset consists of 1155 (360 × 240) photographs of cars taken from the rear. For the purposes of this paper, we take a subset 800 images and use 400 images for learning the PCA projection matrix and visual word centers and 400 as test images. We select 10 image queries and perform three tests: *(i)* we mimic a surveillance test by selecting only the license plates as ROI-queries; *(ii)* we select as mid-scale ROI-queries a section of the car trunk, and *(iii)* use the whole images as queries. An example of the query subset is given in the left part of Fig. 2. For the license plate test, we manually create "good" and "junk" ground-truth files over matching images [8]; "junk" ground truth comprises any image in which the query (i.e., the license plate) is not visible or not distinguishable by the interest point detector.

Holidays + Flickr10k [10]: The Holidays dataset consists of 1491 high resolution images, mainly consisting of personal holiday photos. There are 500 queries of a distinct scene or object. To simulate large-scale retrieval and further diversify the test, we merge the Holidays dataset with a 10,000 (10k) subset from the Flickr1M dataset [10], which we denote Flickr10k. The visual word centers are learnt on an independent image dataset, Flickr60k, and the PCA projection

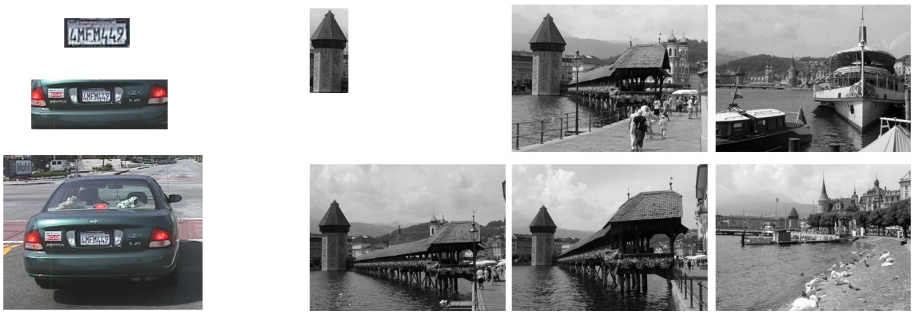

Fig. 2. (Left) Example queries for the Caltech Cars dataset. (Right) Example ROI query (top left) and matching image set for the Holidays dataset (remaining images).

matrix is learnt on a different 10k subset of the Flickr1M image dataset [10]. We use the publicly available SIFT descriptors [10] for both training and testing and select salient regions from a subset of 40 query images as ROI queries into our system. An example ROI query with its corresponding matching image set is shown in the right part of Fig. 2.

Evaluation Process: The retrieval performance is measured by creating a ranked list and computing the mAP over all queries. For VVLAD, we set: $K = 64$, $L = 3$, $V_1 = V_2 = 3$, 128-D VLAD per cell, and the maximum similarity score considered for the ranked list. Additionally, for Fast-VVLAD we set σ^2 to 0.5×10^7 for the weighting function in (2). For VLAD, for the Holidays+Flickr10k, we use: 128-D, (128×4)-D and (128×16)-D sizes, in order to align our VLAD results with the ones reported by Jegou et al. [13]. For the Caltech Cars dataset, given that no previous VLAD results are reported, we use 128-D and (128×13)-D sizes in order to align VLAD with the VVLAD storage requirement. The Multi-VLAD descriptor produces (128×14)-D size per image for both datasets [9]. Finally, per descriptor, we report the matching complexity averaged over all tests and normalized to the baseline 128-D VLAD complexity.

4.2 Performance and Results

Table 1 summarises the retrieval performance of all methods on the Caltech Cars dataset. The first observation is that the larger (128×13)-D VLAD actually performs considerably worse than the 128-D VLAD. Previous work [6] has also reported performance drop when increasing the VLAD dimension, and this can be explained by the fact that, in the case of VLAD, the additional dimensions are adding noise to the descriptor. On the other hand, Fast-VVLAD performs significantly better than the 128-D VLAD on small license plate queries, yielding substantial mAP gain of 26 %. Importantly, both Fast-VVLAD and VVLAD maintain consistently-good mAP even with the larger ROIs of car trunks and whole-image queries. Between them, Fast-VVLAD remains competitive to VVLAD, while requiring half of the average matching complexity. Interestingly, the Fast-VLAD outperforms VVLAD for whole image queries. While this may appear to be counterintuitive, this is due to the Fast-VVLAD similarity score of (2) considering all partition levels, which provides robustness against false

Table 1. Complexity and mAP results for the Caltech Cars (Rear) image dataset [11].

	D	Matching Complexity	License Plates	Trunk	Whole Image
VLAD [13]	128	1	0.504	0.738	0.726
	128×13	13	0.232	0.320	0.383
Proposed VVLAD	128×13	13	0.646	0.801	0.657
Proposed Fast-VVLAD	128×13	6.74	0.636	0.797	0.700
Multi-VLAD [9]	128×14	14	0.610	0.818	0.655

positives. Finally, both Fast-VVLAD and VVLAD outperform Multi-VLAD for small queries (license plates), whilst being lower dimensional.

The Holidays+Flickr10k dataset provides a less controlled test for our system. Crucially, in this dataset, the scale of the selected ROI and the whole images tends to vary more. Table 2 summarises the retrieval performance for the 500 whole-image queries and 40 smaller ROI queries. The Fast-VVLAD is again found to outperform VVLAD for whole image queries due to its superior similarity score of (2). The (128 × 4)-D VLAD performs best for whole image queries, but only by a small margin. Fast-VVLAD outperforms all tested VLADs for ROI queries (gains of 13 % to 48 % in mAP over VLAD). Finally, while Fast-VVLAD achieves only around 6 % higher mAP than Multi-VLAD in this dataset, it provides for more than two-fold reduction in matching complexity.

Table 2. Complexity and mAP results for the Holidays + Flickr10k dataset [10].

	D	Matching Complexity	Query ROI	Whole Image
VLAD [13]	128	1	0.185	0.473
	128 × 4	4	0.242	0.514
	128 × 16	16	0.227	0.449
Proposed VVLAD	128 × 13	13	0.282	0.459
Proposed Fast-VVLAD	128 × 13	6.74	0.274	0.485
Multi-VLAD [9]	128 × 14	14	0.259	0.459

5 Conclusion

We proposed a novel descriptor design, termed Voronoi-based vector of locally-aggregated descriptors (VVLAD), for region-of-interest (ROI) retrieval in very large datasets that is less dependent on the size of ROI. We have shown how VVLAD could fit into a practical large-scale ROI-based retrieval system via the proposed fast search, memory-efficient design and robust similarity scoring mechanisms. Our results show that the proposed VVLAD and its fast version maintain competitive retrieval performance over diverse ROI queries on two datasets and significantly improve on the retrieval performance (or implementation efficiency) of VLAD and Multi-VLAD when dealing with smaller ROI queries.

References

1. Arandjelovic, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2911–2918 (2012)
2. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. of Comput. Vis. **60**(2), 91–110 (2004)

3. Lazebnik, S. et al.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, In: IEEE International Conference on Computer Vision and Pattern Recogonition, vol. 2, pp. 2169–2178 (2006)
4. Philbin, J. et al.: Lost in quantization: improving particular object retrieval in large scale image databases. In: IEEE International Conference on Computer Vision and Pattern Recogonition, pp. 1–8 (2008)
5. Chum, O. et al.: Total recall: automatic query expansion with a generative feature model for object retrieval, In: IEEE International Conference on Computer Vision, pp. 1–8 (2007)
6. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation, In: IEEE International Conference on Computer Vision and Pattern Recogonition, pp. 3304–3311 (2010)
7. Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: IEEE International Conference on Computer Vision and Pattern Recogonition, pp. 1–8 (2007)
8. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: IEEE International Conference on Computer Vision and Pattern Recogonition, pp. 1–8 (2007)
9. Arandjelovic, R., Zisserman, A.: All about VLAD. In: IEEE International Conference on Computer Vision and Pattern Recogonition, pp. 1578–1585 (2013)
10. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
11. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: IEEE International Conference on Computer Vision and Pattern Recogonition, vol. 2, pp. II-264–II-271 (2003)
12. Perronnin, F., Liu, Y., Sánchez, J., Poirier, H.: Large-scale image retrieval with compressed fisher vectors. In: IEEE International Conference on Computer Vision and Pattern Recogonition, pp. 3384–3391 (2010)
13. Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., Schmid, C.: Aggregating local image descriptors into compact codes. In: IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 34, no. 9, pp. 1704–1716 (2012)
14. Jégou, H., Chum, O.: Negative evidences and co-occurences in image retrieval: the benefit of pca and whitening. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 774–787. Springer, Heidelberg (2012)
15. Chum, O., Matas, J.: Unsupervised discovery of co-occurrence in sparse high dimensional data. In: IEEE International Conference on Computer Vision and Pattern Recogonition, pp. 3416–3423 (2010)
16. Mikolajczyk, K., et al.: A comparison of affine region detectors. Int. J. of Comput. Vis. **65**(1–2), 43–72 (2005)
17. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part I. LNCS, vol. 2350, pp. 128–142. Springer, Heidelberg (2002)