All about VLAD

Relja Arandjelović Andrew Zisserman
Department of Engineering Science, University of Oxford
{relja,az}@robots.ox.ac.uk

Abstract

The objective of this paper is large scale object instance retrieval, given a query image. A starting point of such systems is feature detection and description, for example using SIFT. The focus of this paper, however, is towards very large scale retrieval where, due to storage requirements, very compact image descriptors are required and no information about the original SIFT descriptors can be accessed directly at run time.

We start from VLAD, the state-of-the art compact descriptor introduced by Jégou et al. *[8] for this purpose, and make three novel contributions: first, we show that a simple change to the normalization method significantly improves retrieval performance; second, we show that vocabulary adaptation can substantially alleviate problems caused when images are added to the dataset after initial vocabulary learning. These two methods set a new state-of-the-art over all benchmarks investigated here for both mid-dimensional (20k-D to 30k-D) and small (128-D) descriptors.*

Our third contribution is a multiple spatial VLAD representation, MultiVLAD, that allows the retrieval and localization of objects that only extend over a small part of an image (again without requiring use of the original image SIFT descriptors).

1. Introduction

The area of large scale particular object retrieval has seen a steady train of improvements in performance over the last decade. Since the original introduction of the bag of visual words (BoW) formulation [16], there have been many notable contributions that have enhanced the descriptors [1, 15, 19], reduced quantization loss [6, 14, 18], and improved recall [1, 3, 4].

However, one of the most significant contributions in this area has been the introduction of the Vector of Locally Aggregated Descriptors (VLAD) by Jégou *et al*. [8]. This image descriptor was designed to be very low dimensional (e.g. 16 bytes per image) so that all the descriptors for very large image datasets (e.g. 1 billion images) could still fit into main memory (and thereby avoid expensive hard disk access). Its introduction has opened up a new research theme on the trade-off between memory footprint of an image descriptor and retrieval performance, e.g. measured by average precision.

We review VLAD in section 2.1, but here mention that VLAD, like visual word encoding, starts by vector quantizing a locally invariant descriptor such as SIFT. It differs from the BoW image descriptor by recording the *difference* from the cluster center, rather than the number of SIFTs assigned to the cluster. It inherits some of the invariances of the original SIFT descriptor, such as in-plane rotational invariance, and is somewhat tolerant to other transformations such as image scaling and clipping. Another difference from the standard BoW approach is that VLAD retrieval systems generally preclude the use of the original local descriptors. These are used in BoW systems for spatial verification and reranking [6, 13], but require too much storage to be held in memory on a single machine for very large image datasets. VLAD is similar in spirit to the earlier Fisher vectors [11], as both record aspects of the distribution of SIFTs assigned to a cluster center.

As might be expected, papers are now investigating how to improve on the original VLAD formulation [2, 5]. This paper is also aimed at improving the performance of VLAD. We make three contributions:

1. Intra-normalization: We propose a new normalization scheme for VLAD that addresses the problem of burstiness [7], where a few large components of the VLAD vector can adversely dominate the similarity computed between VLADs. The new normalization is simple, and always improves retrieval performance.

2. Multi-VLAD: We study the benefits of recording multiple VLADs for an image and show that retrieval performance is improved for small objects (those that cover only a small part of the image, or where there is a significant scale change from the query image). Furthermore, we propose a method of sub-VLAD localization where the window corresponding to the object instance is estimated at a finer resolution than the the VLAD tiling.

3. Vocabulary adaptation: We investigate the problem of vocabulary sensitivity, where a vocabulary trained on one dataset, A, is used to represent another dataset B, and the performance is inferior to using a vocabulary trained on B. We propose an efficient, simple, method for improving VLAD descriptors via vocabulary adaptation, without the need to store or recompute any local descriptors in the image database.

The first two contributions are targeted at improving VLAD performance. The first improves retrieval in general, and the second partially overcomes an important deficiency – that VLAD has inferior invariance to changes in scale (compared to a BoW approach). The third contribution addresses a problem that arises in real-world applications where, for example, image databases grow with time and the original vocabulary is incapable of representing the additional images well.

In sections 3–5 we describe each of these methods in detail and demonstrate their performance gain over earlier VLAD formulations, using the Oxford Buildings 5k and Holidays image dataset benchmarks as running examples. The methods are combined and compared to the state of the art for larger scale retrieval (Oxford 105k and Flickr1M) in section 6.

2. VLAD review, datasets and baselines

We first describe the original VLAD computation and subsequent variations, and then briefly overview the datasets that will be used for performance evaluation and those that will be used for vocabulary building (obtaining the cluster centers required for VLAD computation).

2.1. VLAD

VLAD is constructed as follows: regions are extracted from an image using an affine invariant detector, and described using the 128-D SIFT descriptor. Each descriptor is then assigned to the closest cluster of a vocabulary of size k (where k is typically 64 or 256, so that clusters are quite coarse). For each of the k clusters, the residuals (vector differences between descriptors and cluster centers) are accumulated, and the k 128-D sums of residuals are concatenated into a single $k \times 128$ dimensional descriptor; we refer to it as the unnormalized VLAD. Note, VLAD is similar to other descriptors that record residuals such as Fisher vectors [11] and super-vector coding [20]. The relationship between Fisher vectors and VLAD is discussed in [12].

In the original scheme [8] the VLAD vectors are L2 normalized. Subsequently, a signed square rooting (SSR) normalization was introduced [5, 9], following its use by Perronnin *et al*. [12] for Fisher vectors. To obtain the SSR normalized VLAD, each element of an unnormalized VLAD is sign square rooted (*i.e*. an element x_i is transformed into $sign(x_i)\sqrt{|x_i|}$) and the transformed vector is L2 normalized. We will compare with both of these normalizations in the sequel, and use them as baselines for our approach.

Chen *et al*. [2] propose a different normalization scheme for the residuals and also investigate omitting SIFT descriptors that lie close to cluster boundaries. Jégou and Chum [5] extend VLAD in two ways: first, by using PCA and whitening to decorrelate a low dimensional representation; and second, by using multiple (four) clusterings to overcome quantization losses. Both give a substantial retrieval performance improvement for negligible additional computational cost, and we employ them here.

2.2. Benchmark datasets and evaluation procedure

The performance is measured on two standard and publicly available image retrieval benchmarks, Oxford buildings and Holidays. For both, a set of predefined queries with hand-annotated ground truth is used, and the retrieval performance is measured in terms of mean average precision (mAP).

Oxford buildings [13] contains 5062 images downloaded from Flickr, and is often referred to as *Oxford 5k*. There are 55 queries specified by an image and a rectangular region of interest. To test large scale retrieval, it is extended with a 100k Flickr images, forming the *Oxford 105k* dataset.

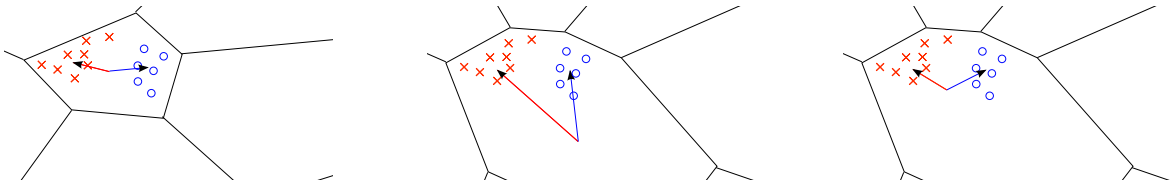
Holidays [6] contains 1491 high resolution images containing personal holiday photos with 500 queries. For large scale retrieval, it is appended with 1 million Flickr images (Flickr1M [6]), forming *Holidays+Flickr1M*.

We follow the standard experimental scenario of [5] for all benchmarks: for Oxford 5k and 105k the detector and SIFT descriptor are computed as in [10]; while for Holidays(+Flickr1M) the publicly available SIFT descriptors are used.

Vocabulary sources. Three different datasets are used for vocabulary building (i.e. clustering on SIFTs): (i) Paris 6k [14], which is analogous to the Oxford buildings dataset, and is often used as an independent dataset from the Oxford buildings [1, 3, 5, 14]; (ii) Flickr60k [6], which contains 60k images downloaded from Flickr, and is used as an independent dataset from the Holidays dataset [6, 7, 8]; and, (iii) 'no-vocabulary', which simply uses the first k (where k is the vocabulary size) SIFT descriptors from the Holidays dataset. As k is typically not larger than 256 whereas the smallest dataset (Holidays) contains 1.7 million SIFT descriptors, this vocabulary can be considered independent from all datasets.

3. Vocabulary adaptation

In this section we introduce cluster adaptation to improve retrieval performance for the case where the cluster centers used for VLAD are not consistent with the dataset – for example they were obtained on a different dataset or because



(a) (b) (c)

Figure 1: **VLAD similarity measure under different clusterings.** The Voronoi cells illustrate the coarse clustering used to construct VLAD descriptors. Red crosses and blue circles correspond to local descriptors extracted from two different images, while the red and blue arrows correspond to the sum of their residuals (differences between descriptors and the cluster center). Assume the clustering in (a) is a good one (*i.e*. it is representative and consistent with the dataset descriptors), while the one in (b) is not. By changing the clustering from (a) to (b), the sign of the similarity between the two images (from the cosine of the angle between the residuals) changes dramatically, from negative to positive. However, by performing cluster center adaptation the residuals are better estimated (c), thus inducing a better estimate of the image similarity which is now consistent with the one induced by the clustering in (a).

new data has been added to the dataset. As described earlier (section 2.1), VLAD is constructed by aggregating differences between local descriptors and coarse cluster centers, followed by L2 normalization. For the dataset used to learn the clusters (by k-means) the centers are *consistent* in that the mean of all vectors assigned to a cluster over the entire dataset is the cluster center. For an individual VLAD (from a single image) this is not the case, or course, and it is also not the case, in general, for VLADs computed over a different dataset. As will be seen below the inconsistency can severely impact performance. An ideal solution would be to recluster on the current dataset, but this is costly and requires access to the original SIFT descriptors. Instead, the method we propose alleviates the problem without requiring reclustering.

The similarity between VLAD descriptors is measured as the scalar product between them, and this decomposes as the sum of scalar products of aggregated residuals for each coarse cluster independently. Consider a contribution to the similarity for one particular coarse cluster k. We denote with $x_k^{(1)}$ and $x_k^{(2)}$ the set of all descriptors in image 1 and 2, respectively, which get assigned to the same coarse cluster k. The contribution to the overall similarity of the two VLAD vectors is then equal to:

$$\frac{1}{C^{(1)}} \sum_i (x_{k,i}^{(1)} - \mu_k)^T \frac{1}{C^{(2)}} \sum_j (x_{k,j}^{(2)} - \mu_k) \qquad (1)$$

where μ_k is the centroid of the cluster, and $C^{(1)}$ and $C^{(2)}$ are normalizing constants which ensure all VLAD descriptors have unit norm. Thus, the similarity measure induced by the VLAD descriptors is increased if the scalar product between the residuals is positive, and decreased otherwise. For example, the sets of descriptors illustrated in figure 1a are deemed to be very different (they are on opposite sides of the cluster center) thus giving a negative contribution to the similarity of the two images.

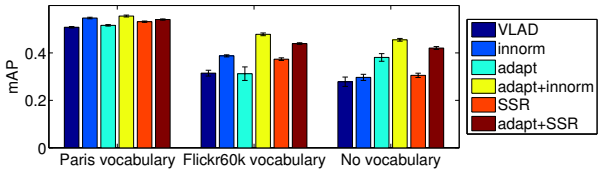
It is clear that the VLAD similarity measure is strongly affected by the cluster center. For example, if a different center is used (figure 1b), the two sets of descriptors are now deemed to be similar thus yielding a positive contribution to the similarity of the two images. Thus, a different clustering can yield a completely different similarity value.

We now introduce *cluster center adaptation* to improve residual estimates for an inconsistent vocabulary, namely, using new adapted cluster centers $\hat{\mu}_k$ that are consistent when computing residuals (equation (1)), instead of the original cluster centers μ_k. The algorithm consists of two steps: (i) compute the adapted cluster centers $\hat{\mu}_k$ as the mean of all local descriptors in the dataset which are assigned to the same cluster k; (ii) recompute all VLAD descriptors by aggregating differences between local descriptors and the adapted centers $\hat{\mu}_k$. Note that step (ii) can be performed without actually storing or recomputing all local descriptors as their assignment to clusters remains unchanged and thus it is sufficient only to store the descriptor sums for every image and each cluster.

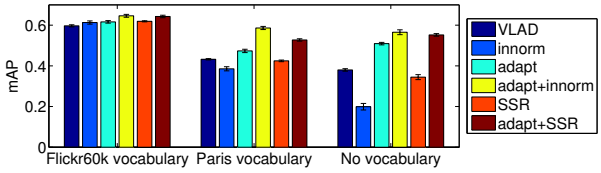
Figure 1c illustrates the improvement achieved with center adaptation, as now residuals, and thus similarity scores, are similar to the ones obtained using the original clustering in figure 1a. Note that for an adapted clustering the cluster center is indeed equal to the mean of all the descriptors assigned to it from the dataset. Thus, our cluster adaptation scheme has no effect on VLADs obtained using consistent clusters, as desired.

To illustrate the power of the adaptation, a simple test is performed where the Flickr60k vocabulary is used for the Oxford 5k dataset, and the difference between the original vocabulary and the adapted one measured. The mean magnitude of the displacements between the $k = 256$ adapted and original cluster centers is 0.209, which is very large keeping in mind that RootSIFT descriptors [1] themselves all have a unit magnitude. For comparison, when the Paris vocabulary is used, the mean magnitude of the difference is only 0.022.

Results. Figure 2 shows the improvement in retrieval performance obtained when using cluster center adaptation (*adapt*) compared to the standard VLAD under various



(a) Oxford 5k benchmark [13]



(b) Holidays benchmark [6]

Figure 2: **Retrieval performance.** Six methods are compared, namely: (i) baseline: the standard VLAD, (ii) intra-normalization, *innorm* (section 4), (iii) center adaptation, *adapt* (section 3), (iv) *adapt* followed by *innorm*, (v) baseline: signed square rooting *SSR*, (vi) aided baseline: *adapt* followed by *SSR*. Each result corresponds to the mean result obtained from four different test runs (corresponding to four different clusterings), while error bars correspond to one standard deviation. The results were generated using RootSIFT [1] descriptors and vocabularies of size $k = 256$.

dataset sources for the vocabulary. Center adaptation improves results in all cases, especially when the vocabulary was computed on a vastly different image database or not computed at all. For example, on Holidays with Paris vocabulary the mAP increases by 9.7%, from 0.432 to 0.474; while for the no-vocabulary case, the mAP improves by 34%, from 0.380 to 0.509. The improvement is smaller when the Flickr60k vocabulary is used since the distribution of descriptors is more similar to the ones from the Holidays dataset, but it still exists: 3.2% from 0.597 to 0.616. The improvement trends are similar for the Oxford 5k benchmark as well.

Application in large scale retrieval. Consider the case of real-world large-scale retrieval where images are added to the database with time. This is the case, for example, with users uploading images to Flickr or Facebook, or Google indexing images on new websites. In this scenario, one is forced to use a fixed precomputed vocabulary since it is impractical (due to storage and processing requirements) to recompute too frequently as the database grows, and reassign all descriptors to the newly obtained clusters. In this case, it is quite likely that the obtained clusters are inconsistent, thus inducing a bad VLAD similarity measure. Using cluster center adaptation fits this scenario perfectly as it provides a way of computing better similarity estimates without the need to recompute or store all local descriptors, as descriptor assignment to clusters does not change.

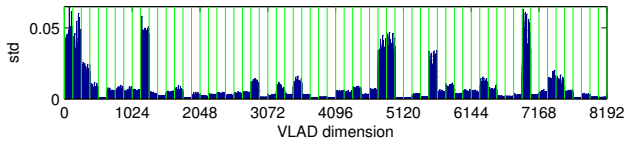
4. Intra-normalization

In this section, it is shown that current methods for normalizing VLAD descriptors, namely simple L2 normalization [8] and signed square rooting [12], are prone to putting too much weight on bursty visual features, resulting in a suboptimal measure of image similarity. To alleviate this problem, we propose a new method for VLAD normalization.

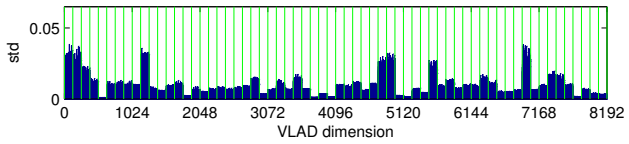
The problem of bursty visual elements was first noted in the bag-of-visual-words (BoW) setting [7]: a few artificially large components in the image descriptor vector (for example resulting from a repeated structure in the image such as a tiled floor) can strongly affect the measure of similarity between two images, since the contribution of other important dimensions is hugely decreased. This problem was alleviated by discounting large values by element-wise square rooting the BoW vectors and re-normalizing them. In a similar manner VLADs are signed square root (SSR) normalized [5, 9]. Figure 3 shows the effects these normalizations have on the average energy carried by each dimension in a VLAD vector.

We propose here a new normalization, termed *intra-normalization*, where the sum of residuals is L2 normalized *within* each VLAD block (*i.e.* sum of residuals within a coarse cluster) independently. As in the original VLAD and SSR, this is followed by L2 normalization of the entire vector. This way, regardless of the amount of bursty image features their effect on VLAD similarity is localized to their coarse cluster, and is of similar magnitude to all other contributions from other clusters. While SSR reduces the burstiness effect, it is limited by the fact that it only *discounts* it. In contrast, intra-normalization fully suppresses bursts, as witnessed in figure 3c which shows absolutely no peaks in the energy spectrum.

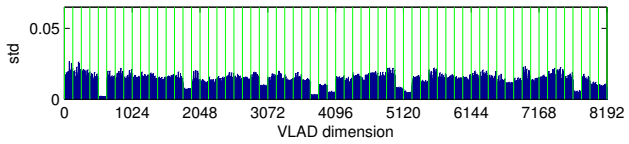
Discussion. The geometric interpretation of *intra-normalization* is that the similarity of two VLAD vectors depends on the *angles* between the residuals in corresponding clusters. This follows from the scalar product of equation (1): since the residuals are now L2 normalized the scalar product depends only on the cosine of the differences in angles of the residuals, not on their magnitudes. Chen *et al.* [2] have also proposed an alternative normalization where the per-cluster mean of residuals is computed instead of the sum. The resulting representation still depends on the magnitude of the residuals, which is strongly affected by the size of the cluster, whereas in *intra-normalization* it does not. Note that all the arguments made in favor of cluster center adaptation (section 3) are unaffected by intra-normalization. Specifically, only the values of $C^{(1)}$ and $C^{(2)}$ change in equation (1), and not the dependence of the VLAD similarity measure on the quality of coarse clustering which is addressed by cluster center adaptation.



(a) Original VLAD normalization (L2)



(b) Signed square rooting (SSR) followed by L2



(c) Intra-normalization (innorm) followed by L2

Figure 3: **The effect of various normalizing schemes for VLAD.** The plots show the standard deviation (*i.e.* energy) of the values for each dimension of VLAD across all images in the Holidays dataset; the green lines delimit blocks of VLAD associated with each cluster center. It can be observed that the energy is strongly concentrated around only a few components in the VLAD vector under the original L2 normalization scheme (3a). These peaks strongly influence VLAD similarity scores, and SSR does indeed manage to discount their effect (3b). However, even with SSR, it is clear that the same few components are responsible for a significant amount of energy and are still likely to bias similarity scores. (c) Intra-normalization completely alleviates this effect (see section 4). The relative improvement in the retrieval performance (mAP) is 7.2% and 13.5% using *innorm* compared to *SSR* and *VLAD*, respectively. All three experiments were performed on Holidays with a vocabulary of size $k = 64$ (small so that the components are visible) learnt on Paris with cluster center adaptation.

Results. As shown in figure 2, intra-normalization (innorm) combined with center adaptation (adapt) always improves retrieval performance, and consistently outperforms other VLAD normalization schemes, namely the original VLAD with L2 normalization and SSR. Center adaptation with intra-normalization (adapt+innorm) significantly outperforms the next best method (which is adapt+SSR); the average relative improvement on Oxford 5k and Holidays is 4.7% and 6.6%, respectively. Compared to SSR without center adaptation our improvements are even more evident: 35.5% and 27.2% on Oxford 5k and Holidays, respectively.

5. Multiple VLAD descriptors

In this section we investigate the benefits of tiling an image with VLADs, instead of solely representing the image by a single VLAD. As before, our constraints are the memory footprint and that any performance gain should not involve returning to the original SIFT descriptors for the image. We target objects that only cover a small part of the image (VLAD is known to have inferior performance for these compared to BoW), and describe first how to improve their retrieval, and second how to predict their localization and scale (despite the fact that VLAD does not store any spatial information).

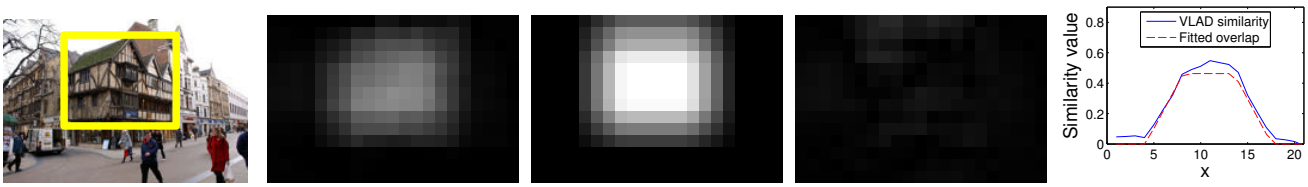
The multiple VLAD descriptors (MultiVLAD) are extracted on a regular 3×3 grid at three scales. 14 VLAD descriptors are extracted: nine (3×3) at the finest scale, four (2×2) at the medium scale (each tile is formed by 2×2 tiles from the finest scale), and one covering the entire image. At run time, given a query image and region of interest (ROI) covering the queried object, a single VLAD is computed over the ROI and matched across database VLAD descriptors. An image in the database is assigned a score equal to the maximum similarity between any of its VLAD descriptors and the query.

As will be shown below, computing VLAD descriptors at fine scales enables retrieval of small objects, but at the cost of increased storage (memory) requirements. However, with 20 bytes per image [8], 14 VLADs per image amounts to 28 GB for a 100 million images, which is still a manageable amount of data that can easily be stored in the main memory of a commodity server.

To assess the retrieval performance, additional ROI annotation is provided for the Oxford 5k dataset, as the original only specifies ROIs for the query images. Objects are deemed to be small if they occupy less than 300×300 pixels squared. Typical images in Oxford 5k are 1024×768, thus the threshold corresponds to the object occupying up to about 11% of an image. We measure the mean average precision for retrieving images containing these small objects using the standard Oxford 5k queries.

We compare to two baselines a single 128-D VLAD per image, and also a $14 \times 128 = 1792$-D VLAD. The latter is included for a fair comparison since MultiVLAD requires 14 times more storage. MultiVLAD achieves a mAP of 0.102, this outperforms the single 128-D VLAD descriptors, which only yield a mAP of 0.025, and also the 1792-D VLAD which obtains a mAP of 0.073, i.e. a 39.7% improvement. MultiVLAD consistently outperforms the 1792-D VLAD for thresholds smaller than 400^2, and then is outperformed for objects occupying a significant portion of the image (more than 20% of it).

Implementation details. The 3×3 grid is generated by splitting the horizontal and vertical axes into three equal parts. To account for potential featureless regions near image borders (*e.g.* the sky at the top of many images often contains no interest point detections), we adjust the outer boundary of the grid to the smallest bounding box which contains all interest points. All the multiple VLADs for an image can be computed efficiently through the use of an integral image of unnormalized VLADs.



(a) Image with the ROI (b) VLAD similarities (c) Region overlaps (d) Residuals (e) 1-D slice

Figure 4: **Variation of VLAD similarity with region overlaps.** (b) The value plotted at each point (x, y) corresponds to the VLAD similarity (scalar product between two VLADs) between the VLAD of the region of interest (ROI) in (a) and the VLAD extracted from the 200×200 pixel patch centered at (x, y). (c) The proportion of each patch from (b) that is covered by the ROI from (a). (d) Residuals obtained by a linear regression of (c) to (b). (e) A 1-D horizontal slice through the middle of (b) and (c). Note that residuals in (d) and (e) are very small, thus VLAD similarities are very good linear estimators of region overlap.

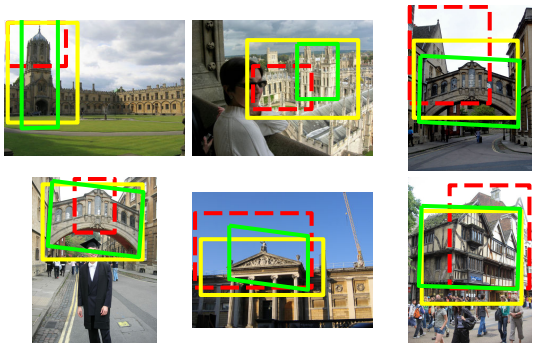


Figure 5: **Fine versus greedy localization.** Localized object: ground truth annotation (green); greedy method (red dashed rectangles); best location using the fine method of section 5.1 (yellow solid rectangles).

5.1. Fine object localization

Given similarity scores between a query ROI and all the VLADs contained in the MultiVLAD of a result image, we show here how to obtain an estimate of the corresponding location within the result image. To motivate the method, consider figure 4 where, for each 200×200 subwindow of an image, VLAD similarities (to the VLAD of the target ROI) are compared to overlap (with the target ROI). The correlation is evident and we model this below using linear regression. The procedure is similar in spirit to the interpolation method of [17] for visual localization.

Implementation details. A similarity score vector $\mathbf{s}$ is computed between the query ROI VLAD and the VLADs corresponding to the image tiles of the result image's MultiVLAD. We then seek an ROI in the result image whose overlap with the image tiles matches these similarity scores under a linear scaling. Here, overlap $\mathbf{v}(r)$ between an ROI r and an image tile is computed as the proportion of the image tile which is covered by the ROI. The best ROI, r_{best}, is determined by minimizing residuals as

$$r_{best} = \underset{r}{\operatorname{argmin}} \min_{\lambda} ||\lambda \mathbf{v}(r) - \mathbf{s}|| \qquad (2)$$

where any negative similarities are clipped to zero. Regressed overlap scores mimic the similarity scores very well, as shown by small residuals in figure 4d and 4e.

Note that given overlap scores $\mathbf{v}(r)$, which are easily computed for any ROI r, the inner minimization in (2) can be solved optimally using a closed form solution, as it is a simple least squares problem: the value of λ which minimizes the expression for a given r is $\lambda = \frac{\mathbf{s}^T \mathbf{v}(r)}{\mathbf{v}(r)^T \mathbf{v}(r)}$.

To solve the full minimization problem we perform a brute force search in a discretized space of all possible rectangular ROIs. The discretized space is constructed out of all rectangles whose corners coincide with a very fine (30 by 30) regular grid overlaid on the image, *i.e.* there are 31 distinct values considered for each of x and y coordinates. The number of all possible rectangles with non-zero area is $\binom{31}{2}^2$ which amounts to 216k.

The search procedure is very efficient as least squares fitting is performed with simple 14-D scalar product computations, and the entire process takes 14 ms per image on a single core 3 GHz processor.

Localization accuracy. To evaluate the localization quality the ground truth and estimated object positions and scales are compared in terms of the overlap score (*i.e.* the ratio between the intersection and union areas of the two ROIs), on the Oxford 5k dataset. In an analogous manner to computing mean average precision (mAP) scores for retrieval performance evaluation, for the purpose of localization evaluation the average overlap score is computed for each query, and averaged across queries to obtain the mean average overlap score.

For the region descriptors we use MultiVLAD descriptors with center adaptation and intra-normalization, with multiple vocabularies trained on Paris and projected down to 128-D. This setup yields a mAP of 0.518 on Oxford 5k.

The *fine* localization method is compared to two baselines: *greedy* and *whole image*. The *whole image* baseline returns the ROI placed over the entire image, thus always falling back to the "safe choice" and producing a non-zero overlap score. For the *greedy* baseline, the MultiVLAD retrieval system returns the most similar tile to the query in terms of similarity of their VLAD descriptors.

The mean average overlap scores for the three systems are 0.342, 0.369 and 0.429 for the *whole image*, *greedy* and

Method	Holidays	Oxford 5k
BoW 200k-D [9, 16]	0.540	0.364
BoW 20k-D [9, 16]	0.452	0.354
Improved Fisher [12]	0.626	0.418
VLAD [8]	0.526	-
VLAD+SSR [9]	0.598	0.378
Improved det/desc: VLAD+SSR [9]	-	0.532
This paper: adapt+innorm (mean)	**0.646**	**0.555**
This paper: adapt+innorm (single best)	**0.653**	**0.558**

Table 1: **Full size image descriptors (*i.e.* before dimensionality reduction): comparison with state-of-the-art.** Image descriptors of medium-dimensionality (20k-D to 32k-D) are compared in terms of retrieval performance (mAP) on the Oxford 5k and Holidays benchmarks. Reference results are obtained from the paper of Jégou *et al*. [9]. For fair comparison, we also include our implementation of VLAD+SSR using the detector [10] and descriptor [1] which give significant improvements on the Oxford 5k benchmark. The mean results are averaged over four different runs (corresponding to different random initializations of k-means for vocabulary building), and the single best result is from the vocabulary with the highest mAP.

fine respectively; the *fine* method improves the two baselines by 25% and 16%. Furthermore, we also measure the mean average number of times that the center of the estimated ROI is inside the ground truth ROI, and the *fine* method again significantly outperforms others by achieving a score of 0.897, which is a 28% and 8% improvement over *whole image* and *greedy*, respectively. Figure 5 shows a qualitative comparison of *fine* and *greedy* localization.

6. Results and discussion

In the following sections we compare our two improvements of the VLAD descriptor, namely cluster center adaptation and intra-normalization, with the state-of-the-art. First, the retrieval performance of the full size VLAD descriptors is evaluated, followed by tests on more compact descriptors obtained using dimensionality reduction, and then the variation in performance using vocabularies trained on different datasets is evaluated. Finally, we report on large scale experiments with the small descriptors. For all these tests we used RootSIFT descriptors clustered into $k = 256$ coarse clusters, and the vocabularies were trained on Paris and Flickr60k for Oxford 5k(+100k) and Holidays(+Flickr1M), respectively.

Full size VLAD descriptors. Table 1 shows the performance of our method against the current state-of-the-art for descriptors of medium dimensionality (20k-D to 30k-D). Cluster center adaptation followed by intra-normalization outperforms all previous methods. For the Holidays dataset we outperform the best method (improved Fisher vectors [12]) by 3.2% on average and 4.3% in the best case, and for Oxford 5k we achieve an improvement of 4.3% and 4.9% in the average and best cases, respectively.

Method	Holidays	Oxford 5k
GIST [9]	0.365	-
BoW [9, 16]	0.452	0.194
Improved Fisher [12]	0.565	0.301
VLAD [8]	0.510	-
VLAD+SSR [9]	0.557	0.287
Multivoc-BoW [5]	0.567	0.413
Multivoc-VLAD [5]	0.614	-
Reimplemented Multivoc-VLAD [5]	0.600	0.425
This paper: adapt+innorm	**0.625**	**0.448**

Table 2: **Low dimensional image descriptors: comparison with state-of-the-art.** 128-D dimensional image descriptors are compared in terms of retrieval performance (mAP) on the Oxford 5k and Holidays benchmarks. Most results are obtained from the paper of Jégou *et al*. [9], apart from the recent multiple vocabulary (Multivoc) method [5]. The authors of Multivoc do not report the performance of their method using VLAD on Oxford 5k, so we report results of our reimplementation of their method.

Small image descriptors (128-D). We employ the state-of-the-art method of [5] (Multivoc) which uses multiple vocabularies to obtain multiple VLAD (with SSR) descriptions of one image, and then perform dimensionality reduction, using PCA, and whitening to produce very small image descriptors (128-D). We mimic the experimental setup of [5], and learn the vocabulary and PCA on Paris 6k for the Oxford 5k tests. For the Holidays tests they do not specify which set of 10k Flickr images are used for learning the PCA. We use the last 10k from the Flickr1M [6] dataset.

As can be seen from table 2, our methods outperform all current state-of-the-art methods. For Oxford 5k the improvement is 5.4%, while for Holidays it is 1.8%.

Effect of using vocabularies trained on different datasets. In order to assess how the retrieval performance varies when using different vocabularies, we measure the proportion of the ideal mAP (*i.e.* when the vocabulary is built on the benchmark dataset itself) achieved for each of the methods.

First, we report results on Oxford 5k using full size VLADs in table 3. The baselines (VLAD and VLAD+SSR) perform very badly when an inappropriate (Flickr60k) vocabulary is used achieving only 68% of the ideal performance for the best baseline (VLAD+SSR). Using adapt+innorm, apart from improving mAP in general for all vocabularies, brings this score up to 86%. A similar trend is observed for the Holidays benchmark as well (see figure 2).

We next report results for 128-D descriptors where, again, in all cases Multivoc [5] is used with PCA to perform dimensionality reduction and whitening. In addition to the residual problems caused by an inconsistent vocabulary, there is also the extra problem that the PCA is learnt on a different dataset. Using the Flickr60k vocabulary with adapt+innorm for Oxford 5k achieves 59% of the ideal performance, which is much worse than the 86% obtained

Method \vocabulary	Ox5k	Paris	Flickr60k
VLAD	0.519	0.508 (98%)	0.315 (61%)
VLAD+SSR	0.546	0.532 (97%)	0.374 (68%)
VLAD+adapt	0.519	0.516 (99%)	0.313 (60%)
VLAD+adapt+SSR	0.546	0.541 (99%)	0.439 (80%)
VLAD+adapt+innorm	**0.555**	**0.555 (100%)**	**0.478 (86%)**

Table 3: **Effect of using different vocabularies for the Oxford 5k retrieval performance.** Column one is the ideal case where retrieval is assessed on the same dataset as used to build the vocabulary. Full size VLAD descriptors are used. Results are averaged over four different vocabularies for each of the tests. The proportion of the ideal mAP (*i.e.* when the vocabulary is built on Oxford 5k itself) is given in brackets.

with full size vectors above. Despite the diminished performance, adapt+innorm still outperforms the best baseline (VLAD+SSR) by 4%. A direction of future research is to investigate how to alleviate the influence of the inappropriate PCA training set, and improve the relative performance for small dimensional VLAD descriptors as well.

Large scale retrieval. With datasets of up to 1 million images and compact image descriptors (128-D) it is still possible to perform exhaustive nearest neighbor search. For example, in [5] exhaustive search is performed on 1 million 128-D dimensional vectors reporting 6 ms per query on a 12 core 3 GHz machine. Scaling to more than 1 million images is certainly possible using efficient approximate nearest neighbor methods.

The same 128-D descriptors (adapt+innorm VLADs reduced to 128-D using Multivoc) are used as described above. On Oxford 105k we achieve a mAP of 0.374, which is a 5.6% improvement over the best baseline, being (our reimplementation of) Multivoc VLAD+SSR. There are no previously reported results on compact image descriptors for this dataset to compare to. On Holidays+Flickr1M, adapt+innorm yields 0.378 compared to the 0.370 of Multivoc VLAD+SSR; while the best previously reported mAP for this dataset is 0.370 (using VLAD+SSR with full size VLAD and approximate nearest neighbor search [9]). Thus, we set the new state-of-the-art on both datasets here.

7. Conclusions and recommendations

We have presented three methods which improve standard VLAD descriptors over various aspects, namely cluster center adaptation, intra-normalization and MultiVLAD.

Cluster center adaptation is a useful method for large scale retrieval tasks where image databases grow with time as content gets added. It somewhat alleviates the influence of using a bad visual vocabulary, without the need of recomputing or storing all local descriptors.

Intra-normalization was introduced in order to fully suppress bursty visual elements and provide a better measure of similarity between VLAD descriptors. It was shown to be the best VLAD normalization scheme. However, we recommend intra-normalization always be used in conjunction with a good visual vocabulary or with center adaptation (as intra-normalization is sometimes outperformed by SSR when inconsistent clusters are used and no center adaptation is performed). Although it is outside the scope of this paper, intra-normalized VLAD also improves image classification performance over the original VLAD formulation.

Acknowledgements. We are grateful for discussions with Hervé Jégou and Karen Simonyan. Financial support was provided by ERC grant VisRec no. 228180 and EU Project AXES ICT-269980.

References

[1] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proc. CVPR*, 2012.
[2] D. Chen, S. Tsai, V. Chandrasekhar, G. Takacs, H. Chen, R. Vedantham, R. Grzeszczuk, and B. Girod. Residual enhanced visual vectors for on-device image matching. In *Asilomar*, 2011.
[3] O. Chum, A. Mikulik, M. Perdoch, and J. Matas. Total recall II: Query expansion revisited. In *Proc. CVPR*, 2011.
[4] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proc. ICCV*, 2007.
[5] H. Jégou and O. Chum. Negative evidences and co-occurrences in image retrieval: the benefit of PCA and whitening. In *Proc. ECCV*, 2012.
[6] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proc. ECCV*, 2008.
[7] H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *Proc. CVPR*, Jun 2009.
[8] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. CVPR*, 2010.
[9] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local images descriptors into compact codes. *IEEE PAMI*, 2012.
[10] M. Perdoch, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *Proc. CVPR*, 2009.
[11] F. Perronnin and D. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proc. CVPR*, 2007.
[12] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *Proc. CVPR*, 2010.
[13] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. CVPR*, 2007.
[14] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proc. CVPR*, 2008.
[15] K. Simonyan, A. Vedaldi, and A. Zisserman. Descriptor learning using convex optimisation. In *Proc. ECCV*, 2012.
[16] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, 2003.
[17] A. Torii, J. Sivic, and T. Pajdla. Visual localization by linear combination of image descriptors. In *International Workshop on Mobile Vision*, 2011.
[18] J. C. van Gemert, J. M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders. Kernel codebooks for scene categorization. In *Proc. ECCV*, 2008.
[19] S. Winder, G. Hua, and M. Brown. Picking the best daisy. In *Proc. CVPR*, 2009.
[20] X. Zhou, K. Yu, T. Zhang, and T. S. Huang. Image classification using super-vector coding of local image descriptors. In *Proc. ECCV*, 2010.