Memory Vectors for Particular Object Retrieval with Multiple Queries

Ronan Sicre, Hervé Jégou

Memory Vectors for Particular Object Retrieval with Multiple Queries

Ronan Sicre
Inria
ronan.sicre@inria.fr

Hervé Jégou
Inria
herve.jegou@inria.fr

ABSTRACT

We address the problem of retrieving all the images containing a specific object in a large image collection, where the input query is given as a set of representative images of the object. This problem is referred to as multiple queries in the literature. For images described with bag-of-visual-words (BOW), one of the best performing approach amounts to simply averaging the query descriptors.

This paper [1] introduces an improved fusion of the object description based on the recent concept of generalized max-pooling and memory vectors, which summarizes a set of vectors by a single representative vector. They have the property of reducing the influence of frequent features. Therefore, we propose to build a memory vector for each set of queries and the membership test is performed with each image descriptor from the database, to determine its similarity with the query representative. This new strategy for multiple queries brings a significant improvement for most of the image descriptors we have considered, in particular with Convolutional Neural Networks (CNN) features.

Categories and Subject Descriptors

H.3.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Information Search and Retrieval

Keywords

image retrieval; multiple queries; memory vectors

1. INTRODUCTION

The last decade has witnessed the rapid increase of multimedia data, which has raised numerous challenges regarding the management of large collections. Multimedia search in large collection, such as Content Based Image Retrieval (CBIR) has received a lot of attention. In this work, we are interested in multiple queries image retrieval, where the CBIR system has a set of representative images of an object as input. Arandjelovic *et al.* [1] show that simply averaging the scores offers a competitive performance, at least as good as taking the best scores across all queries. Another advantage is that, if the image is a vector representation (e.g., bag-of-words or the Fisher vector), one can equivalently (thanks to linearity of average) average the vector representations of the multiple queries. In this case, only the average vector has to be compared against the databases vectors describing the image collection.

In this paper, we are interested in improving this fusion strategy. Our motivation is that, when multiple queries are given as input, some of the images of the query object are very similar, see Figure 1. The redundant information dominates the representation, while rare discriminative features are comparatively overlooked.

Our solution is inspired by recent papers on image representations, namely generalized max-pooling [13] (GMP) and democratic aggregation [12], which improve the aggregation stage of coded local descriptors (such as Fisher) by reducing the influence of frequent, often uninformative, descriptors. Conversely, it gives more importance to rare and potentially highly discriminative patterns. Interestingly, GMP [13] was recently shown to be an effective way to construct a *memory vector* [8] summarizing, in a continuous manner, a vector set by a single vector. This method was introduced to perform approximate nearest neighbor queries in high-dimensional spaces. The memory vector is a compressed representation of the set used to perform a membership test.

In this paper, we model the multiple-queries set with a representative memory vector summarizing all the images. Then, for each database vector we estimate a membership score and use it to rank the images, which amounts to performing a weighted summation of the input vectors. By exploiting this dual interpretation first proposed for generalized max-pooling [13], we extend our method to other matching methods like the selective match kernel [20].

The paper is organized as follows. After a review of related works in Section 2, Section 3 presents our strategy and explains how we use it in the context of multiple-queries. Section 4 presents our experiments. Our results demonstrate the interest of our work for several image representations. Importantly, our method is especially effective with features extracted from a convolution network, outperforming the state-of-the-art of multiple-queries with short vectors.

2. RELATED WORK

Image descriptors. Our method uses on input off-the-shelf vector image representations, such as Bag-of-visual-words (BOW) [18], VLAD [11] or improved Fisher vectors [14].

[1]This work was achieved in the context of the FIRE-ID ANR Project (ANR-12-CORD-016).

ICMR'15, June 23–26, 2015, Shanghai, China.

http://dx.doi.org/10.1145/2671188.2749306 .

We are especially interested in using state-of-the-art short representations, for instance based on democratic aggregation [12]. We also consider descriptors based on Convolutional Neural Networks (CNN), which were recently shown [3] to be very successful on large scale image retrieval tasks, even when reducing the image description to a few hundreds dimensions. These vector representations can be further reduced with binarization [4] or product quantization [10].

Strategies for Multiple-queries. The last few years have witnessed massive progress on the task of finding images depicted a given object in a large database, yet this task still remains challenging when only one image is provided to the system without any annotation. However, in many applications like brand or logo detection, or on output of a text-based search system, several representative images of the objects may be available. This scenario is similar to the instance search task of Trecvid [19] evaluation campaigns.

Multiple image gives a better description of an object to handle problems such as view-point changes, viewing conditions, occlusions, background clutter, shape variation, etc. Arandjelovic & al. [1] presents this problem as well as several possible strategies to exploit these images: average query, SVM on all queries, maximum of multiple queries, average of multiple queries, and exemplar SVM on each query. Their study concludes while averaging the scores, or equivalently averaging the vector representations of the query set, is competitive while reducing the complexity because only one query is submitted. Other works in this setting have been proposed. Chen & al. [6] enrich queries with multiple views in a boosted framework. Fernando & al. [7] use pattern mining to build object specific mid-level image representation. These two strategies aim at explicitly model the query object and are relatively costly. Wu & al. [21] reduces background influence by selecting region of interest appearing in multiple queries.

GMP and memory vectors. Memory vectors [8] build a vector $\mathbf{m}(\mathcal{X})$ summarizing a set $\mathcal{X} = \{\mathbf{x_1}, ..., \mathbf{x_n}\}$ of d-dimensional vectors. This vector optimizes the *membership test*: given a query vector $\mathbf{y}$, it computes the dot product $y^\top \mathbf{m}(\mathcal{X})$ to evaluate whether $\mathbf{y} \in \mathcal{X}$, or close enough to at least one vector in $\mathcal{X}$. The vector are assumed ℓ_2-normalized and uniformly distributed on the unit-hypersphere. In practice, these assumptions do not have necessarily to hold for the method to work well. The test is formalized as the following detection problem:

- Hypothesis H1: $\mathbf{y}$ is not related to $\mathcal{X}$.
- Hypothesis H2: $\mathbf{y}$ is related to $\mathcal{X}$. For example, $\mathbf{y}$ is related to $\mathbf{x_j}$ and we write $\mathbf{y} = \alpha\mathbf{x_j} + \beta\mathbf{z}$, where $\mathbf{z}$ is a random vector orthogonal to $\mathbf{x_j}$ and $||\mathbf{z}|| = 1$. Thus, we have $\alpha^2 + \beta^2 = 1$, because $||\mathbf{y}|| = 1$.

We refer to the dot product $\mathbf{m}^\top\mathbf{y}$ as the membership score. The quality of this score obviously depends on how the vector $\mathbf{m}$ is constructed, for instance simply with the sum vector:

$$m(\mathcal{X}) = \sum_{x \in \mathcal{X}} \mathbf{x}. \quad (1)$$

In that case [8], for a large dimensionality d, the membership score the following distributions:

$$\text{H1: } \langle \mathbf{m}|\mathbf{y} \rangle \sim \mathcal{N}(0, n/d) \quad (2)$$

$$\text{H2: } \langle \mathbf{m}|\mathbf{y} \rangle \sim \mathcal{N}(\alpha, (n-1)/d) \quad (3)$$

However, it is possible to obtain a better hypothesis test. This is achieved by imposing that for any of the vector $\mathbf{x_i}$ in $\mathcal{X}$, $\mathbf{x_i}^\top\mathbf{m}(\mathcal{X}) = 1$, or simply $\mathbf{X}^\top\mathbf{m} = \mathbf{1_n}$ in matrix form, where $\mathbf{1_n}$ is a n-dimensional vector with all values set to 1. This constraint ensures that the interference with remaining vectors of $\mathcal{X}$ are eliminated when the exact same vector is submitted to the system. There are many (a subspace) solutions $\mathbf{m}(\mathcal{X})$ satisfying these constraints.

However, the false positive probability directly depends on the norm of the memory vector and therefore a good way to single a solution is to minimize its norm. In other terms, we want to minimize $||\mathbf{m}||^2$ under the constraint $\mathbf{X}^\top\mathbf{m} = \mathbf{1_n}$. The solution is given by the Moore-Penrose pseudo-inverse:

$$\mathbf{m} = (\mathbf{X}^+)^\top \mathbf{1_n} \quad (4)$$

Since $n < d$, $\mathbf{m} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{1_n}$. We note that if no solution exists, $\mathbf{m}^*$ minimizes $||\mathbf{X}^\top\mathbf{m} - \mathbf{1_n}||$. For large d, this construction leads to the following distributions:

$$\text{H1: } \mathbf{m}^\top Y \sim \mathcal{N}(0, ||\mathbf{m}||^2/d) \quad (5)$$

$$\text{H2: } \mathbf{m}^\top Y \sim \mathcal{N}(\alpha, \beta^2||\mathbf{m}||^2/d) \quad (6)$$

Compared to Equation 1, the improvement comes from the decrease of the variance of the scores under H2 at the cost of increasing the variance under H1. This increase under H1 is small for small n/d, meaning that using this optimized construction is comparatively better for short vectors. For a very large dimensionality, the construction of Equation 4 is asymptotically equivalent to the sum construction.

3. MULTIPLE QUERIES: OUR PROPOSAL

3.1 Memory vectors for multiple queries

Our main idea is to inverse the formulation proposed of memory vectors by summarizing the multiple representations of the object by a single vector. The memory vectors were proposed for test membership and approximate search [8], with one query being tested against a set of vectors based on a membership test. In contrast, we aggregate the queries instead of aggregating the database vectors. The query memory vector is compared to all the database vectors and the membership score is used to order them. Note, if the vector is constructed as the average of the input vectors as in Equation 1, this strategy is equivalent to the well-performing method of averaging the vectors [1]. In contrast, we optimize it by using the construction of Equation 4.

Assumptions. All the image descriptors are ℓ_2-normalized, therefore satisfying the assumption mentioned in Section 2. Additionally, the number n of vectors in the query set is significantly smaller than d. Indeed, there is typically no more than 10 images per query. More precisely, $n = 5$ images per query for Oxford5k and Paris6k used as a multiple-query, see Section 4.1. As a result, n/d is small and the membership test is likely to be effective, which fits well our objective of using short representations ranging from d=128 to 512.

Discussion. With the pseudo-inverse construction, the common parts of the combined descriptors are down-weighted and the norm of the memory vector $||\mathbf{m}||^2$ is minimized. This method has a similar effect as the democratization [12] or the GMP [13] and equalizes the influence of frequent and rare features. The main difference is that this effect is obtained across images, while previous strategies were proposed to address this problem within a given query image.

w_i					
	Gram matrix CNN 512				
0.31	1.00	0.06	0.58	0.02	0.40
0.89	0.06	1.00	0.07	0.08	-0.03
0.55	0.58	0.07	1.00	-0.01	0.29
0.93	0.02	0.08	-0.01	1.00	-0.01
0.75	0.40	-0.03	0.29	-0.01	1.00
	triD 512				
0.68	1.00	0.01	0.43	-0.04	0.28
0.69	0.01	1.00	0.19	0.21	0.06
0.42	0.43	0.19	1.00	-0.04	0.31
0.88	-0.04	0.21	-0.04	1.00	0.03
0.61	0.28	0.06	0.31	0.03	1.00

Figure 1: Visualization of five query images from Oxford5k, and the weights and the Gram matrix for two different types of descriptors.

3.2 Retrieval pipeline

Image representations are computed with VLAD and CNN and other recent single-vector image representations. Since we are mostly interested to improve the initial ranking, we do not use any spatial verification or query expansion techniques, unlike some mentioned previous work [7].

Multi-VLAD – Vlad3c. Features are first extracted using the Hessian-Affine detector and described by RootSIFT [2], and finally rotated with PCA. VLAD are computed using 16, 64, and 256 clusters and compact representations are built by concatenating these three descriptors, power normalization, centering, and PCA to obtain 128, 256, and 512 dimensional "Vlad-3c" descriptors respectively [9, 17].

triD. We use off-the-shelf the descriptors of the recent triangulation embedding and democratization [12].

CNN features. We extract CNN descriptors pre-trained on the Imagenet dataset with the Fast CNN architecture [5]. We follow the recommendation of Babenko *et al.* [3] and use the output of the 6th layer before applying the ReLU. The descriptors are post-processed as Vlad3c and tridD: power normalized, centered, and PCA is applied.

3.3 Dual representation

To evaluate how much an image contributes to the produced memory vector, we employ the dual interpretation proposed by Murray *et al.* [13], which shows that the memory vector of the set $\mathcal{X}$ can be written as a weighted sum:

$$\mathbf{m}(\mathcal{X}) = \sum_{x_i \in \mathcal{X}} w_i \mathbf{x_i}, \tag{7}$$

where w_i is the weight for a given vector of the query set. Denoting $\mathbf{w} = [w_1, \ldots, w_n]^\top$, the weight vector obtained is

$$\mathbf{w} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{1_n}. \tag{8}$$

This formulation computes weights that can be interpreted as quantitative measures of the rarity of the different query images (smaller is more frequent). We can therefore visualize the contribution of each image descriptors to the query object fused descriptor.

Figure 1 gives the weights used to build the memory vectors. Five query images of the same object are shown, as well as the Gram matrix ($\mathbf{X}^\top \mathbf{X}$) and the resulting weights for two different descriptors: CNN and tridD reduced to 512 dimensions. It is interesting to note that the first, third and fifth images are visually similar and have high similarity scores in the Gram matrix. The second and fourth images are relatively different from the other queries and have larger contributions (highest weights) in the fused descriptor.

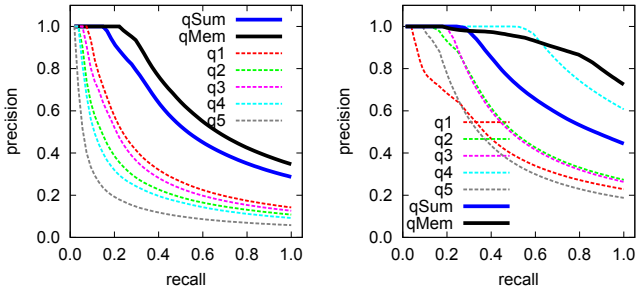

Figure 2: Precision recall curves for two different objects from Oxford5k with five query images, the sum, and the memory vector.

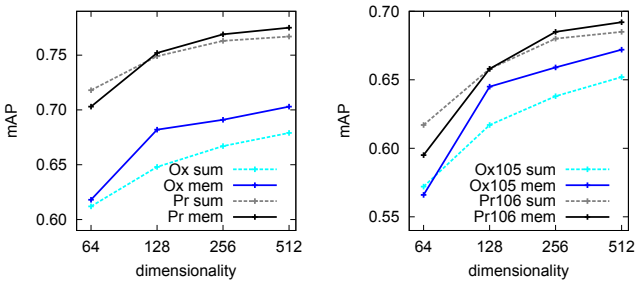

Figure 3: Comparison between sum and memory vectors for the CNN descriptors reduced to [64,128,256,512] dimensions on several datasets.

4. EXPERIMENTS

4.1 Datasets

Oxford5k Buildings [15] is a dataset that consists of 5,062 high-resolution images. There are 55 query images, 5 for each of the 11 chosen Oxford buildings. For each query image, a bounding box on the region of interest is available. The dataset Oxford5k can be further combined with an extra 100k images from Flickr forming Oxford105k dataset.

Paris6k [16] contains 6,412 high-resolution images. The dataset has also 55 query images, with bounding boxes, from 11 landmarks in Paris. This dataset is further combined with the extra 100k images in Paris106k.

To perform multiple queries retrieval, the 5 query image descriptors of a specific landmark are combined into a single representation. These representations are used as query for each specific landmark. Performances are measured in terms of mean average precision (mAP).

Table 1: Averaging vs our method for standard VLAD with 256 clusters, tridD, and CNN on Oxford5k and Oxford105k.

	Oxford5k		Oxford105k	
Representations	Avg [1]	Ours	Avg [1]	Ours
Vlad 256c	0.721	**0.727**	0.685	**0.691**
tridD full	**0.792**	**0.792**	0.768	**0.769**
CNN full	0.603	**0.619**	0.581	**0.589**

Table 2: Averaging vs our method for various reduced representations on Oxford5k and Oxford105k.

	Oxford5k		Oxford105k	
Representations	Avg [1]	Ours	Avg [1]	Ours
Vlad-3c 128	0.557	**0.562**	0.485	**0.496**
Vlad-3c 256	0.620	**0.628**	0.545	**0.556**
Vlad-3c 512	**0.695**	**0.695**	**0.630**	0.627
tridD 128	0.653	**0.660**	0.627	**0.630**
tridD 256	0.710	**0.724**	0.685	**0.696**
tridD 512	0.750	**0.756**	0.725	**0.731**
CNN 128	0.648	**0.682**	0.617	**0.645**
CNN 256	0.667	**0.691**	0.638	**0.659**
CNN 512	0.679	**0.703**	0.652	**0.672**

4.2 Results

Precision-recall. Figure 2 shows precision recall curves for two query objects from Oxford5k. Each query image is evaluated, as well as the fused vector either with sum (equivalent to the averaging [1]) and the optimized memory vector of Equation 4. Observe the benefit of combining multiple queries with our construction compared to sum.

Full descriptors. Table 1 compares the mAP scores for the full representations on Oxford datasets, before any dimensionality reduction. We observe an increase of performance with our strategy for all representations (Vlad, tridD, CNN). It is comparatively smaller for tridD. This can be explained by the fact that the democratization method used in this method already reduces the contribution of frequent features within the image. The CNN features receive the largest improvement, which is appealing because they are the best performing descriptors.

Short vectors. Table 2 presents the results for the reduced representations with PCA to 128, 256, and 512 dimensions. We observe a larger improvement in this setup than for the full representations, especially with CNN descriptors. This is expected because high-dimensional descriptors have more chances to be orthogonal to one to another, in which case sum and optimized memory vectors are identical.

Figure 3 displays the MAP scores against the dimensionality of the CNN descriptors. For the most compact 64-dimensional descriptor, the ratio n/d is not small enough to maintain a low false positive probability and therefore the strategy is not effective. Table 3 shows some supplementary results on Paris datasets, which concur with the ones observed on Oxford buildings.

Finally, we note that centering then applying PCA on CNN descriptors offers a large performance increase in terms of MAP on our various retrieval datasets. For example, reducing the dimensionality to 512 gives an increase of 7% to 9% MAP. Furthermore, we experimentally observed that applying PCA offers better results than PCA with whitening.

Table 3: Experiments on Paris6k and Paris106k.

	Paris6k		Paris106k	
Representations	Avg [1]	Ours	Avg [1]	Ours
Vlad 256	0.654	**0.661**	0.521	**0.529**
Vlad-3c 128	0.565	**0.569**	**0.544**	0.543
Vlad-3c 256	0.598	**0.607**	0.574	**0.577**
Vlad-3c 512	0.629	**0.640**	0.606	**0.615**
CNN full	**0.689**	**0.689**	**0.608**	0.605
CNN 128	0.749	**0.752**	**0.658**	**0.658**
CNN 256	0.763	**0.769**	0.680	**0.685**
CNN 512	0.767	**0.775**	0.685	**0.692**

5. CONCLUSION

We have shown that it is possible to improve the fusion of multiple query descriptors. Our method is based on the recent concept of memory vectors, which summarizes a set of vectors by a single representative vector. The proposed retrieval system offers a consistent increase in performance compared to the average query on several description methods and datasets.

6. REFERENCES

[1] R. Arandjelovic and A. Zisserman. Multiple queries for large scale specific object retrieval. In *BMVC*, 2012.

[2] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, June 2012.

[3] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. *ECCV*, 2014.

[4] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC*, pages 380–388, May 2002.

[5] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *BMVC*, 2014.

[6] Y. Chen, X. Li, A. Dick, and A. van den Hengel. Boosting object retrieval with group queries. *SP Letters*, 2012.

[7] B. Fernando and T. Tuytelaars. Mining multiple queries for image retrieval: On-the-fly learning of an object-specific mid-level representation. In *ICCV*, 2013.

[8] A. Iscen, T. Furon, V. Gripon, M. G. Rabbat, and H. Jégou. Memory vectors for similarity search in high-dimensional spaces. *CoRR*, 2014.

[9] H. Jégou and O. Chum. Negative evidences and co-occurences in image retrieval: The benefit of pca and whitening. In *ECCV*. 2012.

[10] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *Trans. PAMI*, 2011.

[11] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *Trans. PAMI*, 2012.

[12] H. Jégou, A. Zisserman, et al. Triangulation embedding and democratic aggregation for image search. In *CVPR*, 2014.

[13] N. Murray and F. Perronnin. Generalized max pooling. *CVPR*, 2014.

[14] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*. 2010.

[15] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[16] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.

[17] B. Safadi and G. Quénot. Descriptor optimization for multimedia indexing and retrieval. In *CBMI*, 2013.

[18] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[19] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVid. In *MIR*, 2006.

[20] G. Tolias, Y. Avrithis, and H. Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *ICCV*, 2013.

[21] X. Wu and K. Kashino. Interest point selection by topology coherence for multi-query image retrieval. *MTAP*, 2014.