To aggregate or not to aggregate: selective match kernels for image search

Giorgos Tolias, Yannis Avrithis, Hervé Jégou

To aggregate or not to aggregate: Selective match kernels for image search

Giorgos Tolias
INRIA Rennes, NTUA

Yannis Avrithis
NTUA

Hervé Jégou
INRIA Rennes

Abstract

This paper considers a family of metrics to compare images based on their local descriptors. It encompasses the VLAD descriptor and matching techniques such as Hamming Embedding. Making the bridge between these approaches leads us to propose a match kernel that takes the best of existing techniques by combining an aggregation procedure with a selective match kernel. Finally, the representation underpinning this kernel is approximated, providing a large scale image search both precise and scalable, as shown by our experiments on several benchmarks.

1. Introduction

This paper is interested in improving visual recognition of objects, locations and scenes. The best existing approaches rely on local descriptors [14, 15]. Most of them inherit from the seminal Bag-of-Words (BOW) representation [25, 7]. It employs a visual vocabulary to quantize a set of local descriptors and to produce a single vector that represents the image. This offers several desirable properties. For image classification [7], it is compatible with powerful machine learning techniques such as support vectors machines. In this case, it is usually employed with relatively small visual vocabularies. In a query by content scenario [25], which is the focus of our paper, large vocabularies make the search efficient [17, 22, 16], thanks to inverted file structures [24] that exploit the sparsity of the representation. The methods relying on these ingredients are typically able to search in millions of images in a few seconds or less.

Several researchers have built upon this approach to design better systems. In particular, the search is advantageously refined by re-ranking approaches, which operate on an initial short-list. This is done by exploiting additional geometrical information [22, 18, 26] or applying query expansion techniques [6, 27]. This paper focuses on improving the quality of the initial result set. Re-ranking approaches are complementary stages that are subsequently applied.

This work was done in the context of the Project Fire-ID, supported by the Agence Nationale de la Recherche (ANR-12-CORD-0016).

Another important improvement is obtained by reducing the quantization noise. This is done by multiple assignment [23, 12], or by exploiting a more precise representation of the individual local descriptors, such as binary codes in the so-called Hamming Embedding (HE) method [12], or by integrating some information about the neighborhood of the descriptor [31]. All these approaches implicitly rely on approximate pair-wise matching of the query descriptors with those of the database images.

In a concurrent effort to scale to even larger databases, recent encoding techniques such as Fisher kernels [19, 21], local linear coding [30] or the "vector or locally aggregated descriptors" (VLAD) [13], depart from the initial BOW framework by introducing alternative encoding schemes. By compressing the resulting vector representation [13, 20], the local descriptors are not considered individually. Images can be represented by a small number of bytes, similar to coded global descriptors [29], but with the advantage of preserving some key properties inherited from local descriptors, such as rotation and scale invariance.

Our paper introduces a framework to bridge the gap between the "matching-based" approaches, such as HE, and the recent aggregated representations, in particular VLAD. For this purpose, we introduce in Section 2 a class of match kernels that includes both matching-based and aggregated methods for unsupervised image search.

We then discuss and analyze in Section 3 two key differences between matching-based and aggregated approaches. First, we consider the *selectivity* of the matching function, *i.e.*, the property that a correspondence established between two patches contributes to the image-level similarity only if the confidence is high enough. It is explicitly exploited in matching-based approaches only.

Second, the aggregation (or pooling) operator used in BoW, VLAD or in the Fisher vector, is not considered in pure matching approaches such as HE. We show that it is worth doing it even in matching-based approaches, and discuss its relationship with other methods (*e.g.*, [11, 21]) introduced to handle the non-*iid* statistical behavior of local descriptors, also called the burstiness phenomenon [11].

This leads us to conclude that none of the existing schemes combines the best ingredients required to achieve

the best possible retrieval quality. As a result, we introduce a new method that exploits the best of both worlds to produce a strong image representation and its corresponding kernel between images. It combines an aggregation scheme with a selective kernel. This vector representation is advantageously compressed to drastically reduce the memory requirements, while also improving the search efficiency.

Section 4 shows that our method significantly outperforms the state of the art in a comparable setup, *i.e.*, when comparing the quality of the initial result set produced when searching a large collection.

2. A framework for match kernels

This section first describes the class of match kernels that we will analyze in this paper. This framework encompasses several popular techniques published in the literature. In the following, we denote the cardinality of a set $\mathcal{A}$ by $\#\mathcal{A}$.

cardinality: 基数，一个集合的元素数量

Let us assume that an image is described by a set $\mathcal{X} = \{x_1, \ldots, x_n\}$ of n d-dimensional local descriptors. The descriptors are quantized by a k-means quantizer

$$\begin{aligned} q : \mathbb{R}^d &\to \mathcal{C} \subset \mathbb{R}^d \\ x &\mapsto q(x) \end{aligned} \tag{1}$$

where $\mathcal{C} = \{c_1, \ldots, c_k\}$ is a codebook comprising $k = \#\mathcal{C}$ vectors, which are referred to as visual words. We denote by $\mathcal{X}_c = \{x \in \mathcal{X} : q(x) = c\}$ the subset of descriptors in $\mathcal{X}$ that are assigned to a particular visual word c. In order to compare two image representations $\mathcal{X}$ and $\mathcal{Y}$, we consider a family of set similarity functions K of the general form

$$\mathrm{K}(\mathcal{X}, \mathcal{Y}) = \gamma(\mathcal{X})\,\gamma(\mathcal{Y}) \sum_{c \in \mathcal{C}} w_c\, \mathrm{M}\left(\mathcal{X}_c, \mathcal{Y}_c\right), \tag{2}$$

where function M is defined between two sets of descriptors $\mathcal{X}_c, \mathcal{Y}_c$ assigned to the same visual word. Depending on the definition of M, the set similarity function K is or is not a positive-definite kernel.

The scalar w_c is a constant that depends on visual word c, for instance it integrates the inverse document frequency (IDF) weighting term. The normalization factor $\gamma(.)$ is typically computed as

$$\gamma(\mathcal{X}) = \left(\sum_{c \in \mathcal{C}} w_c\, \mathrm{M}\left(\mathcal{X}_c, \mathcal{X}_c\right) \right)^{-1/2}, \tag{3}$$

such that the self-similarity of an image is $\mathrm{K}(\mathcal{X}, \mathcal{X}) = 1$. Several popular methods of the literature can be described by the framework of Equation (2).

Bag-of-words. The BOW representation [25, 7] represents each local descriptor x solely by its visual word. As noticed in [3, 12], bag-of-words with cosine similarity can be expressed in terms of Equation (2), by defining

$$\mathrm{M}(\mathcal{X}_c, \mathcal{Y}_c) = \#\mathcal{X}_c \times \#\mathcal{Y}_c = \sum_{x \in \mathcal{X}_c} \sum_{y \in \mathcal{Y}_c} 1, \tag{4}$$

Other comparison metrics are also possible in this framework. For instance, the histogram intersection would use $\min(\#X_c, \#Y_c)$ instead. In the case of max-pooling [4], $\mathrm{M}(\mathcal{X}_c, \mathcal{Y}_c)$ would be equal to 1 if both $\mathcal{X}_c, \mathcal{Y}_c$ are non-empty, and zero otherwise.

Hamming Embedding (HE) [10, 12] is a matching model that extends BOW by representing each local descriptor x with both its quantized value $q(x)$ and a binary code b_x of B bits. It computes the scores between all pairs of descriptors assigned to the same visual word, as

$$\mathrm{M}\left(\mathcal{X}_c, \mathcal{Y}_c\right) = \sum_{x \in \mathcal{X}_c} \sum_{y \in \mathcal{Y}_c} w\left(h\left(b_x, b_y\right)\right), \tag{5}$$

where h is the Hamming distance and w is a weighting function that associates a weight to each of the $B + 1$ possible distance values. This function was first defined as binary [10], such that $w(h) = 1$ if $h \leq \tau$, and 0 otherwise. A smoother weighting scheme is a better choice [11, 12], such as the (thresholded) Gaussian function [11]

$$w(h) = \begin{cases} e^{-h^2/\sigma^2}, & h \leq \tau \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

We assume that binary codes lie in the Hamming space $\{-1, +1\}^B$ and use the Hamming inner product

$$\langle a, b \rangle_h = \frac{a^\top b}{B} = \hat{a}^\top \hat{b} \in [-1, 1] \tag{7}$$

instead of the Hamming distance presented in the original HE paper [10]. Here $\hat{a}$ denotes the ℓ_2-normalized counterpart of vector a. The two choices are equivalent since $2h(a, b) = B(1 - \langle a, b \rangle_h)$.

VLAD [13] aggregates the descriptors associated with a given visual word to produce a $d \times k$ vector representation. This vector is constructed as the concatenation $\boldsymbol{\mathcal{V}}(\mathcal{X}) \propto [V(\mathcal{X}_{c_1}), \ldots, V(\mathcal{X}_{c_k})]$ of d-dimensional vectors, where

$$V(\mathcal{X}_c) = \sum_{x \in \mathcal{X}_c} r(x), \tag{8}$$

and

$$r(x) = x - q(x) \tag{9}$$

is the residual vector of x. Since the similarity of two VLADs is measured by the dot product, it is easy to show that VLAD corresponds to a match kernel of the form proposed in Equation (2):

$$\boldsymbol{\mathcal{V}}(\mathcal{X})^\top \boldsymbol{\mathcal{V}}(\mathcal{Y}) = \gamma(\mathcal{X})\,\gamma(\mathcal{Y}) \sum_{c \in \mathcal{C}} V(\mathcal{X}_c)^\top V(\mathcal{Y}_c), \tag{10}$$

where Equation (3) determines the normalization factors. Then it appears that

$$\mathrm{M}(\mathcal{X}_c, \mathcal{Y}_c) = V(\mathcal{X}_c)^\top V(\mathcal{Y}_c) \tag{11}$$

$$= \sum_{x \in \mathcal{X}_c} \sum_{y \in \mathcal{Y}_c} r(x)^\top r(y). \tag{12}$$

The power-law normalization proposed for Fisher vectors [21] is also integrated in this framework by modifying the definition of V, however it cannot be expanded as Equation (12). Its effect is similar to burstiness handling in [11].

Burstiness [11] refers to the phenomenon whereby a visual word appears more times in an image than what a statistically independent model would predict. It tends to corrupt the visual similarity measure. Once individual contributions are aggregated per cell as in the HE model of Equation (5), one solution is to down-weight highly populated cells.

For instance, one of the most effective burst weighting models of [11] assumes that the outer sum in Equation (5) refers to query descriptors $\mathcal{X}_c$ in the cell and down-weights the inner sum of the descriptors $\mathcal{Y}_c$ of a given database image by $(\#\mathcal{Y}_c(x))^{-1/2}$, where

$$\mathcal{Y}_c(x) = \{y \in \mathcal{Y}_c : w(h(b_x, b_y)) \neq 0\} \tag{13}$$

is the subset of descriptors in $\mathcal{Y}_c$ that match with x. A more radical option is $(\#\mathcal{Y}_c(x))^{-1}$, effectively removing multiple matches within cells, similarly to max-pooling [4].

3. Investigating selectivity and aggregation

The three match kernels presented above share some similarities, in particular the fact that the set of descriptors is partitioned into cells and that only vectors lying in the same cell contribute to the overall similarity. VLAD and HE have key characteristics that we discuss in this section. This leads us to explore new possible kernels that are thoroughly evaluated in Section 4. We first develop a common model assuming that full descriptors are available in both images, *i.e.*, uncompressed, and then consider the case of binarized representations.

3.1. Towards a common model

The non-aggregated kernels individually match all the elements occurring in the same Voronoi cell. They are defined as the set of kernels M of the form

$$\mathrm{M_N}(\mathcal{X}_c, \mathcal{Y}_c) = \sum_{x \in \mathcal{X}_c} \sum_{y \in \mathcal{Y}_c} \sigma\left(\phi(x)^\top \phi(y)\right). \tag{14}$$

This equation encompasses all the variants discussed so far, excluding the burstiness post-processing considered in Equation (13). Here ϕ is an arbitrary vector representation function, possibly non-linear or including normalization,

Model	$\mathrm{M}(\mathcal{X}_c, \mathcal{Y}_c)$	$\phi(x)$	$\sigma(u)$	$\psi(z)$	$\Phi(\mathcal{X}_c)$
BOW (4)	$\mathrm{M_N}$ or $\mathrm{M_A}$	1	u	z	$\#\mathcal{X}_c$
HE (5)	$\mathrm{M_N}$	$\hat{b}_x$	$w\left(\frac{B}{2}(1-u)\right)$	—	—
VLAD (8)	$\mathrm{M_N}$ or $\mathrm{M_A}$	$r(x)$	u	z	$V(\mathcal{X}_c)$
SMK (20)	$\mathrm{M_N}$	$\hat{r}(x)$	$\sigma_\alpha(u)$	—	—
ASMK (22)	$\mathrm{M_A}$	$r(x)$	$\sigma_\alpha(u)$	$\hat{z}$	$\hat{V}(\mathcal{X}_c)$
SMK⋆ (23)	$\mathrm{M_N}$	$\hat{b}_x$	$\sigma_\alpha(u)$	—	—
ASMK⋆ (24)	$\mathrm{M_A}$	$r(x)$	$\sigma_\alpha(u)$	$\hat{b}(z)$	$\hat{b}(V(\mathcal{X}_c))$

Table 1. Existing and new solutions for the match kernel M. They are classified as non-aggregated $\mathrm{M_N}$ (14) and aggregated kernels $\mathrm{M_A}$ (15), or possibly both. $\phi(x)$: scalar or vector representation of descriptor x. $\sigma(u)$: scalar selectivity of u, where u is assumed normalized in $[-1, 1]$. $\psi(z)$: representation of aggregated descriptor z per cell. $\Phi(\mathcal{X}_c)$ (17): equivalent representation of descriptor set $\mathcal{X}_c$ per cell. Given any vector x, we denote by $\hat{x} = x/\|x\|$ its ℓ_2-normalized counterpart.

and $\sigma : \mathbb{R} \to \mathbb{R}$ is a scalar selectivity function. Options for these functions are presented in Table 1 and discussed later in this section.

The aggregated kernels, in contrast, are written as

$$\mathrm{M_A}(\mathcal{X}_c, \mathcal{Y}_c) = \sigma\left\{\psi\left(\sum_{x \in \mathcal{X}_c} \phi(x)\right)^\top \psi\left(\sum_{y \in \mathcal{Y}_c} \phi(y)\right)\right\} \tag{15}$$

$$= \sigma\left(\Phi(\mathcal{X}_c)^\top \Phi(\mathcal{Y}_c)\right), \tag{16}$$

where ψ is another vector representation function, again possibly non-linear or including normalization. $\Phi(\mathcal{X}_c)$ is the aggregated vector representation of a set $\mathcal{X}_c$ of descriptors in a cell, such that $\Phi(\emptyset) = \mathbf{0}$ and

$$\Phi(\mathcal{X}_c) = \psi\left(\sum_{x \in \mathcal{X}_c} \phi(x)\right). \tag{17}$$

This formulation suggests other potential strategies. In contrast to Equation (14), there is at most a single match between aggregated representations $\Phi(\mathcal{X}_c)$ and $\Phi(\mathcal{Y}_c)$, and selectivity σ is applied after aggregation.

Of the variants discussed so far, BOW and VLAD both fit into Equation (15), with σ simply being identity. This is not the case for HE matching. Note that the aggregation, *i.e.*, computing $\Phi(\mathcal{X}_c)$, is an off-line operation.

3.2. Non-aggregated matching — SMK

We introduce a *selective match kernel* (SMK) in this subsection. It is motivated by the observation that VLAD employs a linear weighting scheme in Equation (12) for the contribution of individual matching pairs (x, y) to M, while HE applies an non-linear weighting function σ to the similarity $\phi(x)^\top \phi(y)$ between a pair of descriptor x and y.

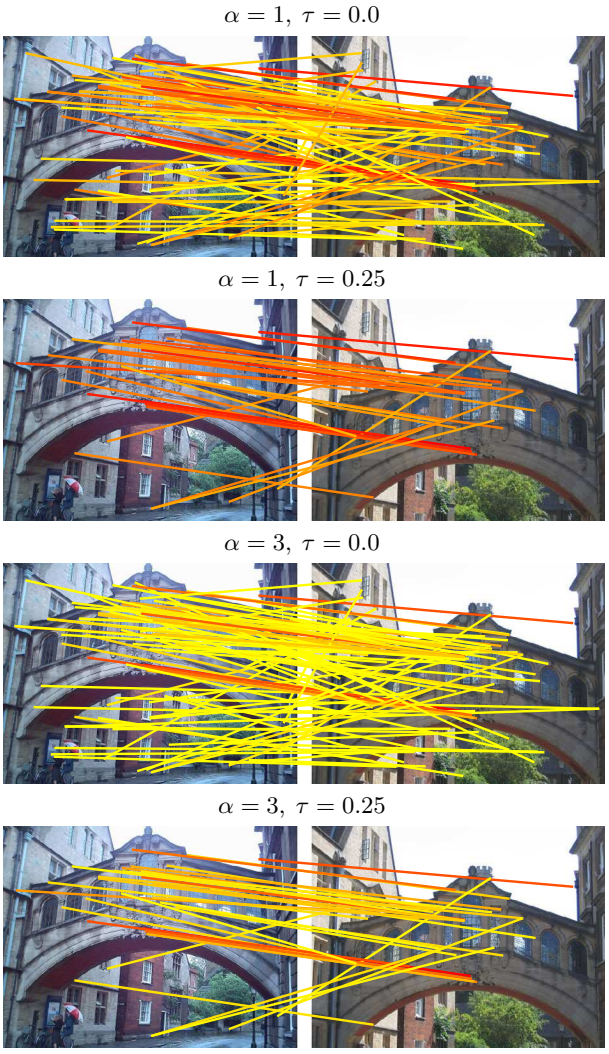

Figure 1. Matching features with descriptors assigned to the same visual word and similarity above the threshold. Examples for different values of α and τ. Color denotes descriptor similarity defined by $\sigma_\alpha(\hat{r}(x)^\top \hat{r}(y))$, with yellow corresponding to 0 and red to the maximum similarity per image pair.

Choice of selectivity function σ. Without loss of generality, we consider a thresholded polynomial selectivity function $\sigma_\alpha : \mathbb{R} \to \mathbb{R}^+$ of the form

$$\sigma_\alpha(u) = \begin{cases} \text{sign}(u)|u|^\alpha & \text{if } u > \tau \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

and typically set $\alpha = 3$. In all our experiments we have used $\tau \geq 0$. It plays the same role as the weighting function w in Equation (5), applied to similarities instead of distances.

Figure 1 shows the effect of this function σ_α when matching features between two images, for different values of the exponent α and of the threshold τ. The descriptor similarity, now measured by σ_α, is displayed in different colors. A larger α increases the selectivity and drastically down-weights false correspondences. This advantageously replaces hard thresholding as initially proposed in HE [10].

Choice of ϕ. We consider a non-approximate representation of the intermediate vector representation $\phi(x)$ in Equation (14), and adopt a choice similar to VLAD by using the ℓ_2-normalized residual $\hat{r}(x)$, defined as

$$\hat{r}(x) = \frac{x - q(x)}{\|x - q(x)\|}. \tag{19}$$

Our SMK kernel is obtained by setting $\sigma = \sigma_\alpha$ and $\phi = \hat{r}$ in Equation (14), as

$$\text{SMK}(\mathcal{X}_c, \mathcal{Y}_c) = \sum_{x \in \mathcal{X}_c} \sum_{y \in \mathcal{Y}_c} \sigma_\alpha(\hat{r}(x)^\top \hat{r}(y)), \tag{20}$$

It differs from HE in that it uses the normalized residual instead of binary vectors. It also differs from VLAD, considered as a matching function, by the selectivity function σ and because we normalize the residual vector. These differences are summarized in Table 1.

3.3. Aggregated selective match kernel ASMK

SMK weights the contributions of individual matches with a non-linear function. We now propose to apply a selective function *after* aggregating the different vectors per cell. Aggregating the vectors per cell has the advantage of producing a more compact representation.

Our ASMK kernel is constructed as follows. The residual vectors are summed as in VLAD, producing a single representative descriptor per cell. This sum is subsequently ℓ_2-normalized. The ℓ_2-normalization ensures that the similarity in input of σ always lies in the range $[-1, +1]$. It means that

$$\Phi(\mathcal{X}_c) = \hat{V}(\mathcal{X}_c) = V(\mathcal{X}_c)/\|V(\mathcal{X}_c)\| \tag{21}$$

describes all the descriptors assigned to the cell c. The selectivity function σ_α is applied after aggregation and normalization, therefore the matching kernel M_A becomes

$$\text{ASMK}(\mathcal{X}_c, \mathcal{Y}_c) = \sigma_\alpha\left(\hat{V}(\mathcal{X}_c)^\top \hat{V}(\mathcal{Y}_c)\right). \tag{22}$$

The database vectors $\hat{V}(\mathcal{X}_c)$ are computed off-line.

Figure 2 illustrates several examples of features that are aggregated. They commonly correspond to repeated structure and textured regions. Such bursty features appear in most urban images, and their matches usually dominate the image level similarity. ASMK handles this by keeping only one representative instance of all bursty descriptors, which , due to normalization, is equal to the normalized mean residual. Normalization per visual word was recently proposed by a concurrent work [2] with comparatively small vocabularies. The choice of normalizing our vector representation

Figure 2. Examples of features mapped to the same visual word, finally being aggregated. Each visual word is drawn with a different color. Top 25 visual words are drawn, based on the number of features mapped to them.

resembles binary BOW [25] or max pooling [4] which both tackle burstiness by accounting at most one vote per visual word. Aggregating without normalizing still allows bursty features to dominate the total similarity score.

3.4. Binarization — SMK⋆ and ASMK⋆

HE relies on the binary vector b_x instead of residual $r(x) = x - q(x)$. Although the choice of binarization was adopted for the sake of compactness, a question arises: What is the performance of the kernel if the full vector are employed instead? This is what has motivated us to develop the SMK and ASMK match kernels, which rely on full d-dimensional descriptors. However, these kernels are costly in terms of memory. That is why we also develop their binary versions (denoted with an additional *) in this section.

SMK⋆ and ASMK⋆. The approximated version SMK⋆ of SMK is similar to HE, the only difference is the inner product formulation and the choice of the selectivity function σ_α in Equation (18):

$$\text{SMK}^\star(\mathcal{X}_c, \mathcal{Y}_c) = \sum_{x \in \mathcal{X}_c} \sum_{y \in \mathcal{Y}_c} \sigma_\alpha\left(\hat{b}_x^\top \hat{b}_y\right). \quad (23)$$

It is an approximation of the full descriptor model of Equation (20), which uses the binary vector $\hat{b}$ instead of $\hat{r}$.

Similarly, the approximation ASMK⋆ of the aggregated version ASMK is obtained by binarizing $V(\mathcal{X}_c)$ before applying the selectivity function:

$$\text{ASMK}^\star(\mathcal{X}_c, \mathcal{Y}_c) = \sigma_\alpha\left\{\hat{b}\left(\sum_{x \in \mathcal{X}_c} r(x)\right)^\top \hat{b}\left(\sum_{y \in \mathcal{Y}_c} r(y)\right)\right\}, \quad (24)$$

where b is an element-wise binarization function $b(x) = +1$ if $x \geq 0, -1$ otherwise. Note that the residual is here computed with respect to the median as in HE, and not the centroid. Moreover, in SMK⋆ and ASMK⋆ all descriptors are projected using the same projection matrix as in HE.

Remark: In LSH, the Hamming distance gives an estimate of the cosine similarity [5] between original vectors (through $\arccos$ function). The differences with HE are that (i) LSH is based on a set of random projections, whereas HE uses a randomly oriented orthogonal basis; (ii) HE binarizes the vectors according to their projected median values.

4. Experiments

This section describes some implementation details and introduce the datasets and evaluation protocol used in our experiments. We further present experiments for measuring the impact of the kernel parameters, and finally compare our methods against state-of-the-art methods. Most of our results are presented without spatial verification or query expansion (QE) to focus on the quality of the initial ranking, before re-ranking by these complementary methods.

4.1. Implementation and experimental setup

Datasets. We evaluate the proposed methods on 3 publicly available datasets, namely Holidays [12], Oxford Buildings [22] and Paris [23]. Evaluation measure is the mean Average Precision (mAP). Due to the randomness introduced to the binarized methods (SMK⋆ and ASMK⋆) by the random projection matrix, the same as the one used in the original Hamming Embedding, we create 3 independent inverted files and measure the average performance.

Features. We have used the Hessian-Affine detector to extract local features. For Oxford and Paris datasets, we have used the modified Hessian-Affine detector of Perdoch *et al.* [18], which includes the gravity vector assumption and

improves retrieval performance. Most of our experiments use the default detector threshold value. We also consider the use of lower threshold values to derive larger sets of features, and show the corresponding benefit in search quality, at the cost of a memory and computational overhead.

We use SIFT descriptors and apply component-wise square-rooting [1, 8]. This has proven to yield superior performance at no cost. In more details, we follow the approach [8] in which component-wise square rooting is applied and the final vector is ℓ_2-normalized. We also center the SIFT descriptors. Our SIFT descriptor post-processing is the same as the one of Tolias and Jégou [27].

Vocabularies. We have used flat k-means to create our visual vocabularies. These are always trained on an independent dataset, different from the one indexed and used for evaluation each time. Using visual vocabularies trained on the evaluation dataset yields superior performance [22, 1] but is more prone to over-fitting. Vocabularies used for Oxford are trained on Paris, and vice versa, while the ones used for Holidays are trained on an independent set of images downloaded from Flickr. Unless stated otherwise, we use a vocabulary of 65k visual words.

Inverted files. In contrast to VLAD, we apply our methods with relatively large vocabularies aiming at best performance for object retrieval, and use an inverted file structure to exploit the sparsity of the BOW based representation. With SMK and ASMK, each dimension of vectors $\phi(x)$ or $\Phi(\mathcal{X}_c)$ respectively, is uniformly quantized with 8 bits and stored in the inverted file. Correspondingly, a binary vector of 128 dimensions is stored along with SMK* and ASMK*.

Multiple assignment. We further combine our proposed method with multiple assignment (MA), which is applied on query side only [12]. We replicate each descriptor vector and assign each instance to a different visual word. When it is stated that multiple assignment is used in our experiment, 5 nearest visual words are used. Single assignment will be referred to as SA.

Burstiness. The non-aggregated versions of the proposed methods allow multiple matches for a given visual word. Thus, we combine them with the intra-image burstiness normalization [11]. This is done to compare with our aggregated methods which also deal with the burstiness phenomenon. We will refer to burstiness normalization as BURST in the experiments.

Query expansion. We combine our methods with local visual query expansion [27] to futher improve the performance. A brief description follows. Similarly to other visual query expansion methods [6, 1], we apply spatial verification [22] to the 100 top ranked images in order to identify the ones that are truly relevant. Images are considered as verified when they are found to have at least 5 inliers with

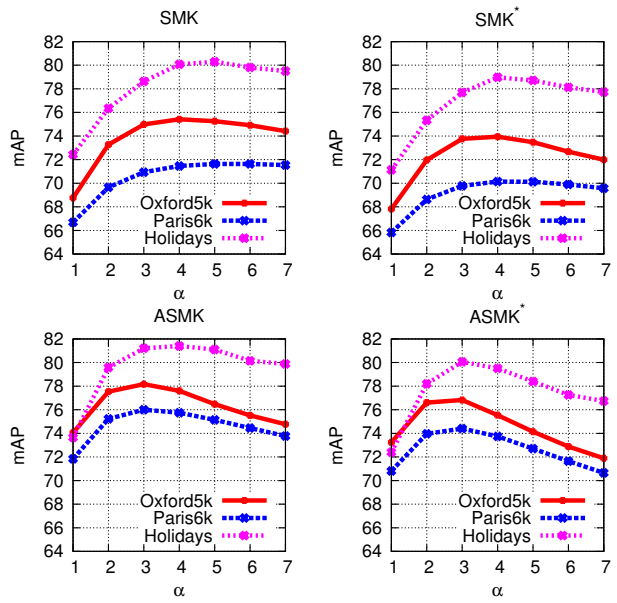

Figure 3. Impact of parameter α for SMK and ASMK (left) and their binarized counterparts (right). In these experiments, $\tau = 0$.

single and 8 with multiple assignment. The estimated geometric transformation estimated for verified images is used to back-project features of database images to the query image. Features projected out of the query region are discarded. We collect visual words of all retained features, sort them based on the number of verified images in which they appear and select the top ranked ones. We select them in a way such that the number of new visual words that are not present in the query image are equal to the number of original visual words of the query image. Descriptors assigned to those visual words are merged with the query features, and aggregation per visual word is applied once more. The new expanded query is of the same nature as the original one and can be issued to the same indexing structure.

Aggregation. For the aggregated methods descriptors of database images are aggregated off-line and then stored in the inverted file. On query time, query descriptors are aggregated in the same way. In the case of multiple assignment, aggregation is similarly applied once the aforementioned replication of descriptors is performed.

Query expansion uses spatial verification which requires the construction of tentative correspondences. In the aggregated scheme, when two aggregated features are matched then correspondences are formed between all original features being aggregated of query and database image.

4.2. Impact of the parameters

Parameter α. Figure 3 shows the impact of the parameter α associated with our selectivity function. It controls the balance between strong and weaker matches. Setting

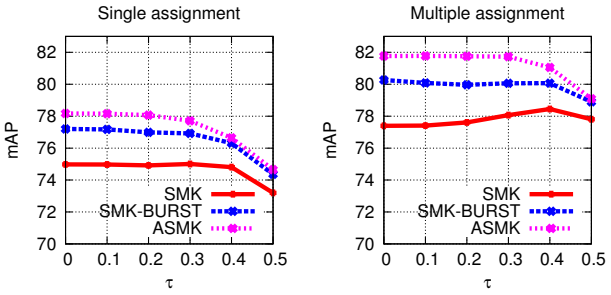

Figure 4. Impact of threshold value τ on Oxford dataset for SMK, SMK with burstiness normalization and ASMK. Results for single (left) and multiple (right) assignment is shown.

k	8k	16k	32k	65k
Oxford	69 %	78 %	85 %	89 %
Paris	68 %	76 %	82 %	86 %
Holidays	55 %	65 %	73 %	78 %

Table 2. Ratio of memory requirements after aggregation (ASMK or ASMK*) to the ones before aggregation (SMK or SMK*), for various vocabulary sizes.

$\alpha = 1$ corresponds to the linear weighting function used by VLAD. The weighting function significantly improves the performance in all cases. In the rest of our experiments, $\alpha = 3$ as a compromise for good performance across all datasets.

Threshold τ. We evaluate the performance on the Oxford dataset for different values of the threshold τ. Figure 4 shows that the performance is stable for small threshold values. In the rest of our experiments we will set the threshold value equal to 0, maintaining best performance but also reducing the number of matches obtained from the inverted file. Remark also that ASMK outperforms SMK combined with burstiness normalization [11].

Vocabulary size k. Approaches which are based on a visual vocabulary are usually too sensitive to its size. We evaluate our proposed methods for different vocabulary sizes and present performance in Figure 5. Our methods being employed with descriptor information and not only visual words, do not appear to be too sensitive to the vocabulary size. ASMK outperforms SMK combined with burstiness normalization. We have computed VLAD with the 8k vocabulary, which achieves 65.5 mAP on Oxford5k with a vector representation of $8192 \cdot 128$ dimensions. SMK and ASMK with single assignment and the 8k vocabulary achieve 74.2 and 78.1 respectively.

We have measured the amount of descriptors being aggregated in each case by the *memory ratio* which is defined as the ratio of the total number of descriptors indexed after aggregation to the ones before aggregation. The memory savings are presented in Table 2. Our aggregated scheme not only improves performance, but also saves memory.

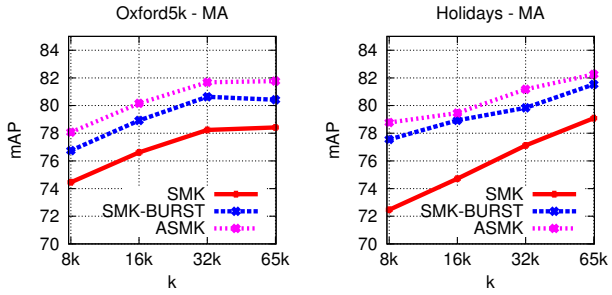

Figure 5. Impact of vocabulary size k measured on Oxford5k and Holidays datasets. Multiple assignment is used.

Dataset	MA	Oxf5k	Oxf105k	Paris6k	Holidays
HE [12]		51.7	-	-	74.5
HE [12]	×	56.1	-	-	77.5
HE-BURST [9]		64.5	-	-	78.0
HE-BURST [9]	×	67.4	-	-	79.6
Fine vocabulary [16]	×	74.2	67.4	74.9	74.9
AHE-BURST [9]		66.6	-	-	79.4
AHE-BURST [9]	×	69.8	-	-	81.9
Rep. structures [28]	×	65.6	-	-	74.9
ASMK*		76.4	69.2	74.4	80.0
ASMK*	×	80.4	**75.0**	77.0	81.0
ASMK		78.1	-	76.0	81.2
ASMK	×	**81.7**	-	**78.2**	**82.2**

Table 4. Performance comparison with state-of-the-art methods ($\alpha = 3$, $\tau = 0$, $k = 65$k), without spatial verification nor QE. Note that both SMK and ASMK rely on full descriptors and do not scale to Oxford105k. Memory used by SMK* (reps., ASMK*) is equal (resp., lower) than in HE. The best ASMK* variant is faster than HE (less features after aggregation).

Larger feature sets. We have conducted experiments using lower detector threshold values than the default one, thus deriving a larger set of features per image. The performance is compared between the two features sets in Table 3, showing that using more features yields superior performance in all cases. The use of the selectivity function allows the use of more features which also includes more false matches, but these are properly down-weighted.

4.3. Comparison to the state of the art

Table 4 summarizes the performance of our methods for single and multiple assignment and compares to state of the art methods which do not apply any spatial re-ranking or query expansion. Only our binarized methods are scalable for Oxford105k. ASMK achieves a better performance than the binarized ASMK* and outperforms all other methods.

We further combine ASMK with the query expansion scheme previously described and achieve mAP equal to **87.9** and **85.4** on Oxford5k and Paris6k respectively. ASMK* with query expansion achieves 85.0 on Oxford105k. These scores are better than the best scores reported with query expansion on these datasets [16].

Dataset	Oxford5k				Paris6k				Holidays			
	Small		Large		Small		Large		Small		Large	
Method	SMK	ASMK	SMK	ASMK	SMK	ASMK	SMK	ASMK	SMK	ASMK	SMK	ASMK
#features	12.5M	11.2M	21.9M	19.2M	15.0M	13.0M	25.1M	21.5M	4.4M	3.5M	16.7M	12.0M
SA	74.9	78.1	78.5	82.0	70.9	76.0	73.2	78.7	78.6	81.2	84.0	**88.0**
MA	77.4	81.7	79.3	**83.8**	71.8	78.2	74.2	**80.5**	79.0	82.2	82.9	86.5

Table 3. Performance evaluation for different feature set sizes, extracted by using different detector threshold values. Small = set with the default threshold, Large = set with lower threshold. Number of features indexed without (SMK) and with (ASMK) aggregation are reported. Performance for single (SA) and multiple (MA) assignment. **These results are without spatial verification and without QE.**

5. Conclusions

This paper draws a framework for well known matching kernels such as BOW, HE and VLAD. We build a common model in which we further incorporate our matching kernels sharing the best properties of HE and VLAD. We exploit the use of a selectivity function and show how aggregation per visual word can deal with burstiness. Finally, our methods combined with a query expansion scheme exhibit superior performance than state of the art methods.

References

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

[2] R. Arandjelović and A. Zisserman. All about VLAD. In *CVPR*, 2013.

[3] L. Bo and C. Sminchisescu. Efficient match kernel between sets of features for visual recognition. In *NIPS*, 2009.

[4] Y. Boureau, F. Bach, Y. Lecun, and J. Ponce. Learning mid-level features for recognition. In *cvpr*, 2010.

[5] M. Charikar. Similarity estimation techniques from rounding algorithms. In *ACM Symposium on Theory of Computing*, 2002.

[6] 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 *ICCV*, Oct. 2007.

[7] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV Workshop Statistical Learning in Computer Vision*, 2004.

[8] M. Jain, R. Benmokhtar, P. Gros, and H. Jégou. Hamming embedding similarity-based image classification. In *ICMR*, Jun. 2012.

[9] M. Jain, H. Jégou, and P. Gros. Asymmetric hamming embedding: Taking the best of our bits for large scale image search. In *ACM Multimedia*, 2011.

[10] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, Oct. 2008.

[11] H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *CVPR*, Jun. 2009.

[12] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *IJCV*, 87(3):316–336, Feb. 2010.

[13] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, Jun. 2010.

[14] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, Nov. 2004.

[15] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Trans. PAMI*, 27(10):1615–1630, 2005.

[16] A. Mikulík, M. Perdoch, O. Chum, and J. Matas. Learning a fine vocabulary. In *ECCV*, Sep. 2010.

[17] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, Jun. 2006.

[18] M. Perdoch, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *CVPR*, Jun. 2009.

[19] F. Perronnin and C. R. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, Jun. 2007.

[20] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier. Large-scale image retrieval with compressed Fisher vectors. In *CVPR*, Jun. 2010.

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

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

[23] 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*, Jun. 2008.

[24] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, Aug. 1988.

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

[26] G. Tolias and Y. Avrithis. Speeded-up, relaxed spatial matching. In *ICCV*, Nov. 2011.

[27] G. Tolias and H. Jégou. Local visual query expansion: Exploiting an image collection to refine local descriptors. Technical Report RR-8325, INRIA, Jul. 2013.

[28] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi. Visual place recognition with repetitive structures. In *CVPR*, 2013.

[29] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large databases for recognition. In *CVPR*, Jun. 2008.

[30] J. Wang, J. Yang, F. L. K. Yu, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[31] Z. Wu, Q. Ke, M. Isard, and J. Sun. Bundling features for large scale partial-duplicate web image search. In *CVPR*, pages 25–32, 2009.