Rotation and translation covariant match kernels for image retrieval

Giorgos Tolias, Andrei Bursuc, Teddy Furon, Hervé Jégou

Rotation and translation covariant match kernels for image retrieval

Giorgos Tolias*, Andrei Bursuc, Teddy Furon, Hervé Jégou

Inria, Rennes

Abstract

Most image encodings achieve orientation invariance by aligning the patches to their dominant orientations and translation invariance by completely ignoring patch position or by max-pooling. Albeit successful, such choices introduce too much invariance because they do not guarantee that the patches are rotated or translated consistently. In this paper, we propose a geometric-aware aggregation strategy, which jointly encodes the local descriptors together with their patch dominant angle or location. The geometric attributes are encoded in a continuous manner by leveraging explicit feature maps. Our technique is compatible with generic match kernel formulation and can be employed along with several popular encoding methods, in particular Bag-of-Words, VLAD and the Fisher vector. The method is further combined with an efficient monomial embedding to provide a codebook-free method aggregating local descriptors into a single vector representation. Invariance is achieved by efficient similarity estimation of multiple rotations or translations, offered by a simple trigonometric polynomial. This strategy is effective for image search, as shown by experiments performed on standard benchmarks for image and particular object retrieval, namely Holidays and Oxford buildings.

Keywords: image retrieval, geometry aware aggregation, match kernels, monomial embedding

1. Introduction

THIS paper considers the problem of particular image or particular object retrieval. This subject has received a sustained attention over the last decade. Many of the recent works employ local descriptors such as SIFT [1] or variants [2, 3] for the low-level description of the images. In particular, approaches derived from the Bag-of-Words framework [4] are especially successful to solve problems like recognizing buildings. They are typically combined with spatial verification [5, 6] or other re-ranking strategies such as query expansion [7, 8].

Our objective is to improve the quality of the first retrieval stage, before any re-ranking is performed. This is critical when considering large datasets, as re-ranking methods depend on the quality of the initial short-list, which typically consists of a few hundred images. The initial stage is improved by better matching rules, for instance with Hamming embedding [9], by learning a fine vocabulary [10], or weighting the distances [11, 12]. Moreover, it is useful to employ some geometrical information associated with the region of interest [9]. All these approaches rely on matching individual descriptors and therefore store some data on a per descriptor basis. Moreover, the quantization of the query's descriptors on a large vocabulary causes delays.

Recently, very short yet effective representations have been proposed based on alternative encoding strategies, such as local linear coding [13], the Fisher vector [14] or VLAD [15]. Most of these representations have been proposed first for image classification, yet also offer very effective properties in the context of extremely large-scale image search. A feature of utmost importance is that they offer vector representations compatible with cosine similarity. The representation can then be effectively binarized [16] with cosine sketches, such as those proposed by Charikar [17] (*a.k.a.* LSH), or aggressively compressed to very short vectors with principal component dimensionality reduction (PCA). Product quantization [18] is another example achieving a very compact representation of a few dozens to hundreds of bytes as well as an efficient search because the comparison is done in the compressed domain.

This paper focuses on such short- and mid-sized vector representations of images. Our objective is to exploit some geometrical information associated with the regions of interest. A popular work in this context is the spatial pyramid kernel [19], which is widely adopted for image classification. However, it is ineffective for particular image retrieval as the grid is too rigid and the resulting representation is not invariant enough, as shown by Douze *et al.* [20].

Here, we aim at incorporating some relative angle information to ensure that the patches are consistently rotated. In other terms, we want to achieve a covariant property similar

*Corresponding author: giorgos.tolias@inria.fr, tel:+33 (0)2 99 84 71 30

to that offered by Weak Geometry Consistency (WGC) [9], but directly implemented into the coding stage of image vector representations like Fisher, or VLAD. We achieve that by jointly encoding the local descriptor with the dominant angle in a continuous way. Some recent works in classification [21] and image search [22] consider a similar objective and proceed by rotation quantization. Encoding of such a rough approximation is not straightforwardly compatible with generic match kernels.

In contrast, we achieve the covariant property for any method provided that it can be written as a match kernel. This holds for the Fisher vector, LLC, Bag-of-Words and efficient match kernels listed in [23]. Our method initially assumes aligned objects and image similarity is computed efficiently for multiple rotations thanks to simple trigonometric identities. Finally, the same methodology yields a continuous alternative to spatial pyramid match kernel by encoding patch positions.

This work is the continuation of our previous work [24]. The new contribution consists of the extension to the translation covariant match kernel and the exploitation of a trigonometric polynomial for efficient similarity computation. The latter was only discussed in our previous work, but not exploited.

This paper is organized as follows. Section 2 discusses related works, while Section 3 introduces notation for generic match kernels. Our approach is presented in Section 4 and Section 5 describes the extension to position-translation. Evaluation is presented in Section 6 on several popular benchmarks for image search, namely Oxford5k [5], Oxford105k and Inria Holidays [25]. These experiments show that our approach gives a significant improvement over the state of the art on image search with vector representations. Interestingly, we further achieve competitive results by combining our approach with monomial embeddings, *i.e.*, with a *codebook-free* approach, as opposed to coding approaches like VLAD.[1]

2. Related work

Our method is inspired by the kernel descriptor of Bo *et al.* [26] but it departs from this in several ways. First, we are interested in aggregating local descriptors to produce a vector image representation, whereas they construct new local descriptors. Our objective is not to encode the pixel gradient orientation but to achieve the property that the patch representation is covariant. Therefore, we encode the dominant orientation or the spatial coordinates of the region of interest jointly with the corresponding SIFT descriptor. Finally, we rely on explicit feature maps [27] to encode the angle, which provides a much better approximation than efficient match kernel [23] for a given number of components.

The well known aggregated representations, such as Bag-of-Words, VLAD and Fisher vectors, only encode appearance and completely discard spatial information. The most popular attempt surpassing this limitation is the Spatial Pyramid Match [19] (SPM). Patch position is quantized and used as a pooling variable. In this fashion, invariance to any geometric transformation is lost, and only a restricted amount of tolerance is attained.

Regarding position encoding, Arandjelovic and Zisserman [28] extract multiple VLAD descriptors per image from horizontal and vertical tiles, aiming at localizing the searched object in the image. This approach is effective for retrieving small objects, but does not solve the afore mentioned shortcomings of image level aggregated descriptors. Recent works in image classification [29, 30, 31] provide lower-dimensional alternatives for SPM via different encodings of spatial information. Krapac *et al.* [29] define a Fisher Kernel integrating location prior. Both appearance and spatial layout of patches are encoded. Spatial Coordinate Coding [30] augments the SIFT descriptors with the corresponding spatial coordinates. Quantization, encoding and pooling take place in the augmented feature space. In a similar work [31], feature scale is encoded in addition to position, leading to encouraging results on PASCAL VOC and ImageNet fine-grained classification benchmarks [32].

The hierarchical kernel descriptor of Bo *et al.* [33] encodes position information at multiple levels. Patch location proximity is evaluated via a Gaussian kernel. In order to keep the representation compact, the positions are expressed as projections on 25 basis vectors uniformly sampled from a 5x5 grid. In contrast, we encode positions in a continuous manner, leading to a richer representation and to reduced quantization artifacts.

Following a similar principle to that of SPM but at a single level, the dominant angle is quantized and considered as a pooling variable in recent works [21, 22]. CVLAD [22] in particular, shifts the sub-vectors corresponding to each angular cell in order to mimic query image rotation and provides some rotation invariance. This strategy increases complexity proportionally to the number of rotations taken into account. In contrast, along with our method we propose a very efficient way to compute similarity for multiple image rotations.

WGC applies geometric constraints on the whole database of images, and not only on a short-list [5, 6]. We achieve the same property with aggregated representations, thus individual descriptors are not indexed. Moreover, we do not need to explicitly form patch correspondences and compute relative angles for each.

[1] Code is available online `https://gforge.inria.fr/frs/download.php/latestzip/4895/PkgAngularmodulation-latest.zip`

3. Background: match kernels and embeddings

We consider the context of match kernels. An image is typically described by a set of local descriptors $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_i, \dots\}$, $\mathbf{x}_i \in \mathbb{R}^d, \|\mathbf{x}_i\| = 1$. Similar to other works [34, 23, 9], two images described by $\mathcal{X}$ and $\mathcal{Y}$ are compared with a match kernel K of the form

$$K(\mathcal{X}, \mathcal{Y}) = \beta(\mathcal{X})\beta(\mathcal{Y}) \sum_{\mathbf{x}\in\mathcal{X}} \sum_{\mathbf{y}\in\mathcal{Y}} k(\mathbf{x}, \mathbf{y}), \tag{1}$$

where k is referred to as the local kernel and where the proportionality factor β ensures that $K(\mathcal{X}, \mathcal{X}) = K(\mathcal{Y}, \mathcal{Y}) = 1$. A convenient way to obtain such a kernel is to map the vectors $\mathbf{x}$ to a higher-dimensional space with a function $\varphi : \mathbb{R}^d \to \mathbb{R}^D$, such that the inner product similarity evaluates the local kernel $k(\mathbf{x}, \mathbf{y}) = \langle \varphi(\mathbf{x}) | \varphi(\mathbf{y}) \rangle$. This approach then represents a set of local descriptors by a single vector

$$\mathbf{X} = \beta(\mathcal{X}) \sum_{\mathbf{x}\in\mathcal{X}} \varphi(\mathbf{x}), \qquad \text{(such that } \|\mathbf{X}\| = 1) \tag{2}$$

because the match kernel is computed with a simple inner product as

$$K(\mathcal{X}, \mathcal{Y}) = \beta(\mathcal{X})\beta(\mathcal{Y}) \sum_{\mathbf{x}\in\mathcal{X}} \sum_{\mathbf{y}\in\mathcal{Y}} \langle \varphi(\mathbf{x}) | \varphi(\mathbf{y}) \rangle = \langle \mathbf{X} | \mathbf{Y} \rangle. \tag{3}$$

This kernelized view encompasses many approaches for aggregating local image descriptors such as Bag-of-Words [4, 35], LLC [13], Fisher vector [14], VLAD [15], or VLAT [36]. A desirable property of k is to have $k(\mathbf{x}, \mathbf{y}) \approx 0$ for unrelated features, so that they do not interfere with the measurements between the true matches. It is somehow satisfied with the classical inner product $k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x} | \mathbf{y} \rangle$. Several authors [34, 36, 12, 37] propose to increase the contrast between related and unrelated features with a monomial match kernel of degree p of the form

$$K(\mathcal{X}, \mathcal{Y}) = \beta(\mathcal{X})\beta(\mathcal{Y}) \sum_{\mathbf{x}\in\mathcal{X}} \sum_{\mathbf{y}\in\mathcal{Y}} \langle \mathbf{x} | \mathbf{y} \rangle^p. \tag{4}$$

All monomial (and polynomial) embeddings admit exact finite-dimensional feature maps whose length rapidly increases with degree p (in $\mathcal{O}(d^p/p!)$). The order $p = 2$ has already demonstrated some benefit, for instance in semantic segmentation [38] or in image classification [36]. In this case, the kernel is equivalent to comparing the set of features based on their covariance matrix [36]. Equivalently, by observing that some components are identical, we can define the embedding $\varphi_2 : \mathbb{R}^d \to \mathbb{R}^{d(d+1)/2}$ mapping $\mathbf{x} = [x_1, \dots, x_d]^\top$ to

$$\varphi_2(\mathbf{x}) = [x_1^2, \dots, x_d^2, x_1 x_2 \sqrt{2}, \dots, x_{d-1} x_d \sqrt{2}]^\top. \tag{5}$$

Similarly, the simplified exact monomial embedding associated with $p = 3$ is the function $\varphi_3 : \mathbb{R}^d \to \mathbb{R}^{(d^3+3d^2+2d)/6}$ defined as

$$\varphi_3(\mathbf{x}) = [x_1^3, \dots, x_d^3, x_1^2 x_2 \sqrt{3}, \dots, x_d^2 x_{d-1} \sqrt{3}, x_1 x_2 x_3 \sqrt{6}, \dots, x_{d-2} x_{d-1} x_d \sqrt{6}]^\top. \tag{6}$$

4. Covariant aggregation of local descriptors

The core idea of the proposed method is to exploit jointly the SIFT descriptors and the dominant orientation θ_x associated with a region of interest. For this purpose, we now assume that an image is represented by a set $\mathcal{X}^\star$ of tuples, each of the form $(\mathbf{x}, \theta_x)$, where $\mathbf{x}$ is a SIFT descriptor and $\theta_x \in [-\pi, \pi]$ is the dominant orientation. Our objective is to obtain an approximation of a match kernel of the form

$$K^\star(\mathcal{X}^\star, \mathcal{Y}^\star) = \beta(\mathcal{X}^\star)\beta(\mathcal{Y}^\star) \sum_{\substack{(\mathbf{x},\theta_x)\in\mathcal{X}^\star \\ (\mathbf{y},\theta_y)\in\mathcal{Y}^\star}} k(\mathbf{x}, \mathbf{y})\, k_\theta(\theta_x, \theta_y) \tag{7}$$

$$= \langle \mathbf{X}^\star | \mathbf{Y}^\star \rangle, \tag{8}$$

where k is a local kernel identical to that considered in Section 2 and k_θ reflects the similarity between angles. The interest of enriching this match kernel with orientation is illustrated by Figure 1, where we show that several incorrect matches are downweighted thanks to this information.

The kernel in (7) resembles that implemented in WGC [9] with a voting approach. In contrast, we intend to approximate this kernel with an inner product between two vectors as in (8), similar to the linear match kernel simplification in (3). Our work is inspired by the kernel descriptors [26] of Bo *et al.*, who also consider a kernel of a similar form, but at the patch level, to construct a local descriptor from pixel attributes, such as gradient and position.

In our case, we consider the coding stage and employ a better approximation technique, namely explicit feature maps [27], to encode $\mathcal{X}^\star$. This section first explains the feature map of the angle, then described how the descriptors and angles are jointly represented, and finally discusses the match kernel design and properties.

4.1. A feature map for the angle

The first step is to find a mapping $\boldsymbol{\alpha} : [-\pi, \pi] \to \mathbb{R}^M$ from an angle θ to a vector $\boldsymbol{\alpha}(\theta)$ such that $\boldsymbol{\alpha}(\theta_1)^\top \boldsymbol{\alpha}(\theta_2) = k_\theta(\theta_1 - \theta_2)$. The function $k_\theta : \mathbb{R} \to [0, 1]$ is a shift invariant kernel which should be symmetric ($k_\theta(\Delta\theta) = k_\theta(-\Delta\theta)$), pseudo-periodic with period of 2π and monotonically decreasing over $[0, \pi]$. The function k_θ is scalar and it allows us to model the behaviour of the match kernel and to design the feature map accordingly. In the work of Vedaldi

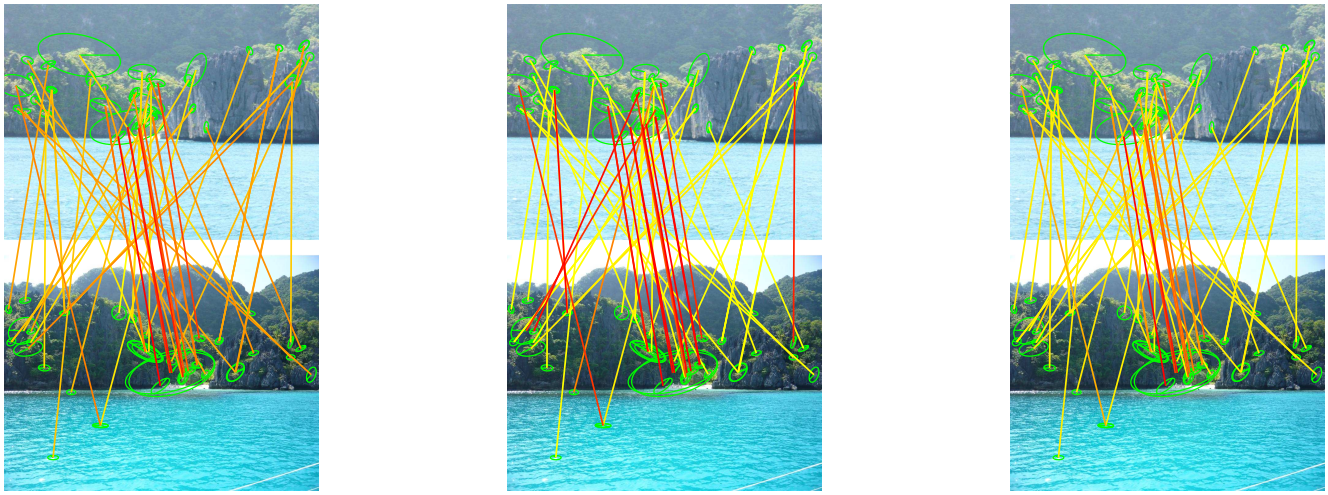

Figure 1: Similarities between regions of interest, based on SIFT kernel k (*left*), angle consistency kernel k_θ (*middle*) and both (*right*). For each local region, we visualize the values $k(\mathbf{x}, \mathbf{y})$, $k_\theta(\Delta\theta)$ and their product by the colors of the link (red=1).

and Zisserman [27] it is termed as kernel signature. We consider in particular the following function:

$$k_{\mathsf{VM}}(\Delta\theta) = \frac{\exp(\kappa\cos(\Delta\theta)) - \exp(-\kappa)}{2\sinh(\kappa)}. \quad (9)$$

It is derived from Von Mises distribution $f(\Delta\theta;\kappa)$, which is often considered as the probability density distribution of the noise of the measure of an angle, and therefore regarded as the equivalent Gaussian distribution for angles. Our function k_{VM} is a shifted and scaled variant of Von Mises, designed such that its range is $[0,1]$, which ensures that $k_{\mathsf{VM}}(\pi) = 0$.

The periodic function k_{VM} can be expressed as a Fourier series whose coefficients are (see [39][Eq. (9.6.19)]):

$$k_{\mathsf{VM}}(\Delta\theta) = \frac{I_0(\kappa) - e^{-\kappa} + 2\sum_{n=1}^{\infty} I_n(\kappa)\cos(n\Delta\theta)}{2\sinh(\kappa)}, \quad (10)$$

where $I_n(\kappa)$ is the modified Bessel function of the first kind of order n. We now consider the truncation $\bar{k}^N_{\mathsf{VM}}$ of the series to the first N terms:

$$\bar{k}^N_{\mathsf{VM}}(\Delta\theta) = \sum_{n=0}^{N} \gamma_n \cos(n\Delta\theta) \quad (11)$$

$$\text{with } \gamma_0 = \frac{(I_0(\kappa) - e^{-\kappa})}{2\sinh(\kappa)} \text{ and } \gamma_n = \frac{I_n(\kappa)}{\sinh(\kappa)} \text{ if } n > 0. \quad (12)$$

We design the feature map $\boldsymbol{\alpha}(\theta)$, mappping an angle θ to a vector, as follows:

$$\boldsymbol{\alpha}(\theta) = (\sqrt{\gamma_0}, \sqrt{\gamma_1}\cos(\theta), \sqrt{\gamma_1}\sin(\theta), \ldots, \sqrt{\gamma_N}\cos(N\theta), \sqrt{\gamma_N}\sin(N\theta))^\top. \quad (13)$$

This vector has $2N+1$ components. Moreover

$$\begin{aligned}\boldsymbol{\alpha}(\theta_1)^\top\boldsymbol{\alpha}(\theta_2) &= \gamma_0 + \sum_{n=1}^{N} \gamma_n(\cos(n\theta_1)\cos(n\theta_2) + \sin(n\theta_1)\sin(n\theta_2)) \\ &= \sum_{n=0}^{N} \gamma_n \cos(n(\theta_1 - \theta_2)) \\ &= \bar{k}^N_{\mathsf{VM}}(\theta_1 - \theta_2) \approx k_{\mathsf{VM}}(\theta_1 - \theta_2) \quad (14)\end{aligned}$$

The design of a feature map is explained in full details by Vedaldi and Zisserman [27]. This feature map gives an approximation of the target function k_{VM}, which is more accurate as N is bigger.

Figure 2 illustrates the function k_{VM} for several values of the parameter κ and its approximation $\bar{k}^N_{\mathsf{VM}}$ for different values of N. First note that $\bar{k}^N_{\mathsf{VM}}$ may not fulfill the original requirements: its range might be wider than $[0,1]$ and it might not be monotonically decreasing over $[0,\pi]$. Larger values of κ produce a more "selective" function of the angle, yet require more components (larger value of N) to obtain an accurate estimation. Importantly, the approximation stemming from this explicit angle mapping is better than that based on efficient match kernels [23], which converges slowly with the number of components. Efficient match kernels are more intended to approximate kernels on vectors than on scalar values. As a trade-off between selectivity and the number of components, we set κ=8 and N=3 (see Section 6). Accordingly, we use $\bar{k}^3_{\mathsf{VM}}$ as k_θ in the sequel. The corresponding embedding $\boldsymbol{\alpha} : \mathbb{R} \to \mathbb{R}^7$ maps any angle to a 7-dimensional vector.

Exact estimation of kernel signature. Instead of approximating a kernel on angles with finite Fourier series, one may

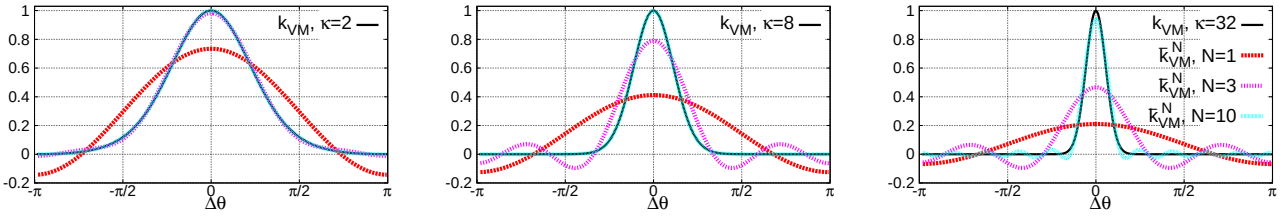

Figure 2: Function k_{VM} for different values of κ and its approximation $\bar{k}^N_{\mathsf{VM}}$ using 1, 3 and 10 frequencies, as implicitly defined by the corresponding mapping $\boldsymbol{\alpha} : [\pi, \pi] \rightarrow \mathbb{R}^{2N+1}$.

rather consider directly designing a function satisfying our initial requirements (pseudo-period, symmetric, decreasing over $[0, \pi]$), such as

$$k_P(\Delta\theta) = \cos(\Delta\theta/2)^P \text{ with } P \text{ even.} \tag{15}$$

Thanks to power reduction trigonometric identities, for even P this function is re-written as

$$k_P(\Delta\theta) = \sum_{p=0}^{P/2} \gamma_p \cos(p\Delta\theta) \tag{16}$$

with

$$\gamma_0 = \frac{1}{2^P}\binom{P}{P/2}, \gamma_p = \frac{1}{2^{P-1}}\binom{P}{P/2-p} \quad 0 < p \leq P/2. \tag{17}$$

Now, applying (13) leads to a feature map $\boldsymbol{\alpha}(\theta)$ with $P+1$ components such that $\boldsymbol{\alpha}(\theta_1)^\top \boldsymbol{\alpha}(\theta_2) = k_P(\theta_1 - \theta_2)$. For this function, the interesting property is that the scalar product is exactly equal to the target kernel value $k_P(\theta_1 - \theta_2)$, and that the original requirements now hold. From our experiments, this function gives reasonable results, but requires more components than $\bar{k}_{\mathsf{VM}}$ to achieve a shape narrow around $\Delta\theta = 0$ and close to 0 otherwise. The results for our image search application task using this function are slightly below our Von Mises variant for a given dimensionality. So, despite its theoretical interest we do not use it in our experiments. Ultimately, one would rather directly learn a Fourier embedding for a targeted task (*e.g.* an embedding per classifier), in the spirit of Fourier kernel learning [40].

4.2. Modulation and covariant match kernel

The vector $\boldsymbol{\alpha}$ encoding the angle θ "modulates"[2] any vector $\mathbf{x}$ (or pre-mapped descriptor $\varphi(\mathbf{x})$) with a function $m : \mathbb{R}^{2N+1} \times \mathbb{R}^D \rightarrow \mathbb{R}^{(2N+1)D}$. Thanks to classical properties of the Kronecker product $\otimes$, we have

$$\begin{aligned} m(\mathbf{x}, \boldsymbol{\alpha}(\theta)) &= \mathbf{x} \otimes \boldsymbol{\alpha}(\theta) \\ &= (x_1\boldsymbol{\alpha}(\theta)^\top, x_2\boldsymbol{\alpha}(\theta)^\top \ldots, x_d\boldsymbol{\alpha}(\theta)^\top)^\top. \end{aligned} \tag{18}$$

[2]By analogy to communications, where modulation refers to the process of encoding information over periodic waveforms.

We now consider two pairs of vectors and angle, $(\mathbf{x}, \theta_x)$ and $(\mathbf{y}, \theta_y)$, and their modulated descriptors $m(\mathbf{x}, \boldsymbol{\alpha}(\theta_x))$ and $m(\mathbf{y}, \boldsymbol{\alpha}(\theta_y))$. In the inner product space $\mathbb{R}^{(2N+1)D}$, the following holds:

$$\begin{aligned} m(\mathbf{x}, \boldsymbol{\alpha}(\theta_x))^\top m(\mathbf{y}, \boldsymbol{\alpha}(\theta_y) &= (\mathbf{x} \otimes \boldsymbol{\alpha}(\theta_x))^\top (\mathbf{y} \otimes \boldsymbol{\alpha}(\theta_y)) \\ &= (\mathbf{x}^\top \otimes \boldsymbol{\alpha}(\theta_x)^\top)(\mathbf{y} \otimes \boldsymbol{\alpha}(\theta_y)) \\ &= (\mathbf{x}^\top \mathbf{y}) \otimes (\boldsymbol{\alpha}(\theta_x)^\top \boldsymbol{\alpha}(\theta_y)) \\ &= (\mathbf{x}^\top \mathbf{y}) k_\theta(\theta_x - \theta_y). \end{aligned} \tag{19}$$

Figure 3 shows the distribution of the similarities between regions of interest before and after modulation, as a function of the difference of angles. Interestingly, there is no obvious correlation between the difference of angle and the SIFT: the similarity distribution based on SIFT is similar for all angles. This suggests that the modulation with angle provides complementary information.

Combination with coding/pooling techniques. Consider any coding method φ that can be written as match kernel (Fisher, LLC, Bag-of-Words, VLAD, etc). The match kernel in (7), with our k_θ approximation, is re-written as

$$\begin{aligned} K^\star(\mathcal{X}^\star, \mathcal{Y}^\star) &\propto \sum_{\substack{(\mathbf{x},\theta_X)\in\mathcal{X}^\star \\ (\mathbf{y},\theta_y)\in\mathcal{Y}^\star}} m(\varphi(\mathbf{x}), \boldsymbol{\alpha}(\theta_x))^\top m(\varphi(\mathbf{y}), \boldsymbol{\alpha}(\theta_y)) \\ &\propto \sum_{(\mathbf{x},\theta_x)} m(\varphi(\mathbf{x}), \boldsymbol{\alpha}(\theta_x))^\top \sum_{(\mathbf{y},\theta_y)} m(\varphi(\mathbf{y}), \boldsymbol{\alpha}(\theta_y), \end{aligned} \tag{20}$$

where we observe that the image can be represented as the summation $\mathbf{X}^\star$ of the embedded descriptors modulated by their corresponding dominant orientation, as

$$\mathbf{X}^\star = \beta(\mathcal{X}^\star) \sum_{(\mathbf{x},\theta_x)\in\mathcal{X}^\star} m(\varphi(\mathbf{x}), \boldsymbol{\alpha}(\theta_x)). \tag{21}$$

This representation encodes the relative angles and is already more discriminative than an aggregation that does not consider them. However, at this stage, the comparison assumes that the images have the same global orientation. This is the case on benchmarks like Oxford5k building, where all images are orientated upright, but this is not true in general for particular object recognition.

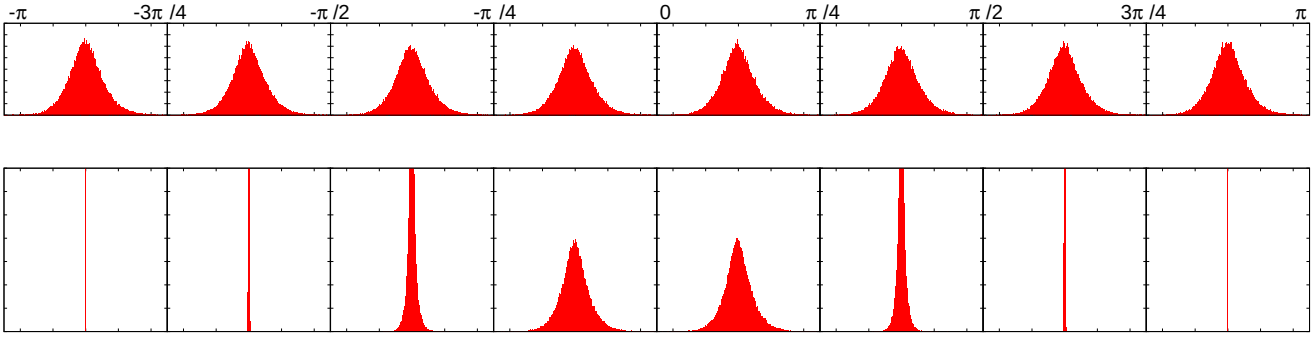

Figure 3: Distribution of patch similarity for different values of orientation difference. In this figure, we split the angular space into 8 equally-sized bins and present the similarity distribution separately for each of these bins. Horizontal axis represents the similarity value between matching features. *Top*: distribution of similarities with kernel on SIFTs. *Bottom:* Distribution after modulation with α.

4.3. Rotation invariance

So far our image representation is rotation covariant. Herein we propose how to achieve rotation invariance via efficient similarity computation for multiple image rotations. Up to now we have assumed that objects are aligned with respect to orientation or, more particularly, that objects are up-right. This implies that true corresponding patches should have similar orientation. We now describe how to produce a similarity score when the orientations of related images may be different. We represent the image vector $\mathbf{X}^\star$ as the concatenation of $2N+1$ D-dimensional subvectors associated to one term of the finite Fourier series: $\mathbf{X}^\star = [\mathbf{X}_0^{\star\top}, \mathbf{X}_{1,c}^{\star\top}, \mathbf{X}_{1,s}^{\star\top}, \ldots, \mathbf{X}_{N,c}^{\star\top}, \mathbf{X}_{N,s}^{\star\top}]^\top$. The vector $\mathbf{X}_0^\star$ is associated with the constant term in the Fourier expansion, $\mathbf{X}_{n,c}^\star$ and $\mathbf{X}_{n,s}^\star$, $1 \leq n \leq N$, correspond to the cosine and sine terms, respectively.

Imagine now that this image undergoes a global rotation of angle θ. Denote $\breve{\mathcal{X}}^\star$ the new set of pairs $(\mathbf{x}, \breve{\theta}_x)$. Since, descriptor $\mathbf{x}$ is by nature rotation invariant, we obtain $\breve{\mathcal{X}}^\star$ by simply shifting all dominant angles by θ, that is $\breve{\theta}_x = \theta_x - \theta$. Denote $\breve{\mathbf{X}}^\star$ the new image vector derived from these local descriptors. It occurs that $\breve{\mathbf{X}}_0^\star = \mathbf{X}_0^\star$ because this term does not depend on the angle, and that, for a given frequency bin n, elementary trigonometry identities lead to

$$\breve{\mathbf{X}}_{n,c}^\star = \mathbf{X}_{n,c}^\star \cos n\theta + \mathbf{X}_{n,s}^\star \sin n\theta \tag{22}$$

$$\breve{\mathbf{X}}_{n,s}^\star = -\mathbf{X}_{n,c}^\star \sin n\theta + \mathbf{X}_{n,s}^\star \cos n\theta. \tag{23}$$

Therefore, we do not need to recompute the image representation of the rotated image. It is efficiently derived by component wise multiplications of the vector describing the original image. It also turns out that $\|\breve{\mathbf{X}}^\star\| = \|\mathbf{X}^\star\|$, meaning that rotation has no effect on the global normalization factor $\beta(\mathcal{X}^\star)$.

When comparing two images with such vectors, the linearity of the inner product ensures that

$$\begin{aligned}\langle \breve{\mathbf{X}}^\star | \mathbf{Y}^\star \rangle =& \langle \mathbf{X}_0^\star | \mathbf{Y}_0^\star \rangle \\ &+ \sum_{n=1}^{N} \cos n\theta \ \left(\langle \mathbf{X}_{n,c}^\star | \mathbf{Y}_{n,c}^\star \rangle + \langle \mathbf{X}_{n,s}^\star | \mathbf{Y}_{n,s}^\star \rangle \right) \\ &+ \sum_{n=1}^{N} \sin n\theta \ \left(-\langle \mathbf{X}_{n,c}^\star | \mathbf{Y}_{n,s}^\star \rangle + \langle \mathbf{X}_{n,s}^\star | \mathbf{Y}_{n,c}^\star \rangle \right).\end{aligned} \tag{24}$$

Here, we stress that the similarity between two images is a real trigonometric polynomial in θ (image rotation angle) of degree N. Its $2N+1$ components are fully determined by computing $\langle \mathbf{X}_0^\star | \mathbf{Y}_0^\star \rangle$ and the inner products between the subvectors associated with each frequency, *i.e.*, $\langle \mathbf{X}_{n,c}^\star | \mathbf{Y}_{n,c}^\star \rangle$, $\langle \mathbf{X}_{n,s}^\star | \mathbf{Y}_{n,s}^\star \rangle$, $\langle \mathbf{X}_{n,c}^\star | \mathbf{Y}_{n,s}^\star \rangle$ and $\langle \mathbf{X}_{n,s}^\star | \mathbf{Y}_{n,c}^\star \rangle$. Finding the maximum of this polynomial amounts to finding the rotation maximizing the image similarity.

Computing the coefficients of this polynomial requires a total of $D \times (1+4N)$ elementary operations for a vector representation of dimensionality $D \times (1+2N)$, that is, less than twice the cost of the inner product between $\mathbf{X}^\star$ and $\mathbf{Y}^\star$. Once these components are obtained, the cost of finding the maximum value achieved by this polynomial is negligible for large values of D, for instance by simply sampling a few values of θ. Therefore, if we offer the orientation invariant property, the complexity of similarity computation is typically twice the cost of that of a regular vector representation (whose complexity is equal to the number of dimensions).

Our trigonometric polynomial of similarity scores can be rewritten as:

$$K^\star(\mathcal{X}^\star, \mathcal{Y}^\star, \theta) = c + \sum_{n=1}^{N} a_n \cos n\theta + \sum_{n=1}^{N} b_n \sin n\theta, \tag{25}$$

with coefficients c, a_n, b_n given by (24). Note that in our experiment it turns out that retrieval performance already saturates at $N = 3$.

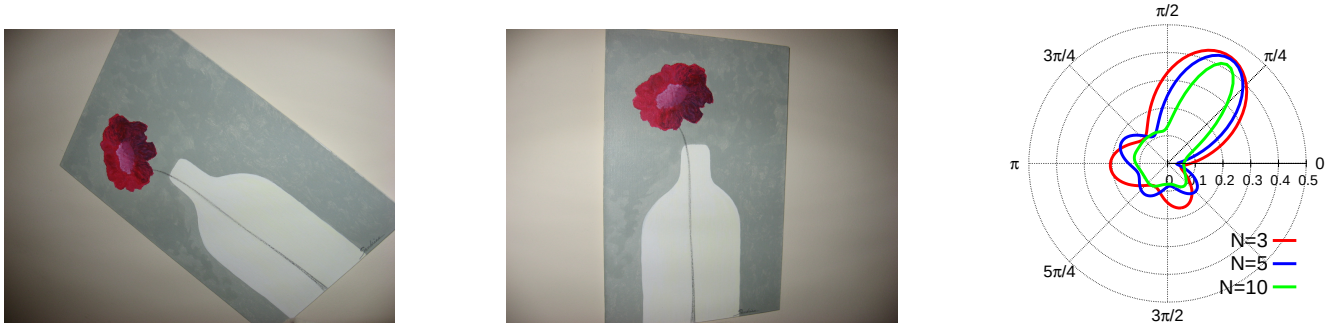

Figure 4: Matching example of two images and similarity estimation for all possible image rotations. The image on the left undergoes rotation. Image similarity versus rotation is shown in polar coordinates, with the angular direction corresponding to the image rotation, and the radial to the image similarity score. This example is computed with angular modulation of VLAD, while using 3, 5 or 10 frequencies.

This strategy for computing the scores for all possible orientations of the query is not directly compatible with non-linear post-processing of $\mathbf{X}^\star$ such as component-wise power-law normalization [41], except for the subvector $\mathbf{X}^\star_0$. We adapt the power-law normalization to become compatible with our strategy: we compute the modulus of the complex number represented by two components ($\sin$ and $\cos$) associated with the same frequency n and the same original component in $\varphi(\mathbf{x})$. These two components are then divided by the square-root (or any power) of this modulus.

In detail, let $\mathbf{X}^\star_{n,c,i}$ and $\mathbf{X}^\star_{n,s,i}$ be the i-th component of subvectors $\mathbf{X}^\star_{n,c}$ and $\mathbf{X}^\star_{n,s}$, respectively. The modified scheme considers the modulus of those components. The power-law normalized version of the former turns out to be equal to $\frac{\mathbf{X}^\star_{n,c,i}}{(\mathbf{X}^{\star\,2}_{n,c,i}+\mathbf{X}^{\star\,2}_{n,s,i})^{\frac{1-l}{2}}}$, where $l \in [0,1]$ is the power-law exponent. The counterpart sine component is obtained similarly, and now the representation of the rotated image is factorized equivalently to (22) and (23).

In our experiments we reduce the dimensionality of the modulated image vector by PCA, as typically done with aggregated representations. It is then not possible to use the efficient polynomial of scores. In this case, we follow the naive strategy, which is to compute the query representation for several hypothesis of angle rotation, typically 8. In theory, this multiplies the query complexity by the same factor 8. However, in practice, it is faster to perform the matrix-matrix multiplication, with the right matrix representing 8 queries, than computing separately the corresponding 8 matrix-vector multiplications. In our former work [24], the naive approach was used in all cases, while now we explore the proposed polynomial on the full vectors.

Figure 4 presents the evaluation of (25) for a pair of images. More frequencies improve the approximation. However, maximum similarity value is observed at a similar point in all cases.

5. Translation covariant aggregation

5.1. Encoding the position

Following the objective of jointly encoding local description and geometry, we now deal with the location of local features. We assume that an image is represented by a set $\mathcal{X}^\diamond$ of triples of the form $(\mathbf{x}, u_x, v_x)$. Local descriptor $\mathbf{x}$ is now accompanied by position coordinates u_x and v_x.

Depending on the use-case and on the desired invariance, u_x and v_x can be cartesian or polar coordinates. Whatever the coordinate system is, our position encoding is performed in a continuous manner, unlike SPM [19] and *visual phrases* [42], where positions, respectively position differences, are quantized to a uniform grid. In the following we consider cartesian coordinates.

We employ once more the angular embedding proposed in Section 4.1, by mapping a spatial coordinate to an angle. The kernel function k_θ defined for angles is periodic, while the spatial coordinates of a local feature are not. Mapping positions directly to $[-\pi, \pi]$ would practically convert the position domain into a torus. In such a case, patches located at opposite edges of the image would be considered close to each other. We simply handle this by mapping to $[-\pi/2, \pi/2]$. Still, when employing multiple translations this undesired effect emerges, but it does not seem to harm the effectiveness of the method in our experiments. Therefore, we convert the position coordinates to angles by

$$\hat{u} = \frac{u - \frac{h}{2}}{\max\{h, w\}}\,\pi\ ,\ \hat{v} = \frac{v - \frac{w}{2}}{\max\{h, w\}}\,\pi, \qquad (26)$$

where h and w are the height and width of the image.

We now employ the procedure of Section 4. We map local coordinates u_x and v_x to $\boldsymbol{\alpha}(\hat{u}_x)$ and $\boldsymbol{\alpha}(\hat{v}_x)$, respectively. Then, each one of them, can be encoded jointly with descriptor $\mathbf{x}$ by $m(\mathbf{x}, \boldsymbol{\alpha}(\hat{u}_x))$ and $m(\mathbf{x}, \boldsymbol{\alpha}(\hat{v}_x))$. We obtain two new match kernels $K_u(\mathcal{X}^\diamond, \mathcal{Y}^\diamond)$ and $K_v(\mathcal{X}^\diamond, \mathcal{Y}^\diamond)$,

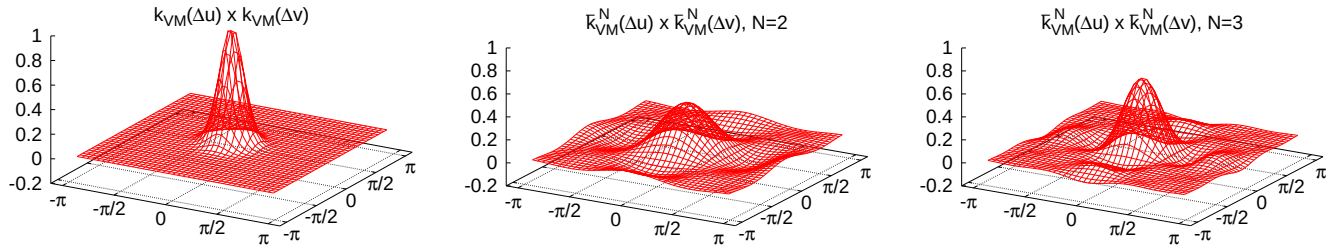

Figure 5: Function $k_{\mathsf{VM}}(\Delta\theta_u) \times k_{\mathsf{VM}}(\Delta\theta_v)$ for $\kappa = 8$ and its approximation $\bar{k}^N_{\mathsf{VM}}(\Delta\theta_u) \times \bar{k}^N_{\mathsf{VM}}(\Delta\theta_v)$ using 2 and 3 frequencies, as implicitly defined by the corresponding mapping $\boldsymbol{\alpha} : [\pi, \pi] \rightarrow \mathbb{R}^{2N+1}$. The higher the number of frequencies, the better the approximation of the signature.

down-weighting or up-weighting matches by the consistency of their u or v coordinate respectively.

We are not limited to modulation only by u or v (single modulation), we can further encode both coordinates by a double modulation. This is achieved by modulating descriptor $\mathbf{x}$ by both $\boldsymbol{\alpha}(\hat{u}_x)$ and $\boldsymbol{\alpha}(\hat{v}_x)$ with a function $m_{u,v} : \mathbb{R}^{2N+1} \times \mathbb{R}^{2N+1} \times \mathbb{R}^D \rightarrow \mathbb{R}^{(2N+1)^2 D}$, where

$$m_{u,v}(\mathbf{x}, \boldsymbol{\alpha}(\hat{u}_x), \boldsymbol{\alpha}(\hat{v}_x)) = \mathbf{x} \otimes \boldsymbol{\alpha}(\hat{u}_x) \otimes \boldsymbol{\alpha}(\hat{v}_x). \quad (27)$$

The match kernel for two sets of local descriptors can be then written as

$$\begin{aligned} K_{u,v}(\mathcal{X}^\diamond, \mathcal{Y}^\diamond) &\propto \sum_{\substack{(\mathbf{x},u_x,v_x)\in\mathcal{X}^\diamond \\ (\mathbf{y},u_y,v_y)\in\mathcal{Y}^\diamond}} k(\mathbf{x},\mathbf{y}) k_\theta(\hat{u}_x,\hat{u}_y) k_\theta(\hat{v}_x,\hat{v}_y) \\ &\propto \langle \mathbf{X}^\diamond | \mathbf{Y}^\diamond \rangle, \end{aligned} \quad (28)$$

where each image is represented by vector $\mathbf{X}^\diamond$. This is the vector of aggregated double modulated local descriptors:

$$\mathbf{X}^\diamond = \beta(\mathcal{X}^\diamond) \sum_{(\mathbf{x},u_x,v_x)\in\mathcal{X}^\diamond} m_{u,v}(\mathbf{x}, \boldsymbol{\alpha}(\hat{u}_x), \boldsymbol{\alpha}(\hat{v}_x)). \quad (29)$$

Figure 5 illustrates the function $k_{\mathsf{VM}}(\Delta\hat{u}) \times k_{\mathsf{VM}}(\Delta\hat{v})$ along with it approximation for 2 and 3 frequencies. We want a "selective" kernel function in order to weight up pairs of similar patches placed at similar locations. The double modulation increases dimensionality too fast with respect to the number of frequencies. As a trade-off between selectivity and the number of components we set κ=8 and N=2. In this case, coordinates (u,v) are mapped to a 25 dimensional vector.

5.2. Translation invariance

We have assumed, up to now, that objects are aligned. Next, we follow the same approach proposed for dominant orientation, in order to offer translation invariance. Note that rotation and scale invariance are lost in this case, but there is some tolerance introduced by the continuous encoding and by the employed similarity function.

In the case of single modulation of u or v, we are able to efficiently evaluate for multiple 1D translations with the same trigonometric polynomial introduced before (25). Detecting maximum similarity aligns objects with respect to u or v, independently. An example is shown in Figure 6. Another choice is to maximize independently and to keep best alignment of both. This is achieved by simply keeping the maximum similarity score of the two.

One step further, we allow for 2D translation along with the double modulation by u and v. Factorization similar to that of (22)-(23) is still possible, resulting into another trigonometric polynomial for efficient 2D translation

$$\begin{aligned} K_{u,v}(\mathcal{X}^\diamond, \mathcal{Y}^\diamond, \hat{u}, \hat{v}) = a^0 &+ \sum_{n=1}^{N} a^1_n \cos n\hat{u} + \sum_{n=1}^{N} a^2_n \sin n\hat{u} \\ &+ \sum_{t=1}^{N} a^3_t \cos t\hat{v} + \sum_{t=1}^{N} a^4_t \sin t\hat{v} \\ &+ \sum_{n=1}^{N}\sum_{t=1}^{N} a^5_{n,t} \cos n\hat{u} \cos t\hat{v} \\ &+ \sum_{n=1}^{N}\sum_{t=1}^{N} a^6_{n,t} \cos n\hat{u} \sin t\hat{v} \\ &+ \sum_{n=1}^{N}\sum_{t=1}^{N} a^7_{n,t} \sin n\hat{u} \cos t\hat{v} \\ &+ \sum_{n=1}^{N}\sum_{t=1}^{N} a^8_{n,t} \sin n\hat{u} \sin t\hat{v}. \end{aligned} \quad (30)$$

The image translation is denoted by $(\hat{u}, \hat{v})$. Coefficients $a^0 \ldots a^8$ are given by inner products of particular subvectors of the two image representation vectors. We skip the details which are in analogy to those of (24).

Its computational cost is, once more, very small comparing to performing the translations in a naive way. Compatibility with power-law normalization is achieved in a similar fashion to that of Section 4, but with groups of 4 components in this case.

Our achievement of fast similarity computation for multiple translations resembles the work of Henriques *et al.* [43] who speed-up learning with multiple shifted versions of negative samples. They do this instead of perform-

Figure 6: Matching example of two images and similarity estimation for all possible image translations. Image on the left undergoes translation independently to the horizontal and the vertical direction. Modulated VLAD is used for the example.

ing costly sliding window based hard negative mining. In our case, we obtain the translated models through latent variables, *i.e.* translation on a given direction, parameterizing a trigonometric polynomial of similarity scores.

6. Experiments

Datasets. We evaluate the performance of the proposed approaches and compare with state-of-the-art methods on two publicly available datasets for image and particular object retrieval, namely Inria Holidays [25] and Oxford Buildings 5k [5]. We also combine the latter with 100k distractor images to measure the performance on a larger scale. The merged dataset is referred to as Oxford105k. Performance is measured with mean Average Precision (mAP) [5].

We further employ the rotated Holidays dataset [44], with images rotated to their natural orientation, in order to evaluate our position covariant kernels. This is necessary since this method is not rotation invariant. It is therefore not applicable when object rotations exist. Note that the rotation covariant kernel does not have such limitations. We refer to the upright oriented Holidays dataset as Holidays$^\wedge$.

Our approach modulates any coding/pooling technique operating as a match kernel. Therefore, we evaluate the benefit of our approach combined with several coding techniques, namely

- VLAD [15], which encodes a SIFT descriptor by considering the residual vector to the centroid.
- The Fisher vector [14, 41, 45]. For image classification, Chatfield *et al.* [46] show that it outperforms concurrent coding techniques, in particular LLC [13]. We adopt the standard choice for image retrieval and use only the gradient with respect to the mean [15].
- Monomomial embeddings of order 2 and 3 applied on local descriptors (See below for pre-processing), *i.e.*, the functions φ_2 in (5) and φ_3 in (6). For the sake of consistency, we also denote by φ_1 the function $\varphi_1 : x \to x$.

We refer to these methods combined with single modulation with the symbol "$\otimes$": VLAD$\otimes$, Fisher$\otimes$, $\varphi_1\otimes$, $\varphi_2\otimes$ and $\varphi_3\otimes$ for the angle modulation. Single position modulations are denoted by VLAD$\otimes_u$ and VLAD$\otimes_v$ and double modulation by VLAD$\otimes_{u,v}$. The particular case for which we independently encode u and v and keep the maximum score of both is referred as VLAD$\otimes_{u/v}$. The same notation is followed for the Fisher and monomial embeddings.

In addition, we compare against the most related work, namely the recent CVLAD [22] method, which also aims at producing an image vector representation integrating the dominant orientations of the patches. Whenever the prior work is not referenced, results are produced using our own implementations of VLAD, Fisher and CVLAD, so that the results are directly comparable with the same features.

6.1. Implementation Details

Local descriptors. We use the Hessian-Affine detector [47] to extract the regions of interest, that are subsequently described by SIFT descriptors [1] post-processed with RootSIFT [48]. Then, we apply PCA and the resulting vector is subsequently ℓ_2-normalized. Following the typical procedure for the Fisher vector [14, 41, 15], when applying PCA

we reduce the vector to 80 components. The same stands for monomial embeddings. An exception is done for VLAD and CVLAD with which we only use the PCA basis to center and rotate descriptors as suggested by Delhumeau [49], without dimensionality reduction.

The optimized Hessian-Affine detector of Perdoch *et al.* [44] improves the retrieval performance. However, it is not compatible with our angular encoding (rotation covariant kernel) by discarding rotations and enforcing the gravity vector assumption (up-right features). For the needs of the angular modulation we use the original Hessian-Affine detector [47], but modify it so that it has similar parameters (enlarged measurement region by a factor of 2) and use a lower detector threshold. In addition, we use the detector of Perdoch *et al.* [44] for evaluating the position encoding. The translation covariant kernel can benefit from the advantages of up-right features when all depicted objects are aligned with respect to rotation. The use of the latter is explicitly stated in each case.

Codebook. For all methods based on codebooks, we only consider distinct datasets for learning. More precisely and following common practice, the k-means and GMM (for VLAD and Fisher, respectively) are learned on Flickr60k for Inria Holidays and Paris6k [50] for Oxford buildings. We rely on the Yael library [51] for codebook construction and VLAD and Fisher encoding.

Post-processing. The final image vector obtained by each method is power-law normalized [11, 41, 15]. This processing improves the performance by efficiently handling the burstiness phenomenon. Exploiting the dominant orientation in our covariant match kernel provides a complementary way to further handle the same problem. We mention that using the dominant orientation is shown effective in a recent work by Torii *et al.* [52]. This post-processing, when applied to the modulated vectors, inherently captures and down weights patches with similar orientation or position.

In addition to power-law normalization, we rotate the aggregated vector representation with a learned PCA rotation matrix [53, 54]. This aims at capturing the co-occurrences to down-weight them either by whitening [53] or a second power-law normalization [54]. We adopt the latter choice (with exponent 0.5) to avoid the sensitivity to eigenvalues (in whitening) when learning PCA with few input data. We refer to this Rotation and Normalization [55] as RN.

Optionally, to produce compact representations, we keep only the first few components (the most energetic ones) and ℓ_2-normalize the shortened vector.

Query rotation. In order to obtain rotation invariance jointly with power-law normalization we exploit the trigonometric polynomial presented in Section 4, along with our modified scheme for power-law normalization. Image similarity is evaluated for multiple query image rotations.

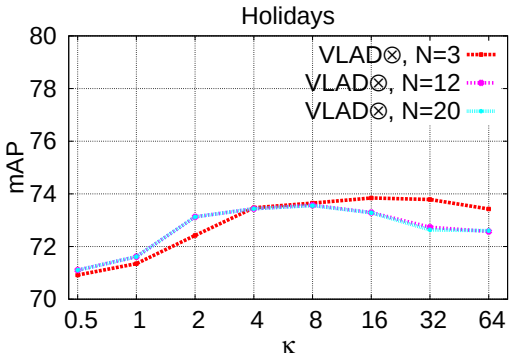

Figure 8: Performance on Holidays dataset of modulated VLAD for different values of κ and for different approximations. Results shown with RN. A codebook of 32 visual words is used.

Since the aforementioned technique is not compatible with RN or dimensionality reduction, in that case, we follow the naive approach and apply rotations of the query image and perform individual queries.

We apply 8 query rotations on Holidays dataset. On Oxford5k, we rather adopt the common choice of not considering other possible orientations, since images are up-right.

Query translation. Our methods for position encoding are evaluated on the upright Holidays$^\wedge$ dataset. On Oxford5k we follow the standard protocol and use the cropped queries. However, we also consider the position of the bonding box in the image as known, in order to properly normalize the patch coordinates.

Once more, we use the trigonometric polynomial for computation of the image similarity score. In the case of 1D translations, we evaluate 25 possible translations on each direction and a step of 10 pixels, that is $1 + 25 \times 2$ translations in total. While for 2D translations, we evaluate 20 translations per direction leading to a total of $(20+20+1)^2$ translations. The chosen step is also set to 10 pixels.

6.2. Impact of power-law normalization

In Figure 7 we present performance for power-law normalization of different exponents. The non-modulated representations appear to have optimal performance around $l = 0.2$, which is in accordance with previous results [49] in the case that the local descriptors are rotated with PCA. The behavior is different for the modulated vectors, where optimal performance appears for $l = 0$. Note that in contrast to the standard power-law normalization scheme, the modified scheme proposed in Section 4.3 does not produce binary vectors for such a choice.

In the rest of our experiments we adopt an exponent equal to 0 and 0.2 for the modulated and non-modulated

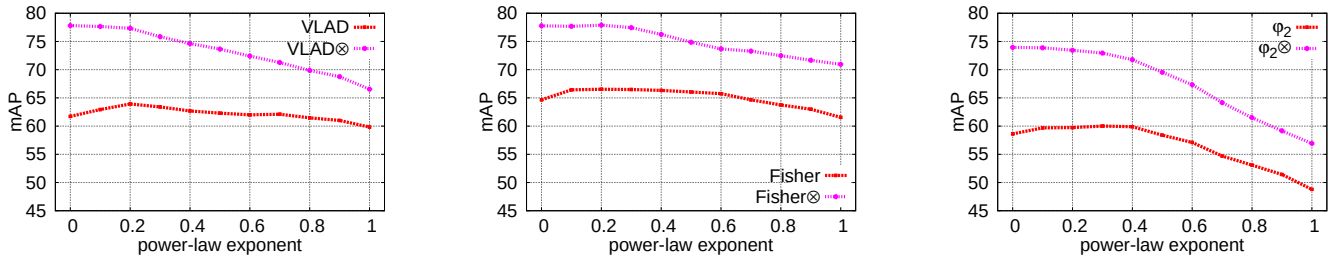

Figure 7: Impact of powerlaw for modulated and non-modulated image representations. Results on Holidays dataset with a codebook of 32 visual words for VLAD and Fisher vectors. We follow the modified power-law normalization for the modulated vectors.

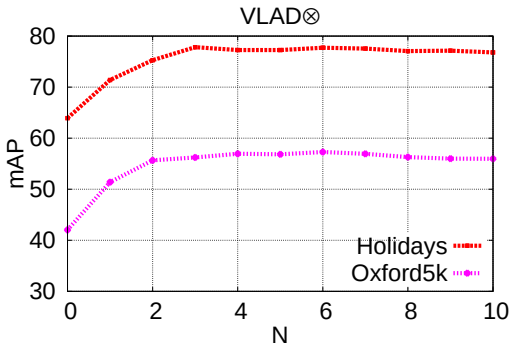

Figure 9: Performance comparison of modulated VLAD for increasing number of components of the angle feature map. Zero corresponds to original VLAD (not modulated). A codebook of 32 visual words is used.

vectors correspondingly. Such choices are not optimal while learning the PCA rotation matrix for RN or for dimensionality reduction, where we apply square-rooting. Any difference observed in the reported performance compared to our previous work [24] is attributed to an optimal power-law value used in this work and to the modified power-law normalization scheme.

6.3. *Impact of the parameters*

The impact of the angle modulation is controlled by the function k_θ parametrized by κ and N. As shown in Figure 2, the value κ typically controls the "bandwidth", *i.e.*, the range of $\Delta\theta$ values with non-zero response. The parameter N controls the quality of the approximation, and implicitly constrains the achievable bandwidth. It also determines the dimensionality of the output vector.

Figure 8 shows the impact of the selectivity of the kernel signature on the performance. As to be expected, there is a trade-off between defining too narrow or too large. The optimal performance is achieved with κ in the range $[2, 8]$.

Figure 9 shows the performance for increasing number of frequencies, which rapidly converges to a fixed mAP. This is the mAP of the exact evaluation of (7). We set $N = 3$ as a compromise between dimensionality expansion and performance. Therefore the modulation multiplies the input dimensionality by 7.

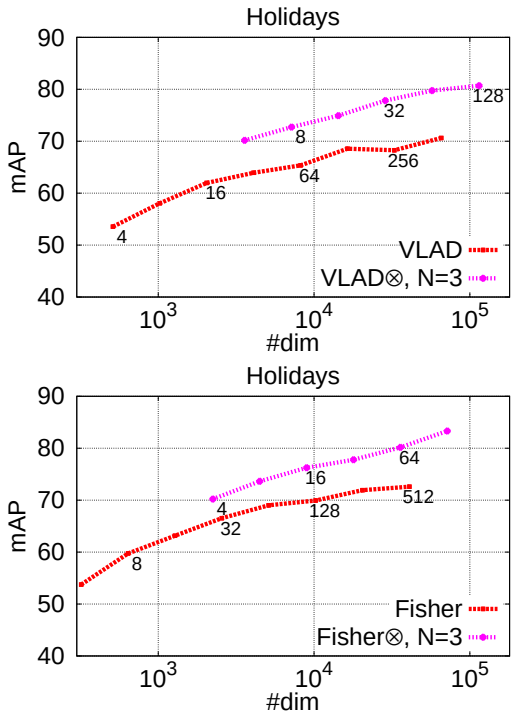

Figure 10: Impact of modulation on VLAD and Fisher: Performance versus dimensionality of the final vector for VLAD (top) and Fisher (bottom) compared to their modulated counterparts. Codebook size is shown with text labels. Results for Holidays dataset.

6.4. *Benefit of angular modulation*

Table 1 shows the benefit of modulation when applied to the monomial embeddings φ_1, φ_2 and φ_3. The results are on par with the recent coding techniques like VLAD or Fisher improved with modulation. We consider the obtained performance as one of our main achievements, because the representation is codebook-free and requires no learning. In addition, we further show the benefit of combining monomial embeddings with RN. This significantly boosts performance with the same vector dimensionality and negligible

Method	φ_1	$\varphi_1\otimes$			φ_2		$\varphi_2\otimes$				φ_3	$\varphi_3\otimes$
RN						×			×	×		
N	–	1	3	6	–	–	1	3	1	3	–	1
#dim	80	240	560	1,040	3,240	3,240	9,720	22,680	9,720	22,680	88,560	265,680
mAP	35.4	47.2	58.5	**62.5**	59.7	74.3	68.7	73.9	76.3	**79.7**	60.0	70.8

Table 1: Impact of modulation on monomial embeddings of order 1, 2 and 3. The performance is reported for Holidays dataset. RN = Rotation and Normalization.

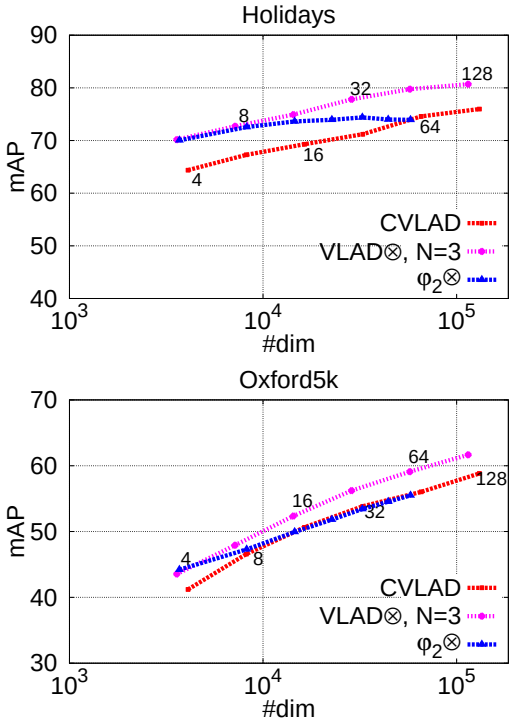

Figure 11: Comparison to CVLAD. We measure performance on Holidays and Oxford5k for CVLAD and our proposed methods for increasing codebook size. The codebook cardinality is shown with text labels for CVLAD and modulated VLAD, while for φ_2 it is the dimensionality of the input vectors after PCA reduction that we vary.

computational overhead.

To demonstrate the benefit of the proposed method, we compare VLAD, Fisher and monomial embeddings to their modulated counterparts. Figure 10 shows that modulation significantly improves the performance for the same codebook size. Given that the modulated vector is 7 times larger (with $N = 3$), the comparison focuses on the performance obtained with the same dimensionality. Even in this case, modulated VLAD⊗ and Fisher⊗ offer a significant improvement. We can conclude that it is better to increase the dimensionality by modulation than using a larger codebook.

6.5. Comparison to other methods

We compare our approach, in particular, to CVLAD, as this work also intends to integrate the dominant orientation into a vector representation. We consistently apply 8 query

Method	#C	#dim	RN	Holidays	Oxf5k	Oxf105k
VLAD [15]	64	4,096		55.6	37.8	-
Fisher [15]	64	4,096		59.5	41.8	-
VLAD [15]	256	16,384		58.7	-	-
Fisher [15]	256	16,384		62.5	-	-
Arandjelovic [28]	256	32,536		65.3	**55.8**	-
Delhumeau [49]	64	8,192		65.8	51.7	45.6
Zhao [22]	32	32,768		**68.8**	42.7	-
VLAD⊗	32	28,672		77.8	56.2	51.4
VLAD⊗	32	28,672	×	80.8	62.1	**53.8**
Fisher⊗	32	17,920		77.7	52.3	47.3
Fisher⊗	32	17,920	×	81.3	61.3	52.6
Fisher⊗	64	35,840	×	**83.6**	**65.1**	-
$\varphi_2\otimes$	n/a	22,680		73.9	51.8	45.7
$\varphi_2\otimes$	n/a	22,680	×	79.7	60.9	51.8
$\varphi_3\otimes$	n/a	265,680		70.8	55.0	-

Table 2: Performance comparison with state of the art approaches. Results with the use of full vector representation. #C: size of codebook. #dim: Number of components of each vector. Modulation is performed with $N = 3$ for all cases, except to φ_3, where $N = 1$. We do not use any re-ranking or spatial verification in any experiment. Results followed by a citation are the ones reported in the original publication.

rotations for both CVLAD and our method on Holidays dataset. Figure 11 shows the respective performance measured for different codebooks. The proposed methods appear to consistently outperform CVLAD, both for the same codebook and for the same dimensionality. Noticeably, the modulated embedded monomial $\varphi_2\otimes$ is on par with or better than CVLAD.

We also compare to other prior works and present results in Table 2 for Holidays, Oxford5k and Oxford105k. We outperform by a large margin the state of the art with full vector representations. Further, our approach is arguably

Method	#dim	full dim	dim→1024	dim→128
VLAD	4,096	40.3	37.3	26.0
VLAD⊗	28,672	53.8	**40.7** (+3.4)	**28.7** (+2.7)
Fisher	2,560	39.6	34.6	24.8
Fisher⊗	17,920	52.6	40.3 (+5.7)	27.6 (+2.8)
φ_2	3,240	37.2	32.4	21.3
$\varphi_2\otimes$	22,680	51.8	40.0 (+7.6)	26.8 (+5.5)

Table 3: Oxford105k: Performance comparison (mAP) after dimensionality reduction with PCA into 128 and 1024 components. The results with the full vector representation are with RN. Observe the consistent gain (in parentheses) brought by our approach for a *fixed* output dimensionality of 1,024 or 128 components.

Method	N	#C	#dim	Holidays^			Oxf5k		
VLAD	n/a	128	16,384	67.5			53.3		
VLAD	n/a	256	32,768	70.1			56.7		
Fisher	n/a	128	10,240	73.4			54.7		
Fisher	n/a	256	20,480	73.7			**57.6**		
χ =				u	v	u/v	u	v	u/v
VLAD$\otimes_\chi$	2	32	20,480	68.8	72.5	72.4	49.3	50.9	50.8
VLAD$\otimes_\chi$	3	32	28,672	69.8	74.3	74.4	50.4	52.5	53.2
Fisher$\otimes_\chi$	2	32	12,800	71.6	75.8	75.7	49.8	51.0	51.5
Fisher$\otimes_\chi$	3	32	17,920	71.5	**76.6**	76.5	50.9	52.1	53.3
χ =				u, v					
VLAD$\otimes_\chi$	2	16	51,200	66.4			47.2		
VLAD$\otimes_\chi$	3	8	50,176	68.5			46.9		
Fisher$\otimes_\chi$	2	16	32,000	75.7			52.1		
Fisher$\otimes_\chi$	3	8	31,360	73.1			51.2		

Table 4: Comparison of the translation covariant match kernel and baseline VLAD and Fisher vectors. Results are reported using the feature detector with the gravity vector assumption by Perdoch *et al.* [44]. Note that these results are not directly comparable to the ones of Table 2 due to different features and different versions of Holidays dataset (rotated vs original).

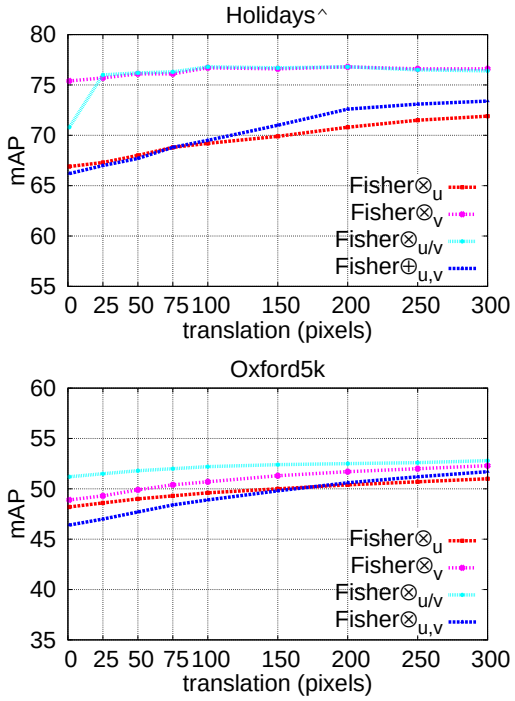

Figure 12: Impact of multiple translations to the performance of our translation covariant match kernel. Results reported on upright Holidays^ and Oxford5k. A codebook of 32 (8) visual words is used for single (double) modulation. Results are reported using the feature detector with the gravity vector assumption Perdoch *et al.* [44].

compatible with these concurrent approaches, which may bring further improvement. Note that RN also boosts performance for VLAD and Fisher. In particular with a codebook of size 32, they achieve 50.4 and 49.8 respectively on Oxford5k. Our scores on Holidays with Fisher$\otimes$ and RN are also competitive to those reported by state-of-the-art methods based on large codebooks [12]. To our knowledge, this is the first time that a vector representation compatible with inner product attains such image search performance.

On Oxford5k and Oxford105k we do not evaluate multiple query rotations for our method. A simple way to enforce up-right objects for our baseline methods is to use up-right features. Performance on Oxford5k achieved by VLAD with codebook of size 256 and with up-right features of the same detector is 53.4, while the corresponding score with rotation invariant features is 52.4. Even though switching off rotation when all objects are aligned seems to slightly increase performance, our method appears to be more effective (VLAD$\otimes$ achieves 56.2 with a codebook of size 32). Moreover, note that up-right features are not applicable when object rotation exists, while our rotation covariant match kernel is.

Table 3 reports the performance after dimensionality reduction to 128 or 1024 components. The same set of local features and codebooks are used for all methods. We observe a consistent improvement over the original encoding.

6.6. Position encoding

The selectivity parameter κ is set equal to 8, similarly to the case of dominant orientation. We evaluate the influence of multiple translations applied to the query image in order to identify the best alignment. In Figure 12 we present the impact of the translation on the performance. Performance improves as we evaluate multiple translations and saturates around 250 pixels. In the rest of our experiments we set the maximum translation value to 250 pixels.

Interestingly, on Holidays^ the vertical coordinate is significantly more informative than the horizontal one. This might be due to the nature of this dataset; images depicting the same landmark have been shot by the same person. Therefore, the objects/scenes have been photographed roughly at the same height, but with horizontal slides in a panoramic manner. On Oxford5k this is not the case.

We evaluate the position encoding for single and double modulation for comparable dimensionality (approximately 30K). The performance increases by taking into account more translations for both methods. Single modulation performs better than the double counterpart for the given dimensionality. The increase of dimensionality for the latter is high, but it uses a much smaller codebook.

In Table 4 we show results for the baseline VLAD, and Fisher vector, and their modulated counterparts with respect to position. By considering 2 instead of 3 frequencies the dimensionality of the modulated descriptor is reduced by 30% with just a small loss (1-2%). We note that, once more, comparing to the baseline the advantage relies on the fact

that the visual codebook is smaller and the assignment to that is faster to compute. Recall that now scale invariance is lost, leading to lower performance on Oxford5k. In the case of scale changes the dominant orientation is more distinctive and reliable. Overall, the orientation information brings higher improvement compared to the spatial one.

In order to provide a direct comparison between the two proposed methods we evaluate the translation covariant match kernel on Oxford5k and with the same features as those of the experiments reported in Table 2. VLAD$\otimes_u$ and VLAD$\otimes_v$ achieve 47.1 and 50.1 respectively by computing similarity for 25 translations on each direction with a codebook of size 32. Concerning the position modulated Fisher vectors the corresponding scores are 43.0 and 45.7. It appears that encoding rotation is more effective for this use-case. However, the translation covariant match kernel opens other possible directions, such as application on image classification, as a continuous alternative to spatial pyramid match.

6.7. Timings

The image representation created by modulating the monomial embedding φ_2 using $N = 3$ takes on average 68 ms for a typical image with 3,000 SIFT descriptors. The resulting aggregated vector representation has 22,680 components. The average query time with such a representation on Holidays is 5.8 ms, assuming no query rotation. Employing the trigonometric polynomial of scores results in 5.9 ms (6.1 ms) for 8 (64) possible fixed rotations. The corresponding timings for Oxford105k and vectors reduced to 128 dimensions are 55 ms (no rotations) and 134 ms (8 fixed rotations). In the case of reduced vectors, the query rotations are computed with the naive way. Note, these timings are better than those achieved by a Bag-of-Words representation with a large vocabulary, for which the quantization typically takes about 1 second with an approximate nearest neighbor search algorithm like FLANN [56]. Our timings are measured with a single threaded implementation on an Intel Xeon E5-4640@2.40GHz.

7. Conclusion

Our modulation strategy integrates geometric information directly in the coding stage. Dominant orientation of local features or their position is jointly encoded with the local descriptor, in a continuous manner. Our method is inspired by and builds upon recent works on explicit feature maps and kernel descriptors. Thanks to a generic formulation provided by match kernels, it is compatible with coding strategies such as Fisher vector or VLAD.

Invariance is offered by estimating maximum similarity for multiple image rotations or translations. The nature of our representation enables this very efficiently with a simple trigonometric polynomial. Our context (datasets) simply demands just a few sampling points on the rotation or translation domain. However, note that our methodology provides high efficiency even for denser search.

Our experiments demonstrate a consistent gain compared to the original coding in all cases. Angular modulation appears to be more promising than that of position for the task that we examine. Interestingly, our method is also very effective with a simple monomial kernel, offering competitive performance for image search with a coding stage not requiring any quantization.

Whatever the coding stage that we use with our approach, the resulting representation is compared with inner product, which suggests that it is compliant with linear classifiers such as those considered in image classification.

Acknowledgments. This work was supported by ERC grant VIAMASS no. 336054 and ANR project Fire-ID.

References

[1] D. Lowe, Distinctive image features from scale-invariant keypoints, IJCV 60 (2) (2004) 91–110.
[2] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, SURF: Speeded up robust features, Computer Vision and Image Understanding 110 (3) (2008) 346–359.
[3] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, Trans. PAMI 24 (7) (2002) 971–987.
[4] J. Sivic, A. Zisserman, Video Google: A text retrieval approach to object matching in videos, in: ICCV, 2003.
[5] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: CVPR, 2007.
[6] G. Tolias, Y. Avrithis, Speeded-up, relaxed spatial matching, in: ICCV, 2011.
[7] O. Chum, J. Philbin, J. Sivic, M. Isard, A. Zisserman, Total recall: Automatic query expansion with a generative feature model for object retrieval, in: ICCV, 2007.
[8] G. Tolias, H. Jégou, Visual query expansion with or without geometry: refining local descriptors by feature aggregation, Pattern Recognition.
[9] H. Jégou, M. Douze, C. Schmid, Improving bag-of-features for large scale image search, IJCV 87 (3) (2010) 316–336.
[10] A. Mikulík, M. Perdoch, O. Chum, J. Matas, Learning a fine vocabulary, in: ECCV, 2010.
[11] H. Jégou, M. Douze, C. Schmid, On the burstiness of visual elements, in: CVPR, 2009.
[12] G. Tolias, Y. Avrithis, H. Jégou, To aggregate or not to aggregate: Selective match kernels for image search, in: ICCV, 2013.
[13] J. Wang, J. Yang, F. L. K. Yu, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: CVPR, 2010.
[14] F. Perronnin, C. R. Dance, Fisher kernels on visual vocabularies for image categorization, in: CVPR, 2007.
[15] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, C. Schmid, Aggregating local descriptors into compact codes, in: Trans. PAMI, 2012.
[16] F. Perronnin, Y. Liu, J. Sánchez, H. Poirier, Large-scale image retrieval with compressed Fisher vectors, in: CVPR, 2010.
[17] M. Charikar, Similarity estimation techniques from rounding algorithms, in: STOC, 2002.
[18] H. Jégou, M. Douze, C. Schmid, Product quantization for nearest neighbor search, Trans. PAMI 33 (1) (2011) 117–128.
[19] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: CVPR, 2006.

[20] M. Douze, H. Jégou, H. Singh, L. Amsaleg, C. Schmid, Evaluation of GIST descriptors for web-scale image search, in: CIVR, 2009.
[21] P. Koniusz, F. Yan, K. Mikolajczyk, Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection, Computer Vision and Image Understanding 17 (5) (2013) 479–492.
[22] W. Zhao, H. Jégou, G. Gravier, Oriented pooling for dense and non-dense rotation-invariant features, in: BMVC, 2013.
[23] L. Bo, C. Sminchisescu, Efficient match kernel between sets of features for visual recognition, in: NIPS, 2009.
[24] G. Tolias, T. Furon, H. Jégou, Orientation covariant aggregation of local descriptors with embeddings, in: ECCV, 2014, pp. 382–397.
[25] H. Jégou, M. Douze, C. Schmid, Hamming embedding and weak geometric consistency for large scale image search, in: ECCV, 2008.
[26] L. Bo, X. Ren, D. Fox, Kernel descriptors for visual recognition, in: NIPS, 2010.
[27] A. Vedaldi, A. Zisserman, Efficient additive kernels via explicit feature maps, Trans. PAMI 34 (3) (2012) 480–492.
[28] R. Arandjelovic, A. Zisserman, All about VLAD, in: CVPR, 2013.
[29] J. Krapac, J. Verbeek, F. Jurie, Modeling spatial layout with fisher vectors for image categorization, in: ICCV, IEEE, Barcelona, Spain, 2011, pp. 1487–1494.
[30] P. Koniusz, K. Mikolajczyk, Spatial coordinate coding to reduce histogram representations, dominant angle and colour pyramid match, in: ICIP, 2011, pp. 661–664.
[31] J. Sánchez, F. Perronnin, T. de Campos, Modeling the spatial layout of images beyond spatial pyramids, Pattern Recognition Letters 33 (16) (2012) 2216 – 2223.
[32] P.-H. Gosselin, N. Murray, H. Jégou, F. Perronnin, Revisiting the Fisher vector for fine-grained classification, Pattern Recognition Letters 49 (2014) 92–98.
[33] L. Bo, K. Lai, X. Ren, D. Fox, Object recognition with hierarchical kernel descriptors, in: CVPR, 2011.
[34] S. Lyu, Mercer kernels for object recognition with local features, in: CVPR, 2005.
[35] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: ECCV Workshop Statistical Learning in Computer Vision, 2004.
[36] D. Picard, P.-H. Gosselin, Efficient image signatures and similarities using tensor products of local descriptors, Computer Vision and Image Understanding 117.
[37] P. Koniusz, F. Yan, P.-H. Gosselin, K. Mikolajczyk, Higher-order occurrence pooling on mid-and low-level features: Visual concept detection, Tech. rep., INRIA (Dec. 2013).
[38] J. C. R. Caseiro, J. Batista, C. Sminchisescu, Semantic segmentation with second-order pooling, in: ECCV, 2012.
[39] M. Abramowitz, I. A. Stegun, Handbook of mathematical functions with formulas, graphs, and mathematical tables, Vol. 55 of National Bureau of Standards Applied Mathematics Series, U.S. Government Printing Office, 1964.
[40] E. G. Bazavan, F. Li, , C. Sminchisescu, Fourier kernel learning, in: ECCV, 2012.
[41] F. Perronnin, J. Sánchez, T. Mensink, Improving the Fisher kernel for large-scale image classification, in: ECCV, 2010.
[42] Y. Zhang, Z. Jia, T. Chen, Image retrieval with geometry-preserving visual phrases, in: CVPR, 2011, pp. 809–816.
[43] J. F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters, Trans. PAMI.
[44] M. Perdoch, O. Chum, J. Matas, Efficient representation of local geometry for large scale object retrieval, in: CVPR, 2009.
[45] T. Jaakkola, D. Haussler, Exploiting generative models in discriminative classifiers, in: NIPS, 1998.
[46] K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, The devil is in the details: an evaluation of recent feature encoding methods, in: BMVC, 2011.
[47] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. V. Gool, A comparison of affine region detectors, IJCV 65 (1/2) (2005) 43–72.
[48] R. Arandjelovic, A. Zisserman, Three things everyone should know to improve object retrieval, in: CVPR, 2012.
[49] J. Delhumeau, P.-H. Gosselin, H. Jégou, P. Pérez, Revisiting the VLAD image representation, in: ACM Multimedia, 2013.
[50] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Lost in quantization: Improving particular object retrieval in large scale image databases, in: CVPR, 2008.
[51] M. Douze, H. Jégou, The Yael library, in: ACM Multimedia, 2014.
[52] A. Torii, J. Sivic, T. Pajdla, M. Okutomi, Visual place recognition with repetitive structures, in: CVPR, 2013.
[53] H. Jégou, O. Chum, Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening, in: ECCV, 2012.
[54] B. Safadi, G. Quenot, Descriptor optimization for multimedia indexing and retrieval, in: CBMI, 2013.
[55] H. Jégou, A. Zisserman, et al., Triangulation embedding and democratic aggregation for image search, in: CVPR, 2014.
[56] M. Muja, D. G. Lowe, Fast approximate nearest neighbors with automatic algorithm configuration, in: VISAPP, 2009.