Orientation covariant aggregation of local descriptors with embeddings

Giorgos Tolias, Teddy Furon, Hervé Jégou

Orientation covariant aggregation of local descriptors with embeddings

Giorgos Tolias, Teddy Furon & Hervé Jégou

Inria

Abstract. Image search systems based on local descriptors typically achieve orientation invariance by aligning the patches on their dominant orientations. Albeit successful, this choice introduces too much invariance because it does not guarantee that the patches are rotated consistently. This paper introduces an aggregation strategy of local descriptors that achieves this covariance property by jointly encoding the angle in the aggregation stage in a continuous manner. It is combined with an efficient monomial embedding to provide a codebook-free method to aggregate local descriptors into a single vector representation.
Our strategy is also compatible and employed with several popular encoding methods, in particular bag-of-words, VLAD and the Fisher vector. Our geometric-aware aggregation 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.

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] for the low-level description of the images. In particular, approaches derived from the bag-of-visual-words framework [3] are especially successful to solve problems like recognizing buildings. They are typically combined with spatial verification [4] or other re-ranking strategies such as query expansion [5].

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 [6], by learning a fine vocabulary [7], or weighting the distances [8, 9]. In addition to the SIFT, it is useful to employ some geometrical information associated with the region of interest [6]. 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 relatively large vocabulary causes delays.

Recently, very short yet effective representations have been proposed based on alternative encoding strategies, such as local linear coding [10], the Fisher vector [11] or VLAD [12]. 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 [13] with cosine sketches, such as those proposed by Charikar [14] (*a.k.a.* LSH), or aggressively compressed with principal component dimensionality reduction (PCA) to very short vectors. Product quantization [15] is another example achieving a very compact representation of a few dozens to hundreds bytes and an efficient search because the comparison is done directly 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 [16], which is widely adopted for image classification. However, it is ineffective for particular image and object retrieval as the grid is too rigid and the resulting representation is not invariant enough, as shown by Douze *et al.* [17].

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 to that offered by Weak Geometry Consistency (WGC) [6], but directly implemented in the coding stage of image vector representations like Fisher, or VLAD. Some recent works in classification [18] and image search [19] consider a similar objective. They suffer from several shortcomings. In particular, they simply quantize the angle and use it as a pooling variable. Moreover the encoding of a rough approximation of the angles 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 [20]. Our method is inspired by the kernel descriptor of Bo *et al.* [21], from which we borrow the idea of angle kernelization. Our method however departs from this work in several ways. First, we are interested in aggregating local descriptors to produce a vector image representation, whereas they construct new local descriptors. Second, we do not encode the gradient orientation but the dominant orientation of the region of interest jointly with the corresponding SIFT descriptor, in order to achieve the covariant property of the local patches. Finally, we rely on explicit feature maps [22] to encode the angle, which provides a much better approximation than efficient match kernel for a given number of components.

This paper is organized as follows. Section 2 introduces notation and discusses some important related works more in details. Our approach is presented in Section 3 and evaluated in Section 4 on several popular benchmarks for image search, namely Oxford5k [4], Oxford105k and Inria Holidays [23]. These experiments show that our approach gives a significant improvement over the state of the art on image search with vector representations. Importantly, we achieve competitive results by combining our approach with monomial embeddings, *i.e.*, with a *codebook-free* approach, as opposed to coding approaches like VLAD.

2 Preliminaries: match kernels and monomial 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 [24, 20, 6], 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 typical 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

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

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 framework encompasses many approaches such as bag-of-words [3, 25], LLC [10], Fisher vector [11], VLAD [12], or VLAT [26]. Note that some non-linear processing, such as power-law component-wise normalization [8, 27], is often applied to the resulting vector. 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 [24, 26, 9] 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 recently for semantic segmentation [28] or in image classification [26]. In this case, the kernel is equivalent to comparing the set of features based on their covariance matrix [26]. 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

$$\varphi_2(\mathbf{x}) = [x_1^2, \dots, x_d^2, x_1x_2\sqrt{2}, \dots, x_{d-1}x_d\sqrt{2}]^\top. \tag{5}$$

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

$$\varphi_3(\mathbf{x}) = [x_1^3, \dots, x_d^3, x_1^2x_2\sqrt{3}, \dots, x_d^2x_{d-1}\sqrt{3}, x_1x_2x_3\sqrt{6}, \dots, x_{d-2}x_{d-1}x_d\sqrt{6}]^\top. \tag{6}$$

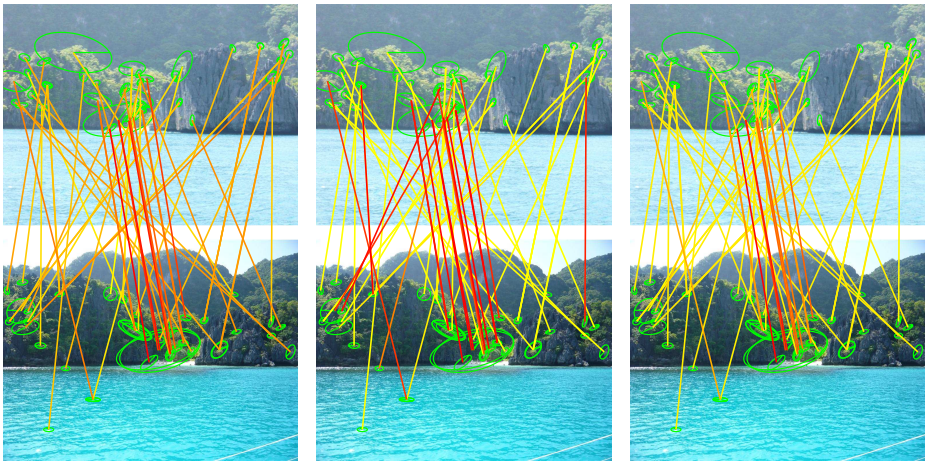

Fig. 1. Similarities between regions of interest, based on SIFT kernel k (*left*), angle consistency kernel k_θ (*middle*) and both (*right*). For each local region, we visualize the values $k(\mathbf{x}, \mathbf{y})$, $k_\theta(\Delta\theta)$ and their product by the colors of the link (red=1).

3 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

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

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

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 [6] 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 [21] 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/pooling stage and employ a better approximation technique, namely explicit feature maps [22], to encode $\mathcal{X}^\star$. This section first explains the feature map of the angle, then how it modulates the descriptors, and finally discusses the match kernel design and properties.

3.1 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 $[0, \pi]$. We consider in particular the following function:

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

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. Although this is not explicitly stated in their paper, the regular Von Mises distribution is the kernel function implicitly used by Bo *et al.* [21] for kernelizing 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 [29][Eq. (9.6.19)]):

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

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:

$$\bar{k}^N_{\mathsf{VM}}(\Delta\theta) = \sum_{n=0}^{N} \gamma_n \cos(n\Delta\theta) \quad \text{with } \gamma_0 = \frac{(I_0(\kappa) - e^{-\kappa})}{2 \sinh(\kappa)} \text{ and } \gamma_n = \frac{I_n(\kappa)}{\sinh(\kappa)} \text{ if } n > 0. \tag{11}$$

We design the feature map $\boldsymbol{\alpha}(\theta)$ as follows:

$$\boldsymbol{\alpha}(\theta) = (\sqrt{\gamma_0}, \sqrt{\gamma_1}\cos(\theta), \ldots, \sqrt{\gamma_N}\cos(N\theta), \sqrt{\gamma_1}\sin(\theta), \ldots, \sqrt{\gamma_N}\sin(N\theta))^\top. \tag{12}$$

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

$$\boldsymbol{\alpha}(\theta_1)^\top \boldsymbol{\alpha}(\theta_2) = \gamma_0 + \sum_{n=1}^{N} \gamma_n (\cos(n\theta_1)\cos(n\theta_2) + \sin(n\theta_1)\sin(n\theta_2)) \tag{13}$$

$$= \sum_{n=0}^{N} \gamma_n \cos(n(\theta_1 - \theta_2)) \tag{14}$$

$$= \bar{k}^N_{\mathsf{VM}}(\theta_1 - \theta_2) \approx k_{\mathsf{VM}}(\theta_1 - \theta_2) \tag{15}$$

This process of designing a feature map is explained in full details by Vedaldi and Zisserman [22]. This feature map gives an approximation of the target function k_{VM}, which is more accurate as N is bigger.

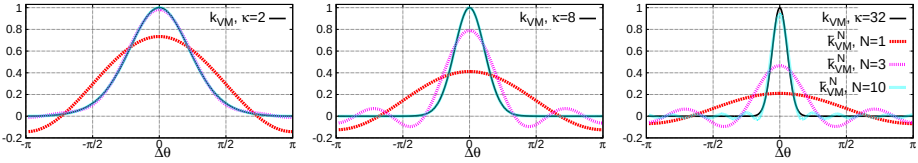

Fig. 2. Function k_{VM} for different values of κ and its approximation $\bar{k}^N_{\mathsf{VM}}$ using 1, 3 and 10 frequencies, as implicitly defined by the corresponding mapping $\boldsymbol{\alpha} : [\pi, \pi] \rightarrow \mathbb{R}^{2N+1}$.

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 N) to obtain an accurate estimation. Importantly, the approximation stemming from this explicit angle mapping is better than that based on efficient match kernels [20], 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 4). Accordingly, we use $\bar{k}^3_{\mathsf{VM}}$ as k_θ in the sequel. The corresponding embedding $\boldsymbol{\alpha} : \mathbb{R} \rightarrow \mathbb{R}^7$ maps any angle to a 7-dimensional vector.

Remark: Instead of approximating a kernel on angles with finite Fourier series, one may rather consider directly designing a function satisfying our initial requirements (pseudo-period, symmetric, decreasing over $[0, \pi]$), such as

$$k_P(\Delta\theta) = \cos(\Delta\theta/2)^P \text{ with } P \text{ even.} \tag{16}$$

This function, thanks to power reduction trigonometric identities for even P, is re-written as

$$k_P(\Delta\theta) = \sum_{p=0}^{P/2} \gamma_p \cos(p\Delta\theta) \tag{17}$$

$$\text{with} \qquad \gamma_0 = \frac{1}{2^P}\binom{P}{P/2}, \gamma_p = \frac{1}{2^{P-1}}\binom{P}{P/2-p} \quad 0 < p \leq P/2. \tag{18}$$

Applying (12) 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 the targeted task, in the spirit of recent works on Fourier kernel learning [30].

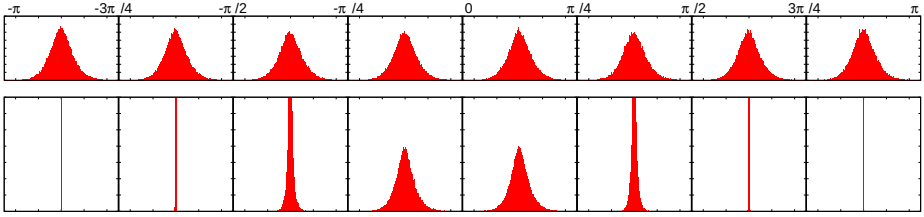

Fig. 3. Distribution of patch similarity for different values of orientation difference. In this figure, we split the angular space into 8 equally-sized bins and present the similarity distribution separately for each of these bins. Horizontal axis represents the similarity value between matching features.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

3.2 Modulation and covariant match kernel

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

$$m(\mathbf{x}, \boldsymbol{\alpha}(\theta)) = \mathbf{x} \otimes \boldsymbol{\alpha}(\theta) = (x_1 \boldsymbol{\alpha}(\theta)^\top, x_2 \boldsymbol{\alpha}(\theta)^\top \ldots, x_d \boldsymbol{\alpha}(\theta)^\top)^\top. \tag{19}$$

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 product space $\mathbb{R}^{(2N+1)D}$, the following holds:

$$\begin{aligned} m(\mathbf{x}, \boldsymbol{\alpha}(\theta_x))^\top m(\mathbf{y}, \boldsymbol{\alpha}(\theta_y) &= (\mathbf{x} \otimes \boldsymbol{\alpha}(\theta_x))^\top (\mathbf{y} \otimes \boldsymbol{\alpha}(\theta_y)) \\ &= (\mathbf{x}^\top \otimes \boldsymbol{\alpha}(\theta_x)^\top)(\mathbf{y} \otimes \boldsymbol{\alpha}(\theta_y)) = (\mathbf{x}^\top \mathbf{y}) \otimes (\boldsymbol{\alpha}(\theta_x)^\top \boldsymbol{\alpha}(\theta_y)) \\ &= (\mathbf{x}^\top \mathbf{y}) k_\theta(\theta_x - \theta_y). \end{aligned} \tag{20}$$

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

$$\begin{aligned} K^\star(\mathcal{X}^\star, \mathcal{Y}^\star) =& \beta(\mathcal{X}^\star)\beta(\mathcal{Y}^\star) \sum_{(\mathbf{x},\theta_x)\in\mathcal{X}^\star} \sum_{(\mathbf{y},\theta_y)\in\mathcal{Y}^\star} m(\varphi(\mathbf{x}), \boldsymbol{\alpha}(\theta_x))^\top m(\varphi(\mathbf{y}), \boldsymbol{\alpha}(\theta_y)), \\ =& \beta(\mathcal{X}^\star) \left(\sum_{(\mathbf{x},\theta_x)} m(\varphi(\mathbf{x}), \boldsymbol{\alpha}(\theta_x)) \right)^\top \beta(\mathcal{Y}^\star) \left(\sum_{(\mathbf{y},\theta_y)} m(\varphi(\mathbf{y}), \boldsymbol{\alpha}(\theta_y) \right), \end{aligned} \tag{21}$$

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

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

$$\mathbf{X}^\star = \beta(\mathcal{X}^\star) \sum_{(\mathbf{x},\theta_x)\in\mathcal{X}^\star} m(\varphi(\mathbf{x}), \boldsymbol{\alpha}(\theta_x)). \tag{22}$$

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.

3.3 Rotation invariance

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 $\mathbf{X}_{n,s}^\star$, $1 \le n \le N$, correspond to the cosine and sine terms, respectively.

Imagine now that this image undergoes a global rotation of angle θ. Denote $\breve{\mathcal{X}}$ the new set of pairs $(\mathbf{x}, \breve{\theta}_x)$ with $\breve{\theta}_x = \theta_x - \theta$, and $\breve{\mathbf{X}}^\star$ is 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

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

$$\breve{\mathbf{X}}_{n,s}^\star = -\mathbf{X}_{n,c}^\star \sin n\theta + \mathbf{X}_{n,s}^\star \cos n\theta. \tag{24}$$

This in turn shows that $\|\breve{\mathbf{X}}^\star\| = \|\mathbf{X}^\star\|$. Therefore the rotation has no effect on the normalization factor $\beta(\mathcal{X}^\star)$.

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

$$\langle \breve{\mathbf{X}}^\star | \mathbf{Y}^\star \rangle = \langle \mathbf{X}_0^\star | \mathbf{Y}_0^\star \rangle + \sum_{n=1}^{N} \cos n\theta \ \left(\langle \mathbf{X}_{n,c}^\star | \mathbf{Y}_{n,c}^\star \rangle + \langle \mathbf{X}_{n,s}^\star | \mathbf{Y}_{n,s}^\star \rangle \right) \tag{25}$$

$$+ \sum_{n=1}^{N} \sin n\theta \ \left(-\langle \mathbf{X}_{n,c}^\star | \mathbf{Y}_{n,s}^\star \rangle + \langle \mathbf{X}_{n,s}^\star | \mathbf{Y}_{n,c}^\star \rangle \right). \tag{26}$$

Here, we stress that the similarity between two images is a real trigonometric polynomial in θ (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 score between the two images.

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 want to 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).

Remark: 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 [27], except for the subvector $\mathbf{X}_0^\star$. We propose two possible options to overcome this problem.

1. The naive strategy is to compute the query 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. We use this simpler approach in the experimental section.
2. Alternately, the power-law normalization is adapted 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. Experimentally, this strategy is as effective as the naive option.

4 Experiments

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 [23] and Oxford Buildings 5k [4]. 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. The retrieval performance is measured with mean Average Precision (mAP) [4].

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 [12], which encodes a SIFT descriptor by considering the residual vector to the centroid.
- The Fisher vector [11, 27, 31]. For image classification, Chatfield *et al.* [32] show that it outperforms concurrent coding techniques, in particular LLC [10]. We adopt the standard choice for image retrieval and use only the gradient with respect to the mean [12].

- 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 our approach with the symbol "$\otimes$": VLAD$\otimes$, Fisher$\otimes$, $\varphi_1\otimes$, $\varphi_2\otimes$ and $\varphi_3\otimes$, correspondingly. In addition, we compare against the most related work, namely the recent CVLAD [19] 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 (improved) implementations of VLAD, Fisher and CVLAD, so that the results are directly comparable with the same features.

4.1 Implementation Details

Local descriptors. We use the Hessian-Affine detector [33] to extract the regions of interest, that are subsequently described by SIFT descriptors [1] post-processed with RootSIFT [34]. Then, following the pre-processing required for the Fisher vector [11, 27, 12], we apply PCA to reduce the vector to 80 components. 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 [35], without dimensionality reduction. The resulting vector is subsequently ℓ_2-normalized.

The improved Hessian-Affine detector of Perdoch *et al.* [36] improves the retrieval performance. However, we do not use it, since it ignores rotations by making the gravity vector assumption. Instead, we use the original detector modified so that it has similar parameters (patch size set to 41).

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 [37] for Oxford buildings. We rely on the Yael library [38] for codebook construction and VLAD and Fisher encoding.

Post-processing. The final image vector obtained by each method is power-law normalized [8, 27, 12]. 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.* [39]. With our angle modulation, this post-processing inherently captures and down weights patches with similar dominant orientation. The power-law exponent is set to 0.4 for Fisher and VLAD and to 0.2 for monomial embeddings. These values give best or close-to-best performance for the initial representations. The resulting vector is ℓ_2-normalized.

In addition to power-law normalization, we rotate the aggregated vector representation with a PCA rotation matrix [40, 41]. This aims at capturing the co-occurrences to down-weight them either by whitening [40] or a second power-law normalization [41]. We adopt the latter choice (with exponent 0.5) to avoid

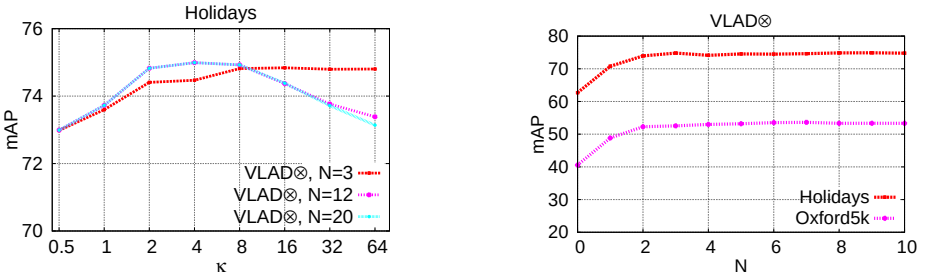

Fig. 4. Left: Performance on Holidays dataset of modulated VLAD for different values of κ and for different approximations. Right: Performance comparison of modulated VLAD for increasing number of components of the angle feature map. Zero corresponds to original VLAD (not modulated). A codebook of 32 visual words is used.

the sensitivity to eigenvalues (in whitening) when learning PCA with few input data. We refer to this Rotation and Normalization as RN in our experiments.

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 and RN, we apply rotations of the query image and apply individual queries as described in Section 3 (option 1). We apply 8 query rotations on Holidays dataset. On Oxford5k, we rather adopt the common choice of not considering other possible orientations: Possible rotation of the query object is usually not considered since all the images are up-right.

4.2 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 "bandwitdh", *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 4 (left) shows the impact of these parameters 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 4 (right) 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.

4.3 Benefit of our approach

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

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	3240	3,240	9,720	22,680	9,720	22,680	88,560	265,680
mAP	35.4	48.9	59.5	**63.2**	59.7	71.6	68.8	73.7	75.3	**79.9**	60.0	72.5

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.

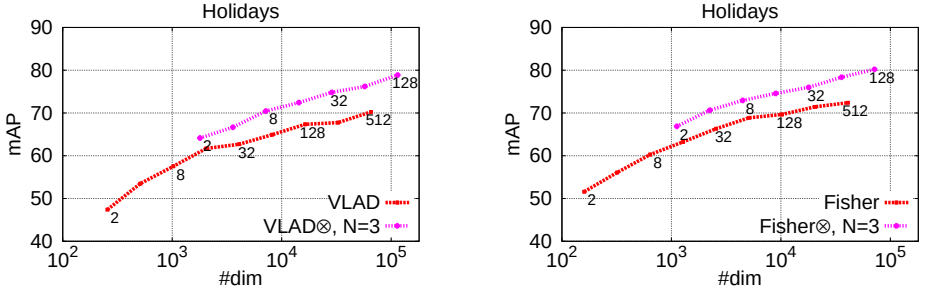

Fig. 5. Impact of modulation on VLAD and Fisher: Performance versus dimensionality of the final vector for VLAD (left) and Fisher (right) compared to their modulated counterparts. Codebook size is shown with text labels. Results for Holidays dataset.

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 computational overhead.

We compare VLAD, Fisher and monomial embeddings to their modulated counterparts. Figure 5 shows that modulation significantly improves the performance for the same codebook size. However, given that the modulated vector is $\times 7$ larger (with $N = 3$), the comparison focuses on the performance obtained with the same dimensionality. Even in this case, modulated VLAD$\otimes$ and Fisher$\otimes$ offer a significant improvement. We can conclude that it is better to increase the dimensionality by modulation than using a larger codebook.

4.4 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 rotations for both CVLAD and our method on Holidays dataset. Figure 6 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 further conduct experiments using oriented dense [19] to compare VLAD$\otimes$ to CVLAD. They achieve 87.2 and 86.5 respectively, on Holidays with codebook of size 512. This score is significantly higher than the one reported in [19]. Corresponding scores on Oxford5k are 50.5 and 50.7, respectively. However, note that it is very costly to densely extract patches aligned with dominant orientation.

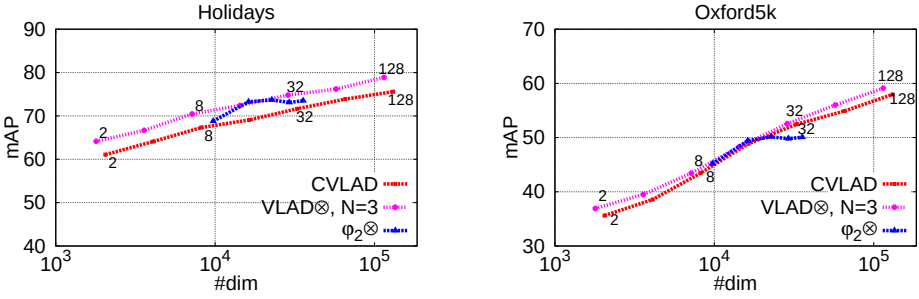

Fig. 6. Comparison to CVLAD. We measure performance on Holidays and Oxford5k for CVLAD and our proposed methods for increasing codebook size. The codebook cardinality is shown with text labels for CVLAD and modulated VLAD, while for φ_2 the number of frequency components (N) used are increased from 1 to 5.

Method	#C	#dim	RN	Holidays	Oxford5k	Oxford105k
VLAD [12]	64	4,096		55.6	37.8	-
Fisher [12]	64	4,096		59.5	41.8	-
VLAD [12]	256	16,384		58.7	-	-
Fisher [12]	256	16,384		62.5	-	-
Arandjelovic [42]	256	32,536		65.3	**55.8**	-
Delhumeau [35]	64	8,192		65.8	51.7	45.6
Zhao [19]	32	32,768		**68.8**	42.7	-
VLAD⊗	32	28,672		74.8	52.5	46.3
VLAD⊗	32	28,672	×	81.0	61.8	**53.9**
Fisher⊗	32	17,920		76.0	51.0	44.9
Fisher⊗	32	17,920	×	81.2	60.7	52.2
Fisher⊗	64	35,840	×	**84.1**	**64.8**	-
φ_2⊗	n/a	22,680		73.7	50.1	44.3
φ_2⊗	n/a	22,680	×	79.9	60.5	51.9
φ_3⊗	n/a	265,680		72.5	53.5	-

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. VLAD⊗ achieves **87.2** on Holidays and 50.5 on Oxford5k with $\#C$=512 and oriented dense.

Method	#dim	full dim	dim→1024	dim→128
VLAD	4,096	40.3	34.7	24.0
VLAD⊗	28,672	53.9	**40.7** (+7.0)	**27.5** (+3.5)
Fisher	2,560	39.3	37.3	25.2
Fisher⊗	17,920	52.2	39.9 (+2.6)	26.5 (+1.3)
φ_2	3,240	35.8	31.1	20.4
φ_2⊗	22,680	51.9	37.7 (+6.6)	24.0 (+3.6)

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.

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 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 50.0 and 48.6 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 [9]. To our knowledge, this is the first time that a vector representation compatible with inner product attains such image search performance.

On Oxford5k we do not evaluate multiple query rotations for our method. A simple way to enforce up-right objects for baseline methods is to use up-right features. Performance of VLAD with codebook of size 256 decreases from 51.3 to 49.4 by doing so, presumably because of small object rotations.

Finally, 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.

4.5 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 using cosine similarity on Oxford5k is 44 ms assuming no query rotation and 257 ms with the use of 8 possible fixed rotations (with the naive strategy discussed in Section 3.3). The corresponding timings for Oxford105k and vectors reduced to 128 dimensions are 55 ms and 134 ms, respectively. Note, these timings are better than those achieved by a bag-of-words representation with a large vocabulary, for which the quantization typically takes above 1 second with an approximate nearest neighbor search algorithm like FLANN [43].

5 Conclusion

Our modulation strategy integrates the dominant orientation directly in the coding stage. It 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. Our experiments demonstrate that it gives a consistent gain compared to the original coding in all cases, even after dimensionality reduction. Interestingly, it 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*, vol. 60, pp. 91–110, Nov. 2004.
2. H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," *Computer Vision and Image Understanding*, vol. 110, pp. 346–359, May 2008.
3. J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *ICCV*, Oct. 2003.
4. J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*, Jun. 2007.
5. 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.
6. H. Jégou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *IJCV*, vol. 87, pp. 316–336, Feb. 2010.
7. A. Mikulík, M. Perdoch, O. Chum, and J. Matas, "Learning a fine vocabulary," in *ECCV*, Sep. 2010.
8. H. Jégou, M. Douze, and C. Schmid, "On the burstiness of visual elements," in *CVPR*, Jun. 2009.
9. G. Tolias, Y. Avrithis, and H. Jégou, "To aggregate or not to aggregate: Selective match kernels for image search," in *ICCV*, Dec. 2013.
10. J. Wang, J. Yang, F. L. K. Yu, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR*, Jun. 2010.
11. F. Perronnin and C. R. Dance, "Fisher kernels on visual vocabularies for image categorization," in *CVPR*, Jun. 2007.
12. H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local descriptors into compact codes," in *Trans. PAMI*, Sep. 2012.
13. F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale image retrieval with compressed Fisher vectors," in *CVPR*, Jun. 2010.
14. M. Charikar, "Similarity estimation techniques from rounding algorithms," in *STOC*, May 2002.
15. H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *Trans. PAMI*, vol. 33, pp. 117–128, Jan. 2011.
16. S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," in *CVPR*, Jun. 2006.
17. M. Douze, H. Jégou, H. Singh, L. Amsaleg, and C. Schmid, "Evaluation of GIST descriptors for web-scale image search," in *CIVR*, July 2009.
18. P. Koniusz, F. Yan, and K. Mikolajczyk, "Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection," *Computer Vision and Image Understanding*, vol. 17, pp. 479–492, May 2013.
19. W. Zhao, H. Jégou, and G. Gravier, "Oriented pooling for dense and non-dense rotation-invariant features," in *BMVC*, Sep. 2013.
20. L. Bo and C. Sminchisescu, "Efficient match kernel between sets of features for visual recognition," in *NIPS*, Dec. 2009.
21. L. Bo, X. Ren, and D. Fox, "Kernel descriptors for visual recognition," in *NIPS*, Dec. 2010.
22. A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *Trans. PAMI*, vol. 34, pp. 480–492, Mar. 2012.

23. H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *ECCV*, Oct. 2008.
24. S. Lyu, "Mercer kernels for object recognition with local features," in *CVPR*, Jun. 2005.
25. G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *ECCV Workshop Statistical Learning in Computer Vision*, May 2004.
26. D. Picard and P.-H. Gosselin, "Efficient image signatures and similarities using tensor products of local descriptors," *Computer Vision and Image Understanding*, vol. 117, Jun. 2013.
27. F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *ECCV*, Sep. 2010.
28. J. C. R. Caseiro, J. Batista, and C. Sminchisescu, "Semantic segmentation with second-order pooling," in *ECCV*, Oct. 2012.
29. M. Abramowitz and 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.
30. E. G. Bazavan, F. Li, , and C. Sminchisescu, "Fourier kernel learning," in *ECCV*, Oct. 2012.
31. T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *NIPS*, Dec. 1998.
32. K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *BMVC*, Sep. 2011.
33. K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *IJCV*, vol. 65, pp. 43–72, Nov. 2005.
34. R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," in *CVPR*, Jun. 2012.
35. J. Delhumeau, P.-H. Gosselin, H. Jégou, and P. Pérez, "Revisiting the VLAD image representation," in *ACM Multimedia*, Oct. 2013.
36. M. Perdoch, O. Chum, and J. Matas, "Efficient representation of local geometry for large scale object retrieval," in *CVPR*, Jun. 2009.
37. 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.
38. M. Douze and H. Jégou, "The Yael library," in *ACM Multimedia*, Nov. 2014.
39. A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual place recognition with repetitive structures," in *CVPR*, Jun. 2013.
40. H. Jégou and O. Chum, "Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening," in *ECCV*, Oct. 2012.
41. B. Safadi and G. Quenot, "Descriptor optimization for multimedia indexing and retrieval," in *CBMI*, Jun. 2013.
42. R. Arandjelovic and A. Zisserman, "All about VLAD," in *CVPR*, Jun. 2013.
43. M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *VISAPP*, Feb. 2009.