Kernel Local Descriptors with Implicit Rotation Matching

Andrei Bursuc, Giorgos Tolias, Hervé Jégou

Kernel Local Descriptors with Implicit Rotation Matching

Andrei Bursuc, Giorgos Tolias, and Hervé Jégou
Inria
{firstname.lastname}@inria.fr

ABSTRACT

In this work we design a kernelized local feature descriptor and propose a matching scheme for aligning patches quickly and automatically. We analyze the SIFT descriptor from a kernel view and identify and reproduce some of its underlying benefits. We overcome the quantization artifacts of SIFT by encoding pixel attributes in a continuous manner via explicit feature maps. Experiments performed on the patch dataset of Brown *et al.* [3] show the superiority of our descriptor over methods based on supervised learning.

Categories and Subject Descriptors

I.4.10 [**Image Processing and Computer Vision**]: Image Representation

General Terms

Algorithms and Experimentation

Keywords

local descriptor; kernel descriptor; rotation invariance

1. INTRODUCTION

In this paper we deal with the design of a local descriptor, which is one of the fundamental problems in computer vision. A variety of tasks, such as image classification, retrieval, detection, and registration, rely on descriptors derived from local patches. The local patches can be originated from dense sampling [5] or from a sparse feature detector [7] which is co-variant to certain image transformations.

A variety of local descriptors builds upon image gradients. A standard choice is to build a histogram of gradient orientations, as in SIFT [7]. Another example concerns methods that rely on pixel intensities, *e.g.* by pairwise comparisons [4]. Learning [3] based on pairs of similar and non-similar patches is employed to select the spatial pooling regions and for dimensionality reduction [10].

In this work, we focus on a kernelized view of patch descriptors. In particular, we are motivated by the kernel descriptor of Bo *et al.* [2]. However, in our method we encode pixels in polar coordinates and rely on explicit feature maps [12], which provide a better approximation by Fourier series. We present a kernelized framework for local descriptors, out of which, known descriptors such as SIFT and our proposed descriptor are derived. This viewpoint of SIFT highlights some of its advantages and drawbacks: namely the encoding of the pixel position and the hard assignment which inevitably inserts some artifacts. We rather offer continuous encoding of pixel position and gradient orientations.

There are local descriptors that are rotation invariant by construction [8]. An alternative approach is to rely on the dominant orientation [7] of the patch in order to provide rotation invariance. The patches are rotated with respect to this dominant angle and transformed to up-right. Therefore, such descriptors are sensitive to this angle. In addition, in several tasks it is necessary to detect several multiple angles per patch, which further increases the computational cost of the procedures that follow. We adapt the latter choice of up-right patches and develop a fast patch alignment method with respect to rotation. Patch similarity is computed for multiple rotations at a slight increase of the computational cost. It allows us to handle the sensitivity to dominant orientation estimation and to dispense with the need for multiple dominant angles. This procedure is similar to the trigonometric polynomial of Tolias *et al.* [11], but at a patch level. It further resembles the way that Henriques *et al.* [6] speed-up learning with multiple shifted versions of negative samples, instead of performing costly sliding window based mining.

Our contribution includes a novel kernel local descriptor that encodes pixel position and gradient orientation in a continuous manner. It is also accompanied by an efficient way to compute patch similarity for multiple rotations, which constitutes our second contribution.

2. LOCAL DESCRIPTORS: KERNEL VIEW

Match kernels have gained an increasing interest after it was shown that it is possible to approximate non-linear kernels with linear ones by using an appropriate feature map [9, 12]. Similarity of sets can be efficiently computed with match kernels just by embedding the features in a suitable feature space in advance. Match kernels provide new grounds for designing better similarity functions.

Here, we define kernel descriptors over sets of pixels belonging to the same patch. An image patch can be considered as a set of pixels $\mathcal{X} = \{\mathbf{x}\}$. Without loss of generality we assume that we are dealing with grayscale patches. The relative position of a pixel $\mathbf{x}$ with respect to the patch center is expressed in polar coordinates and denoted by $(\varphi_{\mathbf{x}}, \rho_{\mathbf{x}})$. We employ gradient information, thus each pixel is described by the gradient magnitude $m_{\mathbf{x}}$ and the gradient orientation $\theta_{\mathbf{x}}$. The latter is expressed relatively to the angle $\varphi_{\mathbf{x}}$ (Figure 3). The pixel position and its gradient information are

referred as *pixel attributes* in the following. Two patches $\mathcal{X}$ and $\mathcal{Y}$ can be now compared via a match kernel of the form

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

where k is the local kernel and γ is the normalization factor ensuring that self-similarity $\mathcal{K}(\mathcal{X},\mathcal{X}) = 1$. The local kernel computes the similarity between two pixels $\mathbf{x}$ and $\mathbf{y}$, while the global kernel $\mathcal{K}$ accumulates similarities of all pairs of pixels. An interesting option is to obtain such a kernel by mapping pixels attributes to a higher-dimensional space with a feature map $\psi : \mathbf{x} \to \psi(\mathbf{x})$, such that the inner product evaluates the local kernel $k(\mathbf{x},\mathbf{y}) = \langle\psi(\mathbf{x})|\psi(\mathbf{y})\rangle$. The match kernel can be now expressed as:

$$\mathcal{K}(\mathcal{X},\mathcal{Y}) = \gamma(\mathcal{X})\gamma(\mathcal{Y}) \sum_{\mathbf{x}\in\mathcal{X}} \sum_{\mathbf{y}\in\mathcal{Y}} \langle\psi(\mathbf{x})|\psi(\mathbf{y})\rangle = \langle\mathbf{X}|\mathbf{Y}\rangle, \quad (2)$$

where $\mathbf{X} = \gamma(\mathcal{X}) \sum_{\mathbf{x}\in\mathcal{X}} \psi(\mathbf{x})$ and $\mathbf{Y} = \gamma(\mathcal{Y}) \sum_{\mathbf{y}\in\mathcal{Y}} \psi(\mathbf{y})$ are the local descriptors of patches $\mathcal{X}$ and $\mathcal{Y}$, respectively. Note that all the pairwise similarities are never explicitly enumerated; the linearity of the inner-product allows us to aggregate in advance the pixel feature vectors per patch.

Several popular methods for patch and image description can be actually described by this framework, among which SIFT. The SIFT descriptor is arguably one of the most popular and effective local feature descriptors in various computer vision applications. Some of the underlying advantages of SIFT can be emphasized better from a kernel perspective.

In the case of SIFT, consider that each pixel $\mathbf{x}$ is mapped to $\psi(\mathbf{x}) \in \mathbb{R}^{128}$, which is a sparse feature map due to the quantization of gradient orientation and spatial location. The aggregation of all pixel feature maps $\psi(\mathbf{x})$ results to the SIFT descriptor. The similarity of two SIFT descriptors can be then computed via inner product. The quantization to the spatial grid and to the orientation bins enforces to take into account only pixels with similar gradient orientations and spatial positions. The hard assignment in the quantization process inevitably inserts some artifacts and leads to a loss in the selectivity of the similarity function.

In the following, we design a kernelized local descriptor that imitates some of the advantages of the SIFT descriptor. At the same time, we alleviate some of drawbacks related to the quantization artifacts by encoding pixel position and gradient orientation in a continuous manner.

3. METHOD

We want to exploit jointly the photometric and position information of all patch elements (pixels). We target a kernel function for patch elements that reflects their resemblance in terms of gradients and their proximity in terms of their spatial position. To this effect, the local kernel $k(\mathbf{x},\mathbf{y})$ can be decomposed into $k_\theta(\theta_\mathbf{x},\theta_\mathbf{y})\, k_\varphi(\varphi_\mathbf{x},\varphi_\mathbf{y})\, k_\rho(\rho_\mathbf{x},\rho_\mathbf{y})$. It captures the similarity of the gradients with k_θ, and the spatial proximity on each coordinate separately with k_φ and k_ρ. Our match kernel now becomes

$$\mathcal{K}(\mathcal{X},\mathcal{Y}) \propto \sum_{\mathbf{x}\in\mathcal{X}} \sum_{\mathbf{y}\in\mathcal{Y}} \tilde{m}_\mathbf{x}\, \tilde{m}_\mathbf{y}\, k_\theta(\theta_\mathbf{x},\theta_\mathbf{y})\, k_\varphi(\varphi_\mathbf{x},\varphi_\mathbf{y})\, k_\rho(\rho_\mathbf{x},\rho_\mathbf{y}),$$

where the magnitude $\tilde{m}_\mathbf{x} = G(\rho_x,\sigma) * \sqrt{m_\mathbf{x}}$ is weighted by a Gaussian window in order to give higher importance to the patch center. In this fashion, our match kernel turns into scalar value comparison, such as angles and radii. Typically, non-linear functions are employed for such a comparison.

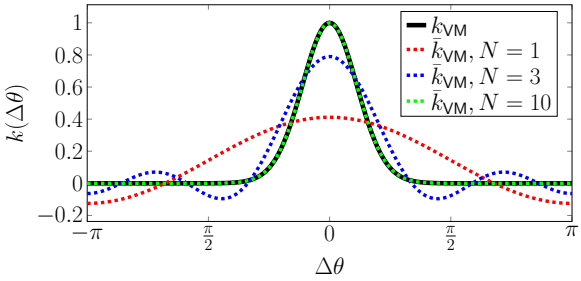

Figure 1: Target weighting function (Von Mises) and the corresponding approximations with 1, 3 and 10 frequencies.

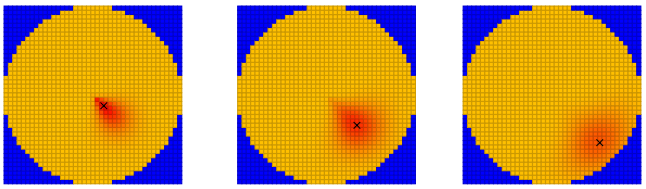

Figure 2: Visualization of the 2D weighting function for 3 sample pixels (shown by a cross). Colors reflect the spatial similarity to other pixels. Red (yellow) color corresponds to maximum (minimum) similarity. Blue color corresponds to the area that is neglected by our descriptor.

3.1 Feature maps for pixel attributes

We employ a normalized version of the Von Mises distribution [11] (normal distribution for angles) in order to compare two angles:

$$k_{\mathsf{VM}}(\theta_1,\theta_2) = k_{\mathsf{VM}}(\Delta\theta) = \frac{\exp(\kappa\cos(\Delta\theta)) - \exp(-\kappa)}{2\sinh(\kappa)}. \quad (3)$$

This is a stationary kernel that depends only on the difference of the two angles $\Delta\theta = \theta_1 - \theta_2$. The selectivity of the function is controlled by parameter κ.

We define a mapping $\phi : [-\pi,\pi] \to \mathbb{R}^M$ of an angle θ to a vector $\phi(\theta)$ such that the inner product of two such vectors approximates the target function, that is $\phi(\theta_1)^\top\phi(\theta_2) = \bar{k}_{\mathsf{VM}}(\theta_1,\theta_2) \approx k_{\mathsf{VM}}(\theta_1,\theta_2)$. For this purpose we make use of explicit feature maps [12] and follow the methodology of Tolias *et al.* [11]. The desired mapping is

$$\phi(\theta) = (\sqrt{\gamma_0}, \sqrt{\gamma_1}\cos(\theta), \sqrt{\gamma_1}\sin(\theta), \ldots, \sqrt{\gamma_N}\cos(N\theta), \sqrt{\gamma_N}\sin(N\theta))^\top, \quad (4)$$

where γ_i is the i-th Fourier coefficient of Von Mises (3). The approximation by N Fourier coefficients (coresponding to N frequencies) produces a vector of $M = 2N+1$ dimensions. The number of frequencies influences the accuracy of the approximation. Figure 1 illustrates the target function and its approximation for different values of N.

We choose the Von Mises to implement all 3 local kernels k_θ, k_φ and k_ρ. The mapping of Equation (4) is trivially used on local kernels k_θ and k_φ, since they deal with angles. We further map the radius of pixel $\mathbf{x}$ to an angle by $\tilde{\rho}_\mathbf{x} = \rho_\mathbf{x}\pi$, with $\rho_\mathbf{x} \in [0,1]$. In this way, we are now able to use the same mapping of scalar to vectors for the radius also. In Figure 2 we visualize the combination of the two local kernels k_ρ and k_φ (by their product) that evaluate the spatial proximity.

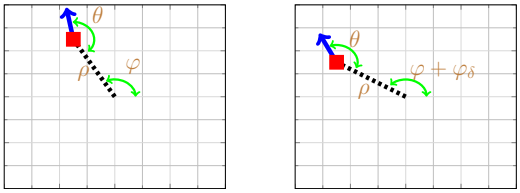

Figure 3: A sample patch and a corresponding pixel denoted by red rectangle. The gradient vector is shown by a blue arrow. The initial patch (left), is rotated by angle φ_δ (right). Both radius ρ and angle θ remain unchanged, while only angle φ changes.

3.2 Wrapping up the descriptor

Each patch element $\mathbf{x}$ is now associated to three vectors describing its attributes, namely $\phi(\theta_\mathbf{x})$, $\phi(\varphi_\mathbf{x})$ and $\phi(\tilde{\rho}_\mathbf{x})$. In order to obtain a single descriptor we propose to describe $\mathbf{x}$ by the Kronecker product of these vectors, defined by $\psi(\mathbf{x}) = \tilde{m}_\mathbf{x}\phi(\theta_\mathbf{x}) \otimes \phi(\varphi_\mathbf{x}) \otimes \phi(\tilde{\rho}_\mathbf{x})$. By aggregating such vectors for all patch elements, the patch descriptor is now formed by $\mathbf{X} \propto \sum_{\mathbf{x}\in\mathcal{X}} \psi(\mathbf{x})$.

By using the Kronecker product properties we can show that comparing two such local descriptors via inner product is equivalent to the approximation of our match kernel:

$$\begin{aligned}\langle \mathbf{X}|\mathbf{Y}\rangle &\propto \sum_{\mathbf{x}\in\mathcal{X}} \psi(\mathbf{x})^\top \sum_{\mathbf{y}\in\mathcal{Y}} \psi(\mathbf{y}) = \sum_{\mathbf{x}\in\mathcal{X}}\sum_{\mathbf{y}\in\mathcal{Y}} \psi(\mathbf{x})^\top\psi(\mathbf{y}) \\ &\approx \sum_{\mathbf{x}\in\mathcal{X}}\sum_{\mathbf{y}\in\mathcal{Y}} \tilde{m}_\mathbf{x}\,\tilde{m}_\mathbf{y}\, k_\theta(\theta_\mathbf{x},\theta_\mathbf{y})\, k_\varphi(\varphi_\mathbf{x},\varphi_\mathbf{y})\, k_\rho(\tilde{\rho}_\mathbf{x},\tilde{\rho}_\mathbf{y}) \\ &= \mathcal{K}(\mathcal{X},\mathcal{Y}). \end{aligned} \quad (5)$$

The desired property of our mapping is the linearity of the inner product used to compare two vectors, which approximates a non-linear function comparing two angles. The pixel representation can be then aggregated in advance for each patch. The dimensionality of the local descriptor is equal to $(2N_\theta+1)(2N_\varphi+1)(2N_\rho+1)$, where different number of frequencies can be used for each pixel attribute (N_θ, N_φ and N_ρ) depending on the use-case. The descriptor is subsequently square-rooted and L_2-normalized.

3.3 Fast rotation alignment

At this stage, our match kernel and, equivalently, the kernel descriptor assume that all patches have the same global orientation. Patches are typically orientated to up-right position according to their dominant orientation. We adopt the same choice. However, this type of alignment can be quite noisy. We propose a method for identifying the rotation that maximizes the patch similarity and aligns them in an optimal way. We achieve this without explicitly computing the descriptors for all possible rotations.

Imagine that a patch $\mathcal{X}$ is rotated by an angle φ_δ into patch $\mathcal{X}_\delta$, and denote the descriptor of the rotated patch by $\mathbf{X}_\delta$. The only pixel attribute that changes is the angle φ, shifted by φ_δ, as illustrated in the toy example of Figure 3. Under this point of view and with respect to variable φ, it can be seen that local descriptor $\mathbf{X}$ is decomposed into the following sub-vectors $[\mathbf{X}_0{}^\top, \mathbf{X}_{1,c}{}^\top, \mathbf{X}_{1,s}{}^\top, \ldots, \mathbf{X}_{N,c}{}^\top, \mathbf{X}_{N,s}{}^\top]^\top$. $\mathbf{X}_0$ is constant for any patch rotation. The sub-vectors $\mathbf{X}_{i,c}$ and $\mathbf{X}_{i,s}$ are related to components of the form $cos(i\varphi_\mathbf{x})$ and $sin(i\varphi_\mathbf{x})$, respectively, for frequency i.

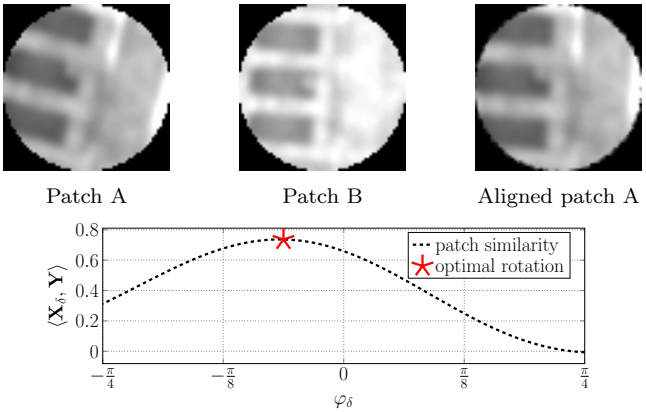

Figure 4: Two sample patches (top) and their similarity (bottom) for multiple rotations of patch A. At the (top) right, patch A is rotated by the optimal orientation.

Interestingly, by using simple trigonometric identities, it turns out [11] that the similarity of two patches, when one of them undergoes rotation, forms a trigonometric polynomial:

$$\begin{aligned}\langle \mathbf{X}_\delta|\mathbf{Y}\rangle =& \langle \mathbf{X}_0|\mathbf{Y}_0\rangle + \sum_{n=1}^{N_\varphi} \cos(n\varphi_\delta)\ (\langle \mathbf{X}_{n,c}|\mathbf{Y}_{n,c}\rangle + \langle \mathbf{X}_{n,s}|\mathbf{Y}_{n,s}\rangle) \\ &+ \sum_{n=1}^{N_\varphi} \sin(n\varphi_\delta)\ (-\langle \mathbf{X}_{n,c}|\mathbf{Y}_{n,s}\rangle + \langle \mathbf{X}_{n,s}|\mathbf{Y}_{n,c}\rangle)\,. \end{aligned} \quad (6)$$

The complexity of computation of the polynomial coefficients in Equation (6) is less than twice the cost to compute the standard similarity between two kernel descriptors, while the cost to evaluate similarity for multiple angles φ_δ is negligible. In Figure 4 we present an example of the similarity of two patches under multiple rotations. The rotation of maximum similarity is used to align the patches.

4. EXPERIMENTS

We compare our descriptor against the state-of-the-art RootSIFT [1], and its counterpart rotated by PCA, noted as RootSIFT-PCA in the following. We further compare with the learned descriptors of Simonyan *et al.* [10] and the ones of Brown *et al.* [3] that rely on pairs of known related and non-related patches. We do not consider in our evaluation the descriptor of Bo *et al.* [2] as it is optimized to be used in image level aggregated manner for image classification, whereas we test patch similarities. We use a patch dataset [3] comprising three subsets, Notre Dame, Liberty and Yosemite, corresponding to different landmarks on which 3D reconstruction was performed. Each subset consists of 450k image patches of size 64×64 which are derived from keypoints detected with Difference-of-Gaussians. They are normalized with respect to scale and rotation. The ground-truth denotes the groups of similar patches, and two patches are considered similar if they are projections of the same 3D point.

4.1 Implementation details

We refer to our kernel descriptor as $KD_{N_\theta,N_\varphi,N_\rho}$ and evaluate for different number of frequencies. Parameter κ is always fixed and equal to 8, except for the case of $N_\rho = 1$ when κ is 2. In order to avoid artifacts for the computation of the rotated patches, we keep only the pixels inside the circle inscribed in the patch, as shown in Fig. 4.

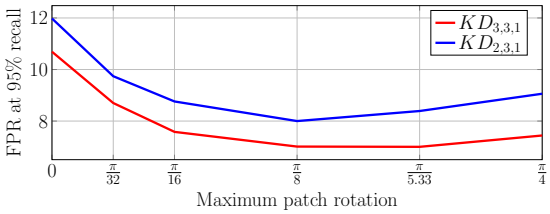

Figure 5: Impact of the rotation alignment on the performance. We evaluate patch similarity for 0, 4, 8, 16, 24 and 32 fixed rotations at each direction (clockwise and counter-clockwise). Results reported on Notre Dame dataset.

Post-processing. The patch descriptor is power-law normalized with the power-law exponent α set to 0.5. For the PCA rotated variant, we obtain better results for $\alpha = 1$. For this case, we apply powerlaw normalization with $\alpha = 0.5$ after the projection. We proceed similarly for RootSIFT-PCA.

Orientation alignment. In order to align two patches we test up to 64 fixed rotations on each direction with a step of $\pi/128$. We find a good trade-off for rotations in the interval $[-\pi/8, \pi/8]$. Since the patches are already up-right, our algorithm reduces the quantization errors in the computation of the dominant orientations.

4.2 Hypothesis test

We evaluate our kernel descriptors following the standard protocol, generate the ROC curves and report false positive rate (FPR) at 95% recall. In Figure 5 we illustrate how performance improves by evaluating similarities for multiple patch rotations. After some extent performance decreases, since patches are already up-right by the rough dominant orientation and we are only introducing noisy matches.

We now consider each of the six possible combinations of training and test sets and we test 100k pairs for each run. Table 1 compares the error rates of our kernel descriptor against other local descriptors. Our descriptor is better than RootSIFT and RootSIFT-PCA and its performance is reaching that of learned descriptors [10]. While our kernel descriptor is higher dimensional, it doesn't require any training. We further evaluate it with PCA learned on a different dataset in order to obtain more compact descriptors. Although we cannot test multiple rotations anymore, the performance improves significantly outperforming more sophisticated methods trained over annotated pairs of similar and non-similar patches.

4.3 Nearest neighbors

We further evaluate our descriptor on a nearest neighbor search task using the patch dataset. We randomly select 1,000 query patches and report recall at the top R retrieved patches. This task is performed on a single subset at a time. Results for using Notre Dame as test set and Yosemite as learning set are reported in Table 2. Our kernel descriptors appear to perform the best, while the rotation alignment improves once more.

5. CONCLUSIONS

We proposed a kernel descriptor equipped with continuous encoding and a fast rotation alignment process. Interestingly, it achieves superior performance even compared to methods that employ supervised learning.[1]

Train	Test	*RootSIFT*	*RootSIFT-PCA*	*Simonyan [10]*	*Brown[3]*	$KD_{2,3,1}$	$KD_{3,3,1}$	$KD_{3,2,2}$-PCA
ND	Lib	29.64	19.34	**12.42**	16.85	21.58	20.06	13.17
Yos	Lib		19.95	14.58	18.27			**14.53**
ND	Yos	26.69	18.37	10.08	13.55	11.07	9.66	**8.31**
Lib	Yos		19.52	11.18	N/A			**9.65**
Lib	ND	22.06	13.90	7.22	N/A	7.99	7.00	**6.36**
Yos	ND		13.98	6.82	11.98			**6.50**
Mean		26.14	17.51	10.38	15.16	13.55	12.24	**9.75**
Dimensions		128	80	73-77	29-36	105	147	80
Learning		N	US	S	S	N	N	US

Table 1: False positive rate (%) at 95% recall. Learning type: *N*-none, *US*-unsupervised, *S*-supervised.

R	1	5	10	100	1000	10000
RootSIFT	8.5	24.4	33.0	62.9	79.7	90.6
RootSIFT-PCA	8.8	23.9	32.7	61.4	78.4	90.4
$KD_{3,3,1}$ (No rotations)	9.1	24.7	34.6	64.9	80.8	91.3
$KD_{3,3,1}$ (16 rotations)	**8.8**	**26.2**	**37.3**	**68.3**	**84.4**	**93.1**
$KD_{3,3,1} - PCA$	9.4	24.9	35.2	66.4	82.9	92.4

Table 2: Recall computed at R top ranked patches for 1000 randomly selected patch queries on Notredame dataset. Learning for PCA is performed on Yosemite dataset, and the dimensionality is reduced to 80 components.

6. REFERENCES

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

[2] L. Bo, X. Ren, and D. Fox. Kernel descriptors for visual recognition. In *NIPS*, Dec. 2010.

[3] M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *Trans. PAMI*, 33(1):43–57, 2011.

[4] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *ECCV*, Oct. 2010.

[5] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.

[6] J. F. Henriques, J. Carreira, R. Caseiro, and J. Batista. Beyond hard negative mining: Efficient detector learning via block-circulant decomposition. In *ICCV*, 2013.

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

[8] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Trans. PAMI*, 24(7):971–987, 2002.

[9] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *NIPS*, 2007.

[10] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *Trans. PAMI*, 2014.

[11] G. Tolias, T. Furon, and H. Jégou. Orientation covariant aggregation of local descriptors with embeddings. In *ECCV*, Sep. 2014.

[12] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *Trans. PAMI*, 34(3):480–492, Mar. 2012.

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