Multiple-Kernel Local-Patch Descriptor

Arun Mukundan
arun.mukundan@cmp.felk.cvut.cz
Giorgos Tolias
giorgos.tolias@cmp.felk.cvut.cz
Ondřej Chum
chum@cmp.felk.cvut.cz

Visual Recognition Group,
Faculty of Electrical Engineering,
Czech Technical University in Prague

Abstract

We propose a multiple-kernel local-patch descriptor based on efficient match kernels of patch gradients. It combines two parametrizations of gradient position and direction, each parametrization provides robustness to a different type of patch miss-registration: polar parametrization for noise in the patch dominant orientation detection, Cartesian for imprecise location of the feature point. Even though handcrafted, the proposed method consistently outperforms the state-of-the-art methods on two local patch benchmarks.

1 Introduction

Representing and matching local features is an essential step of several computer vision tasks. It has attracted a lot of attention in the last decades, when local features still were a required step of most approaches. Despite the large focus on Convolutional Neural Networks (CNN) to process whole images, local features still remain important and necessary for tasks such as Structure-from-Motion (SfM) [10], stereo matching [20], or retrieval under severe change in viewpoint or scale [26].

Recently, the focus has shifted from hand-crafted descriptors to CNN-based descriptors. Learning such descriptors relies on large training sets of patches, that are commonly provided as a side-product of SfM [34]. Remarkable performance is achieved on a standard benchmark [5]. However, recent work [6, 27] shows that CNN-based approaches do not necessarily generalize equally well on different tasks or different datasets. Hand-crafted descriptors still appear an attractive alternative.

We build upon the hand-crafted kernel descriptor proposed by Bursuc *et al*. [9] that is shown to have good performance, even compared to learned alternatives. Its few parameters are easily tuned on some validation set, while it is shown to perform well on multiple tasks, as we confirm in our experiments. Post-processing with PCA and power-law normalization are shown beneficial.

Visualizing and analyzing the parametrization of this kernel descriptor allows us to understand its advantages and disadvantages, mainly the undesirable discontinuity around the patch center. We propose to combine multiple parametrizations and kernels to achieve robustness to different types of patch miss-registration. Experimental evaluation shows that the proposed descriptor outperforms all other approaches on two benchmarks designed to compare local-feature descriptors, specifically on the newly introduced HPatches dataset [6], and on the Phototourism benchmark [34].

2 Related work

We review prior work on local descriptors, covering both hand-crafted and learned ones.

Hand-crafted descriptors attracted a lot of attention for a decade and a variety of approaches and methodologies exists. A popular direction is that of gradient histogram-based descriptors, where the most popular representative is SIFT [17]. Different variants focus on pooling regions [16, 19], efficiency [1, 30], invariance [16] or other aspects [14]. Other are based on filter-bank responses [15], patch intensity [10, 25] or ordered intensity [21].

Kernel descriptors based on the idea of Efficient Match Kernels (EMK) [7] encode entities inside a patch (such a gradient, color, *etc*) in a continuous domain, rather than as a histogram. The kernels and their few parameters are often hand-picked and tuned on a validation set. Kernel descriptors are commonly represented by a finite-dimensional explicit feature maps. Quantized descriptors, such as SIFT, can be also interpreted as kernel descriptors [8, 9].

Learned descriptors commonly require annotation at patch level. Therefore, research in this direction is facilitated by the release of datasets that are originate from an SfM system [22, 34]. Such training datasets allow effective learning of local descriptors, and in particular, their pooling regions [29, 34], filter banks [34], transformations for dimensionality reduction [29] or embeddings [23].

Kernelized descriptors are formulated within a supervised framework by Wang *et al*. [33], where image labels enable kernel learning and dimensionality reduction. In this work, we rather focus on minimal learning in the form of discriminatively learned projections. This is several orders of magnitude faster to learn than other learning approaches.

Recently, learning local descriptor is dominated by deep learning. The network architectures are smaller than the corresponding ones performing on images, and use a large amount of training patches. Among representative examples is the work of Simo-Serra *et al*. [28] training with hard positive and negative examples or the work of Zagoruyko [35] where a central-surround representation is found to be immensely beneficial. CNN-based approaches are seen as joint feature, filter bank, and metric learning [12]. Finally, the state of the art consists of shallower architectures with improved ranking loss [4, 5]. Despite obtaining impressive results on a standard benchmark, CNN-based approaches do not generalize well to other datasets and tasks [6, 27].

A post-processing step is common to both hand-crafted and learned descriptors. This post-processing ranges from simple ℓ_2 normalization, PCA dimensionality reduction, to transformations learned on annotated data.

3 Preliminaries

Kernelized descriptors. In general lines we follow the formulation of Bursuc *et al*. [9]. We represent a patch $\mathcal{P}$ as a set of pixels $p \in \mathcal{P}$ and compare two patches $\mathcal{P}$ and $\mathcal{Q}$ via match kernel

$$\mathcal{M}(\mathcal{P},\mathcal{Q}) = \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}} k(p,q), \tag{1}$$

where kernel $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a similarity function, typically non-linear, comparing two pixels. EMK uses an explicit feature map $\psi : \mathbb{R}^n \to \mathbb{R}^d$ to approximate this result as

$$\mathcal{M}(\mathcal{P},\mathcal{Q}) = \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}} k(p,q) \approx \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}} \psi(p)^\top \psi(q) = \sum_{p\in\mathcal{P}} \psi(p)^\top \sum_{q\in\mathcal{Q}} \psi(q). \tag{2}$$

Vector $\mathbf{V}(\mathcal{P}) = \sum_{p \in \mathcal{P}} \psi(p)$ is a *kernelized descriptor* (KD), associated with patch $\mathcal{P}$, used to approximate $\mathcal{M}(\mathcal{P}, \mathcal{Q})$, whose explicit evaluation is costly. The approximation is given by a dot product $\mathbf{V}(\mathcal{P})^\top \mathbf{V}(\mathcal{Q})$, where $\mathbf{V}(\mathcal{P}) \in \mathbb{R}^d$. To ensure a unit self similarity, ℓ_2 normalization by a factor γ is introduced. The normalized KD is then given by $\bar{\mathbf{V}}(\mathcal{P}) = \gamma(\mathcal{P})\mathbf{V}(\mathcal{P})$, where $\gamma(\mathcal{P}) = (\mathbf{V}(\mathcal{P})^\top \mathbf{V}(\mathcal{P}))^{-1/2}$.

Kernel k comprises product of kernels that act on scalar pixel attributes

$$k(p,q) = k_1(p_1,q_1)k_2(p_2,q_2)\dots k_n(p_n,q_n), \tag{3}$$

where kernel k_n is pairwise similarity function for scalars and p_n are pixel attributes such as position and gradient orientation. Feature map ψ_n corresponds to kernel k_n and feature map ψ is constructed via Kronecker product of individual feature maps $\psi(p) = \psi_1(p_1) \otimes \psi_2(p_2) \otimes \dots \otimes \psi_n(p_n)$. Due to the mixed product property it holds that $\psi(p)^\top \psi(q) \approx k_1(p_1,q_1)k_2(p_2,q_2)\dots k_n(p_n,q_n)$.

Feature maps. As non-linear kernel for scalars we use the normalized Von Mises probability density function[1], which is used for image [31] and patch [9] representation. It is parametrized by κ controlling the shape of the kernel, where lower κ corresponds to wider kernel. We use a stationary kernel that, by definition, depends only on the difference $\Delta_n = p_n - q_n$, *i.e.* $k_{\mathsf{VM}}(p_n,q_n) := k_{\mathsf{VM}}(\Delta_n)$. We adopt a Fourier series approximation with N frequencies that produces a feature map $\psi_{\mathsf{VM}} : \mathbb{R} \to \mathbb{R}^{2N+1}$. It has the property that $k_{\mathsf{VM}}(p_n,q_n) \approx \psi_{\mathsf{VM}}(p_n)^\top \psi_{\mathsf{VM}}(q_n)$. The reader is encouraged to read prior work for details on these feature maps [32], which are previously used in various contexts [9, 31].

Descriptor post-processing. It is known that further descriptor post-processing [3, 9, 24] is beneficial. In particular, KD is further centered and projected as

$$\hat{\mathbf{V}}(\mathcal{P}) = A^\top(\bar{\mathbf{V}}(\mathcal{P}) - \mu), \tag{4}$$

where $\mu \in \mathbb{R}^d$ and $A \in \mathbb{R}^{d \times d}$ are the mean vector and the projection matrix. These are commonly learned by PCA [13] or with supervision [24]. The final descriptor is always ℓ_2-normalized in the end.

4 Method

In this section we consider different patch parametrizations and kernels that result in different patch similarity. We discuss the benefits of each and propose how to combine them. We further learn descriptor transformation with supervision and provide useful insight on how patch similarity is affected.

Patch attributes. We consider a pixel p to be associated with coordinates p_x, p_y in Cartesian coordinate system, coordinates p_ρ, p_ϕ in polar coordinate system, pixel gradient magnitude p_m, and pixel gradient angle p_θ. Angles p_θ, $p_\phi \in [0, 2\pi]$, distance from the center p_ρ is normalized to $[0,1]$, while coordinates p_x, $p_y \in \{1,2,\dots,W\}$ for $W \times W$ patches. In order to use feature map ψ_{VM}, attributes p_ρ, p_x, and p_y are linearly mapped to $[0,\pi]$. The gradient angle is expressed *w.r.t.* the patch orientation, *i.e.* p_θ directly, or *w.r.t.* to the position of the pixel. The latter is given as $p_{\tilde{\theta}} = p_\theta - p_\phi$.

[1] Also known as the periodic normal distribution

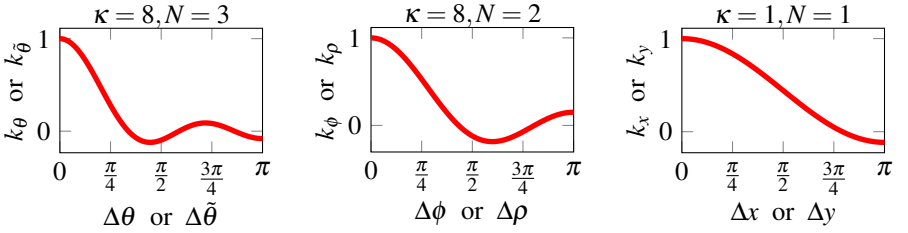

Figure 1: Kernel approximations that we use for pixel attributes. Parameter κ and the number of frequencies N define the final shape. The choice of kernel parameters is guided by [9].

Patch parametrizations. Composing patch kernel k as a product of kernels over different attributes enables easy design of various patch similarities. Correspondingly, this defines different KD. All attributes p_x, p_y, p_ρ, p_θ, p_ϕ, and $p_{\tilde{\theta}}$ are matched by the Von Mises kernel, namely, k_x, k_y, k_ρ, k_θ, k_ϕ, and $k_{\tilde{\theta}}$ parameterized by κ_x, κ_y, κ_ρ, κ_θ, κ_ϕ, and $\kappa_{\tilde{\theta}}$, respectively.

In this work we focus on the two following match kernels over patches. One in *polar* coordinates

$$\mathcal{M}_{\phi\rho\tilde{\theta}}(\mathcal{P},\mathcal{Q}) = \sum_{p\in\mathcal{P}}\sum_{q\in\mathcal{Q}} p_g q_g \sqrt{p_m}\sqrt{q_m} k_\phi(p_\phi,q_\phi) k_\rho(p_\rho,q_\rho) k_{\tilde{\theta}}(p_{\tilde{\theta}},q_{\tilde{\theta}}), \tag{5}$$

and one in *cartesian* coordinates

$$\mathcal{M}_{xy\theta}(\mathcal{P},\mathcal{Q}) = \sum_{p\in\mathcal{P}}\sum_{q\in\mathcal{Q}} p_g q_g \sqrt{p_m}\sqrt{q_m} k_x(p_x,q_x) k_y(p_y,q_y) k_\theta(p_\theta,q_\theta), \tag{6}$$

where $p_g = exp(-p_\rho^2)$ gives more importance to central pixels, in a similar manner to SIFT.

The KD for the two cases are given by

$$\mathbf{V}_{\phi\rho\tilde{\theta}}(\mathcal{P}) = \sum_{p\in\mathcal{P}} p_g p_m \psi_\phi(p_\phi)\otimes\psi_\rho(p_\rho)\otimes\psi_{\tilde{\theta}}(p_{\tilde{\theta}}) = \sum_{p\in\mathcal{P}} p_g\sqrt{p_m}\psi_{\phi\rho\tilde{\theta}}(p) \tag{7}$$

$$\mathbf{V}_{xy\theta}(\mathcal{P}) = \sum_{p\in\mathcal{P}} p_g p_m \psi_x(p_x)\otimes\psi_y(p_y)\otimes\psi_\theta(p_\theta) = \sum_{p\in\mathcal{P}} p_g\sqrt{p_m}\psi_{xy\theta}(p). \tag{8}$$

The $\mathbf{V}_{\phi\rho\tilde{\theta}}$ variant is exactly the one proposed by Bursuc *et al*. [9], considered as a baseline in this work. Different parametrizations result in different patch similarity, which is analyzed in the following. In Figure 1 we present the approximation of kernels used per attribute.

Descriptor post-processing with supervision. Mean vector μ and projection matrix A can be learned in an unsupervised way, *e.g*. by PCA on a sample descriptor set. In such case, matrix A is formed by the eigenvectors as columns. This is the case in prior work, not only for local descriptors [9] but also for global image representation [13]. It was previously observed, and our experiments confirm, that discriminative projection [18] learned on labeled data outperforms post-processing by generative model, such as PCA. The discriminative projection is composed of two parts, a whitening part and a rotation part. The whitening part is obtained from the intraclass (matching pairs) covariance matrix, while the rotation part is the PCA of the interclass (non-matching pairs) covariance matrix in the whitened space. Vector μ is the mean descriptor vector. To reduce the descriptor dimensionality, only eigenvectors corresponding to the largest eigenvalues are used. We refer to this transformation as learned (supervised) whitening (LW) in the rest of the paper.

Visualization of patch similarity. We define pixel similarity $\mathcal{M}(p,q)$ as kernel response between pixels p and q, approximated as $\mathcal{M}(p,q) \approx \psi(p)^\top \psi(q)$. To show a spatial distribution of the influence of pixel p, we define a *patch map* of pixel p. The patch map has the same size as the image patches, for each pixel q of the patch, map $\mathcal{M}(p,q)$ is evaluated for some constant value of q_θ.

For example, in Figure 2 patch maps for different kernels are shown. The position of p is denoted by $\times$ symbol. The value of $p_\theta = 0$ and $q_\theta = 0$ for all spatial locations of q in the top row and $q_\theta = -\pi/8$ in the bottom row. The visualization shows the discontinuity of the pixel similarity impact of the $\mathbf{V}_{\phi\rho\tilde{\theta}}$ descriptor near the center of the patch. This is caused by the polar coordinate system where a small difference in the position near the origin causes large difference in ϕ and $\tilde{\theta}$. Also in the bottom row we see that using the relative gradient direction $\tilde{\theta}$ allows to compensate for imprecision caused by small patch rotation, *i.e.* the most similar pixel is not the one at the location of p with different $\tilde{\theta}$, but a rotated pixel with more similar value of $\tilde{\theta}$. Finally, we observe that the kernel parametrized by Cartesian coordinates and absolute angle of the gradient ($\mathbf{V}_{xy\theta}$, third column) is insensitive to small translations, *i.e.* feature point displacement.

We additionally construct patch maps in the case of descriptor post-processing by a linear transformation, *e.g.* descriptor whitening. Now the contribution of a pixel pair is given by

$$\hat{\mathcal{M}}(p,q) = (A^\top(\psi(p)-\mu))^\top (A^\top(\psi(q)-\mu)) \tag{9}$$
$$= (\psi(p)-\mu)^\top AA^\top(\psi(q)-\mu) \tag{10}$$
$$= \psi(p)^\top AA^\top \psi(q) - \psi(p)^\top AA^\top \mu - \psi(q)^\top AA^\top \mu + \mu^\top AA^\top \mu. \tag{11}$$

The last term is constant and can be ignored, while if A is a rotation matrix then only shifting by μ affects the similarity. After the transformation, the similarity is no longer shift-invariant. The non-linear post-processing, such as power-law normalization or simple ℓ_2 normalization cannot be visualized, as it acts after the pixel aggregation[2].

Figure 3 we shows patch maps for $\mathbf{V}_{\phi\rho\tilde{\theta}}$ in the case of PCA or LW post-processing. PCA is shown to have some small effect on the similarity, while LW significantly changes the derived shape. It implicitly affects the shape of the kernels used; observe that the kernels go wider in the circular direction.

Combining kernel descriptors. We propose to take advantage of both parametrizations $\mathbf{V}_{\phi\rho\tilde{\theta}}$ and $\mathbf{V}_{xy\theta}$, by summing their contribution. This is performed by simple concatenation of the two descriptors. Finally, whitening is jointly learned and dimensionality reduction is performed.

In Figure 4 we show patch maps for the individual and combined representation, before and after applying learned whitening. Observe how the combined one better behaves around the center but also how the final similarity is formed after the whitening.

[2] Details are omitted due to lack of space.

[3] Ten isocontours are sampled uniformly. The similarity is shown in a relative manner and, therefore, the absolute scale is missing (*e.g.* in Figure 2 the maximum value is larger in top row compared to bottom due to $k_\theta(0) > k_\theta(\pi/8)$).

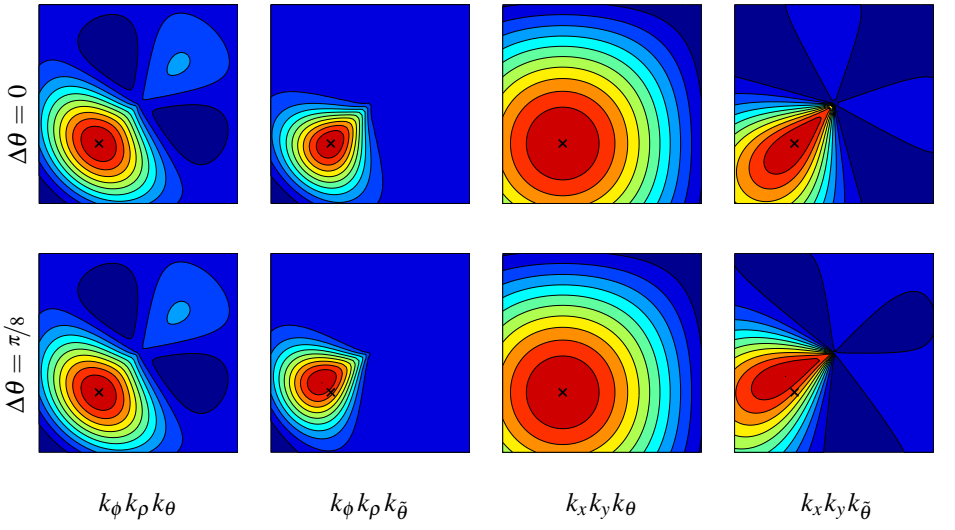

Figure 2: Patch maps for different parametrizations and kernels. We present two parametrizations in polar and two in cartesian coordinates, with absolute or relative gradient angle for each one. $\Delta\theta$ is fixed and pixel p is shown with "$\times$". At the bottom of each column the kernels (patch similarity) approximated are shown.[3]

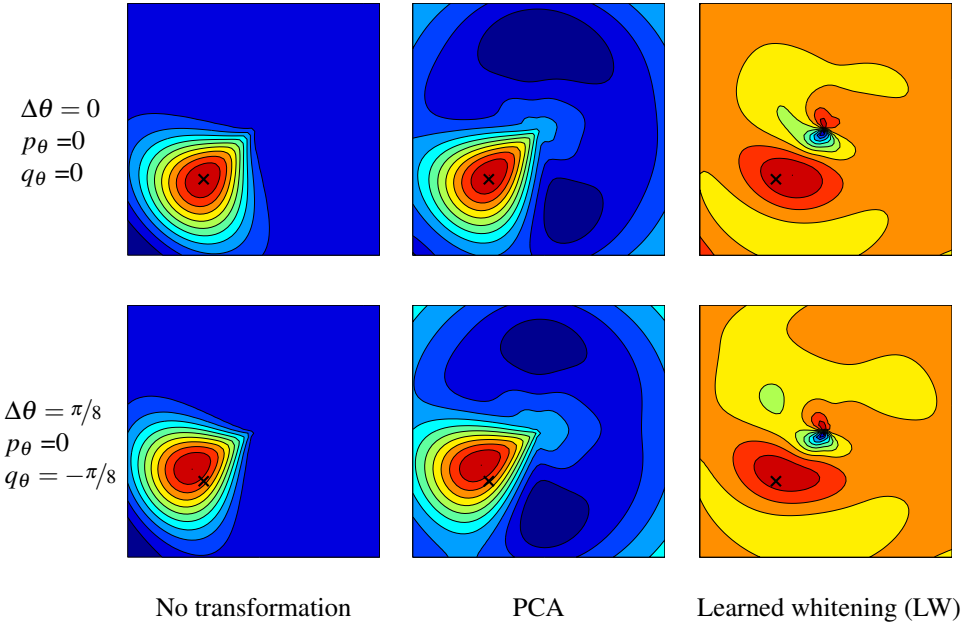

Figure 3: Patch maps for $\phi\rho\tilde{\theta}$ parametrization and kernels. $\Delta\theta$ is fixed by choosing fixed values for p_θ and q_θ. Pixel p is shown with "$\times$". Three different cases are shown: without transformation, with PCA transformation and with transformation by supervised whitening.[3]

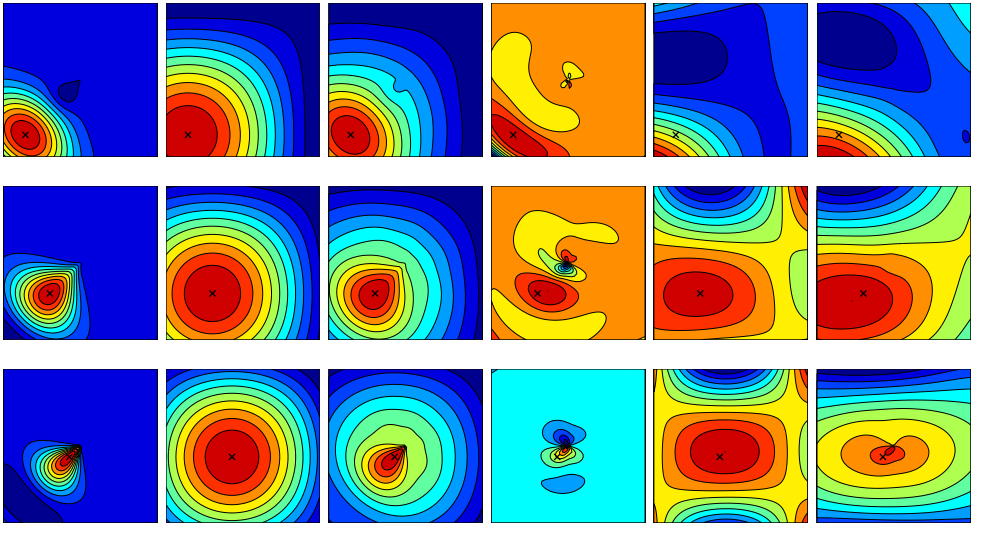

Figure 4: Patch maps for different parametrizations and kernels. We present polar and cartesian parametrization separately, and their combination by descriptor concatenation. We present the case for 3 different pixels p (one pixel per row) shown with "$\times$". $\Delta\theta = 0$ in all examples, in particular $p_\theta = 0$ and $q_\theta = 0$. The cases without descriptor transformation and with transformation by supervised whitening (LW) are shown.[3]

5 Experiments

We evaluate the method on two benchmarks, namely the widely used *Phototourism* (PT) dataset [34], and the recently released *HPatches* (HP) dataset [6]. We first compare the proposed method with the baseline method of Bursuc *et al*. [9] and then with the state-of-the-art methods on the two datasets. In all our experiments with descriptor post-processing the dimensionality is reduced to 128 except for the cases where the input descriptor is already of lower dimension.

Datasets and protocols. The Phototourism dataset contains three sets of patches, namely, Liberty (Li), Notredame (No) and Yosemite (Yo). Additionally, labels are provided to indicate the 3D point that the patch corresponds to, thereby providing supervision. It has been widely used for training and evaluating local descriptors. Performance is measured by the false positive rate at 95% of recall (FPR95). The protocol is to train on one of the three sets and test on the other two. An average over all six combinations is reported.

The HPatches dataset contains local patches of higher diversity, is more realistic, and during evaluation the performance is measured on three tasks: *verification*, *retrieval*, and *matching*. We follow the standard evaluation protocol [6] and report mean Average Precision (mAP). When evaluating on HP, we follow the protocol and learn the whitening on PhotoTourism Liberty, or on a pre-defined split of test and train of the HPatches dataset provided by the authors.

Test			Liberty		Notredame		Yosemite	
Train	*D*	*Mean*	*No*	*Yo*	*Li*	*Yo*	*No*	*Li*
polar [9]	175	22.42	24.34	24.34	16.06	16.06	26.85	26.85
cartes	63	35.87	34.06	34.06	34.10	34.10	39.47	39.47
polar + *cartes*	238	25.37	26.16	26.16	20.04	20.04	29.91	29.91
polar + *PCA* [9]	128	8.30	12.09	13.13	5.16	5.41	7.52	6.49
polar [9] + *LW*	128	7.06	8.55	10.48	4.40	3.94	8.86	6.12
cartes + *LW*	63	15.13	17.31	20.34	10.90	11.85	16.84	13.55
polar + *cartes* + *LW*	128	**5.98**	**7.44**	**9.84**	**3.48**	**3.54**	**6.56**	**5.02**

Table 1: Performance comparison on Phototourism dataset between the baseline approach and our combined descriptor. We further show the benefit of learned whitening (LW) over the standard PCA followed by square-rooting. FPR95 is reported for all methods.

Method	Verification	Matching	Retrieval
polar [9]	80.77	32.51	48.04
cartes	70.67	15.79	30.73
polar + *cartes*	77.97	29.34	44.23
polar + *PCA* [9]	87.11	38.45	54.81
polar [9] + *LW*	88.00	41.91	58.80
cartes + *LW*	85.13	33.77	52.94
polar + *cartes* + *LW*	**88.64**	**43.81**	**61.21**

Table 2: Performance comparison of the baseline approach and our combined descriptor via mAP on HPatches dataset. PCA and LW are learned on a subset of HP.

Comparison with the baseline. The results of the experimental evaluation are shown in Tables 1 and 2 for the PT and HP datasets, respectively. For all compared methods, including the baseline, we observed that in the descriptor post-processing stage, the discriminative whitening (marked LW) outperforms PCA followed by square-rooting (originally proposed in [9]). The difference is observed among 4^{th} and 5^{th} row of Tables 1 and 2.

Polar parametrization with the relative gradient direction (*polar*) significantly outperforms the Cartesian parametrization with the absolute gradient direction (*cartes*). After the descriptor post-processing (*polar* + LW vs. *cartes* + LW), the gap is reduced. The performance of the combined descriptor (*polar* + *cartes*) without descriptor post-processing is worse than the baseline descriptor. That is caused by the fact, that the two descriptors are combined with an equal weight, which is clearly suboptimal. No attempt is made to estimate the mixing parameter explicitly, as this is implicitly done in the post-processing stage. The jointly whitened combination of the two parametrizations (last row of Tables 1 and 2) consistently outperforms the baseline method.

Comparison with the State of the Art. We compare the performance of proposed method with previously published results on Phototourism dataset in Table 3. Our method obtains the best performance, while this is achieved with the supervised whitening which is much faster to learn than CNN descriptors. It only takes less than 10 seconds to compute on a modern computer(4 cores, 2.6Ghz) for the *polar* + *cartes* case on the Phototourism Liberty dataset, as opposed to several hours and GPUs for the deep learning approaches.

Test			Liberty		Notredame		Yosemite	
Train	*D*	*Mean*	*No*	*Yo*	*Li*	*Yo*	*No*	*Li*
DC–S2S [35]	512	9.67	8.79	12.84	4.54	5.58	13.02	13.24
DDESC [28]	128	9.85	8.82	**8.82**	4.54	4.54	16.19	16.19
Matchnet [12]	4096	7.75	6.90	10.77	3.87	5.76	8.39	10.88
TF–M [6]	128	6.47	**7.22**	9.79	**3.12**	3.85	7.08	7.82
polar + cartes + LW	128	**5.98**	7.44	9.84	3.48	**3.54**	**5.02**	**6.56**

Table 3: Performance comparison with the state of the art on Phototourism dataset. FPR95 is reported for all methods and the best score per dataset is shown in bold.

Verification		Matching		Retrieval	
TF–R	81.92	**PCW**	**33.69**	*+TF–R*	40.23
+TF–M	82.69	*+TF–M*	34.29	*+SIFT*	40.36
PCW	**82.94**	*+TF–R*	34.37	*+RSIFT*	43.84
+DC–S2S	83.03	*+DDESC*	35.44	*+DDESC*	44.55
+TF–R	83.24	*+RSIFT*	36.77	**PCW**	**48.26**
PCW⋆	**88.64**	**PCW⋆**	**43.81**	**PCW⋆**	**61.21**

Verification		Matching		Retrieval	
DC–S	70.04	*RSIFT*	27.22	*DC–S2S*	34.76
DC–S2S	78.23	*DC–S2S*	27.69	*DC–S*	34.84
DDESC	79.51	*DDESC*	28.05	*TF–R*	37.69
TF–M	81.90	*TF–R*	30.61	*TF–M*	39.40
TF–R	81.92	*TF–M*	32.64	*DDESC*	39.83
PCW	**82.94**	**PCW**	**33.69**	**PCW**	**48.26**

Table 4: Best performing methods on HP dataset. On the left we compare all methods, while on the right only methods that have **not** used any part of HPatches for training. The "+" refers to ZCA used in [6]. Our method is noted by PCW (*polar* + *cartes* + LW) and shown in bold, while training whitening on a subset of HP is denoted by ⋆. Otherwise it is trained on Liberty (PT). Previously top performing methods are DC-S2S [35], DDESC [28], TF [6], and RSIFT [2]. Top 6 methods per task are ranked and shown. Full list of methods in [6].

The comparison on the HPatches dataset is reported in Table 4. On the left all methods are considered, independently whether the splits of HPatches have been used for training or not. The table on the right compares only those methods that have **not** used any part of HPatches for training. In this case, the post-processing (LW) of our method was learned on Phototourism Liberty, as done in [6] so that the numbers are directly comparable. Note that the proposed method trained on Phototourism Liberty scores high even among the methods that used the split of HPatches in training.

6 Conclusions

We have proposed a multiple-kernel local-patch descriptor combining two parametrizations of gradient position and direction. Each parametrization provides robustness to a different type of patch miss-registration: polar parametrization for noise in the dominant orientation, Cartesian for imprecise location of the feature point. Learning a discriminative whitening implicitly sets the relative weight between the two representations. The proposed method consistently outperforms prior methods on two datasets and three tasks.

Acknowledgments The authors were supported by the MSMT LL1303 ERC-CZ grant, Arun Mukundan was supported by the CTU student grant SGS17/185/OHK3/3T/13.

References

[1] Mitsuru Ambai and Yuichi Yoshida. Card: Compact and real-time descriptors. In *ICCV*, 2011.

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

[3] Artem Babenko and Victor Lempitsky. Aggregating deep convolutional features for image retrieval. In *ICCV*, 2015.

[4] Vassileios Balntas, Edward Johns, Lilian Tang, and Krystian Mikolajczyk. Pn-net: conjoined triple deep network for learning local image descriptors. In *arXiv*, 2016.

[5] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *BMVC*, 2016.

[6] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, 2017.

[7] Liefeng Bo and Cristian Sminchisescu. Efficient match kernels between sets of features for visual recognition. In *NIPS*, 2009.

[8] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Kernel descriptors for visual recognition. In *NIPS*, December 2010.

[9] Andrei Bursuc, Giorgos Tolias, and Hervé Jégou. Kernel local descriptors with implicit rotation matching. In *ICMR*, 2015.

[10] M. Calonder, Vincent Lepetit, C. Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *ECCV*, 2010.

[11] Jan-Michael Frahm, Pierre Fite-Georgel, David Gallup, Tim Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, et al. Building rome on a cloudless day. In *ECCV*, 2010.

[12] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *CVPR*, 2015.

[13] Hervé Jégou and Ondrej Chum. Negative evidences and co-occurrences in image retrieval: The benefit of PCA and whitening. In *ECCV*, 2012.

[14] Takumi Kobayashi and Nobuyuki Otsu. Image feature extraction using gradient local auto-correlations. In *ECCV*, 2008.

[15] Iasonas Kokkinos and Alan Yuille. Scale invariance without scale selection. In *CVPR*, 2008.

[16] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. A sparse texture representation using local affine regions. *IEEE Trans. PAMI*, 27(8):1265–1278, 2005.

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

[18] Krystian Mikolajczyk and Jiri Matas. Improving descriptors for fast tree matching by optimal linear projection. In *ICCV*, 2007.

[19] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Trans. PAMI*, 27(10):1615–1630, 2005.

[20] Dmytro Mishkin, Jiri Matas, Michal Perdoch, and Karel Lenc. Wxbs: Wide baseline stereo generalizations. In *arXiv*, 2015.

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

[22] Mattis Paulin, Matthijs Douze, Zaid Harchaoui, Julien Mairal, Florent Perronin, and Cordelia Schmid. Local convolutional features with unsupervised training for image retrieval. In *ICCV*, 2015.

[23] James Philbin, Michael Isard, Josef Sivic, and Andrew Zisserman. Descriptor learning for efficient retrieval. In *ECCV*, 2010.

[24] Filip Radenović, Giorgos Tolias, and Ondřej Chum. CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. In *ECCV*, 2016.

[25] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, 2011.

[26] Johannes Lutz Schönberger, Filip Radenović, Ondrej Chum, and Jan-Michael Frahm. From single image query to detailed 3D reconstruction. In *CVPR*, 2015.

[27] Johannes Lutz Schönberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *CVPR*, 2017.

[28] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *ICCV*, 2015.

[29] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Learning local feature descriptors using convex optimisation. Technical report, Department of Engineering Science, University of Oxford, 2013.

[30] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Trans. PAMI*, 32(5):815–830, 2010.

[31] Giorgos Tolias, Andrei Bursuc, Teddy Furon, and Hervé Jégou. Rotation and translation covariant match kernels for image retrieval. *CVIU*, 140:9–20, 2015.

[32] Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. In *CVPR*, 2010.

[33] Peng Wang, Jingdong Wang, Gang Zeng, Weiwei Xu, Hongbin Zha, and Shipeng Li. Supervised kernel descriptors for visual recognition. In *CVPR*, 2013.

[34] Simon Winder and Matthew Brown. Learning local image descriptors. In *CVPR*, 2007.

[35] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, 2015.