Learning Local Feature Descriptors Using Convex Optimisation

Karen Simonyan, Andrea Vedaldi and Andrew Zisserman

Abstract—The objective of this work is to learn descriptors suitable for the sparse feature detectors used in viewpoint invariant matching. We make a number of novel contributions towards this goal. First, it is shown that learning the pooling regions for the descriptor can be formulated as a *convex* optimisation problem selecting the regions using sparsity. Second, it is shown that descriptor dimensionality reduction can also be formulated as a *convex* optimisation problem, using Mahalanobis matrix nuclear norm regularisation. Both formulations are based on discriminative large margin learning constraints. As the third contribution, we evaluate the performance of the compressed descriptors, obtained from the learnt real-valued descriptors by binarisation. Finally, we propose an extension of our learning formulations to a weakly supervised case, which allows us to learn the descriptors from unannotated image collections. It is demonstrated that the new learning methods improve over the state of the art in descriptor learning on the annotated local patches dataset of Brown et al. [3] and unannotated photo collections of Philbin et al. [22].

Index Terms—Descriptor learning, feature descriptor, binary descriptor, dimensionality reduction, sparsity, nuclear norm, trace norm, feature matching, image retrieval

1 Introduction

Feature descriptors are an important component of many computer vision algorithms. In large scale matching, such as the Photo Tourism project [27], and large scale image retrieval [21], the discriminative power of descriptors and their robustness to image distortions are a key factor in the performance. During the last two decades a plethora of descriptors have been developed, with SIFT [15] certainly being the most widely used. Most of these methods are hand-crafted, though recently machine learning techniques have been applied to learning descriptors for wide-baseline matching [2], [3], [31], [32] and image retrieval [22]. However, although these methods succeed in improving over the performance of SIFT, they use non-convex learning formulations and this can result in sub-optimal models being learnt.

In this paper we propose a novel framework that, by leveraging on recent powerful methods for large scale learning of sparse models, can learn descriptors more effectively than previous techniques. The contribution of the paper is four-fold.

First, we reformulate the learning of the *configuration* of the spatial pooling regions of a descriptor as the problem of selecting a few regions among a large set of candidate ones (Sect. 4). The significant advantage compared to previous approaches is that selection can be performed by optimising a sparsity-inducing L_1 regulariser, yielding a *convex* problem and ultimately a globally-optimal solution.

Second, we propose to *reduce dimensionality* as well as *improve discrimination* of the descriptors by learning a low-rank metric through penalising the nuclear norm of the Mahalanobis matrix (Sect. 5). The nuclear norm is the equivalent of an L_1 regulariser for subspaces. The advantage on standard techniques such as PCA is the fact that the low-rank subspace is learnt discriminatively to optimise the matching quality, while yielding a convex problem and a globally optimal solution. The learning of the pooling regions and of the discriminative projections are formulated as large-scale max-margin learning problems with sparsity enforcing regularisation terms. In order to optimise such objectives efficiently, we employ an effective stochastic learning technique [36] (Sect. 7).

Third, we show that our learnt low-dimensional real-valued descriptors are amenable to *binarisation* technique based on the Parseval tight frame expansion [10] (linear projection of a specific form) to a *higher*-dimensional space, followed by thresholding (Sect. 8). By changing the space dimensionality, we can explore the trade-off between the binary code length and discriminative ability. The resulting binary descriptors have a low memory footprint, are very fast to match, and achieve state-of-the-art performance.

Finally, we extend our descriptor learning framework to the case of extremely *weak supervision* (Sect. 9), where learning is performed from unannotated image collections. In that case, we rely on automatically estimated homographies, similarly to [22]. We differ in that the problem of ambiguous feature matches (e.g. due to repetitive structure and occlusions) is tackled in a more principled way using the latent variables formalism.

The result is that we have a principled, flexible, and convex framework for descriptor learning which produces both real-valued and binary descriptors with state-of-the-art performance. As we demonstrate in the experiments of Sect. 10, the proposed method outperforms

• *The authors are with the Visual Geometry Group, Department of Engineering Science, University of Oxford, Oxford, OX1 3PJ, UK. E-mail: {karen,vedaldi,az}@robots.ox.ac.uk*

state-of-the-art real-valued descriptors [1], [3], [22], [32] and binary descriptors [2], [31] on two challenging datasets. Furthermore, the descriptor learning is efficient and is able to complete within a few hours on a single core for very large scale problems.

In Sect. 10 we also demonstrate that the choice of affine-covariant region detection method and its parameters strongly affects the image retrieval performance. Namely, we achieve a substantial improvement over the baseline [1], [21] by simply changing the parameters of the Hessian-Affine feature region detector [18] as well as replacing it with affine-adapted DoG detector [15].

This paper extends our earlier work [25] as follows. First, we incorporate descriptor compression into our descriptor computation pipeline, which can be carried out using product quantisation or binarisation. For the latter, we employ the method based on the Parseval tight frame expansion [10] (Sect. 8) and show that the resulting binary descriptors achieve state-of-the-art performance. Second, we substantially expand the description of the method for learning directly from weakly supervised image collections (Sect. 9). Third, we employ a significantly stronger image retrieval baseline, obtained by using affine-adapted DoG regions with a large descriptor measurement region size, and demonstrate that the proposed learning framework still leads to a considerable performance increase (Sect. 10.2). Fourth, we estimate the patch normalisation factor in a more robust manner (Sect. 4), which improves the performance.

2 Related work

The proposed descriptor learning framework consists of two independent algorithms, one for learning descriptor pooling regions, and the other for discriminative dimensionality reduction. Most conventional feature descriptors are hand-crafted and use a fixed configuration of pooling regions, e.g. SIFT [15] and its derivatives [1], [28] use rectangular regions organised in a grid, while DAISY [29] employs a set of multi-size circular regions grouped into rings. In [3] the Powell optimisation technique is employed to find the parameters of a DAISY-like descriptor. The corresponding objective is not convex, making the optimisation prone to local extrema. Recently, pooling region selection using boosting was proposed in [31], [32]. Since the optimisation is greedy, there is no guarantee to reach the global optimum.

Discriminative dimensionality reduction can also be related to metric learning, on which a vast literature exists. Of particular relevance here are the large margin formulations designed for nearest-neighbour classification, such as [35], the reason being that feature matching is usually performed by nearest-neighbour search in the descriptor space. While our ranking constraints are similar to those of [35], the authors themselves do not consider simultaneous dimensionality reduction. One approach to reducing dimension is to optimise directly over a projection matrix of the required size [8], [30], but this leads to non-convex objectives. A similar formulation with application to learning descriptors for image retrieval was used in [22]. In [32] dimensionality reduction is performed using the projections corresponding to the largest eigenvalues of the learnt Mahalanobis matrix. This method is ad hoc as the dimensionality reduction is not taken into account in the learning objective. In our case, we enforce a low rank of the Mahalanobis matrix by penalising its nuclear norm, which is a convex surrogate of the matrix rank. In [23], the nuclear norm was used for the max-margin matrix factorisation, but their implementation resorted to smooth surrogates to simplify the optimisation. We tackle the optimisation problem in a principled way and perform large-scale optimisation of the non-smooth objective using the recently developed Regularised Dual Averaging (RDA) method [20], [36], which we employ for both L_1-regularised learning of pooling regions and nuclear norm regularised learning of discriminative dimensionality reduction.

Binary descriptors have recently attracted much attention [4], [14], [28], [31], [33] due to the low memory footprint and very fast matching times (especially when computing the Hamming distance on the modern CPUs). BRIEF [4] and BRISK [14] descriptors are computed by comparing intensity values at patch locations, which are either randomly selected [4] or hand-crafted [14]. A different approach was used in LDAHash [28], where the binary descriptor is computed by thresholding the SIFT descriptor projected onto a subspace using a learnt projection matrix. Instead of SIFT, [33] used the vectorised image patch. The binarisation algorithm [10], employed in this paper, also performs a linear transformation followed by thresholding. It is thus related to Locality Sensitive Hashing (LSH) through random projections [5] and Iterative Quantisation (ITQ) [7]. It differs in that the binary code length is higher than the original descriptor dimensionality, and the projection matrix forms a Parseval tight frame [12]. It should be noted that apart from binarisation, popular descriptor compression methods, such as Vector Quantisation (VQ) [26] and Product Quantisation (PQ) [9], can be readily applied to our descriptors and are evaluated in Sect. 10.

3 Descriptor computation pipeline

We begin with the outline of our descriptor computation pipeline, which is reminiscent of [3]. The input is an image patch $\mathbf{x}$ which is assumed to be pre-rectified with respect to affine deformation and dominant orientation. A compact discriminative descriptor $\Psi(\mathbf{x})$ of the patch is computed from the local gradient orientations through the following steps:

Gradient orientation binning. First, Gaussian smoothing is applied to the patch $\mathbf{x}$. Then the intensity gradient is computed at each pixel and soft-assigned to the two closest orientation bins, weighted by the gradient magnitude as in [3], [15], [29]. This results in p feature channels for the patch, where p is the number of contrast-sensitive

orientation bins covering the $[0; 2\pi]$ range (we used $p = 8$ as in SIFT).

Spatial pooling and normalisation. The oriented gradients computed at the previous step are spatially aggregated via convolution with a set of kernels (e.g. Gaussian or box filters, normalised to a unit mass) with different location and spatial support (Sect. 4); we refer to them as descriptor *Pooling Regions* (PR). Pooling is applied separately to each feature channel, which results in the descriptor vector $\widetilde{\phi}(\mathbf{x})$ with dimensionality pq, where q is the number of PRs. The output of each filter is divided by the normalisation factor $T(\mathbf{x})$ and thresholded to obtain responses $\phi(\mathbf{x})$ invariant to intensity changes and robust to outliers.

Discriminative dimensionality reduction. After pooling, the dimensionality of the descriptor $\phi(\mathbf{x})$ is reduced by projection onto a lower-dimensional subspace using the matrix W learnt to improve descriptor matching (Sect. 5). The resulting descriptor $\Psi(\mathbf{x}) = W\phi(\mathbf{x})$ can be used in feature matching directly, quantised [9], [26] or binarised (Sect. 8).

4 Learning Pooling Regions

In this section, we present a framework for learning pooling region configurations. First, a large pool of putative PRs is created, and then sparse learning techniques are used to select an optimal configuration of a few PRs from this pool.

The candidate PRs are generated by sampling a large number of PRs of different size and location within the feature patch. In this paper, we mostly consider reflection-symmetric PR configurations, with each PR being a unit-mass isotropic Gaussian kernel

$$k(u, v; \rho, \alpha, \sigma) \sim \exp\left(-\frac{(u - \rho\cos\alpha)^2 + (v - \rho\sin\alpha)^2}{2\sigma^2}\right) \quad (1)$$

where (ρ, α) are the polar coordinates of the centre of the Gaussian relative to the centre of the patch and σ is the Gaussian standard deviation. As shown in Fig. 1, the candidate pooling regions ρ, α, σ are obtained by sampling the parameters in the ranges: $\rho \in [0; \rho_0]$ (half-pixel step), $\alpha \in [0, 2\pi)$ (step of $\pi/16$), $\sigma \in [0.5; \rho_0]$ (half-pixel step), and then reflecting the resulting PRs (ρ_0 is the patch radius).

Rather than working with individual PRs $(\rho_j, \alpha_j, \sigma_j)$, $j = 1, \ldots, M$, these are grouped by symmetry into *rings* Ω of regions that will be either all selected or discarded. Assuming that the detector choses a natural orientation for the image patch (e.g. the direction parallel or orthogonal to an edge), it is natural to consider rings symmetric with respect to vertical, horizontal, and diagonal flips. Of the 32 regions of equal ρ and σ, this results in two groups of four regions and three groups of eight regions, for a total of five rings (Fig. 1). $\rho = 0$ is a special case that has only one pooling region. Since there is a set of five rings for each choice of ρ and σ, the total number of rings is still fairly large, but significantly smaller than the number of individual regions. For example, in Sect. 10.2 the number of candidate rings $\Omega_1, \ldots, \Omega_N$ for 31×31 patches is $N = 4650$.

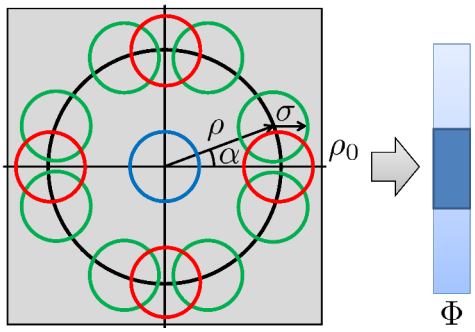

Fig. 1. Pooling region candidate rings. The blue circle shows a ring of a single PR, the red circles – four PRs, the green circles – eight PRs. Each ring corresponds to a sub-vector in the descriptor Φ (shown on the right).

Selecting pooling regions. This paragraph shows how to select a few PR rings from the N available candidates such that the resulting descriptor discriminates between *positive* (correctly matched) and *negative* (incorrectly matched) feature pairs. More formally, let ϕ be the descriptor defined by PRs pool subset encoded by the w vector:

$$\phi_{i,j,c}(\mathbf{x}) = \sqrt{w_i}\Phi_{i,j,c}(\mathbf{x}) \quad (2)$$

where $\Phi(\mathbf{x})$ is the "full" descriptor, induced by **all** PRs from the pool $\{\Omega_i\}$ (i indexes over PR rings Ω_i, j is a PR index within the ring Ω_i, and c is the feature channel number). The elements of w are non-negative, with non-zero elements acting as weights for the PR rings selected from the pool (and zero weights corresponding to PR rings that are not selected). Due to the symmetry of PR configuration, a single weight w_i is used for all PRs in a ring Ω_i. As we will see below, the square root of w_i is taken to ensure the linearity of the squared descriptor distance with respect to w.

We put the following margin-based constraints on the distance between feature pairs in the descriptor space [35]:

$$d(\mathbf{x}, \mathbf{y}) + 1 < d(\mathbf{u}, \mathbf{v}) \quad \forall(\mathbf{x}, \mathbf{y}) \in \mathcal{P}, (\mathbf{u}, \mathbf{v}) \in \mathcal{N} \quad (3)$$

where $\mathcal{P}$ and $\mathcal{N}$ are the training sets of positive and negative feature pairs, and $d(\mathbf{x}, \mathbf{y})$ is the distance between descriptors of features $\mathbf{x}$ and $\mathbf{y}$. To measure the distance, the squared L_2 distance is used (at this point we do not consider descriptor dimensionality reduction):

$$\begin{aligned} d(\mathbf{x}, \mathbf{y}) &= \|\phi(\mathbf{x}) - \phi(\mathbf{y})\|_2^2 = \\ &\sum_{i,j,c} \left(\sqrt{w_i}\Phi_{i,j,c}(\mathbf{x}) - \sqrt{w_i}\Phi_{i,j,c}(\mathbf{y})\right)^2 = \\ &\sum_i w_i \sum_{j,c} \left(\Phi_{i,j,c}(\mathbf{x}) - \Phi_{i,j,c}(\mathbf{y})\right)^2 = \end{aligned} \quad (4)$$

$$\sum_i w_i \psi_i(\mathbf{x}, \mathbf{y}) = w^T \psi(\mathbf{x}, \mathbf{y}),$$

where $\psi(\mathbf{x}, \mathbf{y})$ is an N-dimensional vector storing in the i-th element sums of squared differences of descriptor components corresponding to the ring Ω_i:

$$\psi_i(\mathbf{x}, \mathbf{y}) = \sum_{j,c} \left(\Phi_{i,j,c}(\mathbf{x}) - \Phi_{i,j,c}(\mathbf{y})\right)^2 \ \forall i = 1 \dots N \quad (5)$$

Now we are set to define the learning objective for PR configuration learning. Substituting (4) into (3) and using the soft formulation of the constraints, we derive the following non-smooth *convex* optimisation problem:

$$\arg\min_{w \geq 0} \sum_{\substack{(\mathbf{x},\mathbf{y}) \in \mathcal{P} \\ (\mathbf{u},\mathbf{v}) \in \mathcal{N}}} \mathcal{L}\left(w^T \left(\psi(\mathbf{x}, \mathbf{y}) - \psi(\mathbf{u}, \mathbf{v})\right)\right) + \mu_1 \|w\|_1 \quad (6)$$

where $\mathcal{L}(z) = \max\{z + 1, 0\}$ is the hinge loss, and the L_1 norm $\|w\|_1$ is a sparsity-inducing regulariser which encourages the elements of w to be zero, thus performing PR selection. The parameter $\mu_1 > 0$ sets a trade-off between the empirical ranking loss and sparsity. We note that "sparsity" here refers to the number of PRs, not their location within the image patch, where they are free to overlap. The formulation (6) can be seen as an instance of SVM-rank formulation, which maximises the area under the ROC curve corresponding to thresholding the descriptor distance [11]. However, due to the L_1 regularisation and non-negativity constraints, conventional SVM solvers are not readily applicable to optimising the objective (6). The algorithm for its large-scale optimisation is described in Sect. 7.

During training, all PRs from the candidate rings are used to compute the vectors $\psi(\mathbf{x}, \mathbf{y})$ for training feature pairs $(\mathbf{x}, \mathbf{y})$. While storing the full descriptor Φ is not feasible for large training sets due to its high dimensionality (which equals $n_0 = p \sum_{i=1}^{N} |\Omega_i|$, i.e. the number of channels times the number of PRs in the pool) the vector ψ is just N-dimensional, and can be computed in advance before learning w.

Descriptor normalisation and cropping. Once a sparse w is learnt, at test time only PRs corresponding to the non-zero elements of w are used to compute the descriptor. This brings up the issue of descriptor normalisation, which should be consistent between training and testing to ensure good generalisation. The conventional normalisation by the norm of the pooled descriptor $\widetilde{\phi}$ would result in different normalisation factors, since the whole PR pool is used during training, but only a (learnt) subset of PRs – in testing. Here we explain how to compute the descriptor normaliser $T(\mathbf{x})$ which does not depend on PRs. This ensures that in both training and testing the same normalisation is applied, even though different sets of PRs are used.

Before normalisation, the descriptor $\widetilde{\phi}(\mathbf{x})$ is essentially a spatial convolution of gradient magnitudes distributed across orientation bins. Such a descriptor is invariant to an additive intensity change, but it does vary with intensity scaling. To cancel out this effect, a suitable normalisation factor $T(\mathbf{x})$ can be computed from the patch directly, independently of the PR configuration. Here, we set $T(\mathbf{x})$ to the ζ-quantile of gradient magnitude distribution over the patch. Given $T(\mathbf{x})$, the response of each PR is normalised and cropped to 1 for each PR independently as follows:

$$\phi_i(\mathbf{x}) = \min\left\{\widetilde{\phi}_i(\mathbf{x})/T(\mathbf{x}), 1\right\} \ \forall i. \quad (7)$$

We employ the quantile statistic to estimate the threshold value such that only a small ratio of pixels have the gradient magnitude larger than it. These pixels potentially correspond to high-contrast or overexposed image areas, and to limit the effect of such areas on the descriptor distance, the corresponding gradient magnitude is cropped (thresholded). The thresholding quantile value $\zeta = 0.8$ was estimated on the validation set. An alternative way of computing the threshold $T(\mathbf{x})$ is to use the sum of the gradient magnitude mean and variance, as done in [25]. In this work, we use a more robust quantile statistic, which leads to slight performance improvement, compared to [25]. As a result of the normalisation and cropping procedure, the descriptor $\phi(\mathbf{x})$ is invariant to affine intensity transformation, and robust to abrupt gradient magnitude changes.

5 Learning discriminative dimensionality reduction

This section proposes a framework for learning discriminative dimensionality reduction. The aim is to learn a linear projection matrix W such that (i) W projects descriptors onto a lower dimensional space; (ii) positive and negative descriptor pairs are separated by a margin in that space.

The first requirement can be formally written as $W \in \mathbb{R}^{m \times n}, m < n$ where m is the dimensionality of the projected space and n is the descriptor dimensionality before projection. The second requirement can be formalised using a set of constraints similar to (3):

$$d_W(\mathbf{x}, \mathbf{y}) + 1 < d_W(\mathbf{u}, \mathbf{v}) \quad \forall (\mathbf{x}, \mathbf{y}) \in \mathcal{P}, (\mathbf{u}, \mathbf{v}) \in \mathcal{N} \quad (8)$$

where d_W is the squared L_2 distance in the projected space:

$$d_W(\mathbf{x}, \mathbf{y}) = \|W\phi(\mathbf{x}) - W\phi(\mathbf{y})\|_2^2 = \quad (9)$$
$$\left(\phi(\mathbf{x}) - \phi(\mathbf{y})\right)^T W^T W \left(\phi(\mathbf{x}) - \phi(\mathbf{y})\right) = \theta(\mathbf{x}, \mathbf{y})^T A \theta(\mathbf{x}, \mathbf{y}),$$

with $\theta(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) - \phi(\mathbf{y})$, and $A = W^T W$ is the Mahalanobis matrix.

The constraints (8), (9) are not convex in W, but are convex in A. Therefore, optimisation is performed over the convex cone of positive semi-definite matrices [35]: $A \in \mathbb{R}^{n \times n}$, $A \succeq 0$. The positive semidefiniteness constraint ensures that optimising over A is equivalent to optimising over W, i.e. for the learnt matrix A there exists a projection matrix W such that $A = W^T W$.

The dimensionality reduction constraint on W can be equivalently transformed into a rank constraint on A. Indeed, if $\mathrm{rank}(A) = m$, then an $m \times n$ matrix W can be obtained from the eigen-decomposition $A = VDV^T$, where diagonal matrix $D \in \mathbb{R}^{n\times n}$ has m non-zero elements (positive eigenvalues). Let $D_r \in \mathbb{R}^{m\times n}$ be the matrix obtained by removing the zero rows from D. Then W can be constructed as $W = \sqrt{D_r}V^T$. Conversely, if $W \in \mathbb{R}^{m\times n}$ and $\mathrm{rank}(W) = m$, then $\mathrm{rank}(A) = \mathrm{rank}(W^TW) = \mathrm{rank}(W) = m$. However, the direct optimisation of $\mathrm{rank}(A)$ is not tractable due to its non-convexity. The convex relaxation of the matrix rank is described next.

Nuclear norm regularisation. The nuclear norm $\|A\|_*$ of matrix A (also referred to as the trace norm) is defined as the sum of singular values of A. For positive semi-definite matrices the nuclear norm equals the trace. The nuclear norm performs a similar function to the L_1 norm of a vector – the L_1 norm of a vector is a convex surrogate of its L_0 norm, while the nuclear norm of a matrix is a convex surrogate of its rank [6].

Using the soft formulation of the constraints (8), (9) and the nuclear norm in place of rank, we obtain the non-smooth *convex* objective for learning A:

$$\arg\min_{A\succeq 0} \sum_{\substack{(\mathbf{x},\mathbf{y})\in\mathcal{P}\\(\mathbf{u},\mathbf{v})\in\mathcal{N}}} \mathcal{L}\left(\theta(\mathbf{x},\mathbf{y})^T A\theta(\mathbf{x},\mathbf{y}) - \theta(\mathbf{u},\mathbf{v})^T A\theta(\mathbf{u},\mathbf{v})\right) + \mu_*\|A\|_*, \tag{10}$$

where the parameter $\mu_* > 0$ trades off the empirical ranking loss versus the dimensionality of the projected space: the larger μ_*, the smaller the dimensionality. We note that this formulation gives no *direct* control over the projected space dimensionality. Instead, the dimension can be tuned by running the optimisation with different values of μ_*.

6 Discussion

Our descriptor learning algorithm includes two stages: learning a sparse pooling region configuration (Sect. 4) and learning a low-rank projection for the selected PRs (Sect. 5). It is natural to consider whether the two learning stages can be combined into a joint optimisation problem. Indeed, selecting a small set of PR rings and, simultaneously, performing their dimensionality reduction corresponds to projecting the full descriptor $\Phi \in \mathbb{R}^{n_0}$ (2) with a rectangular matrix $V \in \mathbb{R}^{m\times n_0}, m \ll n_0$, which has a special structure. Namely, to select only a few PR rings from the pool, V must have a column-wise group sparsity pattern, such that the group of columns, corresponding to the i-th PR ring, can only be set to zero all together (indicating that the i-th ring is not selected from the candidate pool).

Unfortunately, the optimisation over the projection matrix V is large-scale (the number of parameters $mn_0 \approx 19M$ for $m = 64$ and $n_0 \approx 298K$) and non-convex (Sect. 5). A convex optimisation of the corresponding Mahalanobis matrix $B = V^TV \in \mathbb{R}^{n_0\times n_0}$ would incur learning $n_0^2 \approx 89 \cdot 10^9$ parameters under non-trivial group sparsity constraints, which is computationally challenging.

Instead, we factorise the projection V as $V = WV_{PR}$, where $V_{PR} \in \mathbb{R}^{n\times n_0}$ is a rectangular diagonal matrix, induced by the PR-selecting sparse vector $w \in R^N, N = 4650$ (Sect. 4), and $W \in \mathbb{R}^{m\times n}$ is further reducing the dimensionality of the selected PRs (Sect. 5). Even though the sequential learning of w and W is sub-optimal, it results in two convex optimisation problems, which are easy to solve.

7 Regularised stochastic learning

In sections 4 and 5 we proposed convex optimisation formulations for learning the descriptor PRs as well as the discriminative dimensionality reduction. However, the corresponding objectives (6) and (10) yield very large problems as the number of summands is $|\mathcal{P}|\,|\mathcal{N}|$, where typically the number of positive and negative matches is in the order of 10^5 – 10^6 (Sect. 10). This makes using conventional interior point methods infeasible.

To handle such very large training sets, we propose to use *Regularised Dual Averaging* (RDA), the recent method by [20], [36]. To the best of our knowledge, RDA has not yet been applied in the computer vision field, where, we believe, it could be used in a variety of applications beyond the one presented here. RDA is a stochastic proximal gradient method effective for problems of the form

$$\min_w \frac{1}{T}\sum_{t=1}^{T} f(w, z_t) + R(w) \tag{11}$$

where w is the weight vector to be learnt, z_t is the t-th training $(\mathrm{sample}, \mathrm{label})$ pair, $f(w, z)$ is a convex loss, and $R(w)$ is a convex regularisation term. Compared to proximal methods for optimisation of smooth losses with non-smooth regularisers (e.g. FISTA), RDA is more generic and applicable to *non-smooth* losses, such as the hinge loss employed in our framework. As opposed to other stochastic proximal methods (e.g. FOBOS), RDA uses more aggressive thresholding, thus producing solutions with higher sparsity. A detailed description of RDA can be found in [36]; here we provide a brief overview.

At iteration t RDA uses the loss subgradient $g_t = \partial f(w, z_t)/\partial w$ to perform the update:

$$w_{t+1} = \arg\min_w \left(\langle \bar{g}_t, w\rangle + R(w) + \frac{\beta_t}{t}h(w)\right) \tag{12}$$

where $\bar{g}_t = \frac{1}{t}\sum_{i=1}^{t} g_i$ is the average subgradient, $h(w)$ is a strongly convex function such that $\arg\min_w h(w)$ also minimises $R(w)$, and β_t is a specially chosen non-negative non-decreasing sequence. We point out that $\bar{g}_t$ is computed by averaging subgradients across iterations, rather than samples (similarly to gradient descent with

the momentum term). If the regularisation $R(w)$ is not strongly convex (as in the case of L_1 and nuclear norms), one can set $h(w) = \frac{1}{2}\|w\|_2^2$, $\beta_t = \gamma\sqrt{t}$, $\gamma > 0$ to obtain the convergence rate of $O(1/\sqrt{t})$.

It is easy to derive the specific form of the RDA update step for the objectives (6) and (10):

$$w_{t+1}^{(i)} = \max\left\{-\frac{\sqrt{t}}{\gamma}\left(\bar{g}^{(i)} + \mu_1\right), 0\right\} \quad (13)$$
$$A_{t+1} = \Pi\left(-\frac{\sqrt{t}}{\gamma}\left(\bar{g} + \mu_* \mathbb{I}\right)\right)$$

where $\bar{g}$ is the average subgradient of the corresponding hinge loss, $\mathbb{I}$ is the identity matrix and Π is the projection onto the cone of positive semi-definite matrices, computed by cropping negative eigenvalues in the eigen-decomposition.

8 Binarisation

In this section we describe how a low-dimensional real-valued descriptor $\Psi \in \mathbb{R}^m$ can be binarised to a code $\beta \in \{0,1\}^q$ with the bit length q *higher* or equal to m. To this end, we adopt the method of [10], which is based on the descriptor expansion using a Parseval tight frame, followed by thresholding (taking the sign).

In more detail, a *frame* is a set of $q \geq m$ vectors generating the space of descriptors $\Psi \in \mathbb{R}^m$ [12]. In the matrix form, a frame can be represented by a matrix $U \in \mathbb{R}^{q \times m}$ composed of the frame vectors as rows. A *Parseval tight frame* has the additional property that $U^\top U = \mathbb{I}$. An expansion with such frames, $U\Psi \in \mathbb{R}^q$, is an overcomplete representation of $\Psi \in \mathbb{R}^m$, which preserves the Euclidean distance. Due to the overcompleteness, binarisation of the expanded vectors leads to a more accurate approximation of the original vectors Ψ. Assuming that the descriptors Ψ are zero-centred, the binarisation is performed as follows:

$$\beta = \mathrm{sgn}(U\Psi), \quad (14)$$

where sgn is the sign function: $\mathrm{sgn}(a) = 1$ iff $a > 0$ and 0 otherwise. Following [10], we compute the Parseval tight frame U by keeping the first m columns of an orthogonal matrix obtained from a QR-decomposition of a random $q \times q$ matrix.

In spite of the binary code dimensionality q being not smaller than the dimensionality m of the real-valued descriptor, the memory footprint of the binary code is smaller if $q < 32m$. Indeed, only 1 bit is required to store each dimension of a binary descriptor, while 32 bits/dimension are required for the real-valued descriptors in the IEEE single precision format. Additionally, the Hamming distance between binary descriptors can be computed very quickly using the XOR and POPCNT (population count) instructions of the modern CPUs. Changing q allows us to generate the binary descriptors with any desired bitrate $q \geq m$, balancing matching accuracy vs memory footprint.

9 Learning from unannotated image collections

In this section we describe a novel formulation for obtaining feature correspondences from image datasets using only extremely weak supervision. Together with the learning frameworks of Sect. 4 and 5 this provides an algorithm for automatically learning descriptors from such datasets. In this challenging scenario, the only information given to the algorithm is that *some* (but unknown) pairs of dataset images contain a common *part*, so that correspondences can be established between them. This assumption is valid for the image collections considered in this paper (Sect. 10.2).

One possible way of obtaining the feature correspondences for descriptor learning would be to compute the 3-D reconstruction [3] of scenes present in the dataset, but this requires a large number of images of the same scene to perform well. A more practical approach [22] relies on the homography estimation between pairs of images via Nearest-Neighbour (NN) SIFT matching and RANSAC. Then, NN inlier matches can be used as positives, and NN outliers and non-NN as negatives for descriptor learning. However, this leads to positives that can already be matched by SIFT, while our goal is to learn a better descriptor. The less biased alternative of ignoring appearance and finding correspondences based on geometry only is also problematic as it may pick up occlusions and repetitive structure, which, being unmatchable based on appearance, would disrupt learning. We address these issues by the latent variables formulation described next.

Pre-processing. This proceeds in two stages: first, homographies are established between randomly sampled image pairs [22] using SIFT descriptor matches and RANSAC; second, detected region correspondences are established between the image pairs using only the homography (not SIFT descriptors). This ensures that the resulting correspondences are independent of SIFT.

In more detail, we begin with automatic homography estimation between the random image pairs. This involves a standard pipeline [19] of: affine-covariant (elliptical) region detection, computing SIFT descriptors for the regions, and estimating an affine homography using the robust RANSAC algorithm on the putative SIFT matches. Only the pairs for which the number of RANSAC inliers is larger than a threshold (set to 50 in our experiments) are retained. Then, in stage two, for each feature $\mathbf{x}$ of the reference image (the first image of the image pair), we compute the sets $P(\mathbf{x})$ and $N(\mathbf{x})$ of putative positive and negative matches in the target image (the second image of the pair). This is done based on the homographies and the descriptor measurement region overlap criterion [19] as follows. Each descriptor measurement region (an upscaled elliptical detected region) in the target image is projected to the reference image plane using the estimated homography, resulting in an elliptical region. Then, the overlap ratio between

this region and each of the measurement regions in the reference image is used to establish the "putative positive" and "negative" matches by thresholding the ratio with high (0.6) and low (0.3) thresholds respectively. Feature matches with the region overlap ratio between the thresholds are considered ambiguous and are not used in training (see Fig. 2 for illustration).

Fig. 2. A close-up of a pair of reference (left) and target (right) images from the Oxford5K dataset. A feature region in the reference image is shown with solid blue. Its putative positive, negative, and ambiguous matches in the target image are shown on the right with green, red, and magenta respectively. Their projections to the reference image are shown on the left with dashed lines of the same colour. The corresponding overlap ratios (with the blue reference region ellipse) are: 0.74 for positive, 0.04 for negative, and 0.33 for ambiguous matches.

Learning framework. We aim at learning a descriptor such that the NN of each feature $\mathbf{x}$ is one of the positive matches from $P(\mathbf{x})$. This is equivalent to enforcing the minimal (squared) distance from $\mathbf{x}$ to the features in $P(\mathbf{x})$ to be smaller than the minimal distance to the features in $N(\mathbf{x})$:

$$\min_{\mathbf{y}\in P(\mathbf{x})} d_\eta(\mathbf{x},\mathbf{y}) < \min_{\mathbf{u}\in N(\mathbf{x})} d_\eta(\mathbf{x},\mathbf{u}), \tag{15}$$

where for brevity η denotes the descriptor parameters, such as PR weights w (Sect. 4) or the metric A (Sect. 5).

In certain cases, the reference image feature $\mathbf{x}$ can not be matched to a geometrically corresponding feature in the target image purely based on appearance. For instance, the target feature can be occluded, or the repetitive structure in the target image can make reliable matching impossible. Using such unmatchable features $\mathbf{x}$ in the constraints (15) introduces an unnecessary noise in the training set and disrupts learning. Therefore, we introduce a binary latent variable $b(\mathbf{x})$ which equals 0 iff the match can not be established. This leads to the optimisation problem:

$$\arg\min_{\eta,b,\mathbf{y}_P} \sum_{\mathbf{x}} b(\mathbf{x})\,\mathcal{L}\left(d_\eta\left(\mathbf{x},\mathbf{y}_P(\mathbf{x})\right) - \min_{\mathbf{u}\in N(\mathbf{x})} d_\eta(\mathbf{x},\mathbf{u})\right) + R(\eta) \tag{16}$$

$$\text{s.t.}\;\; \mathbf{y}_P(\mathbf{x}) = \arg\min_{\mathbf{y}\in P(\mathbf{x})} d_\eta(\mathbf{x},\mathbf{y});\; b(\mathbf{x})\in\{0,1\};\; \sum_{\mathbf{x}} b(\mathbf{x}) = K$$

where $\mathbf{y}_P(\mathbf{x})$ is the nearest-neighbour of the feature $\mathbf{x}$ among the putative positive matches $P(\mathbf{x})$, $R(\eta)$ is the regulariser (e.g. sparsity-enforcing L_1 norm or nuclear norm), and K is a hyper-parameter, which sets the number of samples to use in training and prevents all $b(\mathbf{x})$ from being set to zero.

The objective (16) is related to large margin nearest neighbour [35] and self-paced learning [13], and its local minimum can be found by alternation. Namely, with $b(\mathbf{x})$ and $\mathbf{y}_P(\mathbf{x})$ fixed for all $\mathbf{x}$, the optimisation problem (16) becomes convex (due to the convexity of $-\min$), and is solved for η using RDA (Sect. 7). Then, given η, $\mathbf{y}_P(\mathbf{x})$ can be updated; finally, given η and $\mathbf{y}_P(\mathbf{x})$, we can update $b(\mathbf{x})$ by setting it to 1 for $\mathbf{x}$ corresponding to the smallest K values of the loss $\mathcal{L}\left(d_\eta\left(\mathbf{x},\mathbf{y}_P(\mathbf{x})\right) - \min_{\mathbf{u}\in N(\mathbf{x})} d_\eta(\mathbf{x},\mathbf{u})\right)$. Each of these three steps reduces the value of the objective (16), which gives the convergence guarantee. The optimisation is repeated for different values of K, and the resulting model is selected on the validation set as the one which maximises the feature matching recall, i.e. the ratio of features $\mathbf{x}$ for which (15) holds.

10 Experiments

In this section, we evaluate the proposed descriptor learning framework on two challenging, publicly available datasets with different performance evaluation measures. In both cases, our learnt descriptors achieve state-of-the-art results.

In Sect. 10.1, we rigorously assess the components of the framework (Sect. 4, 5, 8) on the local image patches dataset [3], where feature patches are available together with the ground-truth annotation into matches and non-matches. The descriptor performance in this case is measured based on a fixed operating point on the descriptor matching ROC curve.

In Sect. 10.2, we proceed with evaluating our descriptors in a more challenging scenario, where ground-truth match annotation is not available, so an extremely weakly supervised formulation (Sect. 9) is applied. The learnt descriptors are plugged into a conventional image retrieval engine [21], and the performance is measured using retrieval-specific evaluation protocol on Oxford5K and Paris6K image collections.

10.1 Local Image Patches Dataset

10.1.1 Dataset and evaluation protocol

The dataset [3] consists of three subsets, Yosemite, Notre Dame, and Liberty, each of which contains more than 450,000 image patches (64×64 pixels) sampled around Difference of Gaussians (DoG) feature points. The patches are rectified with respect to the scale and dominant orientation. Each of the subsets was generated from a scene for which 3D reconstruction was carried out using multiview stereo algorithms. The resulting depth maps were used to generate 500,000 ground-truth feature

pairs for each dataset, with equal number of positive (correct) and negative (incorrect) matches.

To evaluate the performance of feature descriptors, we follow the evaluation protocol of [3] and generate ROC curves by thresholding the distance between feature pairs in the descriptor space. We report the false positive rate at 95% recall (FPR95) on each of the six combinations of training and test sets, as well as the mean across all combinations. Considering that in [2], [3] only four combinations were used (with training on Yosemite or Notre Dame, but not Liberty), we also report the mean for those, denoted as "mean 1–4". Following [3], for training we used 500,000 feature matches of one subset, and tested on 100,000 matches of the others. Note that training and test sets were generated from images of different scenes, so the evaluation protocol assesses the generalisation of the learnt descriptors.

10.1.2 Descriptor learning results

We compare our learnt descriptors with the state-of-the-art unsupervised [1] and supervised descriptors [2], [3], [31], [32] in three scenarios. First, we evaluate the performance of the learnt pooling regions (*PR*, Sect. 4) and compare it with the pooling regions of [3]. Second, our complete descriptor pipeline based on projected pooling regions (*PR-proj*, Sect. 4–5) is compared against other real-valued descriptors [1], [3], [32]. Finally, we assess the compression of our descriptors, for which we consider the binarisation method (*PR-proj-bin*, Sect. 8), as well as a conventional product quantisation technique [9] (*PR-proj-pq*). We compare the compressed descriptors with state-of-the-art binary descriptors [2], [31], which were shown to outperform unsupervised methods, such as BRIEF [4] and BRISK [14] as well as earlier learnt descriptors of [28], [33].

In the comparison, apart from the FPR95 performance measure, for each of the descriptors we indicate its memory footprint and type. For real-valued descriptors, we specify their dimensionality as $\langle \text{dim} \rangle$f, e.g. 64f for 64-D descriptors. Assuming that the single-precision float type is used, each real-valued descriptor requires $(32 \times \text{dim})$ bits of storage. For compressed descriptors, their bit length and type are given as $\langle \text{bits} \rangle \langle \text{type} \rangle$, where $\langle \text{type} \rangle$ is "b" for binary, and "pq" for product-quantised descriptors.

To learn the descriptors, we randomly split the set of 500,000 feature matches into 400,000 training and 100,000 validation. Training is performed on the training set for different values of μ_1, μ_* and γ, which results in a set of models with different dimensionality-accuracy tradeoff. Given the desired dimensionality of the descriptor, we pick the model with the best performance on the validation set among the ones whose dimensionality is not higher than the desired one.

Learning pooling regions. Table 1 compares the error rates reported in [3] (5-th column) with those of the PR descriptors learnt using our method. The 4-th column corresponds to the descriptors with the dimensionality limited by 384, so that it is not higher than the one used in [3]; in the 3rd column, the dimensionality was limited by 640 (a threshold corresponding to ≤ 80 PRs selected). In Fig. 3 (top) we plot the error rate of the learnt descriptors as a function of their dimensionality.

TABLE 1
False positive rate (%) (at 95% recall) for learnt pooling regions. Yos: Yosemite, ND: Notre Dame, Lib: Liberty.

Train set	Test set	PR $\leq$ 640-D	PR $\leq$ 384-D	Brown et al. [3]
Yos	ND	**9.49 (544f)**	9.88 (352f)	14.43 (400f)
Yos	Lib	**17.23 (544f)**	17.86 (352f)	20.48 (400f)
ND	Yos	11.11 (576f)	**10.91 (352f)**	15.91 (544f)
ND	Lib	**16.56 (576f)**	17.02 (352f)	21.85 (400f)
Lib	Yos	**11.89 (608f)**	12.99 (384f)	N/A
Lib	ND	**9.88 (608f)**	10.51 (384f)	N/A
mean		12.69	13.20	N/A
mean (1–4)		13.60	13.92	18.17

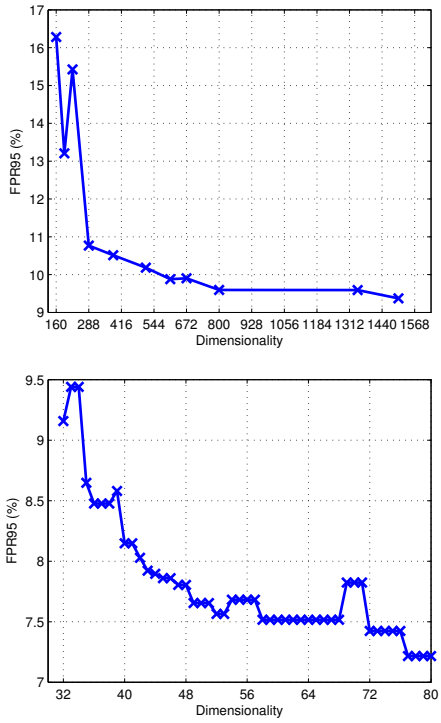

Fig. 3. Dimensionality vs error rate, training on Liberty, testing on Notre Dame. *Top*: learnt pooling regions. *Bottom*: learnt projections for 608-D PR descriptor on the top.

The PR configuration of a 576-D descriptor learnt on the Notre Dame set is depicted in Fig. 4 (left). Pooling regions are shown as circles with the radius equal to their Gaussian σ (the actual size of the Gaussian kernel is 3σ). The pooling regions' weights are colour-coded. Note that σ increases with the distance from the patch centre, which is also specific to certain hand-crafted descriptors, e.g. DAISY [29]. In our case, no prior has been put on the pooling region location and size: the PR parameters space was sampled uniformly, and the optimal configuration was automatically discovered by

TABLE 2
False positive rate (%) (at 95% recall) for real-valued descriptors. Yos: Yosemite, ND: Notre Dame, Lib: Liberty.

Train set	Test set	PR-proj ≤80-D	PR-proj ≤64-D	PR-proj ≤32-D	Brown et al. [3]	Trzcinski et al. [32]	rootSIFT [1]	rootSIFT-proj ≤80-D
Yos	ND	**6.82 (76f)**	7.11 (58f)	9.99 (32f)	11.98 (29f)	13.73 (64f)	22.06 (128f)	14.60 (77f)
Yos	Lib	**14.58 (76f)**	14.82 (58f)	16.7 (32f)	18.27 (29f)	21.03 (64f)	29.65 (128f)	22.20 (77f)
ND	Yos	**10.08 (73f)**	10.54 (63f)	13.4 (32f)	13.55 (36f)	15.86 (64f)	26.71 (128f)	19.00 (70f)
ND	Lib	**12.42 (73f)**	12.88 (63f)	14.26 (32f)	16.85 (36f)	18.05 (64f)	29.65 (128f)	20.11 (70f)
Lib	Yos	**11.18 (77f)**	11.63 (58f)	14.32 (32f)	N/A	19.63 (64f)	26.71 (128f)	19.96 (76f)
Lib	ND	**7.22 (77f)**	7.52 (58f)	9.07 (32f)	N/A	14.15 (64f)	22.06 (128f)	13.99 (76f)
mean		**10.38**	10.75	12.96	N/A	17.08	26.14	18.31
mean (1–4)		**10.98**	11.34	13.59	15.16	17.17	27.02	18.98

TABLE 3
False positive rate (%) (at 95% recall) for compressed descriptors. Yos: Yosemite, ND: Notre Dame, Lib: Liberty.

Train set	Test set	PR-proj-bin 48f→64b	PR-proj-bin 64f→128b	PR-proj-bin 80f→1024b	PR-proj-pq 64f→64pq	PR-proj-pq 80f→1024pq	Trzcinski et al. [31]	Boix et al. [2]
Yos	ND	14.37 (64b)	10.0 (128b)	**7.09 (1024b)**	12.91 (64pq)	**6.82 (1024pq)**	14.54 (64b)	8.52 (1360b)
Yos	Lib	23.48 (64b)	18.64 (128b)	**15.15 (1024b)**	20.15 (64pq)	**14.59 (1024pq)**	21.67 (64b)	15.52 (1360b)
ND	Yos	18.46 (64b)	13.41 (128b)	**8.5 (1024b)**	19.32 (64pq)	**10.07 (1024pq)**	18.97 (64b)	8.81 (1360b)
ND	Lib	20.35 (64b)	16.39 (128b)	**12.16 (1024b)**	17.97 (64pq)	**12.42 (1024pq)**	20.49 (64b)	15.6 (1360b)
Lib	Yos	24.02 (64b)	19.07 (128b)	**14.84 (1024b)**	22.11 (64pq)	**11.22 (1024pq)**	22.88 (64b)	N/A
Lib	ND	15.2 (64b)	11.55 (128b)	**8.25 (1024b)**	14.82 (64pq)	**7.22 (1024pq)**	16.90 (64b)	N/A
mean		19.31	14.84	**11.0**	17.88	**10.39**	19.24	N/A
mean (1–4)		19.17	14.61	**10.73**	17.59	**10.98**	18.92	12.11

learning (under the symmetry constraints). Even though the PR weights near the patch centre are mostly small, the contribution of the pixels in the patch centre is higher than that of the pixels further from it, as shown in Fig. 4 (right). This is explained by the fact that each Gaussian PR filter is normalised to a unit mass, so the relative contribution of pixels is higher for the filters of smaller radius (like the ones selected in the centre). Interestingly, the pattern of pixel contribution, corresponding to the learnt descriptor, resembles the Gaussian weighting employed in hand-crafted methods, such as SIFT.

In Fig. 4 (middle) we show the PR configuration learnt without the symmetry constraint, i.e. individual PRs are not organised into rings. Similarly to the symmetric configurations, the radius of PRs located further from the patch centre is larger than the radius of PRs near the centre. Also, there is a noticeable circular pattern of PR locations, especially on the left and right of the patch, which justifies our PR symmetry constraint. We note that this constraint, providing additional regularisation, dramatically reduces the number of parameters to learn: when PRs are grouped into the rings of 8, a single weight is learnt for all PRs in a ring. In other words, a single element of the w vector (Sect. 4) corresponds to 8 PRs. In the case of asymmetric configurations, each PR has its own weight, so for the same number of candidate PRs, the w vector becomes 8 times longer, which significantly increases the computational burden. In fact, the increased number of parameters makes learning more prone to over-fitting: we observed a slight increase of the error rate by relative 3% after dropping the symmetry constraint.

Learning discriminative dimensionality reduction. For dimensionality reduction experiments, we utilised learnt PR descriptors with dimensionality limited by 640 (third column in Table 1) and learnt linear projections onto lower-dimensional spaces as described in Sect. 5. In Table 2 we compare our results with the best results presented in [3] (6-th column), [32] (7-th column), as well as the unsupervised rootSIFT descriptor of [1] and its supervised projection (rootSIFT-proj), learnt using the formulation of Sect. 5 (columns 8–9). Of these four methods, the best results are achieved by [3]. To facilitate a fair comparison, we learn three types of descriptors with different dimensionality: ≤80-D, ≤64-D, ≤32-D (columns 3–5).

As can be seen, even with low-dimensional 32-D descriptors we outperform all other methods in terms of the average error rate over different training/test set combinations: 13.59% vs 15.16% for [3]. It should be noted that we obtain projection matrices by discriminative supervised learning, while in [3] the best results were achieved using PCA, which outperformed LDA in their experiments. In our case, both PCA and LDA were performing considerably worse than the learnt projection. Our descriptors with higher (but still reasonably low) dimensionality achieve even lower error rates, setting the state of the art for the dataset: 10.75% for ≤64-D, and 10.38% for ≤80-D.

In Fig. 3 (bottom) we show the dependency of the error rate on the projected space dimensionality. As can be seen, the learnt projections allow for significant (order of magnitude) dimensionality reduction, while lowering the error at the same time. In Fig. 5 (left) we visualise the learnt Mahalanobis matrix A (Sect. 5) corresponding to discriminative dimensionality reduction. It has a clear block structure, with each block corresponding to a group of pooling regions. This indicates that the

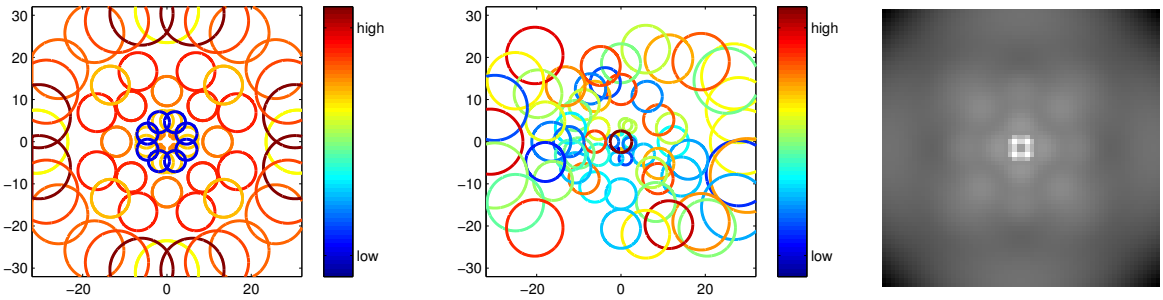

Fig. 4. *Left:* learnt symmetric pooling regions configuration in a 64×64 feature patch. *Middle:* learnt asymmetric pooling regions configuration. *Right:* relative contribution of patch pixels (computed by the weighted averaging of PR Gaussian filters using the learnt weights, shown on the left).

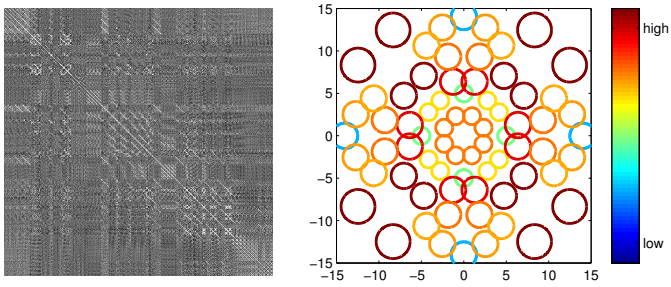

Fig. 5. *Left:* Mahalanobis matrix A (learnt on Notredame), corresponding to projection from 576-D to 73-D space (brighter pixels correspond to larger values). *Right:* Pooling region configuration learnt on Oxford5K.

dependencies between pooling regions within the same ring and across the rings are learnt together with the optimal weights for the neighbouring orientation bins within each PR.

Descriptor compression. The PR-proj descriptors evaluated above are inherently real-valued. To obtain a compact and fast-to-match representation, the descriptors can be compressed using either binarisation or product quantisation. We call the resulting descriptors PR-proj-bin and PR-proj-pq respectively, and compare them with the state-of-the-art binary descriptors of [2], [31]. The binary descriptor of [31] is low-dimensional (64-D), while [2] proposes a more accurate, but significantly longer, 1360-D, representation.

As pointed out in Sect. 8, binarisation based on frame expansion can produce binary descriptors with any desired dimensionality, as long as it is not smaller than the dimensionality of the underlying real-valued descriptor. The dependency of the mean error rate on the dimensionality is shown in Fig. 6 for PR-proj-bin descriptors computed from different PR-proj descriptors. Given a desired binary descriptor dimensionality (bit length), e.g. 64-D, it can be computed from PR-proj descriptors of different dimensionality (32-D, 48-D, 64-D in our experiments). Higher-dimensional PR-proj descriptors have better performance (Table 2), but higher quantisation error (Sect. 8) when compressed to a binary representation. For instance, compressing 48-D PR-proj descriptors to 64 bit leads to better performance than compressing 64-D PR-proj (which has higher quantisation error) or 32-D PR-proj (which has worse initial performance). In general, it can be observed (Fig. 6) that using higher-dimensional (80-D) PR-proj for binarisation consistently leads to best or second-best performance.

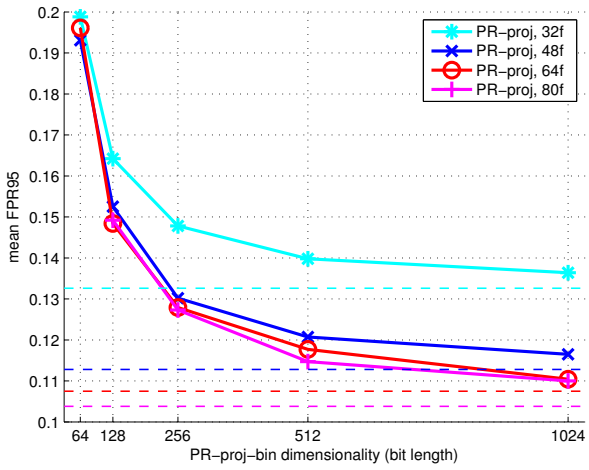

Fig. 6. Mean error rate vs dimensionality for binary PR-proj-bin descriptors computed from real-valued 32-D, 48-D, 64-D, and 80-D PR-proj descriptors. The error rates of the PR-proj descriptors are shown with dashed horizontal lines of the same colour as used for the respective binary descriptors.

In columns 3–5 of Table 3 we report the performance of our PR-proj-bin binary descriptors. The 64-bit descriptor has on average 0.07% higher error rate than the descriptor of [31], but it should be noted that they employed a dedicated framework for binary descriptor learning, while in our case we obtained the descriptor from our real-valued descriptors using a simple, but effective procedure of Sect. 8. Also, in [31] it is mentioned that learning higher-dimensional binary descriptors using their framework did not result in performance im-

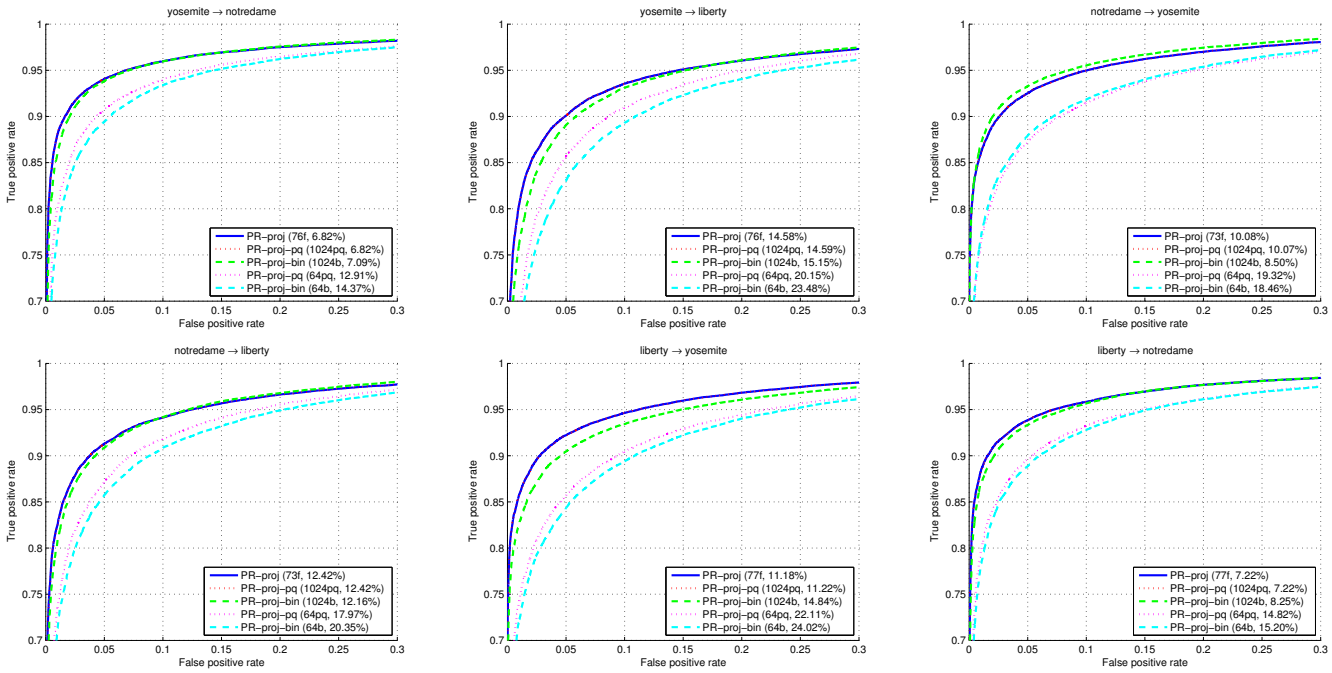

Fig. 7. Descriptor matching ROC curves for six combinations of training and test sets of the Patches dataset [3]. For each of the plots, the sets are indicated in the title as "training→test". For each of the compared descriptors, its dimensionality, type, and false positive rate at 95% recall are given in parentheses (see also Table 2 and Table 3).

provement. In our case, we can explore the "bit length – error rate" trade-off by generating a multitude of binary descriptors with different length and performance. Our 1024-bit descriptor (column 5) significantly outperforms both [31] and [2] (by 8.24% and 1.38% respectively), even though the latter use a higher-dimensional descriptor. We also note that the performance of 1024-bit PR-proj-bin descriptor is close to that of 80-D (2560 bit) PR-proj descriptor, which was used to generate it. Finally, our 128-bit PR-proj-bin descriptor provides a middle ground, with its 4.47% lower error rate than 64-bit descriptor, but still compact representation. Using LSH [5] to compress the same PR-proj descriptor to 128-bit leads to 3.07% higher error rate than frame expansion, which mirrors the findings of [10].

We also evaluate descriptor compression using (symmetric) product quantisation [9]. The error rates for the compressed 64-bit and 1024-bit PR-proj-pq descriptors are shown in columns 6–7 of Table 3. Compression using PQ is more effective than binarisation: 64-bit PR-proj-pq has 1.43% lower error than 64-bit PR-proj-bin, while 1024-bit PR-proj-pq outperforms binarisation by 0.61% and, in fact, matches the error rates of the uncompressed 80-D PR-proj descriptor (column 3 of Table 2).

While PQ compression is more effective in accuracy, in terms of the matching speed binary descriptors are the fastest: average Hamming distance computation time between a pair of 64 bit descriptors was measured to be 1.3ns (1ns=10^{-9}s) on an Intel Xeon L5640 CPU. PQ-compressed descriptors with the same 64 bit footprint (speeded-up using lookup tables) require 38.2ns per descriptor pair. For reference, SSE-optimised L_2 distance computation between 64-D single-precision vectors requires 53.5ns.

Summary. Both our pooling region and dimensionality reduction learning methods significantly outperform those of [3]. It is worth noting that the non-linear feature transform we used (Sect. 3) corresponds to the T1b block in [3]. According to their experiments, it is outperformed by more advanced (and computationally complex) steerable filters, which they employed to obtain their best results. This means that we achieve better performance with a simpler feature transform, but more sophisticated learning framework. We also achieve better results than [32], where a related feature transform was employed, but PRs and dimensionality reduction were learnt using greedy optimisation based on boosting.

Our binary descriptors, obtained from learnt low-dimensional real-valued descriptors, achieve lower error rates than the recently proposed methods [2], [31], [33], where learning was tailored to binary representation.

The ROC curves for our real-valued and compressed descriptors are shown in Fig. 7 for all combinations of training and test sets.

10.2 Oxford Buildings and Paris Buildings Datasets

In this section the proposed learning framework is evaluated on challenging Oxford Buildings (Oxford5K) and Paris Buildings (Paris6K) datasets and compared against the rootSIFT baseline [1], as well as the descriptor learning method of [22].

10.2.1 Dataset and evaluation protocol

The Oxford Buildings dataset consists of 5062 images capturing various Oxford landmarks. It was originally collected for the evaluation of large-scale image retrieval methods [21]. The only available annotation is the set of queries and ground-truth image labels, which define relevant images for each of the queries. The Paris Buildings dataset includes 6412 images of Paris landmarks and is also annotated with queries and labels. Both datasets exhibit a high variation in viewpoint and illumination.

The performance measure is specific to the image retrieval task and is computed in the following way. For each of the queries, the ranked retrieval results (obtained using the framework of [21]) are assessed using the ground-truth landmark labels. The area under the resulting precision-recall curve (average precision) is the performance measure for the query. The performance measure for the whole dataset is obtained by computing the mean Average Precision (mAP) across all queries.

In the comparison, we employed three types of the visual search engine [21]: *tf-idf* uses the tf-idf index computed on quantised descriptors (500K visual words); *tf-idf-sp* additionally re-ranks the top 200 images using RANSAC-based spatial verification. The third engine is based on nearest-neighbour matching of raw (non-quantised) descriptors and RANSAC-based spatial verification. We use tf-idf and tf-idf-sp in the majority of experiments, since using raw descriptors for large-scale retrieval is not practical. Considering that tf-idf retrieval engines are based on vector-quantised descriptors, the descriptor dimensionality is not crucial in this scenario, so we learn the descriptors with dimensionality similar to that of SIFT (128-D).

10.2.2 Feature detector and measurement region size

Here we assess the effect that feature detection and measurement region size have on image retrieval performance. For completeness, we begin with a brief description of the conventional feature extraction pipeline [19] employed in our retrieval framework. In each image, feature detection is performed using an affine-covariant detector, which produces a set of elliptically-shaped feature regions, invariant to the affine transformation of an image. As pointed out in [16], [19], it is beneficial to capture a certain amount of context around a detected feature. Therefore, each detected feature region is isotropically enlarged by a constant scaling factor to obtain the descriptor measurement region. The latter is then transformed to a square patch, which can be optionally rotated w.r.t. the dominant orientation to ensure in-plane rotation invariance. Finally, a feature descriptor is computed on the patch.

In [21], [22], [25] feature extraction was performed using the Hessian-Affine (HesAff) detector [19], $\sqrt{3}$ measurement region scaling factor, and rotation-invariant patches. We make two important observations. First, *not* enforcing patch rotation invariance leads to 5.1% improvement in mAP, which can be explained by the instability of the dominant orientation estimation procedure, as well as the nature of the data: landmark photos are usually taken in the upright position, so in-plane rotation invariance is not required and can reduce the discriminative power of the descriptors. Second, significantly higher performance can be achieved by using a higher measurement region scaling factor, as shown in Fig. 8 (red curve).

One of alternatives to the Hessian operator for feature detection is the Difference of Gaussians (DoG) function [15]. Initially, DoG detector was designed to be (in)variant to the similarity transform, but affine invariance can also be achieved by applying the affine adaptation procedure [17], [24] to the detected DoG regions. We call the resulting detector DoGAff, and evaluate the publicly available implementation in VLFeat package [34]. For DoGAff, not enforcing the patch orientation invariance also leads to 5% mAP improvement. The dependency of the retrieval performance on measurement region scaling factor is shown in Fig. 8 (blue curve). As can be seen, using DoGAff leads to considerably higher retrieval performance than HesAff. It should be noted, however, that the improvement comes at the cost of a larger number of detected regions: on average, HesAff detects 3.5K regions per image on Oxford5K, while DoGAff detects 5.5K regions.

In the sequel, we employ DoGAff feature detector (with 12.5 scaling factor and without enforcing the in-plane rotation invariance) for two reasons: it achieves better performance and the source code is publicly available.[1] The same detected regions are used for all compared descriptors.

10.2.3 Descriptor learning results

In the descriptor learning experiments, we used the Oxford5K dataset for training and both Oxford5K and Paris6K for evaluation. We note that ground-truth matches are not available for Oxford5K; instead, the training data is extracted *automatically* (Sect. 9). The evaluation on Oxford5K corresponds to the use case of learning a descriptor for a particular image collection based on extremely weak supervision. At the same time, the evaluation on Paris6K allows us to assess the *generalisation* of the learnt descriptor to different image collections. Similarly to the experiments in Sect. 10.1, we learn a 576-D PR descriptor (shown in Fig. 5, right) and its discriminative projection onto 127-D subspace.

The mAP values computed using different "descriptor – search engine" combinations are given in Table 4. First, we note that the performance of rootSIFT can be noticeably improved by adding a discriminative linear

1. We used the following MATLAB command for feature extraction using VLFeat 0.9.13:

```
[Regions, Descriptors] = vl_covdet(Image, 'Method', 'DoG', ...
'EstimateAffineShape', true, 'EstimateOrientation', false, ...
'DoubleImage', true, 'PatchRelativeExtent', 12.5, ...
'PatchResolution', 15);
```

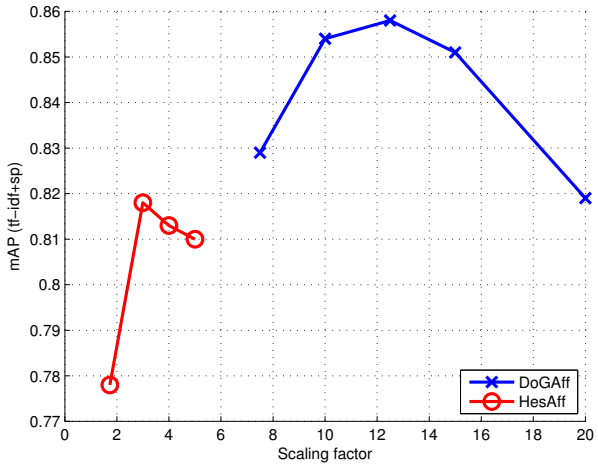

Fig. 8. The dependency of retrieval mAP on the feature detector and measurement region scaling factor (using rootSIFT descriptor and tf-idf-sp retrieval engine).

projection on top of it, learnt using the proposed framework. As a result, the projected rootSIFT (rootSIFT-proj) outperforms rootSIFT on both Oxford5K ($+2.5\%/3.0\%$ mAP using tf-idf/tf-idf-sp respectively) and Paris6K ($+2.2\%/2.1\%$ mAP). Considering that rootSIFT has already moderate dimensionality (128-D), there is no need to perform dimensionality reduction in this case, so we used Frobenius-norm regularisation of the Mahalanobis matrix A in (10), (16).

The proposed PR-proj descriptor (with both pooling regions and low-rank projection learnt) performs similarly to rootSIFT-proj on Oxford5K: $+3.0\%/2.5\%$ compared to the rootSIFT baseline, and $+0.5\%/-0.5\%$ compared to rootSIFT-proj. On Paris6K, PR-proj outperforms both rootSIFT ($+3.0\%/3.1\%$) and rootSIFT-proj ($+0.8\%/1\%$). When performing retrieval using raw descriptors without quantisation, PR-proj performs better than rootSIFT-proj on both Oxford5K (92.6% vs 91.9%) and Paris6K (86.9% vs 86.2%).

In summary, both learnt descriptors, rootSIFT-proj and PR-proj, lead to better retrieval performance compared to the rootSIFT baseline. The mAP improvements brought by the learnt descriptors are consistent for both datasets and retrieval engines, which indicates that our learnt models generalise well.

Comparison with Philbin et al. [22]. We note that our baseline retrieval system (DoGAff–rootSIFT–tf-idf-sp) performs significantly better ($+21.1\%$) than the one used in [22]: 85.8% vs 64.7%. This is explained by the following reasons: (1) different choice of the feature detector (Sect. 10.2.2); (2) more discriminative rootSIFT descriptor [1] used as the baseline; (3) differences in the retrieval engine implementation. Therefore, to facilitate a fair comparison with the best-performing linear and non-linear learnt descriptors of [22], in Table 5 we report the results [25] obtained using our descriptor learnt on top of the same feature detector as used in [21], [22]. Namely, we used HesAff with $\sqrt{3}$ measurement region scaling factor and rotation-invariant descriptor patches. With these settings, our baseline result gets worse, but much closer to [22]: 66.7% using HesAff–SIFT–tf-idf-sp. To cancel out the effect of the remaining difference in the baseline results, in the last two columns of Table 5 we also show the mAP improvement relative to the corresponding baseline for our method and [22].

As can be seen, a linear projection on top of SIFT (SIFT-proj) learnt using our framework results in a bigger improvement over SIFT than that of [22]. Learning optimal pooling regions leads to further increase of performance, surpassing that of non-linear SIFT embeddings [22]. In our case, the drop of mAP improvement when moving to a different image set (Paris6K) is smaller than that of [22], which means that our models generalise better.

The experiments with two different feature detection methods, presented in this section, indicate that the proposed learning framework brings consistent improvement irrespective of the underlying feature detector.

TABLE 4
mAP on Oxford5K and Paris6K for learnt descriptors and rootSIFT [1] using DoGAff feature detector (Sect. 10.2.2).

Descriptor	mAP	
	tf-idf	tf-idf-sp
Oxford5K		
rootSIFT baseline	0.795	0.858
rootSIFT-proj	0.820	**0.888**
PR-proj	**0.825**	0.883
Paris6K		
rootSIFT baseline	0.780	0.796
rootSIFT-proj	0.802	0.817
PR-proj	**0.810**	**0.827**

TABLE 5
mAP on Oxford5K and Paris6K for learnt descriptors (ours and those of [22]) and SIFT. Feature detection was carried out using the HesAff detector to ensure a fair comparison with [22].

Descriptor	mAP		mAP improv. (%)	
	tf-idf	tf-idf-sp	tf-idf	tf-idf-sp
Oxford5K				
SIFT baseline	0.636	0.667	-	-
SIFT-proj	0.673	0.706	5.8	5.8
PR-proj	**0.709**	**0.749**	**11.5**	**12.3**
Philbin et al., SIFT baseline	0.613	0.647	-	-
Philbin et al., SIFT-proj	0.636	0.665	3.8	2.8
Philbin et al., non-linear	0.662	0.707	8	9.3
Paris6K				
SIFT baseline	0.656	0.668	-	-
PR-proj	**0.711**	**0.722**	**8.4**	**8.1**
Philbin et al., SIFT baseline	0.655	0.669	-	-
Philbin et al., non-linear	0.678	0.689	3.5	3

11 Conclusion

In this paper we introduced a generic framework for learning two major components of feature descriptor computation: spatial pooling and discriminative dimensionality reduction. Also, we proposed an extension of

the learning formulation to the case of weak supervision, and demonstrated that the learnt descriptors are amenable to binarisation. Rigorous evaluation showed that the proposed algorithm outperforms state-of-the-art real-valued and binary descriptors on challenging datasets. This was achieved via the use of convex learning formulations coupled with large-scale regularised optimisation techniques.

ACKNOWLEDGEMENTS

This work was supported by Microsoft Research PhD Scholarship Programme and ERC grant VisRec no. 228180. A. Vedaldi was partially supported by the Violette and Samuel Glasstone Fellowship.

REFERENCES

[1] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proc. CVPR*, 2012.
[2] X. Boix, M. Gygli, G. Roig, and L. Van Gool. Sparse quantization for patch description. In *Proc. CVPR*, 2013.
[3] M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *IEEE PAMI*, 33(1):43–57, 2011.
[4] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary robust independent elementary features. In *Proc. ECCV*, 2010.
[5] M. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC*, pages 380–388, 2002.
[6] M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proc. ACC*, pages 4734–4739, 2001.
[7] Y. Gong and S. Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *Proc. CVPR*, 2011.
[8] M. Guillaumin, J. Verbeek, and C. Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *Proc. ECCV*, pages 634–647, 2010.
[9] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. CVPR*, 2010.
[10] H. Jégou, T. Furon, and J.-J. Fuchs. Anti-sparse coding for approximate nearest neighbor search. In *ICASSP*, pages 2029–2032, 2012.
[11] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. KDD*, pages 133–142, 2002.
[12] Jelena Kovacevic and Amina Chebira. An introduction to frames. *Foundations and Trends in Signal Processing*, 2(1):1–94, 2008.
[13] M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *NIPS*, pages 1189–1197, 2010.
[14] S. Leutenegger, M. Chli, and R. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *Proc. ICCV*, pages 2548–2555, 2011.
[15] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
[16] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. BMVC.*, pages 384–393, 2002.
[17] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. ECCV*. Springer-Verlag, 2002.
[18] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *IJCV*, 1(60):63–86, 2004.
[19] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 65(1/2):43–72, 2005.
[20] Y. Nesterov. Primal-dual subgradient methods for convex problems. *J. Math. Prog.*, 120(1):221–259, 2009.
[21] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. CVPR*, 2007.
[22] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Descriptor learning for efficient retrieval. In *Proc. ECCV*, 2010.
[23] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proc. ICML*, pages 713–719, 2005.
[24] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *Proc. ECCV*, volume 1, pages 414–431. Springer-Verlag, 2002.
[25] K. Simonyan, A. Vedaldi, and A. Zisserman. Descriptor learning using convex optimisation. In *Proc. ECCV*, 2012.
[26] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, volume 2, pages 1470–1477, 2003.
[27] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. In *Proc. ACM SIGGRAPH*, volume 25, pages 835–846, 2006.
[28] C. Strecha, Bronstein A. M., M. M. Bronstein, and P. Fua. LDA-Hash: Improved matching with smaller descriptors. *IEEE PAMI*, 34(1), 2012.
[29] E. Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. In *Proc. CVPR*, 2008.
[30] L. Torresani and K. Lee. Large margin component analysis. In *NIPS*, pages 1385–1392. MIT Press, 2007.
[31] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit. Boosting bnary keypoint descriptors. In *Proc. CVPR*, 2013.
[32] T. Trzcinski, M. Christoudias, V. Lepetit, and P. Fua. Learning image descriptors with the boosting-trick. In *NIPS*, pages 278–286, 2012.
[33] T. Trzcinski and V. Lepetit. Efficient discriminative projections for compact binary descriptors. In *Proc. ECCV*, 2012.
[34] A. Vedaldi and B. Fulkerson. VLFeat - an open and portable library of computer vision algorithms. In *ACM Multimedia*, 2010.
[35] K.Q. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, 2006.
[36] L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *J. Machine Learning Research*, 11:2543–2596, 2010.

Karen Simonyan is a Postdoctoral Researcher at the Department of Engineering Science, University of Oxford.

Andrea Vedaldi is a University Lecturer in Engineering Science at the Department of Engineering Science, University of Oxford.

Andrew Zisserman is the Professor of Computer Vision Engineering at the Department of Engineering Science, University of Oxford.