HPatches: A benchmark and evaluation of handcrafted and learned local descriptors

Vassileios Balntas*
Imperial College London
v.balntas@imperial.ac.uk

Karel Lenc*
University of Oxford
karel@robots.ox.ac.uk

Andrea Vedaldi
University of Oxford
vedaldi@robots.ox.ac.uk

Krystian Mikolajczyk
Imperial College London
k.mikolajczyk@imperial.ac.uk

*Authors contributed equally to this work

Abstract

In this paper, we propose a novel benchmark for evaluating local image descriptors. We demonstrate that the existing datasets and evaluation protocols do not specify unambiguously all aspects of evaluation, leading to ambiguities and inconsistencies in results reported in the literature. Furthermore, these datasets are nearly saturated due to the recent improvements in local descriptors obtained by learning them from large annotated datasets. Therefore, we introduce a new large dataset suitable for training and testing modern descriptors, together with strictly defined evaluation protocols in several tasks such as matching, retrieval and classification. This allows for more realistic, and thus more reliable comparisons in different application scenarios. We evaluate the performance of several state-of-the-art descriptors and analyse their properties. We show that a simple normalisation of traditional hand-crafted descriptors can boost their performance to the level of deep learning based descriptors within a realistic benchmarks evaluation.

1. Introduction

Local feature descriptors remain an essential component of image matching and retrieval systems and it is an active area of research. With the success of learnable representations and the availability of increasingly-large labelled datasets, research on local descriptors has seen a renaissance. End-to-end learning allows to thoroughly optimise descriptors for available benchmarks, significantly outperforming fully [20] or semi-handcrafted features [21, 32].

Surprisingly however, the adoption of these purportedly better descriptors has been limited in applications, with

Table 1. Contradicting conclusions reported in literature while evaluating the same descriptors on the same benchmark (Oxford [22]). Rows report inconsistent evaluation results due to variations of the implicit parameters e.g. of feature detectors.

LIOP > SIFT	[24, 36] ,	SIFT > LIOP	[39]
BRISK > SIFT	[18, 24] ,	SIFT > BRISK	[19]
ORB > SIFT	[29] ,	SIFT > ORB	[24]
BINBOOST > SIFT	[19, 32] ,	SIFT > BINBOOST	[5, 39]
ORB > BRIEF	[29] ,	BRIEF > ORB	[19]

SIFT [20] still dominating the field. We believe that is due to the inconsistencies in reported performance evaluations based on the existing benchmarks [22, 38]. These datasets are either small, or lack diversity to generalise well to various applications of descriptors. The progress in descriptor technology and application requirements has not been matched by a comparable development of benchmarks and evaluation protocols. As a result, while learned descriptors may be highly optimised for specific scenarios, it is unclear whether they work well in more general cases e.g. outside the specific dataset used to train them. In fact, just comparing descriptors based on published experiments is difficult and inconclusive as demonstrated in Table 1.

In this paper, we introduce a novel benchmark suite for local feature descriptors, significantly larger, with clearly defined protocols and better generalisation properties, that can supersede the existing datasets. This is inspired by the success of the Oxford matching dataset [22], the most widely-adopted and still very popular benchmark for the evaluation of local features, despite consisting of only 48 images. This is woefully insufficient for evaluating modern descriptors in the era of deep learning and large scale datasets. While some larger datasets exist, as discussed in section 2, these have other important shortcomings in terms of data and task diversity, evaluation metrics and experimental reproducibility. We address these shortcomings by

identifying and satisfying crucial requirements from such a benchmark in section 3.

Data diversity is considered especially important for evaluating various properties of descriptors. To this end, we collect a large number of multi-image sequences of different scenes under real and varying capturing conditions, as discussed in section 4. Scenes are selected to be representative of different use cases and captured under varying viewpoint, illumination, or temporal conditions, including challenging nuisance factors often encountered in applications. The images are annotated with ground-truth transformations, that allow to identify unique correspondences necessary to assess the quality of matches established by descriptors.

Reproducibility and fairness of comparisons is crucial in benchmarks. This is addressed by eliminating the influence of detector parameters. Hence, the benchmark is based on extracted local image patches rather than whole images, which brings important benefits: i) it allows to compare descriptors modulus the choice of detectors, ii) it simplifies the process and makes the experiments reproducible, and, importantly, iii) it avoids various biases, e.g. the number or size of measurement regions or semi-local geometric constraints that make the results from image-based benchmarks incomparable (section 2).

Task diversity is another requirement rarely addressed in exisiting evaluation benchmarks. To this end, we define three complementary benchmarking tasks in section 5: patch verification (classification of patch pairs), image matching, and patch retrieval. These are representative of different use cases and, as we show in the experiments, detectors rank differently depending on the task considered.

While this work is focused on local descriptors, the proposed dataset contains groundtruth, including pairwise geometric transformations, that will allow future evaluations of feature detectors as well. We believe that this benchmark will enable the community to gain new insights in state-of-the-art local feature matching since it is more diverse and significantly larger than any existing dataset used in this field. We assess various methods including simple baselines, handcrafted ones, and state-of-the-art learned descriptors in section 6. The experimental results show that descriptor performance and their ranking may vary in different tasks, and differs from the results reported in the literature. This further highlights the importance of introducing a large, varied and reproducible evaluation benchmark for local descriptors.

All benchmark data and code implementing the evaluation protocols are made publicly available[1].

---

2. Review of existing benchmarks

In this section we review existing datasets and benchmarks for the evaluation of local descriptors and discuss their main shortcomings.

2.1. Image-based benchmarks

In image matching benchmarks, descriptors are used to establish correspondences between images of the same objects or scenes. Local features, extracted from each image by a co-variant detector, are matched by comparing their descriptors, typically with a nearest-neighbor approach. Then, putative matches are assessed for compatibility with the known geometric transformation between images (usually an homography) and the relative number of correspondences is used as the evaluation measure.

The most widely-adopted benchmark for evaluating descriptors and detectors is the *Oxford matching dataset* [22]. It consists of image sequences of 8 scenes, each containing 6 images, and ground-truth homographies. While the Oxford dataset contains images that are all captured by a camera, *Generated Matching dataset* [14] is obtained by generating images using synthetic transformations, and contains 16 sequences of 48 images. However, the synthetic nature of the transformations does not model all noise that typically occurs in the capturing process, thus making this data less challenging than the Oxford data [4]. The *DTU Robots dataset* [1] contains real images of 3D objects, captured using a robotic arm in controlled laboratory conditions, which is suitable for certain application scenarios but of limited diversity in the data. The *Hanover dataset* [11] investigates high-resolution matching and contains images of up to 8 megapixels with highly accurate ground-truth homographies. However, it is also limited by containing only 5 scenes. The *Edge Foci dataset* [42] consists of sequences with very strong changes in viewing conditions, making the evaluation somewhat specialized to extreme cases; furthermore, the groundtruth for non-planar scenes does not uniquely identify the correspondences since the transformations cannot be approximated well by homographies. Similarly, the *WxBs dataset* [25] focuses on very wide baseline matching, with extreme changes in geometry, illumination, and appearance over time.

All these datasets share an important shortcoming that leaves scope for variations in different descriptor evaluations: there is no pre-defined set of regions to match. As a consequence, results depend strongly on the choice of detector (method, implementation, and parameters), making the comparison of descriptors very difficult and unreliable. This is demonstrated in table 1 where different papers reach different conclusions even when they are evaluated on the same data using the same protocol.

Defining centre locations of features to match does not constrain the problem sufficiently either. For example, this

Table 2. Effect of using a different ρ to scale the size of the detected DoG keypoint to the size of the measurement region. Columns 1|X represent matching scores between the first and the X image in the sequence for different scaling factors ρ.

ρ	1|2	1|3	1|4	1|5	1|6
1	0.31	0.13	0.05	0.03	0.01
4	0.68	0.44	0.24	0.15	0.11
12	0.80	0.67	0.54	0.42	0.35
20	0.87	0.77	0.69	0.55	0.50

does not fix the region of the image used to compute the descriptor, typically referred to as the *measurement region*. Usually the measurement region is set to a fixed but arbitrarily set scaling of the feature size, and this parameter is often not reported or varies in papers. Unfortunately, this has a major impact on performance [31]. Table 2 shows matching scores for different scaling factors of the measurement region in the Oxford data.[2] Variations of more than 50% mAP occur; in fact, due to the planarity of such scenes, larger measurement regions lead to improved matching results.

In order to control for the size of the measurement region and other important parameters such as the amount of blurring, resolution of the normalized patch used to compute a descriptor [34], or use of semi-local geometric constraints, we argue that a descriptor benchmark should be based on *image patches* rather than whole images. Thus, all such ambiguities are removed and a descriptor can be represented and evaluated as a function $f(\boldsymbol{x}) \in \mathbb{R}^D$ that maps a patch $\boldsymbol{x} \in \mathbb{R}^{H \times H \times 3}$ to a D-dimensional feature vector. This type of benchmark is discussed next.

2.2. Patch-based benchmarks

Patch based benchmarks consist of patches extracted from interest point locations in images. The patches are then normalised to the same size, and annotated pair- or group-wise with labels that indicate positive or negative examples of correspondence. The annotation is typically established by using image groundtruth, such as geometric transformations between images. In case of image based evaluations the process of extracting, normalising and labelling patches leaves scope for variations and its parameters differ between evaluations.

The first popular patch-based dataset was *Photo-Tourism* [38]. Since its introduction, the many benefits of using patches for benchmarking (section 5.3) became apparent. PhotoTourism introduced a simple and unambiguous evaluation protocol, which we refer to as *patch verification*: given a pair of patches, the task is to predict whether they match or not, which reduces the matching task to a binary classification problem. This formulation is particularity suited for learning-based methods, including CNNs and metric learning in particular due to the large number

---

[2]mAP is computed on the Leuven sequence in the Oxford matching dataset using the DoG detector and SIFT descriptor.

Table 3. Comparison of existing datasets and the proposed HPatches dataset.

dataset	patch	diverse	real	large	multitask
Photo Tourism [37]	✓		✓	✓	
DTU [1]			✓	✓	
Oxford-Affine [22]		✓	✓		
Synth. Matching [14]		✓	✓		
CVDS [9]	✓	✓	✓		
Edge Foci [42]		✓	✓		
RomePatches [26]	✓		✓		
RDED [10]		✓	✓		
HPatches	✓	✓	✓	✓	✓

patches available in this dataset. The main limitation of PhotoTourism is its scarce data diversity (there are only three scenes: Liberty, Notre-Dame and Yosemite), task diversity (there is only the patch verification task), and feature type diversity (only DoG features were extracted). The *CVDS dataset* [9] addresses the data diversity issue by extracting patches from five MPEG-CDVS: Graphics, Paintings, Video, Buildings and Common Objects. Despite its notable variety, experiments have shown that the state-of-the-art descriptors achieve high performance scores on this data [3]. The *RomePatches dataset* [26] consider a query ranking task that reflects image retrieval scenario, but is limited to $10K$ patches, which makes it an order of magnitude smaller than PhotoTourism.

2.3. Metrics

In addition to choosing data, patches, and tasks, the choice of evaluation metric is also important. For classification, the Receiver Operating Characteristic (ROC) curves have often been used [12, 13] as the basis for comparison. However, patch matching is intrinsically highly unbalanced, with many more negative than positive correspondence candidates; ROC curves are less representative for unbalanced data and, as a result, a strong performance in ROC space does not necessarily generalise to a strong performance in applications, such as the nearest-neighbor matching [30, 39, 5, 33]. Several papers [38, 32, 33] reported at a single point on the ROC curve (FPR95, i.e. the false positive rate at 95% true positive recall) which is more appropriate for unbalanced data than the equal error rate or the area under the ROC curve; however, this reduces the information provided by the whole curve. Precision-Recall and mean Average Precision (mAP) are much better choices of metrics for unbalanced datasets – for example DBRIEF [33] is excellent in ROC space but has very low (≈ 0.01) mAP the Oxford dataset [19].

3. Benchmark design

We address the shortcomings of the existing dataset, discussed in section 2, by identifying the following requirements:

Figure 1. Examples of image sequences; note the diversity of scenes and nuisance factors, including viewpoint, illumination, focus, reflections and other changes.

- *Reproducible, patch-based*: descriptor evaluation should be done on patches to eliminate the detector related-factors. This leads to a standardisation across different works and makes results directly comparable.
- *Diverse*: representative of many different scenes and image capturing conditions.
- *Real*: real data have been found to be more challenging than a synthesized one due to nuisance factors that cannot be modelled in image transformations.
- *Large*: to allow accurate and stable evaluation, as well as to provide substantial training sets for learning based descriptors.
- *Multitask*: representative of several use cases, from matching image pairs to image retrieval. This allows cross-task comparison of descriptors performance within the same data.

Based on these desired properties, we introduce a new large-scale dataset of image sequences (section 4) annotated with homographies. This is used to generate a patch-based benchmark suite for evaluating local image descriptors (section 5). Table 3 compares the proposed dataset to existing benchmarks in terms of the properties stated above.

4. Images and patches

Images are collected from various sources, including existing datasets. We have captured 51 sequences by a camera, 33 scenes are from [16], 12 scenes from [1], 5 scenes from [10], 4 scenes from [22], 2 scenes from [35] and 1

REF   E1   E2   E3   E4   H1   H2   H3   H4   T1   T2   T3   T4
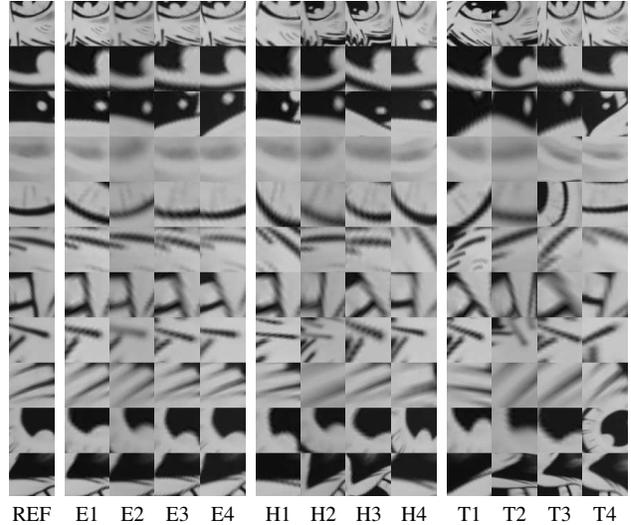
Figure 2. Example of the geometric noise visualized with the extracted patches for the EASY, HARD and TOUGH distributions.

scene from [40]. Some of the sequences are illustrated in fig. 1. In 57 scenes the main nuisance factors are photometric changes and the remaining 59 sequences show significant geometric deformations due to viewpoint change.

A sequence includes a reference image and 5 target images with varying photometric of geometric changes. The sequences are captured such that the geometric transformations between images can be well approximated by homographies from the reference image to each of the target images. The homographies are estimated following [22].

Patches are extracted using the following protocol. Several scale invariant interest point detectors i.e. DoG, Hessian-Hessian and Harris-Laplace are used to extract features[3] for scales larger than $1.6px$, which give stable points. Near-duplicate regions are discarded based on their intersection-over-union (IoU) overlap (> 0.5) and one region per cluster is randomly retained. This keeps regions that overlap less than 0.5 IoU. Approximately 1,300 regions per image are then randomly selected.

For each sequence, patches are detected in the reference image and projected on the target images using the ground-truth homographies. This sidesteps the limitations of the detectors, which may fail to provide corresponding regions in every target images due to significant viewpoint or illumination variations. Furthermore, it allows to extract more patches thus better evaluate descriptors in such scenarios. Regions that are not fully contained in all target images are discarded. Hence, a set of corresponding patches contains one from each image in the sequence. In practice, when a detector extracts corresponding regions in different images, it does so with a certain amount of noise. In order

---

[3]VLFeat implementations of detectors are used.
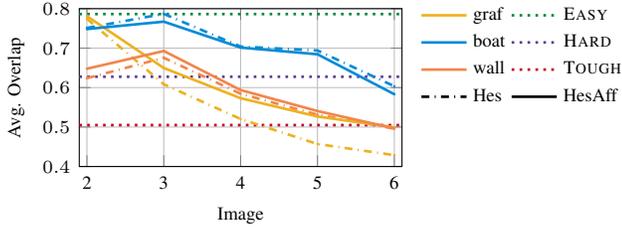
Figure 3. The average overlap accuracy of Hessian and Hessian-Affine detector on the viewpoint sequences of the [23]. Line color encodes dataset and line style a detector. The selected overlaps of the EASY and HARD variants are visualised with a dotted line.

to simulate this noise, detections are perturbed using three settings: EASY, HARD and TOUGH. This is obtained by applying a random transformation $T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ to the region before projection. Assuming that the region centre is the coordinate origin, the transformation includes rotation $R(\theta)$ by angle θ, anisotropic scaling by $s/\sqrt{a}$ and $s\sqrt{a}$, and translation by $[m\, t_x, m\, t_y]$, thus the translation is proportional to the detection scale m. The transformation parameters are uniformly sampled from the intervals $\theta \in [-\theta_{max}, \theta_{max}]$, $t_x, t_y \in [-t_{max}, t_{max}]$, $\log_2(s) \in [-s_{max}, s_{max}]$, $\log_2(a) \in [-a_{max}, a_{max}]$, whose values for each setting are given in table 4. These settings reflect the typical overlap accuracy of the Hessian and Hessian-Affine detectors on Oxford matching benchmark. There, images in each sequence are sorted by increasing transformation, resulting in increased detector noise. Figure 3 shows that the EASY, HARD, and TOUGH groups correspond to regions extracted in images 1-2, 3-4 and 5-6 of such sequences.

Table 4. Range of geometric noise distributions, in units of a patch scale.

Variant	θ_{max}	t_{max}	s_{max}	a_{max}
EASY	$10°$	0.15	0.15	0.2
HARD	$20°$	0.3	0.3	0.4
TOUGH	$30°$	0.45	0.5	0.45

Detected regions are scaled with a factor of 5 (see section 2). The smallest patch size in the reference image is 16×16px since only regions from detection scales above 1.6px are considered. In each region the dominant orientation angle is estimated using a histogram of gradient orientations [20]. Regions are rectified by normalizing the detected affine region to a circle using bilinear interpolation and extracting a square of 65×65 pixels. Example of extracted patches are shown in fig. 2, where the effect of the increasing detector noise is clearly visible.

5. Benchmark tasks

In this section, we define the benchmark metrics, tasks and their evaluation protocols for: patch verification, image matching and patch retrieval.

The tasks are designed to imitate typical use cases of local descriptors. Patch verification (section 5.2) is based on [38] and measures the ability of a descriptor to classify whether two patches are extracted from the same measurement. Image matching (section 5.3), inspired by [22], tests to what extent a descriptor can correctly identify correspondences in two images. Finally, patch retrieval (section 5.4) tests how well a descriptor can match a query patch to a pool of patches extracted from many images, including many distractors. This is a proxy to local feature based image indexing [27, 26].

5.1. Evaluation metrics

We first define the precision and recall evaluation metric used in HPatches. Let $\mathbf{y} = (y_1, \ldots, y_n) \in \{-1, 0, +1\}^n$ be labels of a ranked list of patches returned for a patch query, indicating negative, *to ignore*, and positive match, respectively. Then *precision* and *recall* at rank i are given by[4] $P_i(\mathbf{y}) = \sum_{k=1}^{i} [y_k]_+ / \sum_{k=1}^{i} |y_k|$ and $R_i(\mathbf{y}) = \sum_{k=1}^{i} [y_k]_+ / \sum_{k=1}^{N} [y_k]_+$; the *average precision* (AP) is given by $AP(\mathbf{y}) = \sum_{k:y_k=+1} P_k(\mathbf{y}) / \sum_{k=1}^{N} [y_k]_+$. The main difference w.r.t. the standard definition of PR is the entries that can be ignored *i.e.* $y_i = 0$ which will be used for retrieval task in section 5.4. In this case, let $K \geq \sum_{k=1}^{N} [y_k]_+$ be the total number of positives; recall is computed as $R_i(\mathbf{y}; K) = \sum_{k=1}^{i} [y_k]_+ / K$ and AP as $AP(\mathbf{y}; K) = \sum_{k:y_k=+1} P_k / K$ which corresponds to truncated PR curves).

5.2. Patch verification

In *patch verification* descriptors are used to classify whether two patches are in correspondence or not. The benchmark starts from a list $\mathcal{P} = ((\mathbf{x}_i, \mathbf{x}'_i, y_i),\ i = 1, \ldots, N)$ of positive and negative patch pairs, where $\mathbf{x}_i, \mathbf{x}'_i \in \mathbb{R}^{65 \times 65 \times 1}$ are patches and $y_i = \pm 1$ is their label.

The dataset is used to evaluate a matching approach $\mathcal{A}$ that, given any two patches $\mathbf{x}_i, \mathbf{x}'_i$, produces a confidence score $s_i \in \mathbb{R}$ that the two patches correspond. The quality of the approach is measured as the average precision of the ranked patches, namely $AP(y_{\pi_1}, \ldots, y_{\pi_N})$ where π is the permutation that sorts the scores in decreasing order (i.e. $s_{\pi_1} \geq s_{\pi_2} \geq \cdots \geq s_{\pi_n}$) to apply the formulas from section 5.1.

The benchmark uses four sets of patch pairs extracted by varying the projection noise as discussed in section 4 that is EASY, HARD or TOUGH as well as a set of negative pairs that are either sampled from images within the same sequence or from different sequences. The overall performance of the method $\mathcal{A}$ is then computed as the mean AP for the six patch sets. In total, we generate 2×10^5 positive pairs and 1×10^6 negative pairs per a set.

---

[4]Here $[z]_+ = \max\{0, z\}$.

Note that the benchmark only requires scores s_i computed by the algorithm $\mathcal{A}$; in particular, this unifies the evaluation of a descriptor with a custom similarity metric, including a learned one.

This evaluation protocol is similar to [38]. However, whereas the ROC [13] is used there, AP is preferred here [30] since the dataset is highly unbalanced, with the vast majority (10^6) of patch pairs being negative. The latter is more representative of typical matching scenarios.

5.3. Image matching

In *image matching*, descriptors are used to match patches from a reference image to a target one. In this task an image is a collection of N patches $L_k = (\mathbf{x}_{ik}, \ i = 1, \dots, N)$. Consider a pair of images $\mathcal{D} = (L_0, L_1)$, where L_0 is the reference and L_1 the target. Thus, after matching, $\mathbf{x}_{i0}$ is in correspondence with $\mathbf{x}_{i1}$.

The pair $\mathcal{D}$ is used to evaluate an algorithm $\mathcal{A}$ that, given a reference patch $\mathbf{x}_{i0} \in L_0$, determines the index $\sigma_i \in \{1, \dots, N\}$ of the best matching patch $\mathbf{x}_{\sigma_i 1} \in L_1$, as well as the corresponding confidence score $s_i \in \mathbb{R}$. Then, the benchmark labels the assignment σ_i as $y_i = 2[\sigma_i \stackrel{?}{=} i] - 1$, and computes $AP(y_{\pi_1}, \dots, y_{\pi_N}; N)$, where π is the permutation that sorts the scores in decreasing order (note that the number of positive results is fixed to N; see section 5.1).

We group sequences based on whether they vary by viewpoint or illumination and each group is instantiated with EASY, HARD and TOUGH patches. The overall performance of an algorithm $\mathcal{A}$ is computed as the mean AP for all such image pairs and variants.

Note that the benchmark only requires the indexes σ_i and the scores s_i computed by the algorithm $\mathcal{A}$ for each image pair $\mathcal{D}$. Typically, these can be computed by extracting patch descriptors and comparing with a similarity metric.

This evaluation protocol is designed to closely resemble the one from [22]. A notable difference is that, since the patch datasets are constructed in such a way that each reference patch has a corresponding patch in each target image, the maximum recall is always 100%. Note also that, similarly to the verification task, the benchmark evaluates the combined performance of the descriptor and similarity score provided by the tested algorithm.

5.4. Patch retrieval

In *patch retrieval* descriptors are used to find patch correspondences in a large collection of patches, a large portion of which are distractors, extracted from confounder images. Consider a collection $\mathcal{P} = (\mathbf{x}_0, (\mathbf{x}_i, y_i), i = 1, \dots, N)$ consisting of a query patch $\mathbf{x}_0$, extracted from a reference image L_0, and all patches from images $L_k, k = 1, \dots, K$ in the same sequence (matching images), as well as many confounder images.

Table 5. Basic properties of the selected descriptors. For binary descriptors, the dimensionality is in bits (*), otherwise in number of single precision floats. The computational efficiency is measured in thousands of descriptors extracted per second.

Descr.	MSTD	RESZ	SIFT	RSIFT	BRIEF	BBOOST	ORB	DC-S	DC-S2S	DDESC	TF-M	TF-R
Dims	2	36	128	128	*256	*256	*256	256	512	128	128	128
Patch Sz	65	65	65	65	32	32	32	64	64	64	32	32
Speed CPU	67	3	2	2	333	2	333	0.3	0.2	0.1	0.6	0.6
Speed GPU								10	5	2.3	83	83

In the retrieval protocol, a patch $\mathbf{x}_i$ is given a positive label $y_i = +1$ if it corresponds to the query patch $\mathbf{x}_0$, and negative $y_i = -1$ otherwise. Since there is exactly one corresponding patch in each image L_k of the same sequence, there are exactly K positive patches in $\mathcal{D}$. However, retrieved patches $\mathbf{x}_i$ that do not correspond to the query patch $\mathbf{x}_0$ but at least belong to a matching image L_k are ignored ($y_i = 0$). The idea is that such patches are not detrimental for the purpose of retrieving the correct image, and such innocuous errors may occur frequently in the case of repeated structures in images.

The collection $\mathcal{P}$ is used to evaluate an algorithm $\mathcal{A}$ that assigns to each patch $\mathbf{x}_i$ a confidence score $s_i \in \mathbb{R}$ that the patch matches the query $\mathbf{x}_0$. The benchmark then returns $AP(y_{\pi_1}, \dots, y_{\pi_N}; K)$, where π is the permutation that sorts the scores in decreasing order.

The benchmark extracts 1×10^4 collections $\mathcal{P}$, each corresponding to different query patch $\mathbf{x}_0$ and its corresponding 5 patches as well as 2×10^4 distractors randomly selected from all sequences. Furthermore, there are three variants instantiated for EASY, HARD and TOUGH. The overall performance of an algorithm $\mathcal{A}$ is computed as the mean AP for all such collections and their variants.

The design of this benchmark is inspired by classical image retrieval systems such as [27, 28, 26], which use patches and their descriptors as entries in image indexes. A similar evaluation may be performed by using the PhotoTourism dataset, which includes $\sim 100K$ small sets of corresponding patches. Unfortunately, since these small sets are not maximal, it is not possible to know that a patch *does not* have a correct correspondence without the ground truth, which makes the evaluation noisy.

6. Experimental results

In this section we evaluate local descriptors with the newly introduced benchmark and discuss the results in relation to the literature.

6.1. Descriptors

We evaluate the following descriptors, summarized in table 5. We include two **baselines**: MSTD, $[\mu, \sigma]$ which is the average μ and standard deviation σ of the patch, and RESZ, the vector obtained by resizing the patch to 6×6 pix-

els and normalizing it by subtracting μ and dividing by σ. For **SIFT-based** descriptors we include SIFT [20] and its variant RSIFT [2]. From the family of **binary descriptors** we test BRIEF [8], based on randomised intensity comparison, ORB [29], that uses uncorrelated binary tests, and BBOOST [32], where binary tests are selected using boosting. Finally, we evaluate several recent **deep descriptors** including the siamese variants of DeepCompare [41] (DC-S, DC-S2S) with one and two stream CNN architectures for one or two patch crops, DeepDesc [30] (DDESC), which exploits hard-negative mining, and the TFeat *margin** (TF-M) and *ratio** (TF-R) of the TFeat descriptor [4], based on shallow convolutional networks, triplet learning constraints and fast hard negative mining. All the learning based descriptors were trained on PhotoTourism data, which is different from our new benchmark.

It has been shown in [2, 7, 17] that descriptor normalisation often substantially improves the performance. Thus, we also include post-processed variants of selected descriptors by applying ZCA whitening [6, p. 299-300] with clipped eigen values [15] followed by power law normalisation [2] and L2 normalization. ZCA projection is computed on a subset of the dataset (note that ZCA is unsupervised). The threshold for eigen clipping is estimated for each descriptor separately to maximise its performance on a subset of the dataset. The normalisation is not used for trivial baselines and for the binary descriptors.

Table 5 shows the dimensionality, size of the measurement region in pixels, and extraction time of each descriptor. DeepCompare [41] variants have the highest dimensionality of 256 and 512, otherwise the other real value descriptors are of 128 dimensions except MSTD and RESZ. All binary descriptors are of 256 bits. In terms of speed, the binary descriptors BRIEF and ORB are 4 times faster than the most efficient CNN based features *i.e.*TF-. Other descriptors are at least an order of magnitude slower. Note that MSTD and RESZ are implemented in Matlab therefore their efficiency should be interpreted with caution.

6.2. Results

The descriptors are evaluated on three benchmark tasks: patch verification, image matching, and patch retrieval, as defined in section 5. In all plots in fig. 4, the colour of the marker indicates the amount of geometric noise, i.e. EASY, HARD, and TOUGH, as discussed in section 4. There are two variants of the experimental settings for each task, as explained in the discussion below, and the type of the marker corresponds to the experimental settings. The bars are the means of the six runs given by three variants of noise with two additional settings each. Dashed bar borders and + indicate ZCA projected and normalised features.

Verification. ZCA projected and normalized +TF-R, +DC-S2S, are closely followed by other TF-, +DDESC and +DC-S, with slightly lower scores for post processed SIFT and binary descriptors. The post processing gives a significant boost to DC- as well as SIFT but a smaller improvements to TF- based descriptors. Good performance of CNN features is expected as such descriptors are optimized together with their distance metric to perform well in the verification task. The experiment was run for negative pairs formed by patches from the same sequence SAMESEQ and from different sequences DIFFSEQ. The ones from SAMESEQ are considered more challenging as the textures in different parts of the image are often similar. In fact the results are consistently lower for SAMESEQ. This shows that, not only the noise in positive data poses a challenge, but the performance can also vary depending on what source the negative examples come from.

Matching. The ranking of descriptors changes for this task. Although normalized +DDESC still performs well, surprisingly, +RSIFT comes in front of other descriptors. +TF-also give good matching performance. Overall mAP scores are much lower than for the verification task as the ratio of positive to negative examples is significantly lower here and all the negative ones come from the same sequence. Also the gap between SIFT and deep descriptors is narrow compared to the verification. Another interesting observation is that the results for sequences with photometric changes (ILLUM) are consistently lower than for the viewpoint change (VIEWPT). This is different to what was observed in evaluations on Oxford data [22]. It seems that more progress has been made on geometric invariance in contrast to the robustness to photometric changes. The proposed HPatches dataset includes many sequences with extreme illumination changes.

Retrieval. Top performers in the retrieval scenario are the same as for matching. In particular, SIFT variants are close behind +DDESC. The overall performance is slightly better compared to matching which can again be explained by distractors originating from the same sequence in matching and different sequences in retrieval.

Multitask. There are several interesting observations across the tasks. First, the ranking of the descriptors changes, which confirms that multiple evaluation metrics are needed. Second, SIFT variants, especially when followed by normalisation, perform very well. In fact, +RSIFT is the second-best descriptor in both image matching and patch retrieval. MSTD gives good scores on verification but completely fails for matching and retrieval, as both rely on nearest neighbour matching. Good performance on verification clearly does not generalise well to the other tasks, which much better reflect the practical applications of descriptors. This further highlights the need for using a multitask benchmark to complement training and testing on PhotoTourism, which is done in vast majority of recent papers and is similar to the verification task here. The

Patch Verification mAP [%] — * DiffSeq ♦ SameSeq

Descriptor	mAP
Resz	48.11%
MStd	48.75%
Brief	58.07%
RSIFT	58.53%
ORB	60.15%
SIFT	65.12%
BBoost	66.67%
DC-S	70.04%
+SIFT	74.35%
+RSIFT	76.70%
DC-S2S	78.23%
DDesc	79.51%
+DC-S	81.63%
+DDesc	81.65%
TF-M	81.90%
TF-R	81.92%
+TF-M	82.69%
+DC-S2S	83.03%
+TF-R	83.24%

Image Matching mAP [%] — ◄ Viewpt × Illum

Descriptor	mAP
MStd	0.10%
Resz	7.16%
Brief	10.50%
BBoost	14.77%
ORB	15.32%
DC-S	24.92%
SIFT	25.47%
RSIFT	27.22%
DC-S2S	27.69%
DDesc	28.05%
TF-R	30.61%
+DC-S	31.65%
+DC-S2S	32.34%
TF-M	32.64%
+SIFT	32.76%
+TF-M	34.29%
+TF-R	34.37%
+DDesc	35.44%
+RSIFT	36.77%

Patch Retrieval mAP [%] — ■ Easy ■ Hard ■ Tough

Descriptor	mAP
MStd	1.20%
Resz	13.12%
Brief	16.03%
ORB	18.85%
BBoost	22.45%
SIFT	31.98%
RSIFT	33.56%
DC-S2S	34.76%
DC-S	34.84%
TF-R	37.69%
+DC-S2S	38.23%
TF-M	39.40%
+DC-S	39.68%
DDesc	39.83%
+TF-M	40.02%
+TF-R	40.23%
+SIFT	40.36%
+RSIFT	43.84%
+DDesc	44.55%

Figure 4. Verification, matching and retrieval results. Colour of the marker indicates EASY, HARD, and TOUGH noise. The type of the marker corresponds to the variants of the experimental settings (see section 6.2). Bar is a mean of the 6 variants of each task. Dashed bar borders and + indicate ZCA projected and normalised features.

difference in performance for EASY and TOUGH geometric distortions, as well as for the illumination changes, is up to 30%, which shows there is still scope for improvement in both areas.

The performance of deep descriptors and SIFT varies across the tasks although +DDESC [30] is close to the top scores in each category, however it is the slowest to calculate. In matching and retrieval, ZCA and normalisation bring the performance of SIFT to the top level. Compared to some deep descriptors, SIFT seems less robust to high degrees of geometric noise, with large spread for EASY and TOUGH benchmarks. This is especially evident on the patch verification task, where SIFT is outperformed by most of the other descriptors for the TOUGH data.

The binary descriptors are outperformed by the original SIFT by a large margin for the image matching and patch retrieval task in particular, which may be due to its discriminative power and better robustness to the geometric noise. The binary descriptors are competitive only for the patch verification task. However, the binary descriptors have other advantages, such as compactness and speed, so they may still be the best choice in applications where accuracy is less important than speed. Also +TF perform relatively well, in particular when considering their efficiency.

Post-processing normalisation, in particular square root, has a significant effect. For most of the descriptors, the normalised features perform much better than the original ones.

Finally, patch verification achieves on average much higher mAP score compared to the other tasks. This can be seen mainly from the relatively good performance of the trivial MSTD descriptor. This confirms that patch verification task is insufficient on its own and other tasks are crucial

in descriptor evaluation.

7. Conclusions

With the advent of deep learning, the development of novel and more powerful local descriptors has accelerated tremendously. However, as we have shown in this paper, the benchmarks commonly used for evaluating such descriptors are inadequate, making comparisons unreliable. In the long run, this is likely to be detrimental to further research. In order to address this problem, we have introduced HPatches, a new public benchmark for local descriptors. The new benchmark is patch-based, removing many of the ambiguities that plagued the existing image-based benchmarks and favouring rigorous, reproducible, and large scale experimentation. This benchmark also improves on the limited data and task diversity present in other datasets, by considering many different scene and visual effects types, as well as three benchmark tasks close to practical applications of descriptors.

Despite the multitask complexity of our benchmark suite, using the evaluation is easy as we provide open-source implementation of the protocols which can be used with minimal effort. HPatches can supersede datasets such as PhotoTourism and the older but still frequently used Oxford matching dataset, addressing their shortcomings and providing a valuable tool for researchers interested in local descriptors.

References

[1] H. Aanæs, A. L. Dahl, and K. S. Pedersen. Interesting interest points. *IJCV*, 97(1):18–35, 2012.

[2] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proc. CVPR*, pages 2911–2918, 2012.

[3] V. Balntas. *Efficient learning of local image descriptors*. PhD thesis, University of Surrey, 2016.

[4] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. *Proc. BMVC*, 2016.

[5] V. Balntas, L. Tang, and K. Mikolajczyk. BOLD - binary online learned descriptor for efficient image matching. In *Proc. CVPR*, 2015.

[6] C. M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[7] A. Bursuc, G. Tolias, and H. Jégou. Kernel local descriptors with implicit rotation matching. In *ACM ICMR*, pages 595–598, 2015.

[8] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary robust independent elementary features. In *Proc. ECCV*, pages 778–792, 2010.

[9] V. Chandrasekhar, G. Takacs, D. M. Chen, S. S. Tsai, M. Makar, and B. Girod. Feature matching performance of compact descriptors for visual search. In *Proc. Data Compression Conference*, pages 3–12, 2014.

[10] K. Cordes, B. Rosenhahn, and J. Ostermann. Increasing the accuracy of feature evaluation benchmarks using differential evolution. In *Proc. SDE*, pages 1–8, 2011.

[11] K. Cordes, B. Rosenhahn, and J. Ostermann. High-resolution feature evaluation benchmark. In *Proc. CAIP*, pages 327–334, 2013.

[12] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proc. ICML*, pages 233–240, 2006.

[13] T. Fawcett. Roc graphs: Notes and practical considerations for researchers. 2004.

[14] P. Fischer, A. Dosovitskiy, and T. Brox. Descriptor matching with convolutional neural networks: a comparison to sift. *arXiv preprint arXiv:1405.5769*, 2014.

[15] G. Hua, M. Brown, and S. Winder. Discriminant embedding for local image descriptors. In *Proc. ICCV*, pages 1–8, 2007.

[16] N. Jacobs, N. Roman, and R. Pless. Consistent temporal variations in many outdoor scenes. In *Proc. CVPR*, pages 1–6, 2007.

[17] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proc. CVPR*, volume 2, pages II–506, 2004.

[18] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *Proc. ICCV*, pages 2548–2555, 2011.

[19] G. Levi and T. Hassner. LATCH: learned arrangements of three patch codes. In *Winter Conference on Applications of Computer Vision (WACV)*, 2016.

[20] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, volume 2, pages 1150–1157, 1999.

[21] K. Mikolajczyk and J. Matas. Improving descriptors for fast tree matching by optimal linear projection. In *Proc. ICCV*, 2007.

[22] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE PAMI*, 27(10):1615–1630, 2005.

[23] 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.

[24] O. Miksik and K. Mikolajczyk. Evaluation of local detectors and descriptors for fast feature matching. In *Proc. ICPR*, pages 2681–2684, 2012.

[25] D. Mishkin, J. Matas, M. Perdoch, and K. Lenc. Wxbs: Wide baseline stereo generalizations. In *Proc. BMVC*, pages 12.1–12.12, 2015.

[26] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid. Local convolutional features with unsupervised training for image retrieval. In *Proc. ICCV*, pages 91–99, 2015.

[27] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. ICCV*, pages 1–8, 2007.

[28] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proc. CVPR*, pages 1–8, 2008.

[29] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proc. ICCV*, pages 2564–2571, 2011.

[30] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. *Proc. ICCV*, 2015.

[31] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *IEEE PAMI*, 36(8):1573–1585, 2014.

[32] T. Trzcinski, M. Christoudias, and V. Lepetit. Learning image descriptors with boosting. *IEEE PAMI*, 37(3):597–610, 2015.

[33] T. Trzcinski and V. Lepetit. Efficient discriminative projections for compact binary descriptors. In *Proc. ECCV*, pages 228–242, 2012.

[34] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[35] V. Vonikakis, D. Chrysostomou, R. Kouskouridas, and A. Gasteratos. Improving the robustness in feature detection by local contrast enhancement. In *Proc. IST*, pages 158–163, 2012.

[36] Z. Wang, B. Fan, and F. Wu. Local intensity order pattern for feature description. In *Proc. ICCV*.

[37] S. Winder and M. Brown. Learning local image descriptors. In *Proc. CVPR*, 2007.

[38] S. Winder, G. Hua, and M. Brown. Picking the best daisy. In *Proc. CVPR*, 2009.

[39] T.-Y. Yang, Y.-Y. Lin, and Y.-Y. Chuang. Accumulated stability voting: A robust descriptor from descriptors of multiple scales. In *Proc. CVPR*, pages 327–335, 2016.

[40] G. Yu and J.-M. Morel. ASIFT: an algorithm for fully affine invariant comparison. *Image Processing On Line*, 1, 2011.

[41] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *Proc. CVPR*, 2015.

[42] C. L. Zitnick and K. Ramnath. Edge foci interest points. In *Proc. ICCV*, pages 359–366, 2011.