Comparative Evaluation of Hand-Crafted and Learned Local Features

Johannes L. Schönberger[1] Hans Hardmeier[1] Torsten Sattler[1] Marc Pollefeys[1,2]
[1] Department of Computer Science, ETH Zürich [2] Microsoft Corp.
{jsch,harhans,sattlert,pomarc}@inf.ethz.ch

Abstract

Matching local image descriptors is a key step in many computer vision applications. For more than a decade, hand-crafted descriptors such as SIFT have been used for this task. Recently, multiple new descriptors learned from data have been proposed and shown to improve on SIFT in terms of discriminative power. This paper is dedicated to an extensive experimental evaluation of learned local features to establish a single evaluation protocol that ensures comparable results. In terms of matching performance, we evaluate the different descriptors regarding standard criteria. However, considering matching performance in isolation only provides an incomplete measure of a descriptor's quality. For example, finding additional correct matches between similar images does not necessarily lead to a better performance when trying to match images under extreme viewpoint or illumination changes. Besides pure descriptor matching, we thus also evaluate the different descriptors in the context of image-based reconstruction. This enables us to study the descriptor performance on a set of more practical criteria including image retrieval, the ability to register images under strong viewpoint and illumination changes, and the accuracy and completeness of the reconstructed cameras and scenes. To facilitate future research, the full evaluation pipeline is made publicly available.

1. Introduction

Matching local image features is a crucial step in many computer vision applications, *e.g.*, in Structure-from-Motion (SFM) and Multi-View Stereo (MVS) [1, 17, 33, 37, 39, 40], image retrieval [31, 34, 47, 48], and image-based localization [35, 36, 56]. In many of these applications, the overall performance strongly depends on the quality of the initial feature matching stage. Consequently, determining which local feature descriptors offer the most discriminative power and the best matching performance is of significant interest to a large part of the computer vision community.

For more than a decade, SIFT [26] has arguably been the most popular feature descriptor for such tasks. Recently, the ability of neural networks to learn feature representations from data that are superior to prior hand-crafted ones has led to significant progress in the field of computer vision, *e.g.*, in object detection and recognition [12, 23, 41]. Consequently, neural networks have also been applied to the problem of descriptor learning [3, 14, 24, 42] in order to derive more discriminative representations for local features. The resulting methods demonstrate clear improvements over standard hand-crafted representations, such as SIFT [26], SURF [4], or DAISY [46]. However, there is usually no direct comparison with more advanced hand-crafted SIFT variants such as RootSIFT [2], RootSIFT-PCA [7], or DSP-SIFT [9]. Moreover, learned descriptors are typically evaluated on the patch classification benchmark from Brown *et al*. [6]. The task measures how well a descriptor can distinguish between related and unrelated patches based on their distance in descriptor space. Yet, a better performance on this benchmark does not necessarily imply a better matching quality, as shown by Balntas *et al*. [3]. For example, pruning steps such as Lowe's ratio test [26] or mutual nearest neighbor constraints might compensate for a higher false positive matching rate in terms of descriptor distance. Similarly, reaching a better average matching performance does not automatically imply a better performance in terms of subsequent processing steps. In the context of SFM, finding additional correspondences for image pairs where SIFT already provides enough matches does not necessarily result in more accurate or complete reconstructions. At the same time, descriptors with a better average matching performance might still not find enough correspondences to be able to handle hard image pairs where SIFT fails.

In this paper, we present a thorough experimental evaluation of learned and advanced hand-crafted feature descriptors in order to better understand their performance. In detail, this paper makes the following contributions: i) We provide a more detailed study of the matching performance of the different descriptors using a wider range of evaluation criteria and scenes than previous evaluations such as [3]. ii) Besides analyzing the matching quality in isolation of further processing steps, we also investigate the impact of different descriptors on the challenging and more practical task of image-based reconstruction. For example, this allows us to better determine whether learned descriptors can help to register hard images, *e.g*., photos depicting the scene under strong viewpoint or illumination changes. In addition, we are interested to understand to what extent a better matching performance affects the outcome of further processing stages, *e.g*., the accuracy and completeness of the models produced by SFM and MVS. iii) Our evaluation confirms that, as expected, learned descriptors often surpass SIFT on all evaluation metrics. However, we also observe that advanced versions of hand-crafted descriptors [7, 9] perform on par or better than the state-of-the-art learned feature descriptors, especially in the more complex SFM scenarios. As such, our paper demonstrates that there is still significant room for improvement for learning more powerful feature descriptors. iv) To facilitate further research in developing better descriptors, we make our benchmark publicly available[1]. This includes a large database corresponding patches.

2. Related Work

In the following, we provide a detailed overview of descriptor learning methods and a review of the hand-crafted descriptors used as baselines. In addition, we discuss the existing evaluation protocols and their limitations.

2.1. Descriptor Learning

Descriptor learning is usually formulated as a supervised learning problem. Given a set $\mathcal{P}$ of positive pairs and a set $\mathcal{N}$ of negative pairs, the objective is to learn a representation in which the descriptors belonging to the same physical object are close in descriptor space while unrelated descriptors are far apart. The approaches often differ in the exact definition of this property. For example, Simonyan *et al*. [43] use the *margin constraint*

$$d(\mathbf{p}_1, \mathbf{p}_2) + \tau < d(\mathbf{n}_1, \mathbf{n}_2) \;\; \forall(\mathbf{p}_1, \mathbf{p}_2) \in \mathcal{P}, (\mathbf{n}_1, \mathbf{n}_2) \in \mathcal{N}, \tag{1}$$

where $d(,)$ is a distance metric (usually L_2) and $\tau \in \mathbb{R}_{>0}$ is a margin. This approach can easily be extended to different types of positives and negatives, *e.g*., by using a larger margin τ_2 for random negative pairs and a smaller one $\tau_1 < \tau_2$ for negative pairs with a small initial distance [32]. Enforcing a small intra-class variance for descriptors belonging to the same physical point and a large inter-class variance for unrelated descriptors can also be expressed via a *hinge embedding* [29] or *contrastive* loss [13]

$$l(\mathbf{d}_1, \mathbf{d}_2) = \begin{cases} d(\mathbf{d}_1, \mathbf{d}_2) & \text{if } (\mathbf{d}_1, \mathbf{d}_2) \in \mathcal{P} \\ \max(0, \tau - d(\mathbf{d}_1, \mathbf{d}_2)) & \text{if } (\mathbf{d}_1, \mathbf{d}_2) \in \mathcal{N} \end{cases}, \tag{2}$$

which tries to enforce a minimum distance $\tau > 0$ between unrelated descriptors. As an alternative to working with pairs of descriptors, it is also possible to operate on triplets $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{n})$, with $(\mathbf{p}_1, \mathbf{p}_2) \in \mathcal{P}$ and $(\mathbf{p}_1, \mathbf{n}), (\mathbf{p}_2, \mathbf{n}) \in \mathcal{N}$. Potential cost functions are the *margin ranking loss* [51]

$$l(\mathbf{p}_1, \mathbf{p}_2, \mathbf{n}) = \max(0, \tau + d((\mathbf{p}_1, \mathbf{p}_2) - d(\mathbf{p}_1, \mathbf{n})) \tag{3}$$

and the *ratio loss* [18]

$$l(\mathbf{p}_1, \mathbf{p}_2, \mathbf{n}) = \left(\frac{\mathrm{e}^{\mathrm{d_p}}}{\mathrm{e}^{\mathrm{d_p}} + \mathrm{e}^{\mathrm{d_n}}}\right)^2 + \left(\frac{\mathrm{e}^{\mathrm{d_n}}}{\mathrm{e}^{\mathrm{d_p}} + \mathrm{e}^{\mathrm{d_n}}}\right)^2 , \tag{4}$$

where $d_p = d(\mathbf{p}_1, \mathbf{p}_2)$ and $d_p = d(\mathbf{p}_1, \mathbf{n})$. The latter tries to enforce that the distance between related descriptors is significantly smaller than the distance to an unrelated descriptor, without explicitly specifying a margin.

The input to the descriptor learning algorithm varies between the different approaches. For example, methods based on metric learning [52] often use a fixed descriptor representation as input and learn a discriminative metric for comparing descriptors [6, 32, 43, 44]. In contrast, approaches that learn a new descriptor representation usually operate on raw image patches [3, 42, 43, 55].

One way to obtain the large amount of training data required for learning is to extract positive and negative pairs from 3D models [6,24,44]. As a result of the reconstruction, each 3D point is associated with at least two image descriptors and their corresponding local patches. Consequently, the measurements from a single point form positive pairs while measurements from different 3D points are used to define negative pairs. While SFM already uses a descriptor, *e.g*. SIFT, to compute the pairwise feature matches used for reconstruction, the resulting models can still be used to learn more discriminative descriptors: Due to the transitivity of matching, a 3D point might be associated with patches A, B, and C. Correspondences might initially be obtained between A and B and between B and C, but not between A and C, *e.g*., due to a large viewpoint or illumination change. Thus, the data is suitable to learn a better descriptor that is able to directly match between A and C. An alternative to using SFM or MVS models is to use image retrieval techniques [31] to obtain the positive and negative pairs [32,43].

2.2. Learned Descriptors

Learning Patch and Descriptor Embeddings. Given an image patch, descriptor learning can be formulated as finding a discriminative embedding into a new space. For example, PCA-SIFT [22] uses principal component analysis (PCA) to embed a gradient image of a patch while Lepetit and Fua [25] embed patches using a random forest. Obviously, embeddings can also be applied to already existing descriptors, *e.g*., (Root)SIFT-PCA [7] employs PCA to project (Root)SIFT descriptors into a lower dimensional space. Philbin *et al*. [32] learn both linear and non-linear discriminative projections into lower dimensional spaces

[1] http://www.cvg.ethz.ch/research/local-feature-evaluation/

based on margin constraints. The non-linearity is implemented using a neural network with a single hidden layer. Simonyan *et al*. [43] model the problem of learning a discriminative projection into a low-dimensional space as a convex optimization problem. The resulting linear projections outperform the non-linear ones from Philbin *et al*. [32]. While the other methods learn embeddings into Euclidean spaces, Strecha *et al*. [44] propose a discriminative projection into a binary space where Hamming distances can be computed very efficiently. For our experimental evaluation, we use both RootSIFT-PCA and the projection learned by Simonyan *et al*. (in conjunction with the ConvOpt descriptor [43]). The former serves as a baseline method representing advanced hand-crafted descriptors.

Learning Pooling Regions. Hand-crafted and learned descriptors are constructed by applying a series of filter banks to an image patch, followed by pooling (*e.g*., into histogram bins in the case of SIFT) and normalization. Fixing the arrangement, *e.g*., on a polar grid, and the positions of the pooling regions, Brown *et al*. [6] learn descriptors by optimizing over the remaining pooling parameters such as the size of the regions. Following a similar approach, Trzcinski *et al*. [49] employ a boosting approach to learn binary descriptors from a set of weak learners that represent pooling strategies. Simonyan *et al*. [43] model the problem of learning the pooling regions as a convex energy minimization problem based on the margin constraint from Eq. 1. The size of the resulting *Convex Optimization* (*ConvOpt*) descriptor is controlled by enforcing sparsity when selecting a subset of the pooling regions. More complex descriptors can be trained by combining the learning of pooling regions with learning (linear) discriminative projections [6, 43]. In this paper, we use the *ConvOpt* descriptor, combined with a discriminative projection into a lower dimensional space [43], as a representative of approaches that learn pooling regions. It is selected since it outperforms both Brown *et al*. [6] and Trzcinski *et al*. [49]. As a baseline for hand-crafted descriptors, we employ *DSP-SIFT* [9], a variant of SIFT that pools gradients over multiple scales rather than only the scale at which the SIFT feature was detected.

Learning Filter Banks. While the approaches described in the previous paragraph [6, 43, 49] use a fixed set of filters and learn the pooling regions, approaches based on Convolutional Neural Networks (CNN) [3, 42] fix the pooling strategy and instead learn the filter banks. Simo-Serra *et al*. [42] use a siamese architecture [5] with a 3-layer CNN to minimize the contrastive loss from Eq. 2. Simo-Serra *et al*. notice that most randomly sampled negative patch pairs are easy to separate. In order to train their *Deep Descriptor* (*DeepDesc*), they thus mine for hard positive and negative pairs that can be used during learning. While Simo-Serra *et al*. [42] use pairs of patches, Balntas *et al*. [3] use a triplet network [18] consisting of two convolutional followed by one fully connected layer. Their *TFeat* descriptor is trained using hard-negative mining and Balntas *et al*. propose versions based on the margin ranking loss from Eq. 3 or the ratio loss from Eq. 4. We use both *DeepDesc* and *TFeat* trained with the margin ranking loss for our evaluation.

Joint Descriptor and Metric Learning. The approaches described above learn functions that map local image patches to discriminative descriptors embedded in a Euclidean space. As such, they employ the L_2 distance to compare descriptors. An alternative strategy is to jointly learn a descriptor representation and a distance metric that can be used to compare them [14, 54, 55]. Such approaches are potentially more powerful as deep neural networks can be used to implement a non-linear metric. However, this strength is also a great draw-back as it requires a forward pass through the learned model for comparing each pair of descriptors. Not only is such a pass computationally more complex than computing a single L_2 distance, but the network also prevents the use of traditional spatial subdivision schemes for fast (approximate) nearest neighbor search, such as kd-trees or hierarchical k-means trees [30]. This limits the scalability of methods that jointly learn a descriptor representation and a metric for comparison. In this paper, we evaluate datasets with millions of descriptors. Consequently, we focus on learned descriptors that can be efficiently compared via the L_2 distance.

Joint Detector and Descriptor Learning. The methods discussed above take an image patch as input and compute the corresponding feature descriptor as output. Hence, they are not tied to a single detector providing the patch but could easily be combined with any feature detector. However, jointly optimizing both the descriptor and detector should provide better results as the detector is trained to fire on regions that can be matched by the descriptor and vice versa. Recently, Yi *et al*. [24] proposed such an approach by combining the *DeepDesc* descriptor with a Difference-of-Gaussians (DoG)-like detector [26]. We include their LIFT feature in our evaluation.

2.3. Evaluation Protocols

Mikolajczyk *et al*. [28] evaluate affine region detectors by introducing standard metrics and small-scale datasets under various photometric and geometric image transformation. Later, Mikolajczyk and Schmid [27] extend this evaluation to several local descriptors. As a superset of this evaluation, Heinly *et al*. [16] evaluate binary descriptors and propose additional metrics and datasets.

Most learned descriptors are evaluated on the patch pair classification benchmark [6], which measures the ability of a descriptor to discriminate positive from negative patch pairs. The standard protocol of the benchmark is to generate the ROC curve by thesholding the distance values between pairs of patches. The final reported number is the false positive rate at 95% true positive rate (FPR95). However, as shown by Balntas *et al*. [3], a better FPR95 score does not automatically translate to better nearest neighbor matching because of usual filtering steps, such as Lowe's ratio test or the mutual nearest neighbor constraint. In practice, feature matching is typically followed by a geometric verification stage to prune outliers [1, 17, 31, 34, 36, 37, 39]. Due to the exponential complexity in the number of outliers [10], it is practically more important to have good precision for manageable runtimes of geometric verification. The authors of LIFT and TFeat make a first step to provide more insight into the practicality of the descriptors in a real-world application. Both evaluate their performance in terms of image-based reconstruction on the Strecha benchmark [44]. As we will show in this paper, this dataset is rather easy and provides only little practical insight.

To facilitate comparability with the evaluations by Mikolajczyk and Heinly *et al*., we follow their benchmark protocol to evaluate the raw matching performance on a per image pair basis. As the core contribution of this paper, we also study the impact of matching performance in the more practical setting of an image-based reconstruction pipeline [37, 40] using challenging small- and large-scale datasets. As part of the image-based reconstruction pipeline, SFM uses descriptor matching in the first stage to produce a graph of corresponding features in multiple views. Hence, all subsequent stages strongly rely on a good descriptor representation. Motivated by this, we derive evaluation metrics in all stages of the pipeline: feature matching, geometric verification, image retrieval, and sparse and dense modeling, in order to give new practical insights into the performance of the evaluated descriptors, as detailed in following section.

3. Evaluation

In the first part of this section, we detail and motivate the proposed evaluation protocol. The second part then presents and discusses the results of the evaluation.

3.1. Setup and Protocol

The following paragraphs describe the setup of our evaluation to ensure repeatability of the experiments. The entire protocol is provided to the public as an evaluation framework to foster future research in feature learning.

Evaluated Descriptors. We evaluate the performance of RootSIFT (short *SIFT*) [2] as a baseline descriptor, and RootSIFT-PCA (short *SIFT-PCA*) [7] and *DSP-SIFT* [9] as two representatives of advanced hand-crafted features. To evaluate the learned descriptors, we selected four state-of-the-art methods from the different groups of descriptor learning approaches: *ConvOpt* [43], *DeepDesc* [42], *TFeat* [3], and *LIFT* [24]. All features are evaluated using the same standardized test setup, as specified in the following.

Feature Detection. To ensure comparability between the evaluated descriptors, we use the standard SIFT keypoint detector for all descriptors but LIFT, which implements its own DoG-like detector. The SIFT detector uses DoG and we use 4 octaves starting with a two times up-sampled version of the original image, 3 scales per octave, a peak threshold of $\frac{0.02}{3}$, an edge threshold of 10, and a maximum of 2 detected orientations per keypoint location. These values have been optimized for the purpose of SFM and are, *e.g*., used as defaults in COLMAP [37, 40]. Following standard procedure by the original methods, we then extract 64×64 pixel patches as the input to each descriptor. Note that all descriptors have been learned based on DoG keypoints. We experimented with different detector settings for LIFT and found that the defaults by the authors performed best. On average, DoG detects 5,262 and LIFT 4,173 features for the images in the Oxford5k dataset [31].

Descriptor Matching. Throughout all experiments, the L_2 distance serves as an efficient distance metric to calculate the similarity between two descriptors. To compute the correspondences between pairs of images, we enforce mutual nearest neighbors, *i.e*., a corresponding descriptor in one image must be the nearest neighbor for the corresponding descriptor in the other image and vice versa. This has been shown to reduce the amount of false correspondences for ambiguous structures and significantly improved the results for all descriptors [16, 37]. In contrast to standard practice in SIFT matching, we do not enforce the ratio test by pruning descriptors whose top-ranked nearest neighbors are very similar. The reason being that the ratio test is highly dependent on the distribution of descriptor distances [26]. Preliminary experiments showed that the ratio test is not generally applicable to any of our evaluated descriptors but SIFT. For the smaller datasets with up to 2,000 images, we exhaustively compute correspondences between all pairs of images. For the larger datasets, we use Bag-of-Words (BoW) to match each image only against a fixed number of top-ranked neighbor images. For the nearest neighbor search, we employ a state-of-the-art image retrieval system [38] using Hamming embedding [20] and visual burstiness weighting [21]. Following standard procedure, we ensure that the vocabulary is trained on a completely unrelated image collection. Correspondingly, we use a vocabulary of 262,144 words with a branching factor of 512 trained offline on Oxford5k [31] for all the experiments. To ensure a good quantization of the descriptor space and to evaluate the performance of each descriptor on the task of image retrieval, we train a custom vocabulary for each descriptor.

Geometric Verification. Descriptor matching as described in the previous paragraph is solely based on appearance information. For the purpose of SFM and to quantify the matching performance on a per image pair basis, we estimate the two-view geometry and determine the resulting inlier correspondences using the multi-model geometric verification approach described in [37]. Moreover, we are interested in quantifying the matching performance in the practical context of image-based reconstruction. Towards this goal, we use the successfully verified image pairs with a minimum of 15 inlier feature correspondences as the input to COLMAP [37, 40]. While both the sparse and dense reconstruction results provide insight into the practicality of the descriptors in a real-world application, SFM also implements a much stricter and more accurate geometric verification tool using multi-view information, as compared to the initial two-view verification. Hence, we also evaluate key metrics of the resulting sparse and dense reconstructions produced by SFM and MVS, as detailed in the following.

Matching Metrics. Equivalent to the binary descriptor evaluation by Heinly *et al*. [16], we first evaluate the raw matching performance on a per image pair basis using the standard metrics *Putative Match Ratio*, *Precision*, *Matching Score*, and *Recall*. First, the *Putative Match Ratio = #Putative Matches / #Features* quantifies the selectivity of the descriptor in terms of the fraction of the detected features initially identified as a match. Second, the *Precision = #Inlier Matches / #Putative Matches* defines the inlier ratio of the putative matches, as determined by geometric verification. The *Matching Score = #Inlier Matches / #Features* defines the number of initial features that will result in inlier matches. Last, the *Recall = #Inlier Matches / #True Matches* describes the number of identified ground-truth matches. We refer the reader to Heinly *et al*. [16] for more details and an in-depth motivation of these metrics.

Reconstruction Metrics. In addition to evaluating the raw matching performance on individual image pairs, we also evaluate the performance of the different descriptors in the practical and more challenging setting of image-based reconstruction. Typically, the image-based reconstruction pipeline first uses SFM to calibrate the cameras of the input images and to infer a sparse model of the scene. Then, the output of SFM serves as the input to MVS to obtain a dense representation of the scene, *e.g*., in the form of depth maps, a dense point cloud, or a meshed surface model. Generally, the ultimate goal of image-based reconstruction is to produce high-quality 3D models. The quality of SFM results strongly depends on accurate and complete two-view correspondences as input, and MVS relies on an accurate and complete SFM reconstruction [37]. Thus, SFM and MVS results are good indicators for the descriptor performance in the initial feature matching stage. Furthermore, by chaining two-view correspondences into a graph of feature tracks [37], SFM can exploit multi-view redundancy to more reliably verify the validity of correspondences. To evaluate the completeness and accuracy of the reconstruction results, we determine a number of key metrics: First, the number of *registered images* and *sparse points* quantify the completeness of the reconstruction. A larger number of registered images enables more complete MVS reconstruction and a larger number of 3D points with many image observations constitute a more complete and accurate scene representation. Second, we determine the number of *observations per image*, *i.e*., the number of verified image projections of sparse points, and the *track length*, *i.e*., the number of verified image observations per sparse point. These two metrics are crucial for an accurate calibration of the cameras and reliable triangulation, as they provide redundancy in the estimation. Third, bundle adjustment stands at the core of SFM as a joint non-linear refinement of the cameras and points. The overall *reprojection error* in bundle adjustment indicates the accuracy of the reconstruction and is mainly impacted by the accuracy and redundancy of the input data, which depend on the completeness of the graph of feature correspondences and the keypoint localization accuracy. For a subset of the datasets, ground-truth camera locations are available, and we evaluate the mean *metric pose accuracy* of the camera locations by aligning the reconstructed model to the ground-truth using robust 3D similarity transformation estimation. Last, the MVS problem boils down to dense correspondence estimation between multiple views. To produce accurate and complete results, MVS requires an accurate intrinsic and extrinsic camera calibration. Moreover, more registered images provide additional multi-view photo-consistency constraints and lead to more complete results. Hence, we determine the number of reconstructed *dense points* as a single measure of the overall completeness of the reconstruction and the accuracy of the SFM results. In addition, we have ground-truth depth maps for a subset of the datasets to also directly evaluate the metric accuracy and absolute completeness of the dense reconstruction results.

Datasets. We evaluate all descriptors on existing small- and large-scale benchmark datasets. For the two-view evaluation, we follow the evaluation protocol and the datasets provided by Heinly *et al*. [16]. The benchmark tests the descriptor performance with respect to different types and levels of photometric and geometric image transformations (image blur, exposure, white balance, JPEG compression, scale and/or rotation, planar and non-planar geometry, illumination, *etc*.). For the reconstruction evaluation, we employ various existing benchmark datasets. The well-known MVS benchmark by Strecha *et al*. [45] (*Fountain* and *Herzjesu*) consists of around 10 high-resolution images per dataset with highly accurate ground-truth camera locations and dense depth maps. To evaluate the completeness and accuracy of the depth maps, we follow the evaluation protocol by Hu and Mordohai [19]. Next, we evaluate the performance on the *South Building* dataset [15], which consists of 128 highly overlapping images with mostly repetitive scene structure captured by the same camera in a struc-

	SIFT	SIFT-PCA	DSP-SIFT	ConvOpt	DeepDesc	TFeat	LIFT
Dimensionality	128	80	128	73	128	128	128
Size [bytes]	128	320	512	292	512	512	512
Platform	CPU	CPU	CPU	GPU	GPU	GPU	GPU
Extraction [s]	9.3	10.5	23.7	49.9	24.3	11.8	212.3
Matching [s]	0.14	0.11	0.14	0.10	0.14	0.14	0.14

Table 1. Key properties of the evaluated descriptors. Average timings reported for the Oxford5k dataset. Extraction speed includes keypoint detection and are specified per image. Matching speed is specified per image pair.

tured pattern around the building. Finally, Internet photo collections present the descriptors with more challenges due to the high variance in the input data. We test the descriptors on the large-scale Internet datasets by Wilson and Snavely [53]. Each dataset contains several thousand images of well-known landmarks across the world collected from Flickr. To simulate a harder matching and reconstruction scenario, each dataset is embedded into a distractor set of unrelated images. As such, the descriptors must generalize well to the heterogeneity of Internet data to robustly handle effects such as large illumination and viewpoint changes, repetitive structure, image compression and distortion artifacts, or unrelated distractor images. Finally, we evaluate the reconstruction performance on the large-scale *Cornell* dataset by Crandall *et al*. [8]. The dataset consists of 6,514 unstructured and uncalibrated images of the Cornell campus. The images were taken in a relatively sparse pattern during different seasons and times of the day and thus pose extreme challenges to the descriptors in terms of illumination and viewpoint changes. A subset of 348 images is equipped with ground-truth camera locations obtained through surveying methods that we use to evaluate the pose accuracy. We use the *Oxford5k* dataset [31] to train the visual vocabulary for image matching.

Implementation. To enable comparability in the timings, all experiments were conducted on the same machine with two 14-core Intel E5-2697 2.60GHz CPUs, 512GB of RAM, and 4 NVIDIA Titan X. We use the SIFT implementation by VLFeat [50] and, for all other descriptors, the open-source implementations and models provided by the authors. Traditionally, the descriptor learning models are trained on the multi-view correspondence dataset by Brown *et al*. [6]. We choose their best-performing pretrained models, if multiple are provided. The descriptor matching uses an efficient GPU implementation, and we use COLMAP [37, 40] for the SFM and MVS evaluation, while CMVS [11] is used to cluster the larger datasets into more manageable image clusters for the dense reconstruction.

3.2. Results and Discussion

Performance. Table 1 summarizes the key performance properties for each descriptor including timings, memory requirements, *etc*. on the Oxford5k dataset. The memory footprint and the descriptor dimensionality have important implications for the required storage capacity for large-scale datasets, since we evaluate datasets containing thousands of images with millions of descriptors. For example, the raw SIFT keypoints and descriptors for Cornell already comprise $\approx$ 11GB of data. Furthermore, the descriptor dimensionality impacts the speed of the descriptor matching, which in practice has squared complexity in terms of the number of features per image when using efficient exhaustive GPU matching. Due to its low dimensionality, ConvOpt provides $\approx 40\%$ faster feature matching. Among the different descriptors, there is a large variance in extraction speed. In theory, when implemented efficiently, both SIFT-PCA and DSP-SIFT have only small overhead over standard SIFT. While ConvOpt is relatively slow to extract, it is significantly faster in the matching stage due to its low dimensionality. Conversely, TFeat is relatively fast to extract and slower in the matching stage, similar to the other descriptors with 128 dimensions. LIFT is the slowest method by a large margin. In general, the extraction of the hand-crafted descriptors is much faster as compared to the learned features despite running on the CPU. As such, the learned features are currently not a practical alternative for processing millions of images, such as in the streaming-based reconstruction pipeline by Heinly *et al*. [17] who report a throughput

	SIFT	SIFT-PCA	DSP-SIFT	ConvOpt	DeepDesc	TFeat	LIFT
			Putative Match Ratio in %				
Blur	3.7	**5.7**	**7.0**	5.2	4.6	4.2	**6.5**
JPEG	20.9	**29.3**	**34.0**	26.8	24.4	22.9	**27.5**
Exposure	33.0	**34.1**	**35.3**	32.8	10.4	31.2	**34.9**
Day-Night	5.5	**6.8**	**6.2**	**7.2**	3.6	5.5	5.4
Scale	12.1	**25.2**	**23.4**	**23.8**	23.0	21.5	19.6
Rotation	**12.8**	**17.6**	**17.3**	10.0	11.9	8.7	1.3
Scale-rotation	2.4	**6.0**	**5.8**	**4.7**	4.5	3.7	2.0
Planar	5.9	**10.0**	**10.1**	**9.4**	7.7	8.0	8.0
Non-planar	7.8	**8.8**	**8.7**	**8.4**	7.4	7.2	8.3
Internet	3.2	**4.6**	**4.4**	4.3	2.7	3.4	**4.8**
			Precision in %				
Blur	43.8	**46.5**	**48.4**	45.2	41.9	**46.3**	44.5
JPEG	**98.5**	**96.5**	**98.3**	94.1	91.6	95.8	95.9
Exposure	**99.3**	**98.0**	**98.6**	96.6	68.0	97.3	97.5
Day-Night	**93.8**	**80.4**	**77.8**	73.9	37.8	76.5	71.2
Scale	43.0	**95.5**	**95.5**	92.2	89.1	**94.3**	89.1
Rotation	**33.2**	**33.1**	**33.1**	32.2	32.3	32.3	7.9
Scale-rotation	32.8	**46.7**	**46.8**	42.3	39.1	**43.9**	18.7
Planar	33.9	**37.3**	**39.9**	**34.3**	32.5	33.6	33.2
Non-planar	**43.3**	**42.2**	**43.1**	38.4	34.5	39.3	40.4
Internet	**39.8**	**40.3**	**39.7**	35.6	27.2	36.6	37.1
			Matching Score in %				
Blur	3.7	**5.5**	**6.8**	4.9	4.1	4.0	**6.2**
JPEG	20.8	**28.8**	**33.7**	26.1	23.5	22.6	**27.1**
Exposure	32.8	**33.5**	**34.9**	31.8	9.1	30.5	**34.2**
Day-Night	5.3	**5.9**	**5.5**	**5.8**	1.8	4.7	4.3
Scale	11.7	**24.4**	**22.8**	**22.6**	21.3	20.7	18.2
Rotation	**12.8**	**17.5**	**17.2**	9.7	11.6	8.5	0.9
Scale-rotation	2.4	**5.8**	**5.6**	**4.3**	3.9	3.5	1.6
Planar	5.7	**9.6**	**9.9**	**8.7**	6.9	7.5	7.4
Non-planar	7.7	**8.4**	**8.4**	**7.8**	6.5	6.9	7.7
Internet	3.1	**4.1**	**4.0**	3.5	1.8	2.8	**4.1**
			Recall in %.				
Blur	17.0	**22.4**	**27.2**	**20.0**	16.9	17.0	17.9
JPEG	37.9	**51.6**	**62.8**	46.6	41.0	39.2	**51.5**
Exposure	**79.0**	**81.0**	**84.1**	76.5	18.2	73.1	64.0
Day-Night	25.6	**29.2**	**26.2**	**28.9**	8.4	22.9	19.3
Scale	22.4	**84.0**	73.9	**76.1**	71.9	68.9	**98.4**
Rotation	**20.8**	**28.5**	**28.1**	16.1	19.1	14.1	2.3
Scale-rotation	6.4	**16.4**	**15.2**	**12.0**	10.9	9.6	5.3
Planar	11.4	**18.0**	**18.6**	16.4	13.3	14.2	**17.9**

Table 2. Evaluation results for the descriptor benchmark by Heinly *et al*. [16]. **First**, **second**, **third** best results highlighted in bold.

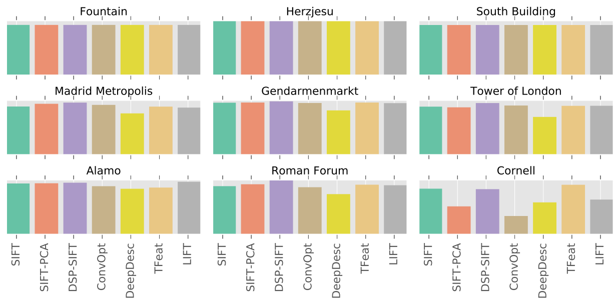

Figure 1. Number of registered images for the different methods.

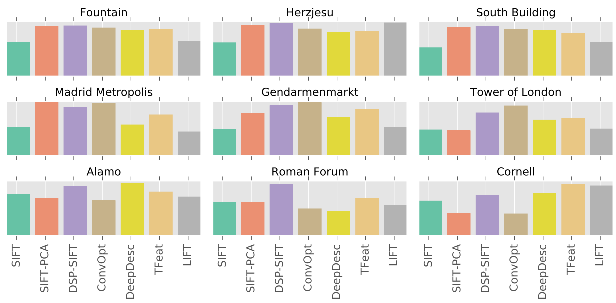

Figure 2. Number of sparse points for the different methods.

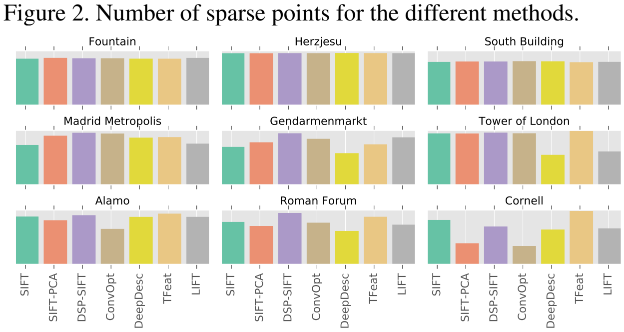

Figure 3. Number of dense points for the different methods.

of 20 images per second on a single GPU.

Image Matching. Table 2 shows the results for the datasets and metrics of the descriptor evaluation benchmark by Heinly *et al*. [16]. The results give insight into which image transformations are particularly challenging for the descriptors. We observe that all descriptors consistently perform worse across the different metrics in the case of image blur, day-night, and large viewpoint change. As expected, the learned descriptors typically outperform SIFT in terms of recall, while SIFT performs better in terms of precision. Surprisingly though, the advanced SIFT variants outperform the learned features for almost all metrics and matching scenarios. Notably, the performance of the learned descriptors often has a high variance across the different datasets, which indicates over-fitting for specific image transformations, *e.g*., due to a lack of training data depicting the entire appearance space of patches. Note that LIFT has problems with matching between rotated images, since it was trained on mostly upright Internet images. Among the learned descriptors, ConvOpt produces overall the best results and has the lowest variance across the different datasets. Table 3 presents the results for the image-based reconstruction benchmark and the *# Inlier Pairs* and *# Inlier Matches* metrics demonstrate a similar matching behavior in the large-scale setting. Next, we discuss, how the isolated matching performance impacts the image-based reconstruction results in practice.

Reconstruction. Table 3 lists the numerical values for the reconstruction evaluation, while Figures 1, 2, and 3 visualize the relative performance of the methods qualitatively. For the two smaller Strecha datasets (Fountain and Herzjesu), which were also evaluated by the authors of LIFT and TFeat, and the South Building dataset, the learned descriptors generally perform on par with or better than SIFT in terms of the number of sparse points, the number of image observations, and the mean track length. As a consequence of a better matching performance, the two advanced SIFT versions produce significantly better results than the other methods in these metrics. However, looking at the number of registered images, and the final dense modeling performance and accuracy metrics, all methods produce roughly the same reconstruction quality. We interpret these results as an indication that the Strecha and South Building datasets are rather easy benchmarks due to the structured camera setup with high overlap, same illumination conditions, *etc*. The higher variance in the results for the larger-scale Internet datasets confirms this interpretation. Here, Madrid Metropolis, Gendarmenmarkt, and Tower of London were matched exhaustively, whereas the images in Alamo, Roman Forum, and Cornell were only matched against the 100 nearest neighbors found using image retrieval. The matching and reconstruction results therefore also test the discriminative power of the descriptors in the context of BoW-based image retrieval. In the more challenging case of Internet photos, the matching performance directly impacts the ability to obtain complete and accurate models. Opposed to our observations in the raw matching evaluation, where SIFT produces inferior results as compared to the learned descriptors, in the reconstruction evaluation, SIFT performs typically on par with the learned descriptors. This implies that a better matching performance does not necessarily lead to better reconstruction results. DSP-SIFT performs best among all the methods, both in terms of sparse and dense reconstruction results. It consistently produces the most complete sparse reconstruction in terms of the number of registered images and reconstructed sparse points, while the dense models have the most points as a result of accurate camera registration. The mean reprojection error is similarly good for the descriptors that use the DoG keypoint detector, with a slightly larger error for DSP-SIFT, which is potentially caused by the descriptor pooling across multiple scales leading to more robustness w.r.t. in-

		# Images	# Registered	# Sparse Points	# Observations	Track Length	Reproj. Error	# Inlier Pairs	# Inlier Matches	# Dense Points	Pose Error	Dense Error
Fountain	*SIFT*	11	11	10,004	44K	4.49	**0.30px**	49	76K	2,970K	0.002m (0.002m)	0.77 (0.90)
	SIFT-PCA		11	**14,608**	**70K**	**4.80**	0.39px	**55**	**124K**	**3,021K**	0.002m (0.002m)	0.77 (0.90)
	DSP-SIFT		11	**14,785**	**71K**	**4.80**	0.41px	54	**129K**	2,999K	0.002m (0.002m)	0.77 (0.90)
	ConvOpt		11	**14,179**	**67K**	**4.75**	0.37px	**55**	**114K**	**2,999K**	0.002m (0.002m)	0.77 (0.90)
	DeepDesc		11	13,519	61K	4.55	**0.35px**	**55**	93K	2,972K	0.002m (0.002m)	0.77 (0.90)
	TFeat		11	13,696	64K	4.68	**0.35px**	54	103K	2,969K	0.002m (0.002m)	0.77 (0.90)
	LIFT		11	10,172	46K	4.55	0.59px	**55**	83K	**3,019K**	0.002m (0.002m)	0.77 (0.90)
Herzjesu	*SIFT*	8	8	4,916	19K	4.00	**0.32px**	27	28K	2,373K	0.004m (0.004m)	0.57 (0.73)
	SIFT-PCA		8	**7,433**	**31K**	**4.19**	0.42px	**28**	**47K**	2,372K	0.004m (0.004m)	0.57 (0.73)
	DSP-SIFT		8	**7,760**	**32K**	**4.19**	0.45px	**28**	**50K**	**2,376K**	0.004m (0.004m)	0.57 (0.73)
	ConvOpt		8	6,939	28K	**4.13**	0.40px	**28**	42K	2,375K	0.004m (0.004m)	0.57 (0.73)
	DeepDesc		8	6,418	25K	3.92	**0.38px**	**28**	34K	**2,380K**	0.004m (0.004m)	0.57 (0.73)
	TFeat		8	6,606	27K	4.09	**0.38px**	**28**	38K	**2,377K**	0.004m (0.004m)	0.57 (0.73)
	LIFT		8	**7,834**	**30K**	3.95	0.63px	**28**	**46K**	2,375K	0.004m (0.004m)	0.57 (0.73)
South Building	*SIFT*	128	128	62,780	353K	5.64	**0.42px**	1K	1,003K	1,972K	–	–
	SIFT-PCA		128	**107,674**	**650K**	**6.04**	0.54px	**3K**	**2,019K**	1,993K	–	–
	DSP-SIFT		128	**110,394**	**664K**	**6.02**	0.57px	3K	**2,079K**	**1,994K**	–	–
	ConvOpt		128	**103,602**	**617K**	5.96	0.51px	**4K**	**1,856K**	**2,007K**	–	–
	DeepDesc		128	101,154	558K	5.53	**0.48px**	**6K**	1,463K	**2,002K**	–	–
	TFeat		128	94,589	566K	**5.99**	**0.49px**	3K	1,567K	1,960K	–	–
	LIFT		128	74,607	399K	5.35	0.78px	3K	1,168K	1,975K	–	–
Madrid Metropolis	*SIFT*	1,344	440	62,729	416K	**6.64**	**0.53px**	14K	1,740K	435K	–	–
	SIFT-PCA		**465**	**119,244**	**702K**	5.89	0.57px	**27K**	**3,597K**	**537K**	–	–
	DSP-SIFT		**476**	**107,028**	**681K**	**6.36**	0.64px	**21K**	**3,155K**	**570K**	–	–
	ConvOpt		**455**	**115,134**	**634K**	5.51	0.57px	**29K**	**3,148K**	**561K**	–	–
	DeepDesc		377	68,110	348K	5.11	**0.53px**	19K	1,570K	516K	–	–
	TFeat		439	90,274	512K	5.68	**0.54px**	18K	2,135K	522K	–	–
	LIFT		430	52,755	337K	**6.40**	0.76px	13K	1,498K	450K	–	–
Gendarmenmarkt	*SIFT*	1,463	950	169,900	1,010K	**5.95**	**0.64px**	28K	3,292K	1,104K	–	–
	SIFT-PCA		**953**	272,118	**1,477K**	**5.43**	0.69px	**43K**	**5,137K**	1,240K	–	–
	DSP-SIFT		**975**	**321,846**	**1,732K**	**5.38**	0.74px	**56K**	**7,648K**	**1,505K**	–	–
	ConvOpt		945	**341,591**	**1,601K**	4.69	0.70px	**56K**	**6,525K**	**1,342K**	–	–
	DeepDesc		809	244,925	949K	3.88	**0.68px**	31K	2,849K	921K	–	–
	TFeat		**953**	**297,266**	1,445K	4.86	**0.66px**	39K	4,685K	1,181K	–	–
	LIFT		942	180,746	964K	5.34	0.83px	27K	2,495K	**1,386K**	–	–
Tower of London	*SIFT*	1,576	702	142,746	963K	6.75	**0.53px**	18K	3,211K	1,126K	–	–
	SIFT-PCA		692	137,800	1,090K	**7.91**	0.60px	12K	2,455K	1,124K	–	–
	DSP-SIFT		**755**	**236,598**	**1,761K**	**7.44**	0.64px	**33K**	**8,056K**	**1,143K**	–	–
	ConvOpt		**719**	**274,987**	**1,732K**	6.30	0.62px	**39K**	**7,542K**	**1,129K**	–	–
	DeepDesc		551	196,990	964K	4.90	**0.55px**	25K	2,745K	653K	–	–
	TFeat		714	**206,142**	**1,424K**	6.91	**0.57px**	**28K**	**5,333K**	**1,182K**	–	–
	LIFT		**715**	147,851	1,045K	**7.07**	0.72px	23K	4,079K	729K	–	–
Alamo	*SIFT*	2,915	743	120,713	1,384K	11.47	0.54px	**23K**	**7,671K**	**611K**	–	–
	SIFT-PCA		**746**	108,553	1,377K	**12.69**	0.55px	12K	4,669K	564K	–	–
	DSP-SIFT		**754**	**144,341**	**1,815K**	**12.58**	0.66px	**16K**	**10,115K**	**629K**	–	–
	ConvOpt		703	102,044	1,001K	9.81	**0.48px**	3K	850K	452K	–	–
	DeepDesc		665	**152,537**	1,207K	7.92	**0.48px**	16K	4,196K	607K	–	–
	TFeat		683	**127,642**	**1,443K**	11.31	**0.52px**	16K	6,356K	**648K**	–	–
	LIFT		**768**	112,984	**1,477K**	**13.08**	0.73px	**23K**	**9,117K**	607K	–	–
Roman Forum	*SIFT*	2,364	1,407	242,192	1,805K	**7.45**	0.61px	**25K**	**6,063K**	**3,097K**	–	–
	SIFT-PCA		**1,463**	**244,556**	**1,834K**	**7.50**	0.61px	16K	4,322K	2,799K	–	–
	DSP-SIFT		**1,583**	**372,573**	**2,879K**	**7.73**	0.71px	**26K**	**9,685K**	**3,748K**	–	–
	ConvOpt		1,376	195,305	1,173K	6.01	**0.55px**	11K	2,111K	3,043K	–	–
	DeepDesc		1,173	174,532	1,275K	7.31	**0.60px**	9K	1,834K	2,434K	–	–
	TFeat		**1,450**	**271,902**	**1,963K**	7.22	**0.61px**	**19K**	**5,584K**	**3,477K**	–	–
	LIFT		1,434	220,026	1,608K	7.31	0.75px	17K	4,732K	2,898K	–	–
Cornell	*SIFT*	6,514	**4,999**	1,010,544	6,317K	6.25	**0.53px**	71K	25,603K	**12,970K**	**1.537m (0.793m)**	–
	SIFT-PCA		3,049	640,553	4,335K	**6.77**	**0.54px**	26K	13,793K	6,135K	11.498m (1.088m)	–
	DSP-SIFT		**4,946**	1,177,916	**7,233K**	6.14	0.67px	73K	26,150K	**11,066K**	**2.943m (1.001m)**	–
	ConvOpt		1,986	632,613	4,747K	**7.50**	0.57px	42K	18,615K	5,321K	5.824m (0.904m)	–
	DeepDesc		3,489	**1,225,780**	6,977K	5.69	**0.55px**	**73K**	**28,845K**	10,159K	3.832m (0.695m)	–
	TFeat		**5,428**	**1,499,117**	**9,830K**	**6.56**	0.59px	**89K**	**40,640K**	**15,605K**	**2.126m (0.593m)**	–
	LIFT		3,798	**1,455,732**	**7,377K**	5.07	0.71px	**81K**	**39,812K**	10,512K	3.113m (0.712m)	–

Table 3. Results for our reconstruction benchmark. Pose error as mean (median) over all images. Dense error for 2cm (10cm) threshold [19]. **First**, **second**, **third** best results highlighted in bold. Number of images, sparse points, and dense points visualized in Figs. 1, 2, and 3.

accurate keypoint localization. Surprisingly, LIFT produces the largest reprojection error and relatively short tracks for all datasets, indicating inferior keypoint localization performance as compared to the hand-crafted DoG method. In addition, even though it was trained on the Roman Forum model, it does not perform better than DSP-SIFT or TFeat.

4. Conclusion

This paper presented a thorough experimental evaluation of learned and advanced hand-crafted feature descriptors to better understand their performance across a wide range of scenarios. The evaluation demonstrated that advanced hand-crafted features still perform on par or better than recent learned features in the practical context of image-based reconstruction. The current generation of learned descriptors shows a high variance across different datasets and applications. This clearly evidences the necessity to evaluate a descriptor's discriminative power over a wide range of datasets. In addition, to overcome the demonstrated limitations, we believe that the next generation of learned descriptors needs more training data. To facilitate further research, we make our full evaluation pipeline and a large training dataset of patches publicly available.

Acknowledgements This project received funding from the European Union's Horizon 2020 research and innovation program under grant No. 688007 (TrimBot2020).

References

[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Comm. ACM*, 2011. 1, 4

[2] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012. 1, 4

[3] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *BMVC*, 2016. 1, 2, 3, 4

[4] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded-Up Robust Features. In *ECCV*, 2006. 1

[5] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature Verification using a "Siamese" Time Delay Neural Network. In *NIPS*. 1994. 3

[6] M. Brown, G. Hua, and S. Winder. Discriminative Learning of Local Image Descriptors. *PAMI*, 2011. 1, 2, 3, 6

[7] A. Bursuc, G. Tolias, and H. Jégou. Kernel local descriptors with implicit rotation matching. In *ACM Multimedia*, 2015. 1, 2, 4

[8] D. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher. Discrete-Continuous Optimization for Large-Scale Structure from Motion. *CVPR*, 2011. 6

[9] J. Dong and S. Soatto. Domain-size pooling in local descriptors: DSP-SIFT. In *CVPR*, 2015. 1, 2, 3, 4

[10] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 1981. 4

[11] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *PAMI*, 2010. 6

[12] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 1

[13] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006. 2

[14] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. MatchNet: Unifying feature and metric learning for patch-based matching. *CVPR*, 2015. 1, 3

[15] C. Hane, C. Zach, A. Cohen, R. Angst, and M. Pollefeys. Joint 3d scene reconstruction and class segmentation. In *CVPR*, 2013. 5

[16] J. Heinly, E. Dunn, and J.-M. Frahm. Comparative Evaluation of Binary Features. In *ECCV*, 2012. 3, 4, 5, 6, 7

[17] J. Heinly, J. L. Schönberger, E. Dunn, and J.-M. Frahm. Reconstructing the World* in Six Days *(As Captured by the Yahoo 100 Million Image Dataset). In *CVPR*, 2015. 1, 4, 6

[18] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In *SIMBAD*, 2015. 2, 3

[19] X. Hu and P. Mordohai. Least commitment, viewpoint-based, multi-view stereo. In *3DIMPVT*, 2012. 5, 8

[20] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008. 4

[21] H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *CVPR*, 2009. 4

[22] Y. Ke and R. Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. *CVPR*, 2004. 2

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012. 1

[24] M. Kwang, E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned Invariant Feature Transform. *ECCV*, 2016. 1, 2, 3, 4

[25] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *PAMI*, 2006. 2

[26] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 3, 4

[27] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 2005. 3

[28] 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*, 2005. 3

[29] H. Mobahi, R. Collobert, and J. Weston. Deep learning from temporal coherence in video. *ICML*, 2009. 2

[30] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*, 2009. 3

[31] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. 1, 2, 4, 6

[32] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Descriptor Learning for Efficient Retrieval. In *ECCV*, 2010. 2, 3

[33] F. Radenović, J. L. Schönberger, D. Ji, J.-M. Frahm, O. Chum, and J. Matas. From dusk till dawn: Modeling in the dark. In *CVPR*, 2016. 1

[34] T. Sattler, M. Havlena, K. Schindler, and M. Pollefeys. Large-scale location recognition and the geometric burstiness problem. In *CVPR*, 2016. 1, 4

[35] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *ICCV*, 2011. 1

[36] T. Sattler, B. Leibe, and L. Kobbelt. Efficient effective prioritized matching for large-scale image-based localization. *PAMI*, 2016. 1, 4

[37] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 4, 5, 6

[38] J. L. Schönberger, T. Price, T. Sattler, J.-M. Frahm, and M. Pollefeys. A vote-and-verify strategy for fast spatial verification in image retrieval. In *ACCV*, 2016. 4

[39] J. L. Schönberger, F. Radenović, O. Chum, and J.-M. Frahm. From single image query to detailed 3d reconstruction. In *CVPR*, 2015. 1, 4

[40] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 1, 4, 5, 6

[41] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *CVPR Workshops*, 2014. 1

[42] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *ICCV*, 2015. 1, 2, 3, 4

[43] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *PAMI*, 2014. 2, 3, 4

[44] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua. LDA-Hash: Improved Matching with Smaller Descriptors. *PAMI*, 2012. 2, 3, 4

[45] C. Strecha, W. von Hansen, L. V. Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *CVPR*, 2008. 5

[46] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *PAMI*, 2010. 1

[47] G. Tolias, Y. Avrithis, and H. Jégou. Image search with selective match kernels: aggregation across single and multiple images. *IJCV*, 2016. 1

[48] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, 2015. 1

[49] T. Trzcinski, M. Christoudias, and V. Lepetit. Learning Image Descriptors with Boosting. *PAMI*, 2015. 3

[50] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008. 6

[51] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning Fine-Grained Image Similarity with Deep Ranking. In *CVPR*, 2014. 2

[52] K. Q. Weinberger and L. K. Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *J. Mach. Learn. Res.*, 2009. 2

[53] K. Wilson and N. Snavely. Robust global translations with 1dsfm. *ECCV*, 2014. 6

[54] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. *CVPR*, 2015. 3

[55] A. R. Zamir, T. Wekel, P. Agrawal, C. Wei, J. Malik, and S. Savarese. Generic 3D Representation via Pose Estimation and Matching. In *ECCV*, 2016. 2, 3

[56] B. Zeisl, T. Sattler, and M. Pollefeys. Camera Pose Voting for Large-Scale Image-Based Localization. In *ICCV*, 2015. 1