This CVPR2015 paper is the Open Access version, provided by the Computer Vision Foundation.
The authoritative version of this paper is available in IEEE Xplore.

Domain-Size Pooling in Local Descriptors: DSP-SIFT

Jingming Dong Stefano Soatto

UCLA Vision Lab, University of California, Los Angeles, CA 90095
{dong,soatto}@cs.ucla.edu

Abstract

We introduce a simple modification of local image descriptors, such as SIFT, based on pooling gradient orientations across different domain sizes, in addition to spatial locations. The resulting descriptor, which we call DSP-SIFT, outperforms other methods in wide-baseline matching benchmarks, including those based on convolutional neural networks, despite having the same dimension of SIFT and requiring no training.

1. Introduction

Local image descriptors, such as SIFT [24] and its variants, are designed to reduce variability due to illumination and vantage point while retaining discriminative power. This facilitates finding correspondence between different views of the same underlying scene. In a wide-baseline matching task on the Oxford benchmark [28, 27], nearest-neighbor SIFT descriptors achieve a mean average precision (mAP) of 27.50%, a 71.85% improvement over direct comparison of normalized grayscale values. Other datasets yield similar results [29]. Functions that reduce sensitivity to nuisance variability can also be learned from data [26, 38, 40, 42, 30]. Convolutional neural networks (CNNs) can been trained to "learn away" nuisance variability while retaining class labels using large annotated datasets. In particular, [15] uses (patches of) natural images as surrogate classes and adds transformed versions to train the network to discount nuisance variability. The activation maps in response to image values can be interpreted as a descriptor and used for correspondence. [15, 12] show that the CNN outperforms SIFT, albeit with a much larger dimension. Here we show that a simple modification of SIFT, obtained by pooling gradient orientations across different domain sizes ("scales"), in addition to spatial locations, improves it by a considerable margin, also outperforming the best CNN. We call the resulting descriptor "domain-size pooled" SIFT, or DSP-SIFT.

Pooling across different domain sizes is implemented in few lines of code, can be applied to any histogram-based method (Sect. 3), and yields a descriptor of the same size that outperforms the original essentially uniformly (Fig. 4). Yet combining histograms of images of different sizes is counterintuitive and seemingly at odds with the teachings of scale-space theory and the resulting established practice of *scale selection* [23] (Sect. 1.1). It is, however, rooted in classical sampling theory and anti-aliasing. Sect. 2 describes *what* we do, Sect. 3 *how* we do it, and Sect. 5 *why* we do it. Sect. 4 validates our method empirically.

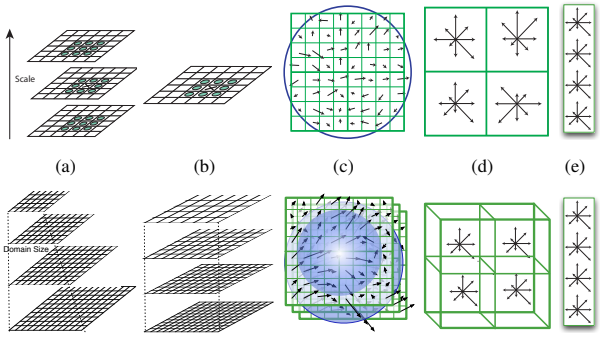

Figure 1. In SIFT (top, recreated according to [24]) isolated scales are selected (a) and the descriptor constructed from the image at the selected scale (b) by computing gradient orientations (c) and pooling them in spatial neighborhoods (d) yielding histograms that are concatenated and normalized to form the descriptor (e). In DSP-SIFT (bottom), pooling occurs across different domain sizes (a): Patches of different sizes are re-scaled (b), gradient orientation computed (c) and pooled across locations *and* scales (d), and concatenated yielding a descriptor (e) of the same dimension of ordinary SIFT.

1.1. Related work

A single, un-normalized cell of the "scale-invariant feature transform" SIFT [24] and its variants [8, 2, 7] can be written compactly as a formula [41, 10]:

$$h_{\text{SIFT}}(\theta|I,\hat{\sigma})[x] = \int \mathcal{N}_\epsilon\left(\theta - \angle\nabla I(y)\right)\mathcal{N}_{\hat{\sigma}}(y-x)d\mu(y) \tag{1}$$

where I is the image restricted to a square domain, centered at a location $x \in \Lambda(\hat{\sigma})$ with size $\hat{\sigma}$ in the lattice Λ determined by the response to a difference-of-Gaussian (DoG) operator across all locations and scales (SIFT *detector*). Here $d\mu(y) \doteq \|\nabla I(y)\| dy$, θ is the independent variable, ranging from 0 to 2π, corresponding to an orientation histogram bin of size ϵ, and $\hat{\sigma}$ is the *spatial pooling scale*. The kernel $\mathcal{N}_\epsilon$ is bilinear of size ϵ and $\mathcal{N}_{\hat{\sigma}}$ separable-bilinear of size $\hat{\sigma}$ [41], although they could be replaced by a Gaussian with standard deviation $\hat{\sigma}$ and an *angular Gaussian* with dispersion parameter ϵ. The SIFT descriptor is the concatenation of 16 cells (1) computed at locations $x \in \{x_1, x_2, \ldots, x_{16}\}$ on a 4×4 lattice Λ, and normalized.

The spatial pooling scale $\hat{\sigma}$ and the size of the image domain where the SIFT descriptor is computed $\Lambda = \Lambda(\hat{\sigma})$ are *tied* to the photometric characteristics of the image, since $\hat{\sigma}$ is derived from the response of a DoG operator on the (single) image.[1] Such a response depends on the *reflectance* properties of the scene and *optical characteristics* and *resolution* of the sensor, neither of which is related to the size and shape of co-visible (corresponding) regions. Instead, how large a portion of a scene is visible in each corresponding image(s) depends on the *shape* of the scene, the *pose* of the two cameras, and the resulting visibility (*occlusion*) relations. Therefore, we propose to *untie* the size of the domain where the descriptor is computed ("scale") from photometric characteristics of the image, departing from the teachings of scale selection (Fig. 8). Instead, we use basic principles of classical sampling theory and *anti-aliasing* to achieve robustness to domain size changes due to occlusions (Sect. 5).

Pooling is commonly understood as *the combination of responses of feature detectors/descriptors at nearby locations, aimed at transforming the joint feature representation into a more usable one that preserves important information* (intrinsic variability) *while discarding irrelevant detail* (nuisance variability) [4, 19]. However, precisely how pooling trades off these two conflicting aims is unclear and mostly addressed empirically in end-to-end comparisons with numerous confounding factors. Exceptions include [4], where intrinsic and nuisance variability are combined and abstracted into the variance and distance between the means of scalar random variables in a binary classification task. For more general settings, the goals of reducing nuisance variability while preserving intrinsic variability is elusive as a *single image* does not afford the ability to separate the two [10].

An alternate interpretation of pooling as anti-aliasing [36] clearly highlights its effects on intrinsic and nuisance variability: Because one cannot know what portion of an object or scene will be visible in a test image, a scale-space ("semi-orbit") of domain sizes ("receptive fields") should be marginalized or searched over ("max-out"). Neither can be computed in closed-form, so the semi-orbit has to be sampled. To reduce complexity, only a small number of samples should be retained, resulting in undersampling and aliasing phenomena that can be mitigated by anti-aliasing, with quantifiable effects on the sensitivity to nuisance variability. For the case of histogram-based descriptors, anti-aliasing planar translations consists of spatial pooling, routinely performed by most descriptors. Anti-aliasing visibility results in *domain-size aggregation*, which no current descriptor practices. This interpretation also offers a way to quantify the effects of pooling on discriminative (reconstruction) power directly, using classical results from sampling theory, rather than indirectly through an end-to-end classification experiment that may contain other confounding factors.

Domain-size pooling can be applied to a number of different descriptors or convolutional architectures. We illustrate its effects on the most popular, SIFT. However, we point out that proper marginalization requires the availability of multiple images of the same scene, and therefore cannot be performed in a single image. While most local image descriptors are computed from a single image, exceptions include [22, 10]. Of course, multiple images can be "hallucinated" from one, but the resulting pooling operation can only achieve invariance to modeled transformations.

In neural network architectures, there is evidence that abstracting spatial pooling hierarchically, *i.e.,* aggregating nearby responses in feature maps, is beneficial [4]. This process could be extended by aggregating across different neighborhood sizes in feature space. To the best of our knowledge, the only architecture that performs some kind of pooling across scales is [31], although the justification provided in [5] only concerns translation within each scale. The same goes for [6], where pooling (low-pass filtering) is only performed within each scale, and not across scales. Other works learn the regions for spatial pooling, for instance [19, 34], but still restrict pooling to within-scale, similar to [20], rather than across scales as we advocate.

We distinguish *multi-scale methods* that concatenate descriptors computed *independently at each scale*, from *cross-scale pooling*, where statistics of the image at different scales are combined directly in the descriptor. Examples of the former include [18], where ordinary SIFT descriptors computed on domains of different size are assumed to belong to a linear subspace, and [34], where Fisher vectors are computed for multiple sizes and aspect ratios and spatial pooling occurs within each level. Also bag-of-word (BoW) methods [35], as mid-level representations, aggregate different low level descriptors by counting their frequency after discretization. Typically, vector quantization

[1] Approaches based on "dense SIFT" forgo the detector and instead compute descriptors on a regular sampling of locations and scales (Fig. 9). However, no existing dense SIFT method performs domain-size pooling.

or other clustering technique is used, each descriptor is associated with a cluster center ("word"), and the frequency of each word is recorded in lieu of the descriptors themselves. This can be done for domain size, by computing different descriptors at the same location, for different domain sizes, and then counting frequencies relative to a dictionary learned from a large training dataset (Sect. 4.4).

Aggregation across time, which may include changes of domain size, is advocated in [17], but in the absence of formulas it is unclear how this approach relates to our work. In [13], weights are shared across scales, which is not equivalent to pooling, but still establishes some dependencies across scales. MTD [21] appears to be the first instance of pooling across scales, although the aggregation is global in scale-space with consequent loss of discriminative power. Most recently, [16] advocates the same but in practice space-pooled VLAD descriptors obtained at different scales are simply concatenated. Also [3] can be thought of as a form of pooling, but the resulting descriptor only captures the mean of the resulting distribution. In addition, [39] exploits the possibility of estimating the proper scales for nearby features via scale propagation but still no pooling is performed across scales. Additional details in related prior work are discussed in [11].

2. Domain-Size Pooling

If SIFT is written as (1), then DSP-SIFT is given by

$$h_{\mathrm{DSP}}(\theta|I)[x] = \int h_{\mathrm{SIFT}}(\theta|I,\sigma)[x]\mathcal{E}_s(\sigma)d\sigma \quad x \in \Lambda \quad (2)$$

where $s > 0$ is the size-pooling scale and $\mathcal{E}$ is an exponential or other unilateral density function. The process is visualized in Fig. 1. Unlike SIFT, that is computed on a scale-selected lattice $\Lambda(\hat{\sigma})$, DSP-SIFT is computed on a *regularly sampled* lattice Λ. Computed on a different lattice, the above can be considered as a recipe for DSP-HOG [8]. Computed on a tree, it can be used to extend deformable-parts models (DPM) [14] to DSP-DPM. Replacing h_{SIFT} with other histogram-based descriptor "X" (for instance, SURF [2]), the above yields DSP-X. Applied to a hidden layer of a convolutional network, it yields a DSP-CNN, or DSP-Deep-Fisher-Network [33]. The details of the implementation are in Sect. 3.

While the implementation of DSP is straightforward, its justification is less so. We report the summary in Sect. 5 and the detailed derivation in [11], that provides a theoretical justification and conditions under which the resulting descriptors are valid. In Sect. 4 we compare DSP-SIFT to alternate approaches. Motivated by the experiments of [28, 29] that compare local descriptors, we choose SIFT as a paragon and compare it to DSP-SIFT on the standard benchmark [28]. Motivated by [15] that compares SIFT to both supervised and unsupervised CNNs trained on Imagenet and Flickr respectively on the same benchmark [28], we submit DSP-SIFT to the same protocol. We also run the test on the new synthetic dataset introduced by [15], that yields the same qualitative assessment.

Clearly, domain-size pooling of under-sampled semi-orbits cannot outperform fine sampling, so if we were to retain all the scale samples instead of aggregating them, performance would further improve. However, computing and matching a large collection of SIFT descriptors across different scales would incur significantly increased computational and storage costs. To contain the latter, [18] assumes that descriptors at different scales populate a linear subspace and fit a high-dimensional hyperplane. The resulting Scale-less SIFT (SLS) outperforms ordinary SIFT as shown in Fig. 7. However, the linear subspace assumption breaks when considering large scale changes, so SLS is outperformed by DSP-SIFT despite the considerable difference in (memory and time) complexity.

3. Implementation and Parameters

Following other evaluation protocols, we use *Maximally Stable Extremal Regions* (MSER) [25] to detect candidate regions, affine-normalize, re-scale and align them to the dominant orientation. For a detected scale $\hat{\sigma}$, DSP-SIFT samples $N_{\hat{\sigma}}$ scales within a neighborhood $(\lambda_1\hat{\sigma}, \lambda_2\hat{\sigma})$ around it. For each scale-sampled patch, a single-scale unnormalized SIFT descriptor (1) is computed on the SIFT scale-space octave corresponding[2] to the sampled scale σ. By choosing $\mathcal{E}_s$ to be a uniform density, these raw histograms of gradient orientations at different scales are accumulated and normalized[3] (2). Fig. 2(a) shows that mean average precision (defined in Sect. 4.2) for different domain size pooling ranges. Improvements are observed as soon as more than one scale is used, with diminishing return: Performance decreases with domain size pooling radius exceeding $\hat{\sigma}/2$. Fig. 2(b) shows the effect of the number of size samples used to construct DSP-SIFT. Although the more samples the merrier, three size samples are sufficient to outperform ordinary SIFT, and improvement beyond 10 samples is minimal. Additional samples do not further increase the mean average precision, but incur more computational cost. In the evaluation in Sect. 4, we use $\lambda_1 = 1/6, \lambda_2 = 4/3$ and $N_{\hat{\sigma}} = 15$. These parameters are empirically selected on the Oxford dataset [27, 28].

[2] This is an updated version of the protocol described in [15], as discussed in detail in Appendix D of [11].

[3] We follow the practice of SIFT [24] to normalize, clamp and renormalize the histograms, with the clamping threshold set to 0.067 empirically.

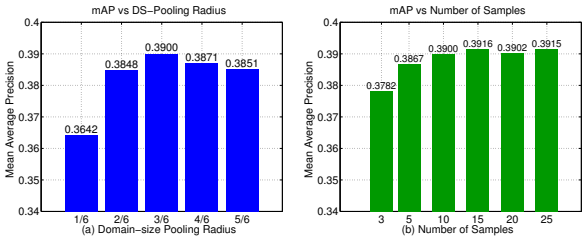

Figure 2. *Mean Average Precision for different parameters.* (a) shows that mAP changes with the radius s of DS pooling. The best mAP is achieved at $\hat{s} = \hat{\sigma}/2$; (b) shows mAP as a function of the number of samples used within the best range $(\hat{\sigma} - \hat{s}, \hat{\sigma} + \hat{s})$.

4. Validation

As a baseline, the RAW-PATCH descriptor (named following [15]) is the unit-norm grayscale intensity of the affine-rectified and resized patch of a fixed size (91×91).

The standard SIFT, which is widely accepted as a paragon [27, 29], is computed using the VLFeat library [41]. Both SIFT and DSP-SIFT are computed on the SIFT scale-space corresponding to the detected scales. Instead of mapping all patches to an arbitrarily user-defined size, we use the area of each selected and rectified MSER region to determine the octave level in the scale-space where SIFT (as well as DSP-SIFT) is to be computed.

Scale-less SIFT (SLS) is computed using the source code provided by the authors [18]: For each selected and rectified patch, the standard SIFT descriptors are computed at 20 scales from a scale range of $(0.5, 12)$, and the standard PCA subspace dimension is set to 8, yielding a final descriptor of dimension 8256 after a subspace-to-vector mapping.

To compare DSP-SIFT to a convolutional neural network, we use the top-performer in [15], an unsupervised model pre-trained on 16000 natural images undergoing 150 transformations each (total 2.4M). The responses at the intermediate layers 3 (CNN-L3) and 4 (CNN-L4) are used for comparison, following [15]. Since the network requires input patches of fixed size, we tested and report the results on both 69×69 (PS69) and 91×91 (PS91) as in [15].

Although no direct comparison with Multiscale Template Descriptors (MTD) [21] is performed, SLS can be considered as dominating it since it uses all scales without collapsing them into a single histogram. The derivation in Sect. 5 suggests, and empirical evidence in Fig. 2(a) confirms, that aggregating the histogram across *all* scales significantly reduces discriminative power. Sect. 4.4 compares DSP-SIFT to a BoW which pools SIFT descriptors computed at different sizes at the same location.

4.1. Datasets

The Oxford dataset [27, 28] comprises 40 pairs of images of mostly planar scenes seen under different pose, distance, blurring, compression and lighting. They are organized into 8 categories undergoing increasing magnitude of transformations. While routinely used to evaluate descriptors, this dataset has limitations in terms of size and restriction to mostly planar scenes, modest scale changes, and no occlusions. Fischer *et al.* [15] recently introduced a dataset of 400 pairs of images with more extreme transformations including zooming, blurring, lighting change, rotation, perspective and nonlinear transformations.

4.2. Metrics

Following [27], we use precision-recall (PR) curves to evaluate descriptors. A *match* between two descriptors is called if their Euclidean distance is less than a threshold τ_d. It is then labeled as a *true positive* if the area of intersection over union (IoU) of their corresponding MSER-detected regions is larger than 50%. Both datasets provide ground truth mapping between images, so the overlapping is computed by warping the first MSER region into the second image and then computing the overlap with the second MSER region. *Recall* is the fraction of true positives over the total number of correspondences. *Precision* is the percentage of true matches within the total number of matches. By varying the distance threshold τ_d, a PR curve can be generated and *average precision* (AP, *a.k.a area under the curve*, AUC) can be estimated. The average of APs provides the *mean average precision* (mAP) scores used for comparison.

4.3. Comparison

Fig. 3 shows the behavior of each descriptor for varying degree of severity of each transformation. DSP-SIFT consistently outperforms other methods when there are large scale changes (zoom). It is also more robust to other transformations such as blur, lighting and compression in the Oxford dataset [28], and to nonlinear, perspective, lighting, blur and rotation in Fischer's [15]. DSP-SIFT is not at the top of the list of all compared descriptors in viewpoint change cases, although "viewpoint" is a misnomer as MSER-based rectification accounts for most of the viewpoint variability, and the residual variability is mostly due to interpolation and rectification artifacts. The fact that DSP-SIFT outperforms CNN in nearly all cases in Fischer's dataset is surprising, considering that the neural network is trained by augmenting the dataset using similar types of transformations.

Fig. 4 shows head-to-head comparisons between these methods, in the same format of [15]. DSP-SIFT outperforms SIFT by 43.09% and 18.53% on Oxford and Fischer respectively. Only on two out of 400 pairs of images in Fischer dataset does domain-size pooling negatively affect the performance of SIFT, but the decrease is rather small. DSP-SIFT improves SIFT on every pair of images in the Oxford dataset. The improvement of DSP-SIFT comes without

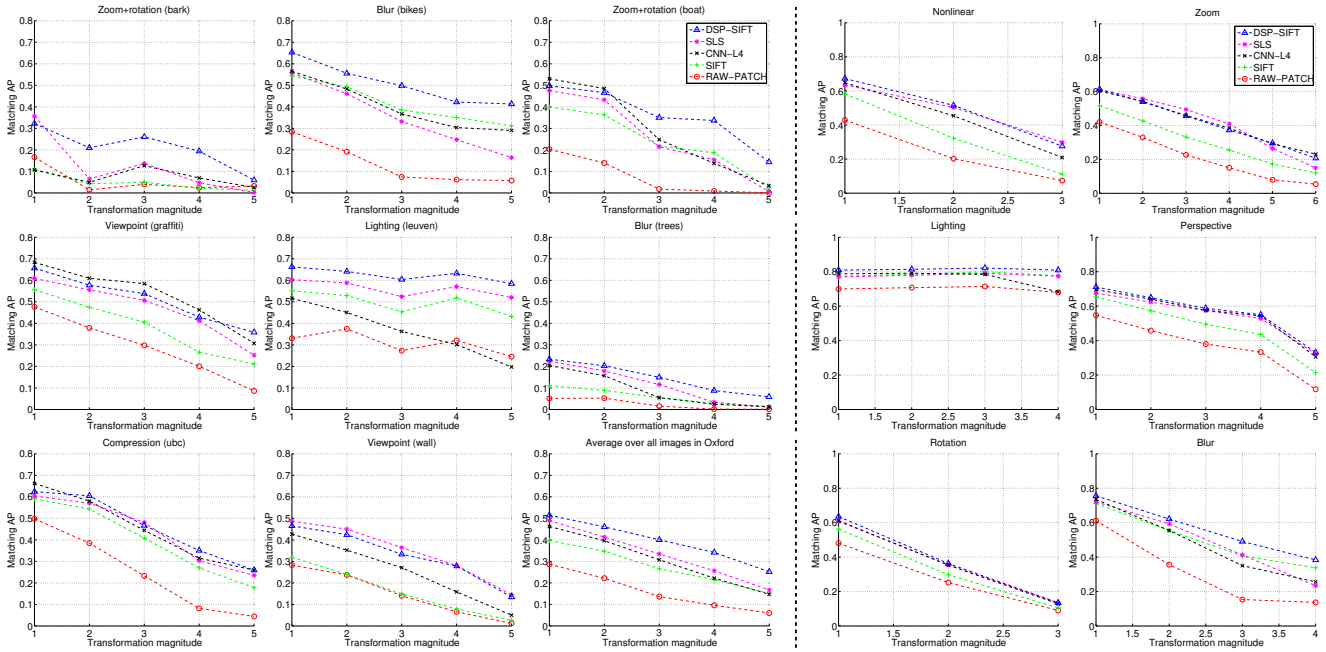

Figure 3. *Average Precision for different magnitude of transformations.* The left 9 panels show (AP) for increasing magnitude of the 8 transformations in the Oxford dataset [27]. The mean AP over all pairs with corresponding amount of transformation are shown in the middle of the third row. The right 6 panels show the same for Fischer's dataset [15].

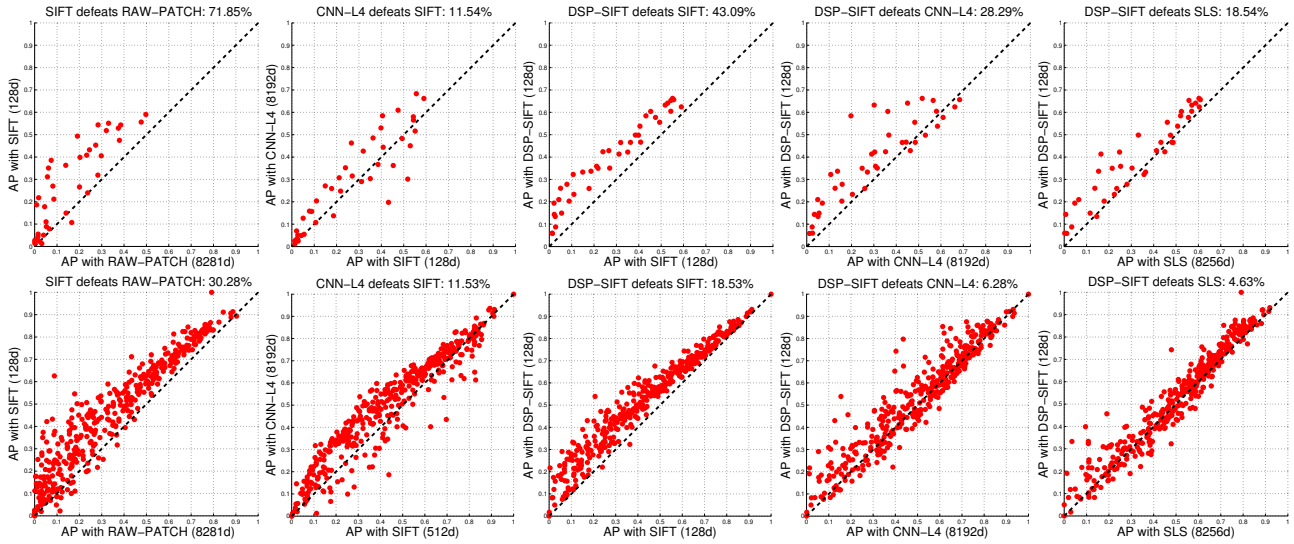

Figure 4. *Head-to-head comparisons.* Similarly to [15], each point represents one pair of images in the Oxford (top) and Fischer (bottom) datasets. The coordinates indicate average precision for each of the two methods under comparison. SIFT is superior to RAW-PATCH, but is outperformed by DSP-SIFT and CNN-L4. The right two columns show that DSP-SIFT is better than SLS and CNN-L4 despite the difference in dimensions (shown in the axes). The relative performance improvement of the winner is shown in the title of each panel.

increase in dimension. In comparison, CNN-L4 achieves 11.54% and 11.53% improvements over SIFT by increasing dimension 64-fold. On both datasets, DSP-SIFT also consistently outperforms CNN-L4 and SLS despite its lower dimension.

4.4. Comparison with Bag-of-Words

To compare DSP-SIFT to BoW we computed SIFT at 15 scales on concentric regions with dictionary sizes ranging from 512 to 2048, trained on over 100K SIFT descriptors computed on samples from ILSVRC-2013 [9]. To make the

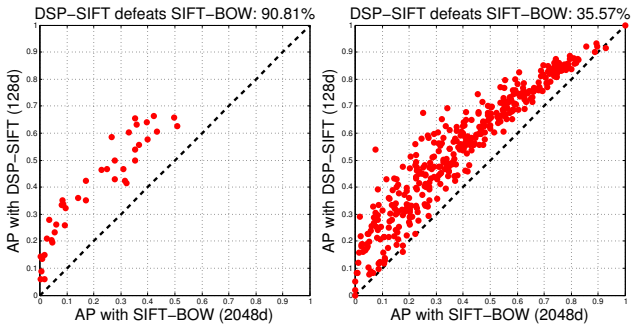

Figure 5. *DSP-SIFT vs. SIFT-BOW.* Similarly to Fig. 4, each point represents one pair of images in the Oxford (left) and Fischer (right) datasets. The coordinates indicate average precision for each of the two methods under comparison. The relative performance improvement of the winner is shown in the title of each panel. DSP-SIFT outperforms SIFT-BOW by a wide margin on both datasets.

comparison fair, the same 15 scales are used to compute DSP-SIFT. By doing so, the only difference between these two methods is *how* to pool across scales rather than *what* or *where* to pool. In SIFT-BOW, pooling is performed by encoding SIFTs from nearby scales using the quantized visual dictionary, while DSP-SIFT combines the histograms of gradient orientations across scales directly. To compute similarity between SIFT-BOWs, we tested both the intersection kernel and ℓ_1 norm, and achieved a best performance with the latter at 20.62% mAP on Oxford and 39.63% on Fischer. Fig. 5 shows the direct comparison between DSP-SIFT and SIFT-BOW with the former being a clear winner.

4.5. Complexity and Performance Tradeoff

Fig. 7 shows the complexity (descriptor dimension) and performance (mAP) tradeoff. Table 1 summarizes the results. In Fig. 7, an "ideal" descriptor would achieve mAP $= 1$ by using the smallest possible number of bits and land at the top-left corner of the graph. DSP-SIFT has the same lowest complexity as SIFT and is the best in mAP among all the descriptors. Looking horizontally in the graph, DSP-SIFT outperforms all the other methods at a fraction of complexity. SLS achieves the second best performance but at the cost of a 64-fold increase in dimension. In general, the performance of CNN descriptors is worse than DSP-SIFT but, interestingly, their mAPs do not change significantly if the network responses are computed on a resampled patch of size 69×69 to obtain lower dimensional descriptors.

4.6. Comparison with SIFT on Larger Domain Sizes

Descriptors computed on larger domain sizes are usually more discriminative, up to the point where the domain straddles occluding boundaries (Fig. 10). When using a detector, the size of the domain is usually chosen to be a factor of the detected scale, which affects performance in a way that depends on the dataset and the incidence of occlusions. In our experiments, this parameter (dilation factor) is set at 3, following [27], and we note that DSP-SIFT is less sensitive than ordinary SIFT to this parameter. Since DSP-SIFT aggregates domains of various sizes (smaller and larger) around the nominal size, it is important to ascertain whether the improvement in DSP-SIFT comes from size pooling, or simply from including larger domains. To this end, we compare DSP-SIFT by pooling domain sizes from $1/6$th through $4/3$rd of the scale determined by the detector, to a single-size descriptor computed at the largest size (SIFT-L). This establishes that the increase in performance of DSP-SIFT over ordinary SIFT comes from pooling across domain sizes, not just by picking larger domain sizes. In the example in Fig. 6, the largest domain size yields an even worse performance than the detected scale (Fig. 6(b)). In a more complex scene where the test images exhibit occlusion, this will be even more pronounced as there is a tradeoff between discriminative power (calling for a larger size) and the probability of straddling an occlusion (calling for a smaller size).

Method	Dim.	mAP	
		Oxford	Fischer
SIFT	**128**	.2750	.4532
DSP-SIFT	**128**	**.3936**	**.5372**
CNN-L4-PS69	512	.3059	.4779
SIFT-BOW	2048	.2062	.3963
CNN-L3-PS69	4096	.3164	.4858
CNN-L4-PS91	8192	.3068	.5055
SLS	8256	.3320	.5135
RAW-PATCH	8281	.1600	.3479
CNN-L3-PS91	9216	.3056	.4899

Table 1. *Summary of complexity (dimension) and performance (mAP) for all descriptors* sorted in order of increasing complexity. The lowest complexities and the best performances are highlighted in bold. We also report mAP for CNN descriptors computed on 69×69 patches as in [15]. The fourth row shows comparison with a bag-of-words of SIFT descriptors computed at the same location but different domain sizes, described in detail in Sect. 4.4.

5. Derivation

In this section we describe the trace of the derivation of DSP-SIFT, which is reported in the appendix of [11]. Crucial to the derivation is the interpretation of a descriptor as a likelihood function [36].

1. The likelihood function of the scene given images is a minimal sufficient statistic of the latter for the purpose of answering questions on the former [1]. Invariance to nuisance transformations induced by (semi-)group actions on the data can be achieved by representing orbits, which are maximal invariants [32]. The planar translation-scale group

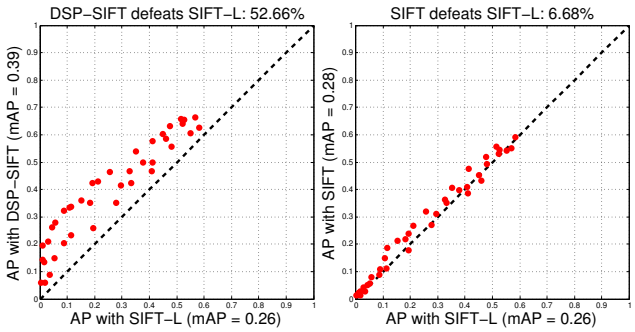

Figure 6. *DSP-SIFT vs. SIFT-L.* Similarly to Fig. 4, each point represents one pair of images in the Oxford dataset. The coordinates indicate average precision for each of the two methods under comparison. The relative performance improvement of the winner is shown in the title of each panel. 6(a) shows that DSP-SIFT outperforms SIFT computed at the largest domain size. This shows that the improvement of DSP-SIFT comes from the pooling across domain sizes rather than choosing a larger domain size. 6(b) shows that choosing a larger domain size actually decreases the performance on the Oxford dataset.

can be used as a crude first-order approximation of the action of the translation group in space (viewpoint changes) including scale change-inducing translations along the optical axis. This draconian assumption is implicit in most single-view descriptors.

2. Comparing (semi-)orbits entails a continuous search (non-convex optimization) that has to be discretized for implementation purposes. The orbits can be sampled adaptively, through the use of a co-variant detector and the associated invariant descriptor, or regularly - as customary in classical sampling theory.

3. In adaptive sampling, the *detector* should exhibit high sensitivity to nuisance transformations (*e.g.,*small changes in scale should cause a large change in the response to the detector, thus providing accurate scale localization) and the *descriptor* should exhibit small sensitivity (so small errors in scale localization cause a small change in the descriptor). Unfortunately, for the case of SIFT (DoG detector and gradient orientation histogram descriptor), the converse is true.

4. Because correspondence entails search over samples of each orbit, time complexity increases with the number of samples. Undersampling introduces structural artifacts, or "aliases," corresponding to topological changes in the response of the detector. These can be reduced by "anti-aliasing," an averaging operation. For the case of (approximations of) the likelihood function, such as SIFT and its variants, anti-aliasing corresponds to *pooling*. While spatial pooling is common practice, and reduces sensitivity to translation parallel to the image plane, scale pooling – which would provide insensitivity to translation orthogonal to the image plane – and domain-size pooling – which

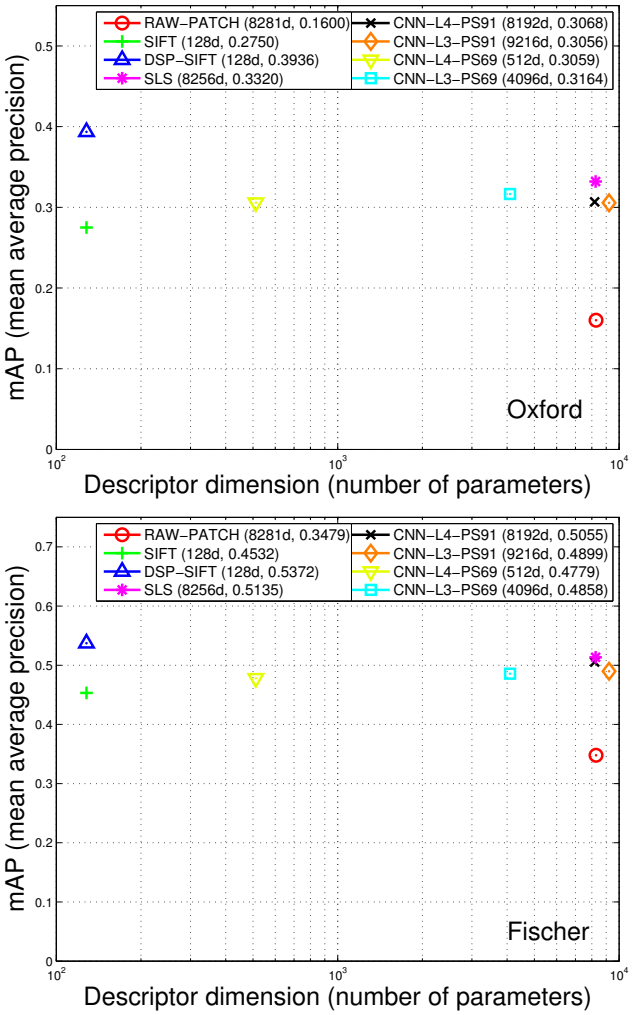

Figure 7. *Complexity-Performance Tradeoff.* The abscissa is the descriptor dimension shown in log-scale, the ordinate shows the mean average precision.

would provide insensitivity to small changes of visibility, are not. This motivates the introduction of DSP-SIFT, and the rich theory on sampling and anti-aliasing could provide guidelines on what and how to pool, as well as bounds on the loss of discriminative power coming from undersampling and anti-aliasing operations.

6. Discussion

Image matching under changes of viewpoint, illumination and partial occlusions is framed as a hypothesis testing problem, which results in a non-convex optimization over continuous nuisance parameters. The need for efficient test-time performance has spawned an industry of engineered descriptors, which are computed locally so the effects of occlusions can be reduced to a binary classification (co-visible, or not). The best known is SIFT, which has been

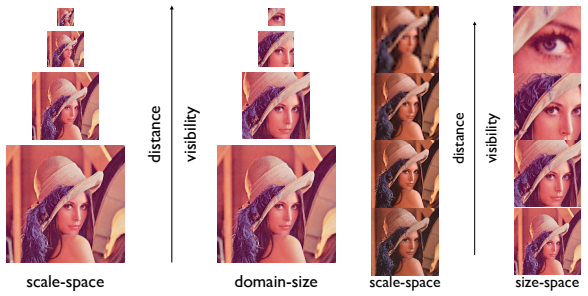

Figure 8. *Scale-space vs. Size-space.* Scale-space refers to a continuum of images obtained by smoothing and downsampling a base image. It is relevant to searching for correspondence when the distance to the scene changes. Size-space refers to a scale-space obtained by maintaining the same scale of the base image, but considering subsets of it of variable size. It is relevant to searching for correspondence in the presence of occlusions, so the size (and shape) of co-visible domains are not known.

shown to work well in a number of independent empirical assessments [27, 29], that however come with little analysis on *why* it works, or indications on how to improve it. We have made a step in that direction, by showing that SIFT can be derived from sampling considerations, where spatial binning and pooling are the result of anti-aliasing operations. However, SIFT and its variants only perform such operations for planar translations, whereas our interpretation calls for anti-aliasing domain-size as well. Doing so can be accomplished in few lines of code and yields significant performance improvements. Such improvements even place the resulting DSP-SIFT descriptor above a convolutional neural network (CNN), that had been recently reported as a top performer in the Oxford image matching benchmark [15]. Of course, we are not advocating replacing large neural networks with local descriptors. Indeed, there are interesting relations between DSP-SIFT and convolutional architectures, explored in [36, 37].

Domain-size pooling, and regular sampling of scale "unhinged" from the spatial frequencies of the signal is divorced from scale selection principles, rooted in scale-space theory, wavelets and harmonic analysis. There, the goal is to reconstruct a signal, with the focus on photometric nuisances (additive noise). In our case, the size of the domain where images correspond depends on the three-dimensional shape of the underlying scene, and visibility (occlusion) relations, and has little to do with the spatial frequencies or "appearance" of the scene. Thus, we do away with the linking of domain size and spatial frequency ("uncertainty principle", Fig. 9).

DSP can be easily extended to other descriptors, such as HOG, SURF, CHOG, including those supported on structured domains such as DPMs [14], and to network architectures such as convolutional neural networks and scattering

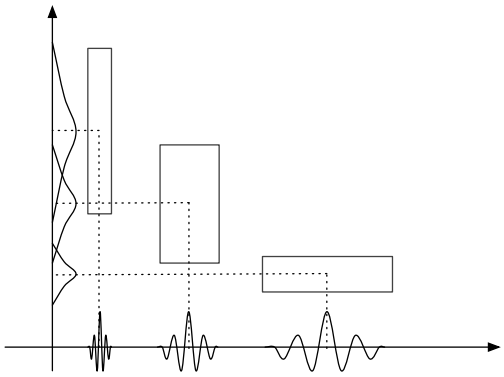

Figure 9. The "uncertainty principle" links the size of the domain of a filter (ordinate) to its spatial frequency (abscissa): As the data is analyzed for the purpose of compression, regions with high spatial frequency must be modeled at small scale, while regions with smaller spatial frequency can be encoded at large scale. When the task is correspondence, however, the size of the co-visible domain is independent of the spatial frequency of the scene within. While approaches using "dense SIFT" forgo the detector and compute descriptors at regularly sampled locations and scales, they perform spatial pooling by virtue of the descriptor, but fail to perform pooling across scales, as we propose.

networks [6], opening the door to multiple extensions of the present work. In addition, a number of interesting open theoretical questions can now be addressed using the tools of classical sampling theory, given the novel interpretation of SIFT and its variants introduced in this paper.

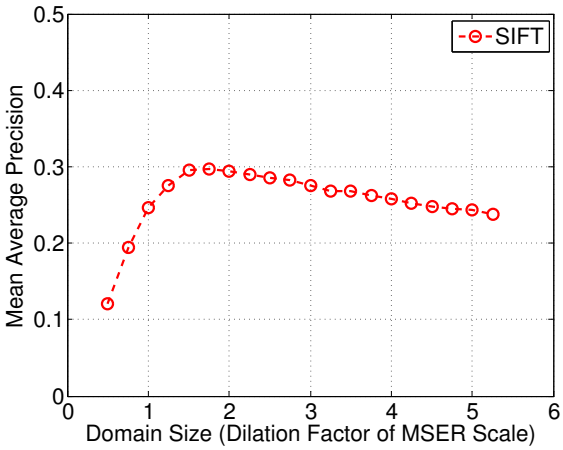

Figure 10. The discriminative power of a descriptor (*e.g.,* mAP of SIFT) increases with the size of the domain, but so does the probability of straddling an occlusion and the approximation error of the imaging model implicit in the detector/descriptor. This effect, which also depends on the detected scale, is most pronounced when occlusions are present, but is present even on the Oxford dataset, shown above.

Acknowledgments

We are thankful to Nikolaos Karianakis for conducting the comparison with various forms of CNNs, and to Philipp Fischer, Alexey Dosovitskiy and Thomas Brox for sharing their dataset, evaluation protocol and comments. Research sponsored in part by NGA HM02101310004, leveraging on theoretical work conducted under the aegis of ONR N000141110863, NSF RI-1422669, ARO W911NF-11-1-0391, and FA8650-11-1-7156.

References

[1] R. R. Bahadur. Sufficiency and statistical decision functions. *Annals of Mathematical Statistics*, 25(3):423–462, 1954. 6

[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417, 2006. 1, 3

[3] A. Berg and J. Malik. Geometric blur for template matching. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001. 3

[4] Y.-L. Boureau, J. Ponce, and Y. LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proc. of the International Conference on Machine Learning (ICML)*, pages 111–118, 2010. 2

[5] J. V. Bouvrie, L. Rosasco, and T. Poggio. On invariance in hierarchical models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 162–170, 2009. 2

[6] J. Bruna and S. Mallat. Classification with scattering operators. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 2, 8

[7] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod. Chog: Compressed histogram of gradients a low bit-rate feature descriptor. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2504–2511, 2009. 1

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 1, 3

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 5

[10] J. Dong, N. Karianakis, D. Davis, J. Hernandez, J. Balzer, and S. Soatto. Multi-view feature engineering and learning. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015; ArXiv preprint:1311.6048*, 2013. 1, 2

[11] J. Dong and S. Soatto. Domain-size pooling in local descriptors: DSP-SIFT (extended version of this paper), *ArXiv preprint:1412.8556*, 2014. 3, 6

[12] A. Dosovitskiy, J. T. Springenberg, and T. Brox. Unsupervised feature learning by augmenting single images. *ArXiv preprint:1312.5242*, 2013. 1

[13] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Scene parsing with multiscale feature learning, purity trees, and optimal covers. *ArXiv preprint:1202.2160*, 2012. 3

[14] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 3, 8

[15] P. Fischer, A. Dosovitskiy, and T. Brox. Descriptor matching with convolutional neural networks: A comparison to SIFT. *ArXiv preprint:1405.5769*, 2014. 1, 3, 4, 5, 6, 8

[16] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. *ArXiv preprint:1403.1840*, 2014. 3

[17] P. Hamel, S. Lemieux, Y. Bengio, and D. Eck. Temporal pooling and multiscale learning for automatic annotation and ranking of music audio. In *The International Society for Music Information Retrieval*, pages 729–734, 2011. 3

[18] T. Hassne, V. Mayzels, and L. Zelnik-Manor. On sifts and their scales. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1522–1528, 2012. 2, 3, 4

[19] Y. Jia, C. Huang, and T. Darrell. Beyond spatial pyramids: Receptive field learning for pooled image features. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3370–3377, 2012. 2

[20] Y. LeCun. Learning invariant feature hierarchies. In *European Conference on Computer Vision*, pages 496–505, 2012. 2

[21] T. Lee and S. Soatto. Learning and matching multiscale template descriptors for real-time detection, localization and tracking. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1457–1464, 2011. 3, 4

[22] T. Lee and S. Soatto. Video-based descriptors for object recognition. *Image and Vision Computing*, Vol. 29(10), pages 639-652, 2011. 2

[23] T. Lindeberg. Principles for automatic scale selection. *Technical report, KTH, Computational Vision and Active Perception laboratory*, 1998. 1

[24] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 1, 3

[25] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. of the British Mach. Vis. Conf.*, 2002. 3

[26] R. Memisevic. Learning to relate images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), pages 1829–1846, 2013. 1

[27] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. 1, 3, 4, 5, 6, 8

[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. *International Journal of Computer Vision*, 1(60):63–86, 2004. 1, 3, 4

[29] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3d objects. *International Journal of Computer Vision*, 73(3):263–284, 2007. 1, 3, 4, 8

[30] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. 1

[31] T. Serre, A. Oliva, and T. Poggio. A feedforward architecture accounts for rapid categorization. In *Proceedings of the National Academy of Sciences*, 104(15):6424–6429, 2007. 2

[32] J. Shao. *Mathematical Statistics*. Springer Verlag, 1998. 6

[33] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep fisher networks for large-scale image classification. In *Advances in Neural Information Processing Systems (NIPS)*, pages 163–171, 2013. 3

[34] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(4), 2014. 2

[35] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1470–1477, 2003. 2

[36] S. Soatto and A. Chiuso. Visual scene representations: Sufficiency, minimality, invariance and deep approximation. *ArXiv preprint:1411.7676*, 2014. 2, 6, 8

[37] S. Soatto, J. Dong, and K. Karianakis. Visual scene representations: Scaling and occlusion in convolutional architectures. *ArXiv preprint:1412.6607*, 2014. 8

[38] J. Susskind, R. Memisevic, G. E. Hinton, and M. Pollefeys. Modeling the joint density of two images under a variety of transformations. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2793–2800, 2011. 1

[39] M. Tau and T. Hassner. Dense correspondences across scenes and scales. *ArXiv preprint:1406.6323*, 2014. 3

[40] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *European Conference on Computer Vision*, pages 140–153. 2010. 1

[41] A. Vedaldi and B. Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the International Conference on Multimedia*, pages 1469–1472. ACM, 2010. 1, 2, 4

[42] S. A. Winder and M. Brown. Learning local image descriptors. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007. 1