SIFTing through Scales

Tal Hassner, Shay Filosof, Viki Mayzels, and Lihi Zelnik-Manor

Abstract—Scale invariant feature detectors often find stable scales in only a few image pixels. Consequently, methods for feature matching typically choose one of two extreme options: matching a sparse set of scale invariant features, or dense matching using arbitrary scales. In this paper, we turn our attention to the overwhelming majority of pixels, those where stable scales are not found by standard techniques. We ask, is scale-selection necessary for these pixels, when dense, scale-invariant matching is required and if so, how can it be achieved? We make the following contributions: (i) We show that features computed over different scales, even in low-contrast areas, can be different and selecting a single scale, arbitrarily or otherwise, may lead to poor matches when the images have different scales. (ii) We show that representing each pixel as a set of SIFTs, extracted at multiple scales, allows for far better matches than single-scale descriptors, but at a computational price. Finally, (iii) we demonstrate that each such set may be accurately represented by a low-dimensional, linear subspace. A subspace-to-point mapping may further be used to produce a novel descriptor representation, the Scale-Less SIFT (SLS), as an alternative to single-scale descriptors. These claims are verified by quantitative and qualitative tests, demonstrating significant improvements over existing methods. A preliminary version of this work appeared in [1].

Index Terms—I.2.10 Vision and Scene Understanding, I.2.10.g Representations, data structures, and transforms

1 Introduction

Over the past decade and a half, scale invariant feature detectors, such as the Harris-Laplace [2] and robust descriptors such as the SIFT [3], have played pivotal roles in maturing Computer Vision systems. The key idea is that at each interest point, one (or few) scales are selected based on a scale covariant function (e.g., the Laplacian of Gaussians). Presumably, local extrema of this function occur at the same scales for the same feature in different images allowing the features to be matched across images in different scales [4]. A typical image, however, often has relatively few pixels for which such scales may be reliably selected. Consequently, matching of scale invariant features has so far been applied mostly to a few pixels in each image.

When dense correspondences are required, traditional methods restrict themselves to using pixels or pixel patches, filtered or otherwise (see, e.g., [5]). Alternatively, feature descriptors may be computed for all the pixels in the image (e.g., [6]). These are designed to be robust to a range of geometric and photometric image transformations. One such example is the Dense-SIFT (DSIFT) descriptor [7] which is extracted at a single scale for all the pixels in the image. Establishing correspondences between two images is then performed either locally or by using global optimization schemes such as SIFT flow [8], [9]. Such methods, however, all implicitly assume that features in the two images share the same, or sufficiently similar, scales. As shown in Fig. 1, when this does not hold, correspondence estimation fails.

Fig. 1. **Dense matches of different objects in different scales. Top:** Source and Target input images. **Bottom:** Source image warped onto Target using the recovered flows: Using DSIFT (bottom left) and our SLS descriptor (bottom right), overlaid on the Target and manually cropped to demonstrate the alignment. DSIFT fails to capture the scale differences and produces an output in the same scale as the input. SLS captures scale changes at each pixel: the output produced by using SLS has the appearance of the Source image in the scale and position of the Target.

In this paper, we focus on those pixels for which a method for selecting well defined scales is not

- *T. Hassner and S. Filosof are with the Department of Mathematics and Computer Science, Open University of Israel. E-mail: hassner@openu.ac.il shayfilosof@gmail.com*
- *V. Mayzels and L. Zelnik-Manor are with the Department of Electrical Engineering, Technion. E-mail: mviki@techunix.technion.ac.il lihi@ee.technion.ac.il*

known. Making up most of the image, these are the pixels for which local image intensities do not vary sufficiently to provide strong extrema in the scale selection function. This work presents the following contributions:

1) We show that even in low contrast areas of the image, where scale-selection is difficult, descriptors may change their values from one scale to the next. Consequently, selecting an arbitrary single scale may lead to false matches when two images have different scales.
2) We propose representing each pixel by a set of SIFT descriptors extracted at multiple scales and matched from one image to the next using set-to-set similarities. The computational cost of matching more descriptors is balanced by a substantial boost in accuracy.
3) We demonstrate that each such set of SIFTs resides on a low-dimensional subspace. We further show that the subspace-to-point mapping of [10], [11], provides a means of representing these subspaces as a novel feature descriptor, the Scale-Less-SIFT (SLS).

These set-based, multiscale SIFT representations are tested on dense correspondence estimation problems with images separated by wide scale differences and changing viewing conditions. They are shown to significantly outperform existing methods both qualitatively and quantitatively.

2 Previous work

Objects and scenes appear in images in different scales. In order to correctly describe features when these scales are unknown, one must consider multiple scales for each feature point. Since the early 1990s, automatic scale selection techniques have been proposed which seek for each feature point a stable, *characteristic* scale. They, therefore, augment earlier scale-space methods by choosing one scale for each feature for the purpose of both reducing the computational burden of higher level visual systems, as well as improving their performance by focusing on more relevant information (See [12] for more on these early approaches).

Lindeberg [13] suggested seeking for each feature its "interesting scales"; that is, scales which reflect a characteristic size of a feature. He proposed selecting these scales by choosing the extrema in the Laplacian of Gaussian (LoG) function computed over the image scales. Pixels of local extrema may additionally be rejected if their LoG value is lower than a predefined threshold. This is applied in order to ensure that unstable, low-contrast points are not selected. An efficient approximation to the LoG function is based on differences of Gaussian (DoG) filters (e.g., [3]). For a given image, three sets of sub-octave, DoG filters are produced. The resulting 3D structure (x, y and $scale$) is then scanned and searched for pixels with higher or lower values than their 26 space-scale neighbors. Coordinate localization is then performed in order to obtain more accurate pixel locations as well as, again, to reject unstable detections located in low contrast areas or near edges.

Scale selection is sometimes performed concurrently with spatial localization. The Harris-Laplace detector [2], for example, uses a scale-adapted Harris corner detector to localize points spatially and LoG filter extrema to localize points in scale. These two steps are performed in an iterative procedure which searches for the joint peaks of these two functions. Here too, points are rejected if they fail to produce responses stronger than a given threshold.

The methods mentioned above, as well as similar techniques, all typically produce a small set of interest points located near corner structures in the image. Mikolajczyk [14] reports that under a scale change factor of 4.4, the percent of pixels for which a scale is detected is as low as 38% for the DoG detector of which in only 10.6% of the cases, the detected scale was correct.

Several existing methods use few invariant features to seed a search for dense matches between different views of a wide-baseline stereo system [15], [16], [17]. As far as we know, however, none of these methods is designed to provide dense correspondences across scale differences. A noteworthy exception is the work of [18] which uses few scale-invariant features to locate an object in an image and then produces dense matches along with accurate segmentations. Their method, however, relies on a global alignment scheme to overcome the main scale differences before dense matching. It is thus unclear how it performs when no such alignment is possible (e.g., several independent scene motions).

In [19] scale invariant descriptors (SID) are proposed without requiring the estimation of image scale. A main advantage of SID is that they are applicable to a broader range of image structures, such as edges, for which scale selection is unreliable. Our experiments here show that SID are less capable of matching across different scenes than the SIFT descriptors underlying our representation. In [20], scale selection is avoided by computing multiscale fractal features, developed for the purpose of texture classification.

Dense SIFT - no scale selection. When dense matching is required, a common approach is to forgo scale estimation; producing, instead, descriptors on a regular grid using constant, typically arbitrarily selected scales. One such example is the efficient DAISY descriptors of [6] or, more related to this work, Dense-SIFT (DSIFT) descriptors [7].

In object recognition tasks, such regular sampling strategies for descriptor generation have been shown to outperform systems utilizing invariant features generated at stable coordinates and scales [21]. This

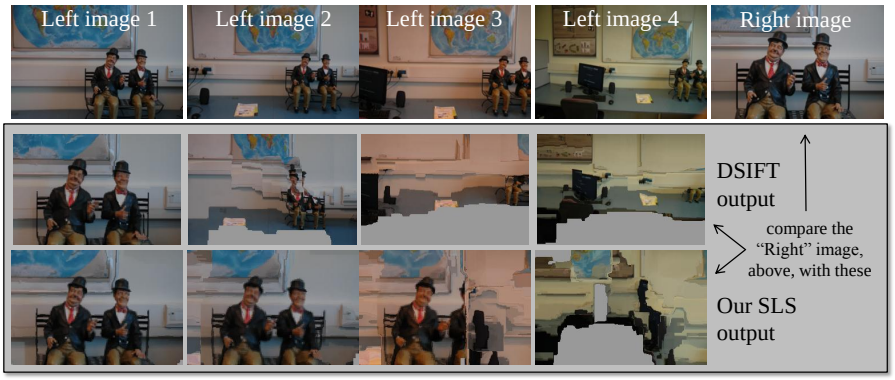


Fig. 2. **Effects of scale differences on DSIFT vs. our own SLS descriptor.** Source images warped onto Target image using correspondences obtained by the SIFT flow algorithm [8], [9] and the DSIFT descriptor, compared against the SLS descriptor (Sec. 3.3). The results in the bottom two rows should appear similar to the top-right image. DSIFT descriptors provide some scale invariance despite a single arbitrary scale selection (left column, middle row). The SLS descriptors provide scale invariance across far greater scale differences (bottom).

may be due to the benefit of having many descriptors with possibly inaccurate scales over having a few descriptors extracted where accurate scales are available.

Existing work on dense matching between two images has thus far largely ignored the issue of scale invariance. The SIFT flow system of [8], [9], for example, produces DSIFT descriptors at each pixel location. These descriptors are then matched between two images, taking advantage of the robustness of the SIFT representation, without attempting to provide additional scale invariance. Matching is performed using a modified optical flow formulation [22]. Although the DSIFT descriptors used by the SIFT flow algorithm provide some scale invariance, this quickly degrades as the scale differences between the two images increase (Fig. 2). An additional related method is the Generalized Patch-Match [23], designed for matching descriptors extracted at each pixel with an emphasis on speed.

The methods described above provide the means for matching descriptors produced on dense regular grids. In the absence of per-pixel scale-invariant descriptors, they are not designed to handle large scale differences. In this paper, we extend these approaches by discussing the utility of multiple SIFT descriptors at each pixel, and their representations.

Dense scale selection. A number of very recent methods have been proposed which, similar to our own work, attempt to address the issue of scale selection on a dense grid. In [24], a modified SIFT flow process is described which attempts to assign all image pixels with scale estimates. These scales are then used to extract regular (scale varying) SIFT descriptors. A different optimization altogether was earlier proposed by [25] for the specific task of optical flow estimation, when the two images are of the same scene. Also designed for optical flow scenarios, the method of [26] attempts to match pixel regions going beyond scale differences and making an assumption of smoothly varying affine transformations between image regions. Finally, rather than estimating scales during the correspondence estimation process, Tau and Hassner [27] propose propagating the scales of sparse interest points to all image pixels, thereby providing a way of assigning pixels in homogenous regions with scale estimates.

3 The Behavior of SIFT Across Scales

We begin by considering how the values of multiple SIFT descriptors vary through scales. The scale space $L(x, y, \sigma)$ of an image $I(x, y)$ is defined by the convolution of $I(x, y)$ with the variable-scale Gaussian $G(x, y, \sigma)$ [28], where:

$$\begin{aligned} L(x, y, \sigma) &= G(x, y, \sigma) \star I(x, y) \\ G(x, y, \sigma) &= \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \end{aligned}$$

Typically (Sec. 2), a feature detector selects coordinates in space x, y and scale σ, from which a single SIFT descriptor $h_\sigma = h(x, y, \sigma)$ is then extracted [3]. Although sometimes more than one scale is selected, they are usually treated independently of each other.

Here, we consider instead all the descriptors $h_{\sigma_i} = h(x, y, \sigma_i)$, where σ_i is taken from a discrete set of scales $\{\sigma_1, ..., \sigma_k\}$. Our chief assumption is that corresponding pixels should exhibit a similar behavior throughout scales. In other words, the same pattern of SIFT descriptors $h(x, y, \sigma_i)$ should be apparent when examining corresponding pixels. The challenge then becomes how to effectively capture this pattern of change across scales?

3.1 SIFT sets

Rather than selecting a single scale for each pixel, we compute multiple descriptors at multiple scales and represent pixels as sets of SIFT descriptors. Formally,

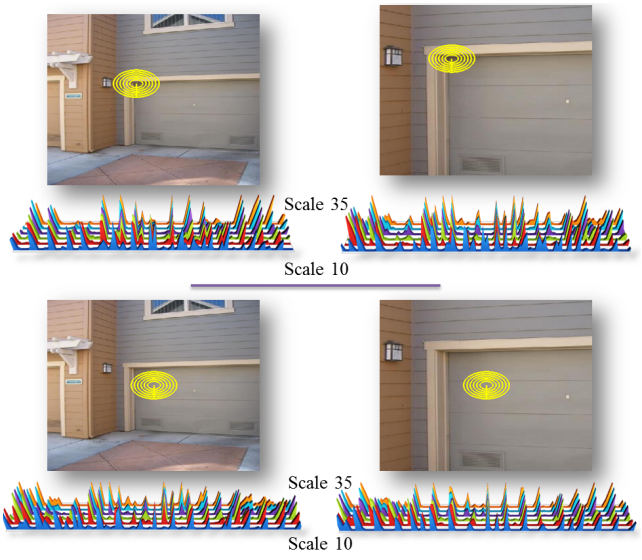


Fig. 3. **SIFT behavior through scales.** Two images separated by a $\times 2$ scale factor. Top: SIFT descriptors extracted at a detected interest point, near a corner structure in the image. Bottom: Descriptors extracted at a low contrast region where *no* interest point was detected. In both cases, SIFTs were extracted at scales ranging from 10 to 35. We illustrate the SIFT descriptor histogram values for each set of descriptors. These demonstrate that (a) Even in low contrast areas, SIFT values are not uniform and (b) the values of the SIFT descriptors gradually change through scales.

denote by p and p' a pair of corresponding pixels in images I and I', respectively. For a set of scales $\sigma_1, \ldots, \sigma_k$, the two pixels are represented by the sets $H = [h_{\sigma_1}, \ldots, h_{\sigma_k}]$ and $H' = \left[h'_{\sigma_1}, \ldots, h'_{\sigma_k}\right]$.

To match the pixels of two images, a set-to-set similarity definition is required. There are several such measures available, e.g., [29]. As we will show in Sec. 4, however, highly accurate matching results are obtained by considering the straightforward "min-dist" measure [29], defined as follows.

$$mindist(p, p') = \min_{i,j} dist(h_{\sigma_i}, h'_{\sigma_j}). \quad (1)$$

Comparing two pixels represented as n SIFT descriptors, would require $O(128 \times n^2)$ operations, which may be prohibitive if the sets are large. Often, however, only a few scales are required to provide accurate representations (Sec. 4). This is explained by the following assumption.

Assumption 1 - Corresponding points are similar at multiple scales. Our underlying assumption is that there exists a set of scales $\sigma_1, \ldots, \sigma_k$ for image I and a set of scales $\sigma'_1, \ldots, \sigma'_k$ for image I', such that the descriptors produced at the two pixels are equal (or else sufficiently similar): $h_{\sigma_i} = h'_{\sigma'_i}$. Let $H = [h_{\sigma_1}, \ldots, h_{\sigma_k}]$ and $H' = \left[h'_{\sigma'_1}, \ldots, h'_{\sigma'_k}\right]$, then we can write $H \sim H'$ (in practice, $H = H'$, up to small, potential, image sampling differences).

This equality, however, holds only when all the scales $\sigma_1, \ldots, \sigma_k$ and $\sigma'_1, \ldots, \sigma'_k$ correspond exactly. In practice, we do not have these correspondences and instead sample the scales at fixed intervals for all images. Thus, the set of scales in one image may be interleaved with the other. Because SIFT values change gradually with scale, only a few scales need to be sampled to provide similar descriptors even in such cases. This is illustrated in Fig. 4 which demonstrates SIFT values in multiple scales of two images separated by a $\times 2$ scale factor. SIFTs in the Target image match the SIFTs in the Source image by a scale offset.

3.2 SIFT subspaces

An alternative, geometric representation for sets of SIFT descriptors, is obtained by considering the linear subspace on which these SIFTs reside. Subspaces have often been used to represent varying information. Some recent examples are listed in [10], [11]. Here, we show that low-dimensional linear subspaces are highly capable of capturing the scale-varying values of SIFT descriptors.

Assumption 2 - Descriptors computed at multiple scales of the same point span a linear subspace. The SIFT descriptor consists of gradient histograms. In many cases, the local statistics of these gradients are equivalent at different scales. For example, in homogeneous, low-contrast regions or areas of stationary textures, the size of the local neighborhood does not change the distribution of gradients. In these cases, we get $h_{\sigma_i} = h_{\sigma_j}$ for $\sigma_i \neq \sigma_j$.

In other cases, the statistics do change with the scale. However, if we sample the scales densely enough, these changes are gradual and monotonic as illustrated in Fig. 3 and empirically demonstrated in Fig. 4. When the descriptor is a smooth function of scale, then, ideally, a descriptor from any scale can be approximated well with a linear interpolation from neighboring scales, i.e.: $h_{\sigma_i} = \sum_j w_{ij} h_{\sigma_j}$, where w_{ij} are scalars. In other words, each descriptor can be represented as a linear combination of several other descriptors at different scales. This occurs when the regions surrounding the patch are piecewise stationary. Enlarging the window size by small steps maintains similar statistics within each window.

The observations above suggest that the set of descriptors $h_{\sigma_1}, \ldots, h_{\sigma_k}$, when sampled densely enough, *approximately* lie on a linear subspace:

$$H = [h_{\sigma_1}, \ldots, h_{\sigma_k}] = \left[\hat{h}_1, \ldots, \hat{h}_b\right] W = \hat{H} W \quad (2)$$

where $\hat{h}_1, \ldots, \hat{h}_b$ are orthonormal basis vectors spanning the space of descriptors, $\hat{H}$ is the matrix with these vectors as its columns and W is a matrix of coefficients.

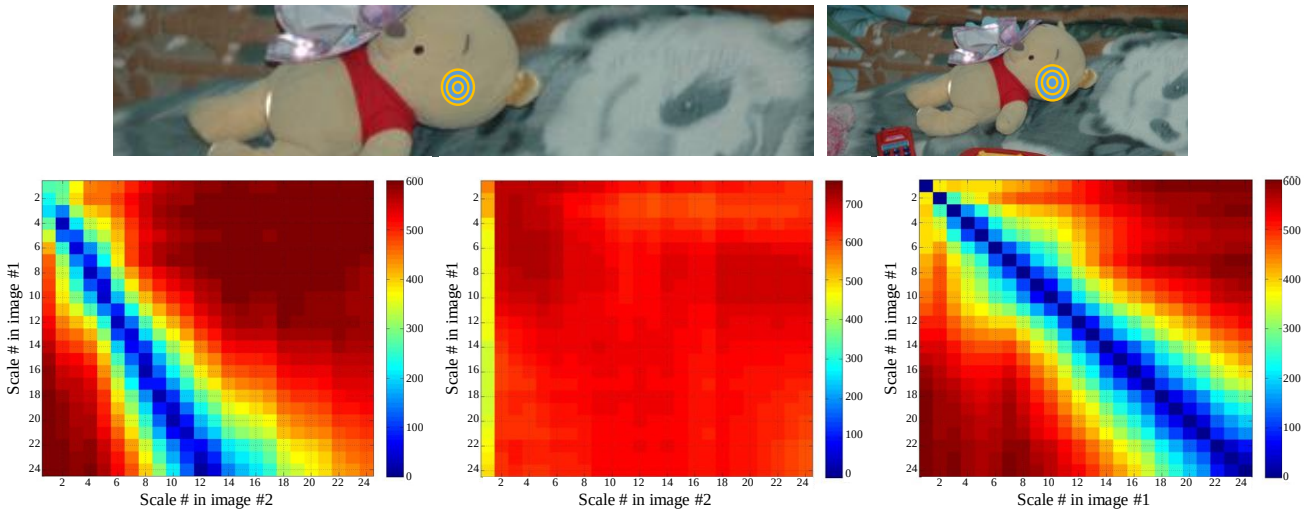


Fig. 4. **SIFT-to-SIFT distances between two sets. Top:** Two images of different size. SIFT descriptors are extracted at a low contrast area where *no* interest point was detected at 24 scales. **Bottom:** SIFT descriptor distance matrix for the various scales. It demonstrates that matching differently scaled descriptors around (a) corresponding points: SIFTs from the Target image match those at higher scales in the Source, implying that setting the same scale to all pixels in both images may lead to poor matches. (b) non-corresponding points: the distance between these descriptors is significantly larger, suggesting that they would not match. (c) the same point: the self SIFT distance matrix shows that SIFTs change gradually across scales, suggesting that descriptors are a smooth function of scale.

Descriptor to subspace mean distance. Our goal for feature matching applications is for the descriptor to be significantly closer to its own subspace rather than to other subspaces. We demonstrate distance comparison in Fig. 5. For each pixel in the image, a SIFT subspace is estimated. The mean distance $m_{\mathbf{p}}$ of the descriptors in the set to the corresponding subspace represented as the matrix $\hat{H}$ with orthonormal columns,

$$m_p = \frac{1}{k} \sum_{1<i<k} ||\hat{H}_p^T \hat{H}_p h_{\sigma_i} - h_{\sigma_i}||^2, \qquad (3)$$

is illustrated for each pixel. This can be compared with the mean distance of each descriptor to a non-corresponding subspace across the image (not shown). While the maximum descriptor to self-subspace Euclidean distance is ≈ 40, the mean distance of the descriptors to the non-corresponding subspaces is ≈ 300. This will enable proper feature matching, as non-corresponding subspace distance is significantly higher relatively to subspace matching inaccuracy. The figure also shows us that large parts of the images do not have the corner structures typically required for the stable scale selection and scale invariant descriptors. It is in those regions where scale is hard to estimate that subspace fitting works best.

Combining the two assumptions. According to assumption 1, for two corresponding pixels, if we knew the set of corresponding scales we would have $H \sim H'$ (or, ideally, $H = H'$). This implies that the two sets of descriptors share the same spanning basis, i.e., $\hat{H}$ and $\hat{H}'$ represent the same subspace. While we do not know the scales required to construct H and H', according to assumption 2 this is not crucial. As long as we sample the scale densely enough we can compute the bases $\hat{H}$ and $\hat{H}'$. Of course, differently scaled images would sometimes imply that scales existing in one image are not present in the other, and vice versa. If image statistics change monotonically, however, introduction of new scales should not substantially change the subspace representation, as we have observed in practice.

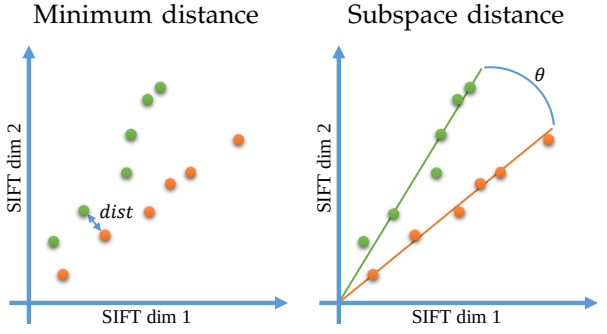


Fig. 6. **If SIFTs were 2D**: A visualization of matching 2d SIFT descriptors, looking at multiple SIFTS taken at different scales. (Left) The distance between two sets is the distance between the two nearest points. (Right) The distance between the two subspaces is related to the angle between them. See Sec. 3.2.

The distance between a pair of pixels, p and p', can

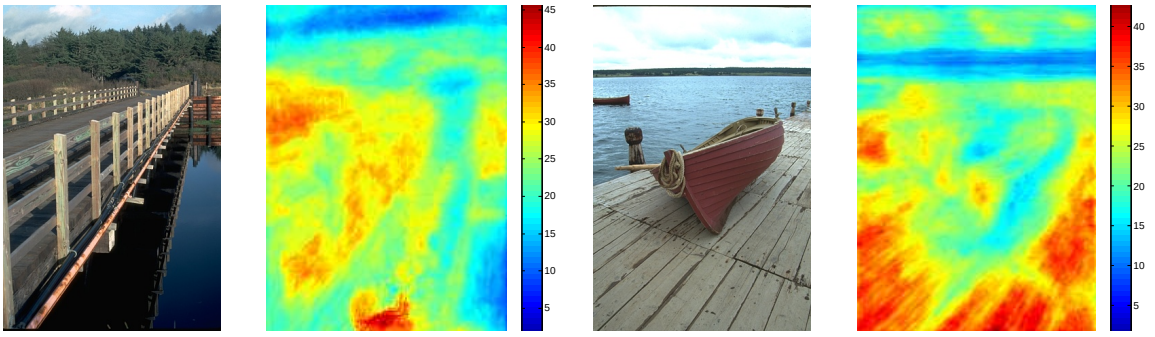


Fig. 5. **Descriptor to SIFT subspace mean distance. Left:** The images. **Right:** Mean distance from the descriptors in the set to the corresponding SIFT Subspace per pixel (Eq. 3). Large portions of the images do not have the corner structures necessary for accurate scale selection and SIFT descriptor extraction. It is in those image regions that subspaces fit best.

be measured by the distance between the corresponding subspaces $\mathcal{H}_p$ and $\mathcal{H}_{p'}$ represented as matrices $\hat{H}$ and $\hat{H}'$ with orthonormal columns. There are several possible definitions to the distance $\mathrm{dist}^2(\mathcal{H}_p, \mathcal{H}_{p'})$ between two linear subspaces [30]. Very often, however, this is expressed by considering the *principle angles between subspaces* (PABS) [31], defined as follows. The principle angles

$$\boldsymbol{\theta}(\mathcal{H}_p, \mathcal{H}_{p'}) = [\theta_1, \ldots, \theta_b], \theta_i \in [0, \pi/2], i = 1, \ldots, b$$

between our subspaces $\mathcal{H}_p$ and $\mathcal{H}_{p'}$, are defined recursively as

$$s_i = cos(\theta_i) = \max_{x \in \mathcal{H}_p} \max_{y \in \mathcal{H}_{p'}} |x^T y| = |x_i^T y_i|,$$

subject to

$$||x|| = ||y|| = 1, x^T x_j = 0, y^T y_j = 0, j = 1, \ldots, b-1.$$

It can be shown [32], [31] that for the matrices $\hat{H}$ and $\hat{H}'$ above, if $U\Sigma V^T = SVD(\hat{H}^T\hat{H}')$, is the singular value decomposition (SVD) of $\hat{H}^T\hat{H}'$ into unitary matrices U and V, and Σ is a $b \times b$ diagonal matrix with real elements $s_1, \ldots, s_b$ in nonincreasing order, then

$$\cos \boldsymbol{\theta}(\mathcal{H}_p, \mathcal{H}_{p'})^{\uparrow} = \mathbf{S}(\hat{H}^T\hat{H}') = [s_1, \ldots, s_b]^T$$

Here, $\cos \boldsymbol{\theta}(\mathcal{H}_p, \mathcal{H}_{p'})^{\uparrow}$ is the vector of principle angles between the two subspaces, $\mathcal{H}_p$ and $\mathcal{H}_{p'}$, arranged in nondecreasing order, and $\mathbf{S}(\hat{H}^T\hat{H}')$ is the vector of singular values of $\hat{H}^T\hat{H}'$ similarly arranged.

This is often used in practice to obtain the related measure of subspace similarity, the *Projection Frobenius Norm* (Projection F-Norm), which is defined by:

$$\mathrm{dist}^2(\mathcal{H}_p, \mathcal{H}_{p'}) = ||\sin \boldsymbol{\theta}(\mathcal{H}_p, \mathcal{H}_{p'})||_2^2, \qquad (4)$$

where the vector of sines, $\sin \boldsymbol{\theta}(\mathcal{H}_p, \mathcal{H}_{p'})$, is obtained following the result above in $O(128 \times d^2)$ operations using SVD, with d being the subspace dimension. Fig. 6 illustrates the different interpretations of the distances between the two sets of SIFT descriptors.

3.3 The Scale-Less SIFT (SLS) representation

It is often beneficial to have a *point* representation for each pixel, rather than a subspace. Such is the case when, for example, efficient indexing is required. We, therefore, employ the subspace-to-point mapping proposed by Basri et al. [33], [10], [11] to produce the Scale-Less SIFT (SLS) descriptor for each such subspace.

Specifically, the Projection F-Norm defined above is named so, as it is closely related to the Frobenius Norm of the orthographic projection matrices of the two subspaces:

$$\mathrm{dist}^2(\mathcal{H}_p, \mathcal{H}_{p'}) = 1/2(||\hat{H}\hat{H}^T - \hat{H}'\hat{H}'^T||_F^2) \qquad (5)$$

We should note that if the two subspaces were of different dimensions, then an additional additive constant reflecting the dimensions of the two subspaces and the difference in dimensionality would also be included on the right hand side of Eq. 5. Since here we only use subspaces of the same intrinsic dimensions, b, this constant equals zero [11].

Basri et al. noted that the Frobenius norm of a square matrix $\mathbf{A}$ can be computed by summing the squares of its entries, or $||\mathbf{A}||_F^2 = \sum_{i,j} \mathbf{A}_{i,j}^2$, and that this can in turn be computed from the L_2 norm of a vector, $\mathbf{a}$, obtained from rearranging the values of $\mathbf{A}$ into one long vector. Since orthographic projection matrices are symmetric, elements outside of the diagonal need only appear once in the vector representation of $\mathbf{A}$. This would additionally require that diagonal elements be scaled by $1/\sqrt{2}$ in order be correctly represented in the expression of Eq. 5.

By using these results, we obtain the following mapping of the subspace $\mathcal{H}_p$, produced at pixel p and represented as a $128 \times d$ matrix $\hat{H}$ with orthonormal columns, to a point representation P – our SLS representation, as follows. The elements of the projection matrix $A = \hat{H}\hat{H}^T$ are rearranged, removing duplicate elements outside the diagonal and scaling diagonal values. More formally, we apply the operator:

$$SLS(\hat{H}_p) = \left(\frac{a_{11}}{\sqrt{2}}, a_{12}, ..., a_{1d}, \frac{a_{22}}{\sqrt{2}}, a_{23}, ..., \frac{a_{dd}}{\sqrt{2}} \right)^T, \quad (6)$$

where a_{ij} is the element (i, j) in matrix A.

In summary, we get that the distance between two mapped subspaces, P and P' is monotonic with respect to the Projection F-Norm between the original subspaces $\mathcal{H}_p$ and $\mathcal{H}_{p'}$ [10], [11]. That is:

$$||P - P'||^2 = 1/2 \operatorname{dist}^2(\mathcal{H}_p, \mathcal{H}_{p'}) \quad (7)$$

Point P thus captures the behavior of SIFT descriptors throughout scale space at pixel p, with a quadratic cost in the dimension of the descriptors. Here, we employ the SLS descriptor, P, as a surrogate for the subspace $\mathcal{H}_p$ without making further adjustments to the method used to compute correspondences.

3.4 SLS and dimensionality reduction

The subspace to point mapping which we use produces a representation which is quadratic in the size of the original representation, 128D for the SIFTs used here. When produced for each and every pixel in the image, storage requirements can quickly grow to be unreasonable. In [11], to address this issue, the original data points were randomly projected, multiple times, to very low dimensions, before the mapping was applied. This resulted in a substantial reduction in size of the mapped subspaces.

We found this procedure to be unsuitable for two reasons. First, random projections require multiple projections (and hence, multiple representations) for each subspace; here, for reasons discussed above, we aim for a single, point representation for each pixel. Second, and more important, we have found that better performance, in terms of both the size of the final descriptor and the accuracy of the obtained flow, can be obtained by an alternative approach.

Specifically, we perform dimensionality reduction of the original SIFT descriptors descriptors *before* mapping them to points. To this end, given two images, we begin by computing dense SIFT descriptors, at multiple scales. The resulting pool of descriptors, obtained from the two image pixels, in all scales, are then used to compute an eigenspace of SIFTs. All SIFTs are then projected to a lower dimension and only then are the final SLS descriptors extracted – using the dimension-reduced SIFTs.

The benefits of reducing the dimensionality of SIFT descriptors by PCA is well known [35]. Here, however, we compute the eigenspace of SIFTs using descriptors extracted densely over space and scale, to our knowledge, for the first time. Key to this is the observation that if the descriptors extracted from multiple scales around a particular pixel, all reside on a low-dimensional subspace (assumption 2 in Sec. 3.2), then their linear projections will likewise span a subspace.

In practice, we use PCA to reduce SIFT dimensions to 32D, which in turn, produces SLS descriptors of 528D (slightly more than four times the size of the original SIFT). For computational purposes, the PCA projection matrix was computed using a random subset of all SIFT descriptors, from both images, at all scales. We next provide results of a wide range of experiments comparing the dimensionality reduced SLS descriptor, PCA-SLS, to the full descriptor, as well as a range of alternative representations.

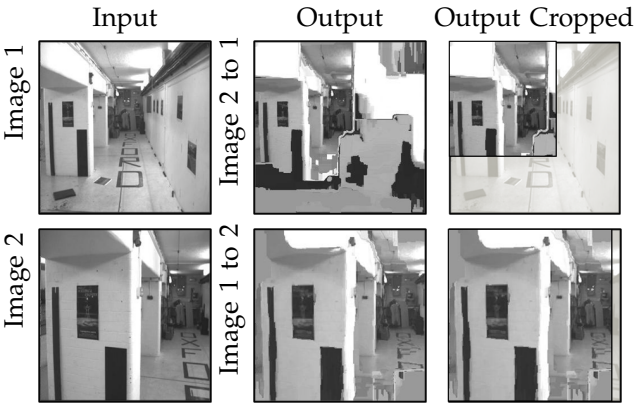


Fig. 7. **Auto-crop to the ROI.** Dense matches directly formed, without estimating Epipolar Geometry, between the first and last images of the Oxford Corridor sequence [34] (left column). On the right, notice the large areas where no information is available in Image 2 to correspond with parts of Image 1. These areas are automatically cropped to include only the area onto which pixels from the second image were warped.

3.5 Cropping to the region of interest

When matching views of significantly different scales, warping one image to the other introduces the problem of cropping the image to its region of interest (ROI). In [18] this problem is avoided by assuming that the high resolution image is neatly cropped. Without this knowledge, the warped high resolution image would include noisy, "smeared" areas where it does not overlap the low resolution image (see Fig. 7).

Here we automatically select the region of high confidence matches, as follows. Given images I and I', we compute the two dense flows, from I to I' and then back, from I' to I. In both cases, we count for each pixel in the target image, the number of source image pixels which were mapped onto it. We threshold the pixels by these numbers and then apply morphological operators to remove small clusters of target pixels. Finally, the ROI of image I is selected as the bounding box of the remaining target pixels obtained by warping image I', and vice versa. This is demonstrated in Fig. 7. No optimization was

TABLE 1
Results on the scaled-Middlebury benchmark. Angular errors (AE) and endpoint errors (EE), ± SD, on resized images from the Middlebury benchmark [36]. Lower scores are better; bold numbers are best scoring.

Data	DSIFT [7]	SID [19]	Seg. SIFT [37]	Seg. SID [37]	SLS	SLS-PCA	SLS-PCA-1
			Angular Errors ± SD				
Dimetrodon	3.13 ± 4.0	**0.16 ±0.3**	2.45 ± 2.8	0.23 ± 0.7	0.17 ±0.5	0.18 ± 0.5	0.19 ± 0.4
Grove2	3.89 ± 11.9	0.66 ±4.4	4.77 ± 15.3	0.22 ± 0.6	**0.15 ±0.3**	0.17 ± 0.4	0.17 ± 0.4
Grove3	2.67 ± 2.8	1.62 ±6.9	8.93 ± 15.6	0.22 ± 0.6	**0.15 ±0.4**	0.18 ± 0.5	0.18 ± 0.5
Hydrangea	9.76 ± 18.0	0.32 ±0.6	7.10 ± 10.6	0.23 ± 0.7	**0.22 ±0.8**	0.23 ± 0.6	**0.22 ± 0.5**
RubberWhale	5.27 ± 8.6	0.16 ±0.3	6.13 ± 17.2	0.16 ± 0.3	**0.15 ±0.3**	0.17 ± 0.3	0.17 ± 0.3
Urban2	3.65 ± 10.7	0.37 ±2.7	2.82 ± 4.1	**0.25 ± 1.1**	0.32 ±1.3	0.31 ± 0.1	0.40 ± 1.4
Urban3	3.87 ± 5.1	0.27 ±0.6	3.53 ± 4.4	0.31 ± 1.0	0.35 ±0.9	**0.25 ± 0.5**	**0.25 ± 0.5**
Venus	2.66 ± 2.9	0.24 ±0.6	2.77 ± 6.7	**0.23 ± 0.5**	**0.23 ±0.5**	0.27 ± 0.6	0.27 ± 0.6
			Endpoint Errors ± SD				
Dimetrodon	10.97 ± 8.7	**0.71 ±0.3**	10.34 ± 7.5	0.97 ± 1.1	0.80 ±0.4	0.87 ± 0.5	0.87 ± 0.5
Grove2	14.38 ± 11.5	1.5 ±5.0	15.50 ± 11.0	1.05 ± 1.9	**0.77 ±0.4**	0.83 ± 0.4	0.83 ± 0.4
Grove3	13.83 ± 9.7	4.48 ±10.5	24.33 ± 20.0	1.37 ± 3.3	**0.87 ±0.4**	0.95 ± 0.5	0.95 ± 0.5
Hydrangea	25.32 ± 17.1	1.59 ±2.8	24.21 ± 17.3	0.88 ± 0.6	0.91 ±1.1	0.87 ± 0.5	**0.85** ± 0.5
RubberWhale	22.59 ± 15.8	**0.73 ±1.1**	17.33 ± 14.8	**0.73 ± 0.4**	0.8 ±0.4	0.88 ± 0.5	0.86 ± 0.5
Urban2	18.96 ± 17.5	1.33 ±3.8	13.36 ± 10.3	**1.21 ± 3.7**	1.51 ±5.4	1.46 ± 4.1	1.83 ± 5.9
Urban3	19.83 ± 17.1	1.55 ±3.7	15.44 ± 11.5	1.47 ± 4.1	9.41 ±24.6	**1.03 ± 0.7**	1.06 ± 0.7
Venus	9.86 ± 8.7	1.16 ±3.8	11.86 ± 11.4	**0.74 ± 0.5**	**0.74 ±0.3**	0.87 ± 0.5	0.87 ± 0.5

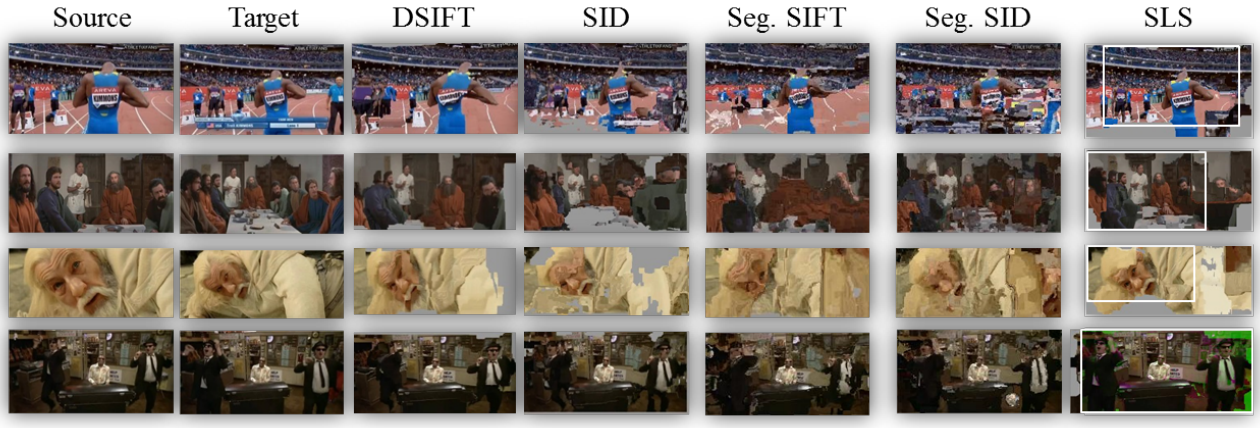


Fig. 8. **Dense flow with scene motion**. Image pairs presenting different scale changes in different parts of the scene, due to camera and scene motion. Correspondences from Source to Target images estimated using [9], comparing DSIFT [8], SID [19], Segmented SID and segmented SIFT, both from [37] and our SLS, shown here with the automatically determined crop region in white (Sec. 3.5).

performed on this process and it is applied without modification to all our images.

4 Experiments

Our evaluation code was written in MATLAB, using the SIFT code of [7], the SID code of [19] and the segmented SIFT and SID code from [37]. Flow was estimated using the original SIFT flow code [8], [9], with either its original DSIFT, or alternatively using SID, segmented SIFT and SID, and our own SLS descriptor. Our SLS results were produced using 8D, linear subspaces obtained by standard PCA. We used 20 scales at each pixel, linearly distributed in the range $[0.5, 12]$. Note that the size of the SLS representation and the matching time depends only on the dimension of the underlying SIFT descriptor (Sec. 3.3).

To promote reproducibility, we publicly released our code, including the new dimensionality reduced SLS descriptors, PCA-SLS, described in Sec. 3.4. Our implementation is available from [1].

4.1 Dense correspondence estimation

Quantitative results on Middlebury data [36]. We compare our SLS and SLS-PCA with DSIFT, SID and the segmented versions of SID and SIFT, Seg. SID and Seg. SIFT, on the Middlebury optical flow set. Since the image pairs in the Middlebury do not exhibit significant scale changes, we modify the data by rescaling the Source and Target images by factors of 0.7 and 0.2, resp. The quality of an estimated match was measured using both angular and endpoint errors (± SD) [36]. Table 1 shows that both multiscale approaches outperform single-scale DSIFT significantly.

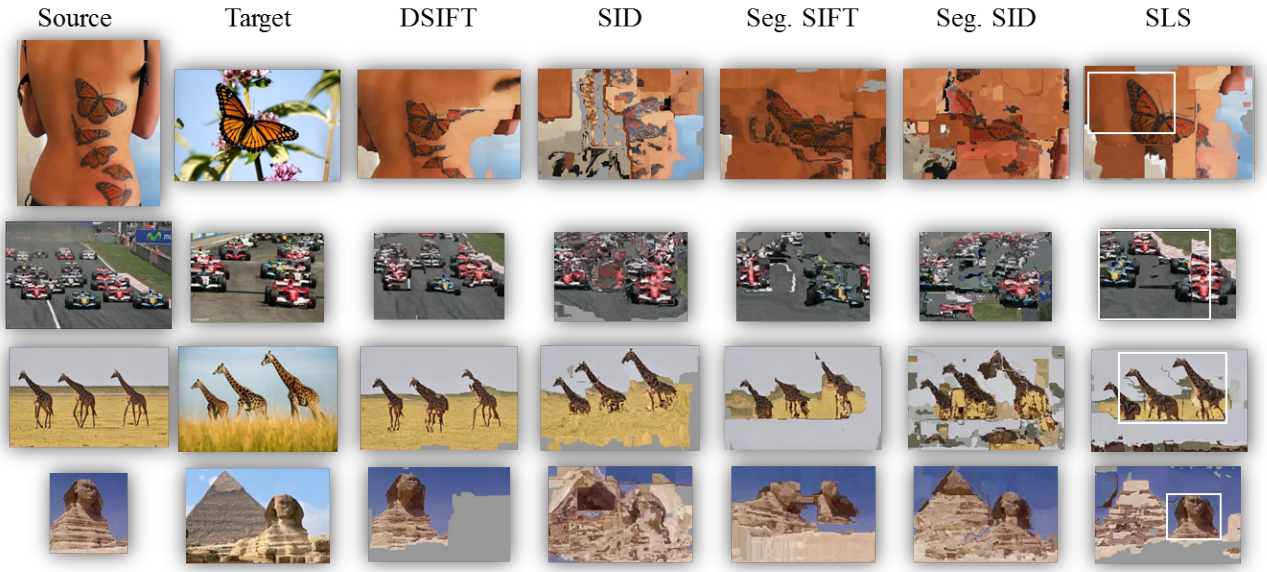


Fig. 9. **Dense flow between different scenes in different scales**. Correspondences from Source to Target images estimated using [9], comparing DSIFT [8], SID [19], Segmented SID and segmented SIFT, both from [37] and our SLS, shown here with the automatically determined crop region in white (Sec. 3.5).

Furthermore, our SLS descriptors lead to lower errors when compared to the descriptors of [19] and [37].

SLS-PCA uses basis vectors computed from descriptors extracted from both images in order to reduce the dimensionality of the representation. In many practical cases, one image, available for preprocessing, is repeatedly compared against others. Table 1 also additionally provides results obtained by SLS-PCA-1, which denotes the use of PCA computed using SIFT descriptors from only one image, and the resulting projection to lower dimension then applied to the descriptors of both images. Evidently, pre-computing the dimensionality reduction projections using a single image, results in only a minor compromise in accuracy.

Qualitative results. We present a visual comparison of the quality of the estimated flows, using each of the three alternatives: DSIFT, SID and our SLS descriptor. Our results present a Source image warped onto the Target image according to the estimated flows. SLS results in Fig. 8 and 9 are further cropped to show areas of high confidence matches (see below).

We ran tests on image pairs with independent scene motion (Fig. 8) and images of different scenes with similar appearances (Fig. 9). All pairs include scale differences, often extreme. We know of no previous method which successfully presents dense correspondences on such challenging image pairs. Our results show that the SLS enables accurate dense correspondences even under extreme changes in scale.

In Fig. 8 DSIFT typically manages to lock onto a single scale quite well, while missing other scale changes in the scene. The SLS descriptor better captures the scale-varying behavior at each pixel and so manages to better match pixels at different scales with only local misalignments.

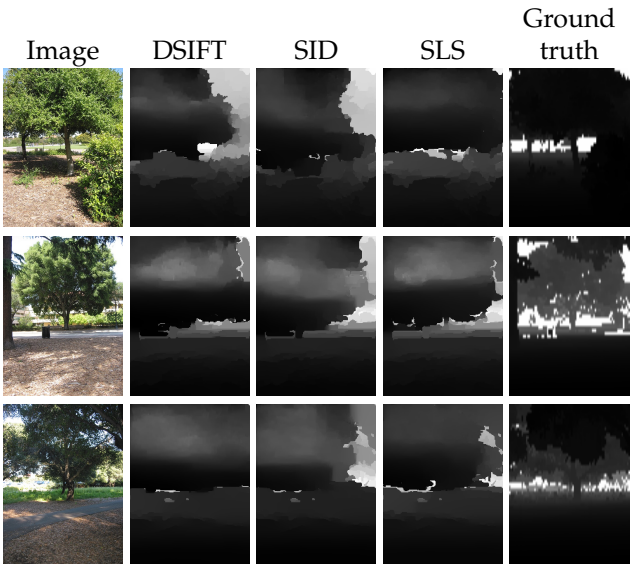


Fig. 10. **Make3d depth transfer.** Estimated depth maps of an image from the Make3d data [38][39]. The SLS result is the most similar to the ground truth.

Fig. 1 and 9 present matches estimated between images of *different* scenes. A good result would have the appearance of the Source (left) images, in the scales and poses of the Target (right) images. As can be seen, the DSIFT and SID descriptors either leave the source in its original scale, unchanged, or else completely fail to produce coherent matches. Although some artifacts are visible in the SLS results (right column), the results

TABLE 2
Depth transfer on Make3d data - Relative Error, Log-10 Error and RMSE. Testing data was rescaled to 0.1, while the training data was rescaled to 0.1, 0.2, 0.3 and 0.4. SLS and Seg. SID descriptors obtain the lowest errors when scale difference is introduced. Using dimension reduction, SLS-PCA can be run on larger images as well. Results are missing for representations and image sizes which were too big to run on our system. Please see text for more details.

	Training data rescale factor											
	0.1			0.2			0.3			0.4		
Method	Rel.	log10	RMSE	Rel.	log10	RMSE	Rel.	log10	RMSE	Rel.	log10	RMSE
DSIFT [8]	0.419	0.165	15.127	0.479	0.287	21.467	0.643	0.243	19.819	0.780	0.250	19.996
SID [19]	0.420	0.174	15.340	0.486	0.261	20.564	N/A	N/A	N/A	N/A	N/A	N/A
Seg.SID [37]	**0.391**	**0.154**	14.785	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
SLS	0.400	0.164	15.396	**0.449**	**0.251**	**20.499**	N/A	N/A	N/A	N/A	N/A	N/A
SLS-PCA	0.411	0.159	**14.692**	0.471	0.268	20.919	**0.618**	**0.239**	**19.791**	**0.726**	**0.242**	**19.699**

present coherent scenes in the target image scales.

4.2 Correspondences on the "Oxford" set

We compare our SLS descriptor against DSIFT and SID also on images from the Oxford, "Mikolajczyk" data set [40]. We use the *Bikes* and *Trees* sets which present slight rotation and translation, but mostly scale (blur) changes and the *Leuven* set, whose images mostly vary in illumination. Since our SLS descriptors are designed to work well on the vast majority of the image pixels – those outside the set of detected interest points – rather than comparing matching at interest points, we compare matching accuracy at 100 randomly selected pixels in the image.

Specifically, we compare the percent of times that a descriptor extracted at a random pixel in the first image of each series, is matched to its ground truth pixel in subsequent images of the series. Matching is performed using nearest neighbor computed using L2 distances between descriptors. Ground truth uses the known homographies between the images of each sequence to find ground truth correspondences. The descriptors tested are DSIFT [7], SID [37] and our SLS.

Fig. 11 presents these results. Evidently, the two representations designed to capture scales outperform DSIFT. Both SID and SLS perform abut the same in the presence of gradual lighting changes, SLS outperforming SID in matching pixels across scale changes.

4.3 Depth estimation from a single image

We ran a test on the Make3d data [38][39], using evaluation code by [41]. The dataset includes 400 training images and 134 testing images with known depth data. The evaluation code finds $k = 7$ similar images from the train set, computes descriptors, and calculates the SIFT flow between the query image and each of these k images using [8], [9]. The flows are applied to the k ground truth depths which are then merged together for the final depth.

The testing data was resized to 0.1 (10%) of the original image size, and the training data was resized to 0.1, 0.2, 0.3 and 0.4. For this test, we selected a random subset of 30 testing images and used SIFT [7], SID [19], the Seg. SID of [37] and our own SLS descriptors. We also provide results for dimension reduced SID and SLS descriptors (see Sec 4.1) The larger sizes (0.3 and 0.4) were only executed with SIFT and SLS-PCA due to memory limitations. Results were omitted for the Seg. SIFT descriptor [37] as it performed substantially worst than the others.

Table 2 reports actual reconstruction accuracy. Apparently, SLS descriptors are comparable to the best performing alternative method, yet can be applied to a greater range of scales by applying dimensionality reduction with PCA. We provide a number of example reconstructions along with their respective ground truths in Fig. 10.

4.4 SIFT points, sets, and subspaces

We next assess the quality of traditional, single scale SIFT descriptors against the various multiscale representations discussed here. This is performed twice: First, we compare the various representations at detected interest points where stable scales can be estimated allowing for scale invariant SIFT descriptors to be extracted. We then repeat our experiment, but this time extracting our representations on a dense grid, throughout the image and without scale selection available at each pixel.

To this end, we use images from the Berkeley image set [42]. Each image is matched against itself, scaled by a random scale factor in the range of $[1.5 \dots 4]$. Our results are reported in Fig. 12. It provides the performance of the various representations by their true and false positive rates:

$$True\ Positives\ Rate = \frac{\#correct\ matches}{\#possible\ correct\ matches}$$

$$False\ Positives\ Rate = \frac{\#incorrect\ matches}{\#possible\ incorrect\ matches}$$

Here, a false match is any time a pixel is not matched to its known, ground truth correspondence, computed using the scale factor used for each pair.

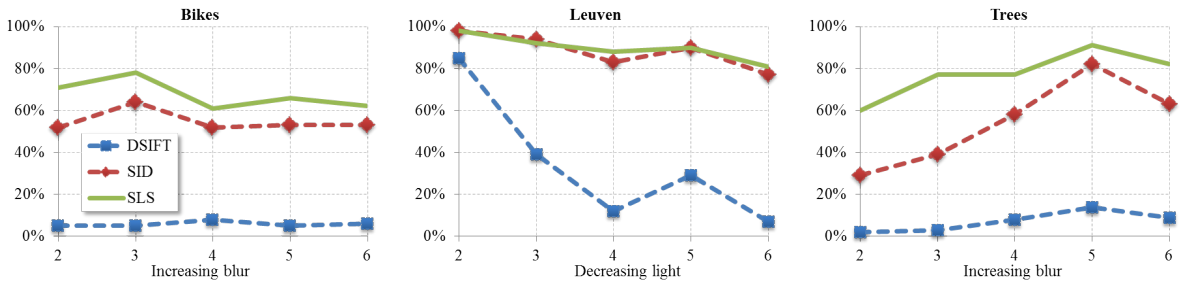
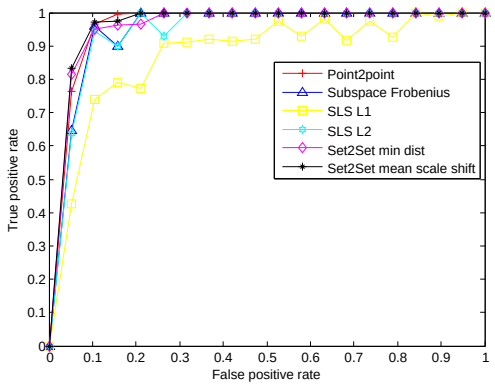
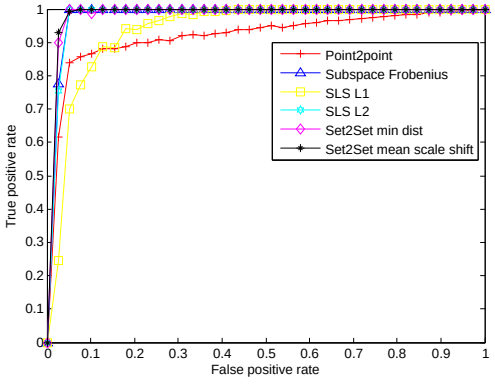


Fig. 11. **Oxford data set results [40].** Comparing descriptor matching accuracy of 100 randomly selected image points represented by DSIFT [7], SID [37] and our own SLS. Accuracy denotes the percent of the times that descriptors in image ♯1 of each set are matched with their true corresponding pixel in each of the subsequent images, using L2 distances between descriptors. Please see text for more details.



(a) Interest points



(b) Dense sampling

Fig. 12. **Interest points vs. dense sampling on Berkeley data set.** We examine different descriptors on: (a) Interest points and (b) Dense sampling. Clearly, point-to-point matching of SIFT descriptors performs better at detected interest points. This advantage should be weighed against the small number of such points in the image. In all other pixels, set-to-set representations outperform single scale, point-to-point representations.

The following methods were compared: SIFT descriptors from one image were matched to those of the other based on the closest L2 neighbor (*point2point*). SIFT sets were matched using the min-dist of [29] (*set2set min dist*) and the minimum over the average distances computed for all discrete scale shifts between the two images (*set2set mean scale shift*). The distances between subspace representations were computed using the Projection Frobenius Norm (*Subspace Frobenius*). Finally, we measure the L1 and L2 distances between our SLS descriptors (*SLS L1* and *SLS L2*, resp.).

Fig. 12 (a) shows that single scale SIFT representations, when extracted at stable scales are very discriminative and they can be matched reliably, even compared to the multi scale representations. This, of course, is not surprising, and is the reason why SIFT descriptors have become so popular in computer vision systems. This performance, however, should be weighed against the ability to extract effective SIFTs throughout the image: in 500×500 pixel images, roughly 2,000 SIFT descriptors can be extracted at stable scales in order to achieve such accuracy.

Fig. 12 (b) compares the same methods on a dense grid. Here, single scale SIFT descriptors were extracted using the scale corresponding to the maximum DoG value at each pixel. Clearly, the matching accuracy of single scale SIFTs drops considerably to well below those of the multi scale representations. This implies that in order to avoid ignoring the majority of the image in favor of a small number of interest points, multiscale representations should be used, rather than single scale SIFTs.

4.5 Parameter evaluation

We next evaluate the influence of various parameters on feature matching accuracy and run-time. We again use images from the Berkeley set [42], rescaled by a randomly determined scale factor uniformly distributed in the range $[1.5 \ldots 4]$. We report the mean±SD accuracy and run-time of matching pixels on regular grids, between each such image pair. Accuracy is measured as the ratio of the times a pixel's nearest neighbor is its ground truth matching pixel, to

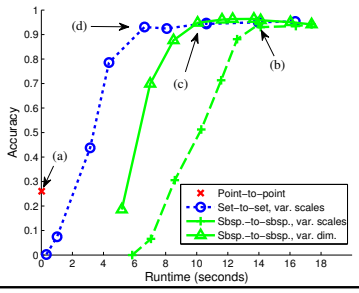
the total number of pixels. Runtime measures the time required for matching.

Fig. 13 presents the following results. **(1) Point-to-point with scale selection**: A single scale is selected for each pixel and is used to extract a DSIFT descriptor. Scale selection follows [3], by choosing the extremum DoG scale, but ignoring any additional filtering. **(2) Set-to-set, variable number of scales**: Using the min-dist measure (Eq. 1) to compute pixel similarities. The number of scales sampled was varied, sampling one to ten DSIFT descriptors from scales distributed linearly in the range of $[0.5, 12]$ using the MATLAB expression `linspace(0.5,12,num_sigma)`. **(3) Subspace-to-subspace, variable number of scales**: Using the same sets as in (2) to fit a linear subspace for each pixel (using PCA). Subspace dimensions equal the number of scales sampled. The distance between two subspaces was computed using Eq. (4). **(4) Subspace-to-subspace, variable dimension**: Same as (3), but here 10 DSIFT descriptors were used to fit subspaces varying in dimension from 1 to 10.

From Fig. 13 it can be seen that when few scales are sampled, a single, carefully selected scale provides better performance than an arbitrarily selected scale. This advantage disappears at 3 scales; accuracy increasing rapidly with more scales sampled. By 5 scales, the matching quality is near perfect for the multiscale representations. The accuracy of the subspace-to-subspace method testifies that these SIFT sets indeed lie close to a low dimensional linear subspace. In fact, it seems that a 4D linear subspace manages to accurately capture scale varying SIFT values. We note that when a single scale is considered, the set-to-set similarity is equivalent to comparing DSIFT descriptors at an arbitrary scale and the subspace-to-subspace distance reduces to a sine similarity of these two DSIFT descriptors. Both are far worse than choosing the single scale at each pixel.

Run-times for the set-based methods are higher than comparing single points. We made no attempt to optimize our code, using built-in MATLAB functions for all our processing, and so better performance may likely be obtained. The complexity of directly comparing two sets (Sec. 3.1) or two subspaces (Sec. 3.2), however, limits the effectiveness of such optimizations. Yet although the set based methods are more computationally expensive, their significantly higher accuracy makes them an alternative worth considering.

Fig. 14 visualizes the results of Fig. 13. In it, we use the target images scaled $\times 2$ as sources, and estimate flow from source to target. Flow vectors, displayed on the source images, were computed using the following three representations: Point-to-point correspondences of SIFT descriptors computed in scales selected using DoG ((a) in Fig. 13); the subspace-to-subspace distance (Eq. (7) in the paper) between 4D subspaces produced by sampling 10 scales linearly distributed in the range of $\sigma = [0.5, 12]$ ((c) in Fig. 13); finally, min-dist (Eq. (4) in the paper) with set representations, using 5 scales samples linearly in the same range as used to produce the subspaces ((d) in Fig. 13). Both multiscale representations provide better correspondences than single descriptors. This is particularly true in low contrast areas where interest points are not typically detected.



Method	Acc.± SD	Runtime ± SD
(a) Point-to-point	0.26±0.06	0.02±0.00
(b) Sub.-to-sub. 7 scales, 7D	0.94±0.05	14.00±0.09
(c) Sub.-to-sub., 10 scales, 4D	0.94±0.05	10.05±0.06
(d) Set-to-set, 5 scales	0.93±0.05	6.64±0.11

Fig. 13. **Accuracy vs. runtime**. See text for details.

TABLE 3
Run-time comparison. Parameters and run-times (in seconds) for the images of Fig. 13, here, rescaled to 133×200 (± SD omitted for clarity). We compare some of the parameter configurations tested in this paper, as well as the time required for using SIFT flow [8]. Subspace representations in row 5–6 represented as SLS descriptors. Rows 5–7 use the same descriptor dimensions.

Method	♯Scales	♯ Dims.	Desc. extraction	Flow
1 DSIFT [7]	-	-	0.68	6.37
2 SID [19]	-	-	131.82	77.00
3 Seg. SID [37]	-	-	149.66	74.89
4 Seg. SIFT [37]	-	-	0.65	6.17
5 Fig. 13 (b)	7	7	538.22	178.52
6 Fig. 13 (c)	10	4	644.1	178.52
7 SLS	20	8	1291.52	178.52
8 SLS-PCA	25	8	1742.74	13.78

Finally, Table 3 provides comparisons of the computational requirements made by the representations used in this paper. In it, run-times in seconds are reported for the process of extracting each of the various dense representations considered here, as well as the time required for SIFT flow to estimate correspondences using each representation. All numbers are reported on the same system with 133×200 pixel images. As expected, our SLS descriptor requires the most time to compute. Its extraction time, however, is balanced with its more accurate results reported throughout this paper. Further evident is the flow estimation run-time advantage of the SLS-

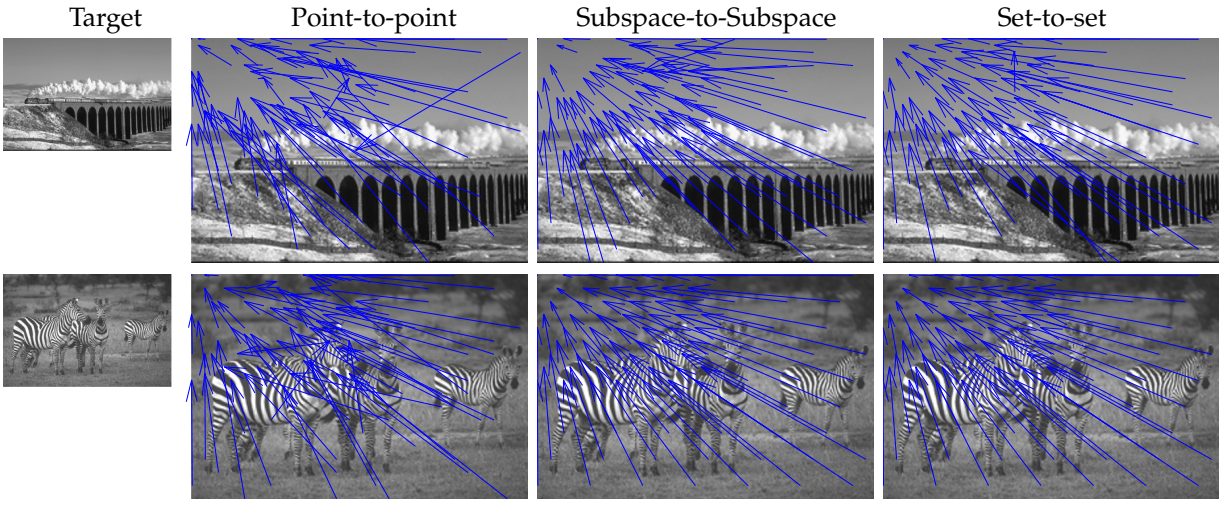


Fig. 14. **Visualization of the quantitative tests on the Berkeley set [42].** Target images are shown in left column. We use the target images, here scaled $\times 2$, as sources, and estimate flow from source to target. Please see text for more details.

PCA representation compared to the other multiscale representations.

5 Conclusion

The scale selection methods that have developed since the early 1990s were largely motivated by a need to reduce computational cost as well as the assumption that few scales can be reliably matched [12]. In this paper, we show that images contain valuable information in *multiple* scales. Thus, scale selection may be detrimental to the quality of the results when dense correspondences are required. The alternative, extracting SIFT descriptors at multiple scales, significantly improves results but at a computational price. We examine how such multiple scales may be compared, representing them as sets or low-dimensional, linear subspaces. In both cases, multiple SIFTs outperform single descriptors in pixel matching tests by wide margins. Finally, we present a point representation for these subspaces, the SLS descriptor, which we use as a stand-in for DSIFT in the SIFT flow method, improving correspondences on a wide range of challenging viewing conditions.

We focus on the SIFT descriptor because of its popularity and its convenient property of changing gradually through scales. It remains to be seen how well the same approach carries over to other successful descriptors, including DAISY [6], SURF [43], LATCH [44], and others. Extensions to affine invariance also require study. Lastly, we intend to examine the impact of this approach in other Computer Vision problems such as those covered by [45].

Acknowledgments

Lihi Zelnik-Manor was supported in part by the Ollendorf foundation, the Israel Ministry of Science, and by the Israel Science Foundation under Grant 1179/11.

References

[1] T. Hassner, V. Mayzels, and L. Zelnik-Manor, "On sifts and their scales," in *Proc. Conf. Comput. Vision Pattern Recognition*, June. 2012. [Online]. Available: http://www.openu.ac.il/home/hassner/projects/siftscales

[2] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *Int. J. Comput. Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[3] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[4] J. Morel and G. Yu, "Is sift scale invariant?" *Inverse Problems and Imaging (IPI)*, vol. 5, no. 1, pp. 115–136, 2011.

[5] H. Hirschmüller and D. Scharstein, "Evaluation of stereo matching costs on images with radiometric differences," *Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1582–1599, 2009.

[6] E. Tola, V. Lepetit, and P. Fua, "Daisy: An efficient dense descriptor applied to wide-baseline stereo," *Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815–830, 2010.

[7] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," in *Proc. int. conf. on Multimedia*, 2010, pp. 1469–1472, available: www.vlfeat.org/.

[8] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, 2011.

[9] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. Freeman, "Sift flow: dense correspondence across different scenes," in *European Conf. Comput. Vision*, 2008, pp. 28–42, people.csail.mit.edu/celiu/ECCV2008/.

[10] R. Basri, T. Hassner, and L. Zelnik-Manor, "A general framework for approximate nearest subspace search," in *Proc. Int. Conf. Comput. Vision Workshop*. IEEE, 2009, pp. 109–116.

[11] ——, "Approximate nearest subspace search," *Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 266–278, 2010.

[12] T. Lindeberg, "Feature detection with automatic scale selection," *Int. J. Comput. Vision*, vol. 30, no. 2, pp. 79–116, 1998.

[13] ——, "Principles for automatic scale selection," *Handbook on Computer Vision and Applications*, vol. 2, pp. 239–274, 1999.

[14] K. Mikolajczyk, "Detection of local features invariant to affine transformations," Ph.D. dissertation, Institut National Polytechnique de Grenoble, France, 2002.

[15] T. Brox, C. Bregler, and J. Malik, "Large displacement optical flow," in *Proc. Conf. Comput. Vision Pattern Recognition*, 2009, pp. 41–48.

[16] C. Strecha, T. Tuytelaars, and L. Gool, "Dense matching of multiple wide-baseline views," in *Proc. Int. Conf. Comput. Vision*, 2003.

[17] J. Yao and W. Cham, "3D modeling and rendering from multiple wide-baseline images by match propagation," *Signal*

processing. Image communication, vol. 21, no. 6, pp. 506–518, 2006.
[18] I. Simon and S. Seitz, "A probabilistic model for object recognition, segmentation, and non-rigid correspondence," in *Proc. Conf. Comput. Vision Pattern Recognition*, 2007, pp. 1–7.
[19] I. Kokkinos and A. Yuille, "Scale invariance without scale selection," in *Proc. Conf. Comput. Vision Pattern Recognition*, 2008, pp. 1–8, available: vision.mas.ecp.fr/Personnel/iasonas/code/distribution.zip.
[20] M. Varma and R. Garg, "Locally invariant fractal features for statistical texture classification," in *Proc. Int. Conf. Comput. Vision*, 2007, pp. 1–8.
[21] E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," in *European Conf. Comput. Vision*, 2006, pp. 490–503.
[22] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods," *Int. J. Comput. Vision*, vol. 61, no. 3, pp. 211–231, 2005.
[23] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized PatchMatch correspondence algorithm," in *European Conf. Comput. Vision*, Sep. 2010.
[24] W. Qiu, X. Wang, X. Bai, A. Yuille, and Z. Tu, "Scale-space sift flow," in *Proc. Winter Conf. on Applications of Comput. Vision*. IEEE, 2014.
[25] L. Xu, Z. Dai, and J. Jia, "Scale invariant optical flow," in *European Conf. Comput. Vision*. Springer, 2012, pp. 385–399.
[26] J. Kannala, E. Rahtu, S. S. Brandt, and J. Heikkilä, "Object recognition and segmentation by non-rigid quasi-dense matching," in *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2008.
[27] M. Tau and T. Hassner, "Dense correspondences across scenes and scales," *Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 5, pp. 875–888, 2016.
[28] T. Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales,"," *J. of App. stat.*, vol. 21, no. 2, pp. 225–270, 1994.
[29] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proc. Conf. Comput. Vision Pattern Recognition*, 2011, pp. 529–534.
[30] A. Edelman, T. Arias, and S. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, pp. 303—353, 1998.
[31] P. Zhu and A. V. Knyazev, "Principal angles between subspaces and their tangents," *arXiv preprint arXiv:1209.0523*, 2012.
[32] k. Björck and G. H. Golub, "Numerical methods for computing angles between linear subspaces," *Mathematics of computation*, vol. 27, no. 123, pp. 579–594, 1973.
[33] R. Basri, T. Hassner, and L. Zelnik-Manor, "Approximate nearest subspace search with applications to pattern recognition," in *Proc. Conf. Comput. Vision Pattern Recognition*, June 2007.
[34] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.
[35] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. Conf. Comput. Vision Pattern Recognition*, vol. 2. IEEE, 2004.
[36] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *Int. J. Comput. Vision*, vol. 92, no. 1, pp. 1–31, 2001.
[37] E. Trulls, I. Kokkinos, A. Sanfeliu, and F. Moreno-Noguer, "Dense segmentation-aware descriptors," in *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2013, pp. 2890–2897.
[38] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Neural Inform. Process. Syst.*, 2005, pp. 1161–1168.
[39] A. Y. N. Ashutosh Saxena, Min Sun, "Make3d: Learning 3d scene structure from a single still image," *IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI)*, vol. 30, no. 5, pp. 824–840, 2009.
[40] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, 2005.
[41] K. Karsch, C. Liu, and S. B. Kang, "Depth extraction from video using non-parametric sampling," in *ECCV*, 2012.
[42] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. Int. Conf. Comput. Vision*, vol. 2, July 2001, pp. 416–423.
[43] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vision Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
[44] G. Levi and T. Hassner, "Latch: learned arrangements of three patch codes," in *Proc. Winter Conf. on Applications of Comput. Vision*. IEEE, 2016.
[45] T. Hassner and C. Liu, *Dense Image Correspondences for Computer Vision*. Springer, 2015.

Tal Hassner received a B.A. in computer science from the Academic College of Tel-Aviv Yaffo, 1998, and M.Sc. and Ph.D. degrees in applied mathematics and computer science from the Weizmann Institute of Science in 2002 and 2006, resp. He later completed a postdoctoral fellowship, also at the Weizmann Institute. In 2008 he joined the faculty of the Department of Mathematics and Computer Science, The Open University of Israel, where he is currently an Associate Professor. Since 2015, he is also a Senior Computer Scientist at the University of Southern California, Information Sciences Institute (ISI).

Shay Filosof Shay Filosof has received his BA degree in computer science from the Academic College of Tel-Aviv Yaffo in 2006 and is now completing his M.Sc in computer science from the Open University of Israel.

Viki Mayzels Viki Mayzels received the BSc degree (cum laude) and the MSc degree in electrical engineering from the Technion in 2001 and 2014, respectively.

Lihi Zelnik-Manor Lihi Zelnik-Manor is an Associate Progessor in the Faculty of Electrical Engineering in the Technion, Israel. Prior to the Technion, she worked as a postdoctoral fellow in the Department of Engineering and Applied Science in the California Institute of Technology (Caltech). She holds a PhD and MSc (with honors) in Computer Science from the Weizmann Institute of Science and a BSc (summa cum laude) in Mechanical Engineering from the Technion. Prof. Zelnik-Manors awards and honors include the Israeli high-education planning and budgeting committee (Vatat) scholarship for outstanding Ph.D. students, the Sloan-Swartz postdoctoral fellowship, the best Student Paper Award at the IEEE SMI'05, the AIM@SHAPE Best Paper Award 2005 and the Outstanding Reviewer Award at CVPR'08. She is also a recipient of the Gutwirth prize for the promotion of research and several grants from ISF, MOST, the 7th European R&D Program, and others.