Chapter 1
Dense Segmentation-aware Descriptors

Eduard Trulls, Iasonas Kokkinos, Alberto Sanfeliu, Francesc Moreno-Noguer

Abstract Dense descriptors are becoming increasingly popular in a host of tasks, such as dense image correspondence, bag-of-words image classification, and label transfer. However the extraction of descriptors on generic image points, rather than select geometric features, e.g. blobs, requires rethinking how to achieve invariance to nuisance parameters. In this work we pursue invariance to occlusions and background changes by introducing segmentation information within dense feature construction. The core idea is to use the segmentation cues to downplay the features coming from image areas that are unlikely to belong to the same region as the feature point. We show how to integrate this idea with dense SIFT, as well as with the dense Scale- and Rotation-Invariant Descriptor (SID). We thereby deliver dense descriptors that are invariant to background changes, rotation and/or scaling. We explore the merit of our technique in conjunction with large displacement motion estimation and wide-baseline stereo, and demonstrate that exploiting segmentation information yields clear improvements.

1.1 Introduction

Dense descriptors can be understood as replacing the convolution operations used in traditional image filterbanks [4, 23] with local descriptors, such as SIFT [20], that are better-suited to tasks such as image correspondence, classification, or labelling. Starting from [25], who demonstrated the merit of replacing sparse with

Eduard Trulls, Alberto Sanfeliu, Francesc Moreno-Noguer
Institut de Robòtica i Informàtica Industrial (UPC/CSIC), C/ Llorens i Artigas 4-6. 08028 Barcelona, Spain, e-mail: {etrulls,sanfeliu,fmoreno}@iri.upc.edu e-mail: etrulls,sanfeliu,fmoreno}@iri.upc.edu

Iasonas Kokkinos
Ecole Centrale Paris, Grande Voie des Vignes, 92295 Chatenay-Malabry, France, e-mail: iasonas.kokkinos@ecp.fr

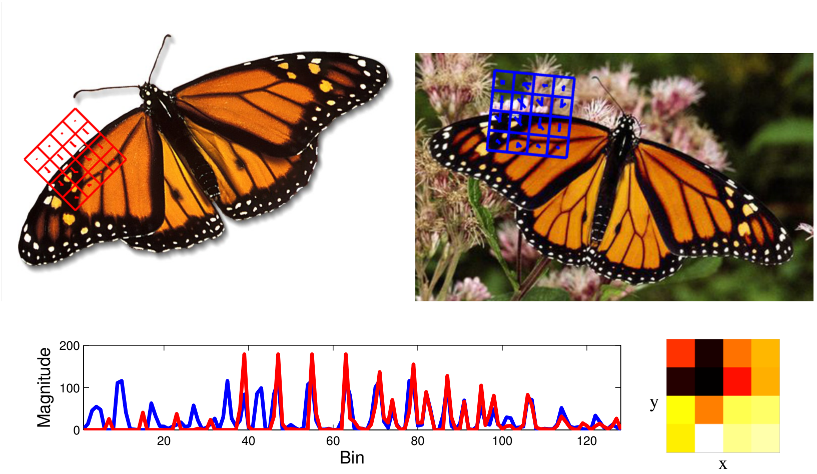


Fig. 1.1: Matching with background interference. We show images for two monarch butterflies, strikingly similar in appearance and pose (accounting for rotation), one over a white background and another in the wild. We plot the SIFT descriptors for two corresponding points found by hand, in red (left image) and blue (right image), close to object boundaries. On the bottom left we plot the descriptor values, ordered: orientation bin first, *x*-bin second, *y*-bin third. On the bottom right we plot the similarity between descriptors, averaged over the orientation bins (white is better, red is worse). Notice that the orientation histograms match well for the cells which lie on the foreground, but not for the background cells. This hurts correspondences around object boundaries.

regularly sampled SIFT descriptors for the task of image classification, a line of subsequent works [40, 18, 11] quickly established dense descriptors as a versatile, efficient, and high-performing front-end for vision tasks. In particular for dense correspondence, the seminal work of SIFT-flow [18] demonstrated that replacing the common ‘brightness constancy’ constraint of optical flow with a SIFT-based notion of similarity facilitates novel applications, such as scene correspondence or label transfer.

Using dense descriptors however requires rethinking how to achieve invariance. Sparse descriptor techniques [30, 20, 2, 24, 46, 34, 35] first use an interest point detector that finds stable scale- and rotation-invariant points, and then pool over coordinate systems adapted around these points to extract scale- and rotation-invariant descriptors. This strategy however is impossible in the dense setting, as image scale and orientation cannot be reliably estimated in most image areas: for instance defining scale is problematic around 1D edges, while defining orientation is problematic on flat image areas.

Several recent works have addressed the scale- and/or rotation-invariance issue in the dense setting, either by adapting global image registration techniques to local image descriptors [13, 15, 31, 19] or by searching for invariant subspaces, as in the Scale-Less SIFT (SLS) descriptor of [12]. We will elaborate on such techniques in Sec. 1.2, where we present in more detail the Scale-Invariant-Descriptor (SID) [13, 15] which our work builds on.

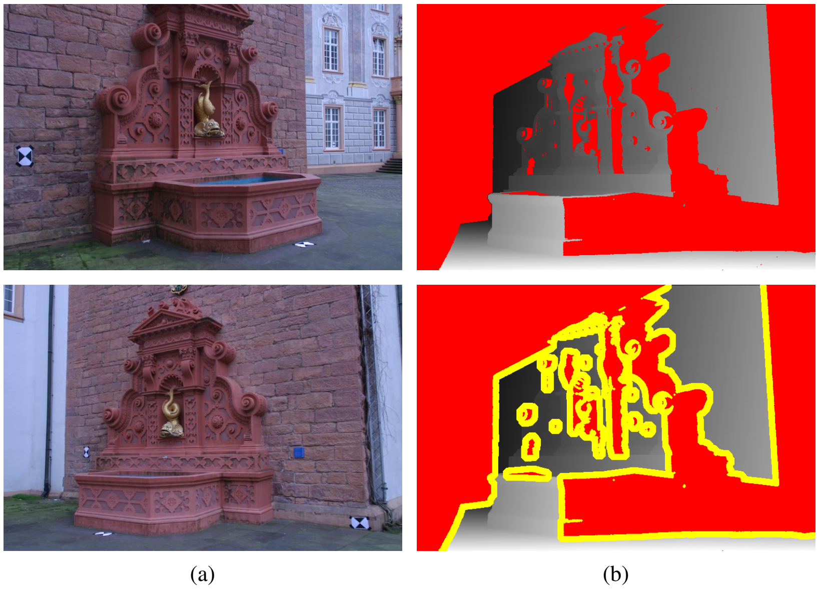


Fig. 1.2: Matching with occlusions. **(a)** Two images from wide-baseline stereo, with large occluded areas. **(b)** On the top row, we show the ground truth depth map, from the view farthest to the right. Occlusions, determined from ground truth visibility maps, are marked in red. On the bottom row, we dilate the occlusion map by 25 pixels and plot the result in yellow. We can think of these as the coordinates of the pixels that will be affected if we compute descriptors over circular areas of radius 25 pixels, as occluded areas creep into their domain—pixels closer to occlusion boundaries will suffer more. As the baseline increases, more and more pixels are beset by this problem: over 30% of visible pixels in this case.

In this Chapter we push this line of works a step further, achieving invariance to 'background changes'. In particular, some of the measurements used to construct a descriptor around a point may often come from distinct, independent objects, and will therefore differ in a new instantiation of the same point. One such case is illustrated in Fig. 1.1, where the object is roughly planar, but its background changes and therefore descriptors computed near its boundaries differ. Another case is shown in Fig. 1.2, where the scene stays the same, but due to its complicated geometry we have self-occlusions yielding again different descriptors across the two scenes. We will henceforth be using the term 'background variability' as a common term for these two, and other cases (e.g. occlusion by objects lying closer to the camera), where observations that do not stem from the same planar region as the feature point may affect the point's descriptor.

While this is an unavoidable problem when constructing local descriptors, we are aware of only a few works addressing it. In an early work in this direction[1], [36] introduced a local membership function to eliminate background information from (sparse) SIFT descriptors; the image gradients that are pooled for the per-

[1] We were unaware of this work when first publishing [43].

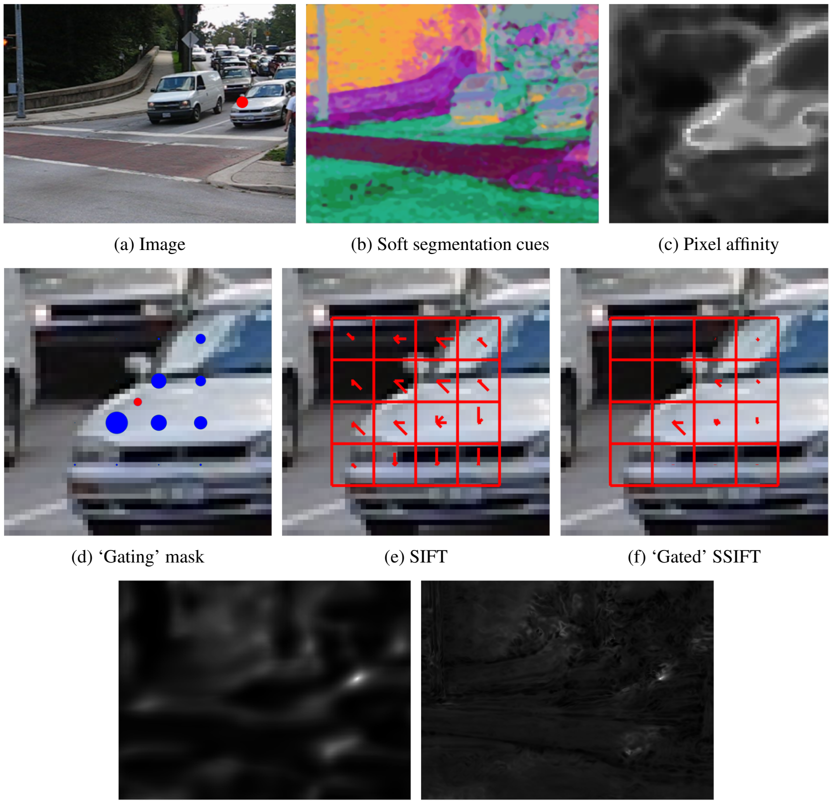
bin histograms were pre-multiplied by this function, resulting in a shunning of the background image gradients. In [40] the authors reported substantial performance improvements in multi-view stereo by treating occlusion as a latent variable in an iterative stereo matching algorithm: at every iteration, each pixel chooses a (discretized) orientation variable and discards the feature measurements coming from the half-disk lying opposite to it. When interleaved with successive rounds of stereo matching this yields increasingly refined depth estimates.

Interestingly in image processing the idea of blocking the interference between different regions is a long-established idea that can be traced back to the structure-preserving filtering used in nonlinear diffusion [26, 29], the bilateral filter [41], and the segmentation-sensitive normalized convolution of [28], while also bearing some resemblance to the self-similarity descriptor of [32]. In this light our work can be understood as bringing to the problem of descriptor construction the insight of blocking the flow of information across distinct regions by using the image to construct a local region membership function.

In particular, our aim is to eliminate, or at least reduce, the effects of background changes when extracting descriptors. For this we use a 'mid-level' segmentation module to reason about which pixels go together: as illustrated in Fig. 1.3, we incorporate segmentation information through a soft 'gating' mask that modulates local measurements, effectively shunning those parts of the image which apparently do not belong to the same object/surface as the center of the descriptor. In particular, we use segmentation cues (Fig. 1.3-b) to compute the affinity of a point with its neighbors (Fig. 1.3-c), and downplay image measurements most likely to belong to a different regions or objects (Fig. 1.3-d).

We argue that our approach has the following favorable aspects: firstly, it is fairly *general*. We apply it to two different descriptors (SIFT and SID), with three different segmentation techniques (Spectral Clustering [21], Generalized Boundaries [17], Structured Edge Forests [9]), for two different applications (motion and stereo). In all cases we demonstrate that the introduction of segmentation results in systematically better results over the respective baselines. Secondly, it is *simple*. It requires no training, and simply modifies the values of an existing feature descriptor. As such it can be used in any application that relies on descriptors. Thirdly, it incurs *minimal overhead*. The affinity masks can be computed and applied efficiently, in the order of a few seconds [17] or even a fraction of a second [9], for *dense* descriptors. Fourthly, our method has a *single parameter*, which can be used to adjust the 'hardness' of the masks. We fix it once and use it throughout our experiments—even across different applications.

In Sec. 1.2 we describe the SID descriptor, which achieves scale- and rotation-invariance in dense descriptor construction and serves as the starting point of our work. In Sec. 1.3, we present how we extract segmentation information from images in a manner that makes it straightforward to extract dense soft segmentation masks and, in turn, dense segmentation-aware descriptors. Lastly, we benchmark our descriptors on two different applications: large displacement motion estimation, and wide-baseline stereo. We demonstrate that the introduction of segmentation cues yields systematic improvements.



(a) Image

(b) Soft segmentation cues

(c) Pixel affinity

(d) ‘Gating’ mask

(e) SIFT

(f) ‘Gated’ SSIFT

(g) SIFT similarity

(h) SSIFT similarity

Fig. 1.3: We exploit segmentation information to construct feature descriptors that are robust to background variability. **(a)** Input image: we want to compute a descriptor for the pixel indicated by the red dot. **(b)** Segmentation embeddings, visualized in terms of their first three coordinates in RGB space; pixels with similar segmentation embeddings are likely to belong to the same region. **(c)** Affinity between the pixel represented by the red dot and its neighbors, measured in terms of the euclidean distance in embedding space. **(d)** A ‘gating’ mask that encodes the reliability, i.e. region similarity, of the locations used to compute the SIFT descriptor **(e)**. We can thus construct a Segmentation-aware SIFT (SSIFT) descriptor **(f)** by ‘gating’ the descriptor features with the mask. Figures **(g-h)** show the distance between the descriptors shown in **(e-f)** and respective dense SIFT/SSIFT descriptors over the whole image domain; we note that the SSIFT similarity function peaks more sharply around the pixel, indicating its higher distinctiveness.

1.2 SID: a Dense Scale- and Rotation-Invariant Descriptor

In several applications it can be desirable to construct a scale-invariant descriptor densely, for instance when establishing dense image correspondences in the presence of scale changes. In such cases scale selection is not appropriate, as it is only reliably applicable around a few singular points (e.g. blobs or corners).

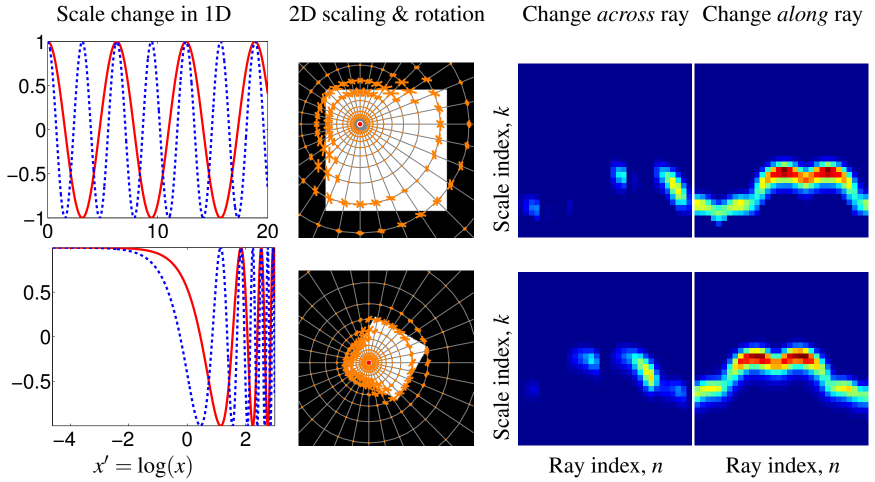


Fig. 1.4: Turning scaling into translations for 1D and 2D Signals: The left column demonstrates for a 1D signal how the logarithmic transformation $x' = \log(x)$ turns scaling into translation: the red-solid ($f(x) = \cos(x)$) and blue-dashed ($g(x) = \cos(2x)$) functions differ by a scale factor of two; the transformation $f'(x) = f(\log(x))$ delivers $f'(x) = \cos(\log(x)), g'(x) = \cos(\log(x) - \log(2))$, which differ by a translation. The next columns illustrate the same effect for 2D signals. The second column shows the descriptors computed on a point before and after scaling and rotating an image; the needle length indicates directional derivative magnitude. The next two columns show the respective magnitudes across and along the ray direction, demonstrating that image scaling and rotation are turned into translations. The point is arbitrary (i.e. not a corner/junction/blob center), and therefore scale selection around it would not be reliable, or even feasible.

We argue that one can instead adapt the Fourier Transform Modulus-based image registration technique [6, 27, 47] to the construction of descriptors and thereby guarantee scale- and rotation-invariance at any image point, withour requiring scale selection. Starting with an illustration of the technique for a one-dimensional signal, we will then briefly present how it applies to image descriptors; a more extensive presentation and evaluation is contained in [15], while we had originally presented this technique for sparse descriptors in [13].

We first consider describing a one-dimensional signal $f(x),\ x > 0$ in a manner that will not change when the signal is scaled as $f(x/a),\ a > 0$. Using the domain transformation $x' = \log(x)$ we can define a new function f' such that

$$f'(x') \doteq f(x), \quad \text{where } x' = \log x, \tag{1.1}$$

which is what we will be referring to as the 'logarithmically transformed' version of f; this is illustrated also in the left column of Fig. 1.4. For this particular transformation, dilating f by a will amount to translating f' by a constant, $\log(a)$:

$$f'(x' - \log(a)) = f(x/a), \tag{1.2}$$

meaning that we turn dilations of f to translations of f'.

We can thus extract a scale-invariant quantity based on the fact that if $g(x)$ and $G(\omega)$ are a Fourier transform pair, $g(x-c)$ and $G(\omega)e^{-j\omega c}$ will be a transform pair as well (by the shifting-in-time property). Defining $f_a(x') = f'(x' - \log(a))$, and denoting by $F_a(\omega)$ the Fourier Transform of $f_a(x)$ we then have:

$$\mathscr{F}_a(\omega) = \mathscr{F}_1(\omega)e^{-j\log(a)\omega}, \quad \text{or,} \tag{1.3}$$

$$|\mathscr{F}_a(\omega)| = |\mathscr{F}_1(\omega)|. \tag{1.4}$$

From Eq. 1.4 we conclude that changing a will not affect the Fourier Transform Modulus $|\mathscr{F}_a(\omega)|$ of f_a, which can thus be used as a scale-invariant descriptor of f.

As shown in the next columns of Fig. 1.4, a 2D scaling and rotation can similarly be converted into a translation with a log-polar transformation of the signal—and then eliminated with the FTM technique. The principle behind this approach is commonly used in tasks involving global transformations such as image registration [6, 47] and texture classification [27].

Adapting the FTM technique to the construction of local descriptors requires firstly a discrete formulation. We construct a descriptor around a point $\mathbf{x} = (x_1, x_2)$ by sampling its neighborhood along K rays leaving $\mathbf{x}$ at equal angle increments $\theta_k = 2\pi k/K,\ k = 0, \ldots, K-1$. Along each ray we use N points whose distances from $\mathbf{x}$ form a geometric progression $r_n = c_0 a^n$. The signal measurements on those points provide us with a $K \times N$ matrix:

$$h[k,n] = f\left[x_1 + r_n \cos(\theta_k), x_2 + r_n \sin(\theta_k)\right], \tag{1.5}$$

By design, image scalings and rotations of the image amount to translations over the radial and angular dimensions, respectively, of this descriptor. From the time-shifting property of the Discrete-Time Fourier Transform (DTFT) we know that if $h[k,n] \overset{\mathscr{F}}{\leftrightarrow} H(j\omega_k, j\omega_n)$ are a DTFT pair, we will then have:

$$h[k-c, n-d] \overset{\mathscr{F}}{\leftrightarrow} H(j\omega_k, j\omega_n)e^{-j(\omega_k c + \omega_n d)}, \tag{1.6}$$

therefore taking the absolute of the DTFT yields a scale- and rotation-invariant quantity.

We can apply the Fourier Transform only over scales to obtain a scale-invariant but rotation-dependent quantity (and vice-versa, for a rotation-invariant and scale-sensitive quantity). This can be useful in scenes with scaling changes but no rotations, where we would be discarding useful information. In our evaluations we will refer to the scale- and rotation-invariant descriptor as **SID** and to the scale-invariant but rotation-sensitive descriptor as **SID-Rot**.

Apart from a discrete formulation, we also need to preprocess the image so as to (a) discount illumination changes, (b) allow for efficient dense computation, and (c) account for sampling effects. Regarding (a), for invariance to additive illumination changes we use the *directional derivatives* of the signal along a set of orientations offset by the current ray's orientation (see e.g. Fig. 1.4 for the components along,

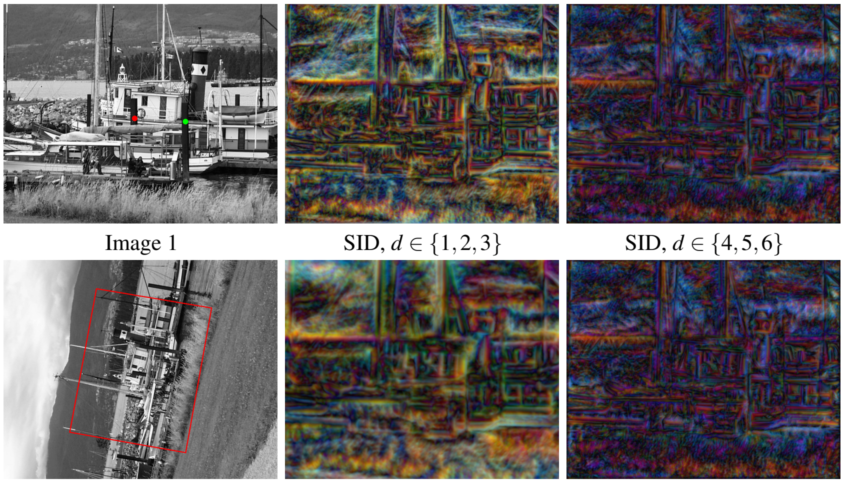


Image 1 SID, $d \in \{1,2,3\}$ SID, $d \in \{4,5,6\}$

Image 2 SID, $d \in \{1,2,3\}$ SID, $d \in \{4,5,6\}$

Fig. 1.5: Visualization of dense SID: the location of the image 1 within image 2 is indicated by the red box; the scaling transformation amounts to an area change in the order of four. We align the descriptors for the bottom row (red box) with the top row image after computing them, and visualize three of their dimensions in RGB space—demonstrating that they are effectively invariant.

and perpendicular to the ray). We extract features at H' orientations, preserving polarity, so that the effective number of orientation histogram bins is $H = 2 \cdot H'$. To discount multiplicative illuminaton changes we normalize the features to unit L_2 norm. For (b), memory- and time-efficient dense computation: we combine Daisy [40] with steerable filtering [10] and recursive Gaussian convolutions [7]. Finally for (c), dealing with sampling effects, as proposed in [13, 40] we use a 'foveal' smoothing pattern, with a smoothing scale that is linear in the distance from the descriptor's center.

In Fig. 1.5 we show the values of the lowest-frequency coefficients of densely computed descriptors on two images related by scaling and rotation. We see that the descriptor values are effectively invariant, despite a scaling factor in the order of 2.

Before proceeding we note that apart from SID, the Scale-Less SIFT (SLS) descriptor was introduced in [12], also with the goal of achieving scale-invariance for dense descriptors. The SLS approach is to compute a set of dense SIFT descriptors at different scales for each point, and project them into an invariant low-dimensional subspace that ellicits the scale-invariant aspects of these descriptors. SLS gives clearly better results than dense SIFT in the presence of scaling transformations, but SID can also be rotation-invariant and is substantially faster to compute. We include this state-of-the-art descriptor in our benchmarks in Sec. 1.4, and refer to Chapter XXX for further details on SLS construction.

1.3 Dense Segmentation-aware descriptors

In this Section we turn to discarding background variability by exploiting segmentation. Our goal is to construct feature descriptors that are contained within a single surface/object ('region' from now on). In this way changes in the background, e.g. due to layered motion, will not affect the description of a point in the interior of a region. Similarly, when a region is occluded by another region in front of it, even though we cannot recover its missing information, we can at least ignore irrelevant occluders.

Achieving this goal can benefit SIFT as well as any other descriptor, but its merit is most pertinent to SID. In particular, image scaling does not necessarily result in a cyclic permutation of the SID elements: the finest- and coarsest-level entries can change. As such the (circular) shifting relationship required to obtain the DTFT pair in Eq. 1.6 does not strictly hold. To remedy this issue SID typically uses large image patches and many rings, so that the percentage of points where this change happens eventually becomes negligible; this however limits SID's applicability, since background structures and occlusions can easily creep into its construction.

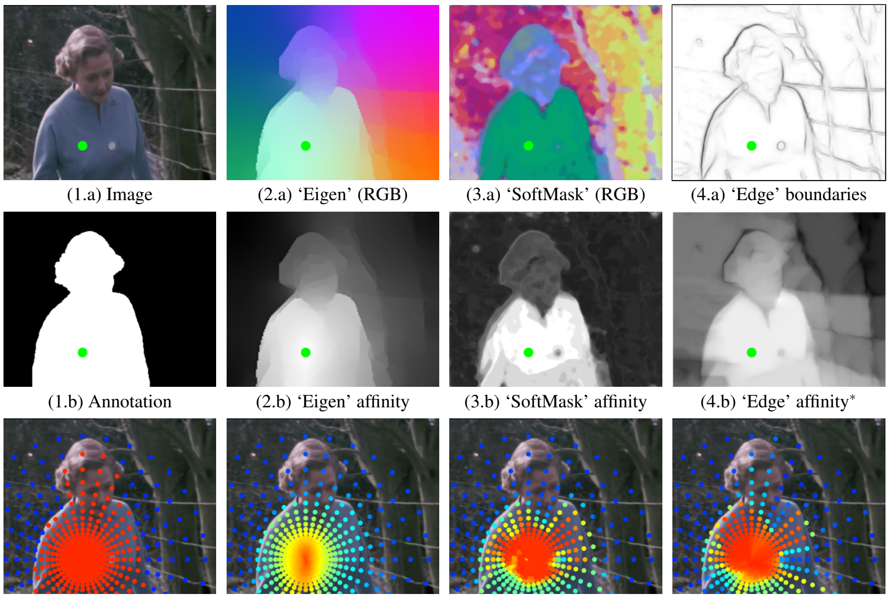
If we were able to use information only from the region containing a point, we could make a descriptor invariant to background changes: as shown in the first column of Fig. 1.6, given the support of the region containing a pixel we can identify the descriptor elements that come from different regions and set them to zero. However, since segmentation is far from solved we turn to algorithms that do not strongly commit to a single segmentation, but rather determine the affinity of a pixel to its neighbors in a soft manner. This soft affinity information is then incorporated into descriptor construction, in the form of a 'gating' signal.

Namely, when constructing a descriptor around a point $\mathbf{x}$ we measure an affinity $w[k,n] \in [0,1]$ between $\mathbf{x}$ and every other grid coordinate $\mathbf{x}[k,n]$, and multiply with it the respective measurements $\mathbf{d}$ extracted around $[k,n]$:

$$\mathbf{d}'[k,n] = w[k,n]\mathbf{d}[k,n]. \tag{1.7}$$

In Eq. 1.7 $\mathbf{d}[k,n]$ represents for SID the concatenation of the H polarized-smoothed derivatives at $[k,n]$ and for SIFT the respective 8-dimensional orientation histogram. Multiplying by $w[k,n]$ effectively shuns measurements which come from the background; as such, the descriptor extracted around a point is affected only by points belonging to its region and remains robust to background variability. As our results indicate, replacing $\mathbf{d}$ by $\mathbf{d}'$ yields noticeable performance improvements.

Having provided the general outline of our method, we now describe three alternative methods to obtain the affinity function $w[k,n]$ used in Eq. 1.7.



(1.a) Image (2.a) 'Eigen' (RGB) (3.a) 'SoftMask' (RGB) (4.a) 'Edge' boundaries

(1.b) Annotation (2.b) 'Eigen' affinity (3.b) 'SoftMask' affinity (4.b) 'Edge' affinity*

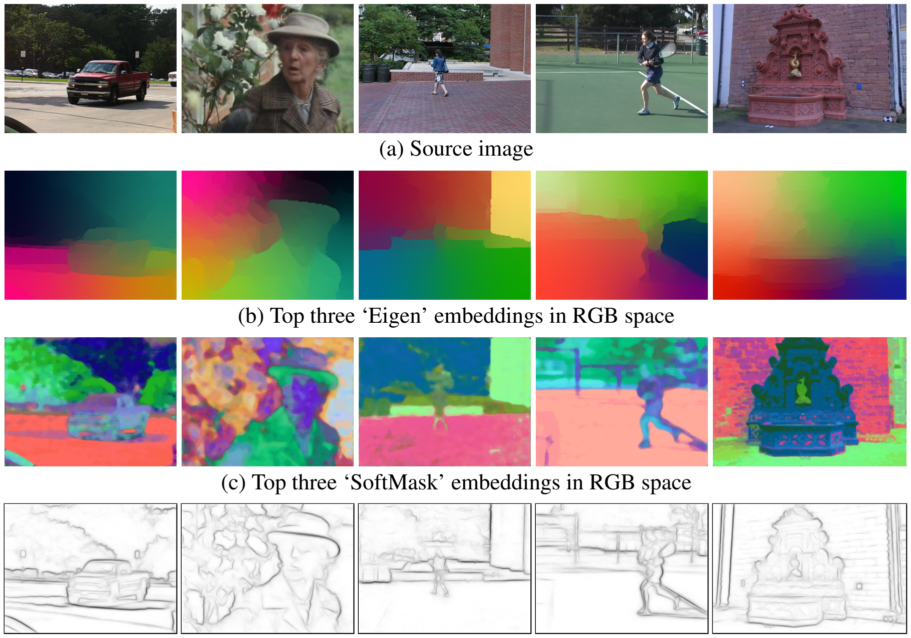
(1.c) 'Ideal' mask (2.c) 'Eigen' mask (3.c) 'SoftMask' mask (4.c) 'Edge' mask

Fig. 1.6: Segmentation-aware descriptor construction. **Column 1:** Given image (1.a) and a 'perfect' figure-ground segmentation (1.b), separating foreground and background measurements would be trivial: (1.c); grid points are marked in red if enabled and blue if disabled. This is unattainable—we propose alternative solutions based on different segmentation cues. **Columns 2-4:** Given segmentation cues ({2-4}.a), we can measure the 'affinity' between pairs of pixels: in ({2-4}.b) we show the affinity between the point represented by the green dot and the rest of the image. We use this per-pixel affinity to design 'gating' masks ({2-4}.c). We present procedures to leverage the Normalized-cut eigenvectors of [21] ('Eigen', column 2), the Generalized Pb soft segmentations of [17] ('SoftMask', column 3) and the Structured Forests boundaries of [9] ('Edge', column 4). Notice that (2.b) and (3.b) are for illustration: in practice we do this only for grid coordinates $\mathbf{x}[k,n]$ (pictured in the bottom row), smoothing the embeddings with filters of size increasing with n, prior to sampling. For the 'Edge'masks we use an affinity measure computed over the radial coordinates n, which does not lend itself to an image-based representation—for illustration purposes in (4.b) we show a distance transform instead [3]. Please refer to Sec. 1.3.3 for details.

1.3.1 *'Eigen' soft segmentations (gPb detector)*

First, we use the image segmentation approach of [21]: we treat the image as a weighted graph, with nodes corresponding to pixels and weights corresponding to the low-level affinity between pixels. The latter is obtained in terms of the *intervening contour* cue [33], which measures the presence of strong boundaries between two pixels. One can phrase the segmentation problem as a global optimization of the *normalized cut* [33] objective defined on the (discrete) labelling of this graph, which is NP-hard. However relaxing this problem yields a generalized eigenvector problem of the form:

$$(D-W)\mathbf{v} = \lambda D\mathbf{v}, \tag{1.8}$$



(a) Source image

(b) Top three 'Eigen' embeddings in RGB space

(c) Top three 'SoftMask' embeddings in RGB space

(d) Structured Forest 'Edge' boundaries

Fig. 1.7: Segmentation cues used in this work.

where $\mathbf{v} \in R^P$ is the relaxed solution for the P image pixels, W is an $P \times P$ affinity matrix encoding the low-level affinity between pixels, and D is a diagonal degree matrix with $D_{ii} = \sum_j W_{ij}$.

Even though this eigenvector problem can be solved exactly (albeit slowly), turning the computed eigenvectors into a segmentation is not obvious. Several post-processing techniques have been proposed (e.g. eigenvector-level clustering, by [33], or eigenvector-based features in [22]), but do not necessarily stem from optimizing the original problem. Instead, we propose to use the eigenvectors directly, as continuous *pixel embeddings*; namely every point $\mathbf{x}$ is mapped to a higher-dimensional space $\mathbf{y}$ where euclidean distances indicate the probability of two pixels falling in different regions. This is closer in spirit to 'Laplacian eigenmaps' [1].

In particular we construct $\mathbf{y}(\mathbf{x})$ by weighting the first $M = 10$ eigenvectors by a quantity dependent on their corresponding eigenvalues:

$$\mathbf{y}(\mathbf{x}) = \left[\frac{1}{\sqrt{\lambda_1}} \mathbf{v_1}((\mathbf{x})), \ldots, \frac{1}{\sqrt{\lambda_M}} \mathbf{v_M}((\mathbf{x})) \right]^T \tag{1.9}$$

so that lower-energy eigenvectors (global structures) have a stronger weight. Indicative examples of the first three dimensions of $\mathbf{y}(\mathbf{x})$ can be seen in Fig. 1.7-(b).

Based on the assumption that euclidean distance in $\mathbf{y}$ indicates how likely two points are to belong to the same region, we measure the descriptor-level affinity, w, between two points $\mathbf{x}, \mathbf{x}'$ as:

$$w = \exp\left(-\lambda \, \|y(\mathbf{x}) - y(\mathbf{x}')\|_2^2\right). \tag{1.10}$$

Here λ is a single scalar design parameter determining the sharpness of the affinity masks, which we set experimentally in Sec. 1.4. We show some pixel affinities (euclidean distances over the embedded space) and segmentation masks in Fig. 1.6. For simplicity, we use 'Eigen' to refer to the embeddings and masks we derive from this approach.

1.3.2 'SoftMask' soft segmentations (*Gb* detector)

As an alternative to the 'Eigen' embeddings, we also explore the soft segmentations employed in the *generalized* boundary detector *Gb* of [17]. Their method uses local color models, built around each pixel, to construct a large set of figure-ground segmentations. These segmentations are then projected onto a lower dimensional subspace through PCA. As before, we can take the top $M = 8$ components, which provides us again with low-dimensional pixel embeddings.

The main advantage of these features is that they are obtained at a substantially smaller computational cost, whereas building and solving for the eigenvectors in Eq. 1.8 is expensive for any but the smallest images. We refer to these embeddings, and their associated masks, as 'SoftMask'.

Fig. 1.7-(c) shows the *Gb* soft segmentations for several images. We notice that the 'SoftMask' embeddings have higher granularity than the 'Eigen' embeddings: on the one hand this makes them more noisy, but on the other hand this also allows them to better capture small features.

1.3.3 'Edge' boundaries (Structured Forest detector)

The third method we explore uses the state-of-the-art Structured Forest boundary detector of [9], which has excellent detection performance while operating at multiple frames per second. Unlike the previous two methods, this method does not provide an embedding, but rather measures the probability that two adjacent pixels may belong to different regions. In order to efficiently extend this measurement beyond adjacent pixels we adapt the 'intervening contour' technique of [33] to the descriptor coordinate system.

We start by sampling the boundary responses on the log-polar grid of SID; we use smoothing prior to sampling, so as to achieve scale-invariant processing. This sampling provides us with a boundary strength signal $B[k,n]$ that complements the descriptor features in Eq. 1.5. We then obtain the affinity function $w[k,n]$ in Eq. 1.7 in terms of the running sum of $w[k,n]$ along the radial coordinate k:

$$w[k,n] = \exp(-\lambda' \sum_{k'=0}^{k-d} B[k,n]). \tag{1.11}$$

We have introduced an additional quantity, d, in Eq. 1.11, which acts like a 'mask dilation' parameter. Namely, this allows us to postpone (by d) the decay of the affinity function w around region boundaries, thereby letting the descriptor profit from the shape information contained around boundaries. We have empirically observed that setting $d = 2$ or $d = 1$ yields a moderate improvement over $d = 0$. We refer to the segmentation masks computed in this manner as 'Edge', for convenience. Fig. 1.7-(d) shows some boundaries obtained with the Structured Forest detector.

1.4 Experimental evaluation

We consider two scenarios: video sequences with multi-layered motion, and wide baseline stereo. We explore the use of the different segmentation cues described in Sec. 1.3, and several dense descriptors (SID, Segmentation-aware SID, Dense SIFT, Segmentation-aware Dense SIFT, SLS, and Daisy). We use the 'S' prefix to indicate 'Segmentation-aware', so that for instance 'SSID' stands for our variant of SID.

1.4.1 Large displacement, multi-layered motion

In this experiment we estimate the motion of objects across time over the image plane—i.e. optical flow. This problem is usually formulated as an optimization over a function that combines a data term, that assumes constancy of some image property (e.g. intensity), with a spatial term that models the expected variation of the flow fields across the image (e.g. piecewise-smoothness).

Traditional optical flow methods rely on pixel data to solve the correspondence problem. We use SIFT-flow [18], which follows a formulation similar to that of optical flow but exploits densely sampled SIFT descriptors rather than raw pixel values. SIFT-flow is designed for image alignment, and unlike optical flow it is amenable not only to different views of the same scene, but also to different instances of scenes belonging to the same category—hence the use of dense SIFT, which has proven successful in image registration. Moreover, this approach can be applied to any feature descriptor that can be computed densely, and was previously paired with SLS in [12]. We use this framework to estimate the motion between pairs of frames, using different descriptors as features.

We test our approach on the Berkeley Motion Dataset (MOSEG) [5], which itself is an extension of the Hopkins 155 dataset [42]. The MOSEG dataset contains 10 sequences of outdoor traffic taken with a handheld camera, 3 sequences of people in movement, and 13 sequences from the TV series *Miss Marple*. All of them exhibit multi-layered motion. For these experiments we consider only the traffic sequences, as in many of the others the ‘objects’ in the scene (e.g. people) disappear or occlude themselves (e.g. turn around)—the dataset is geared towards long-term *tracking*. Ground truth object annotations (segmentation masks) are given for a subset of frames in every sequence—roughly one annotation every ten frames.

For each sequence, we pair the first frame with all successive frames for which we have ground truth annotations, yielding 31 frame pairs. The images are resized to 33%, in particular to permit comparison with SLS, which has high memory requirements. We design an evaluation procedure based on the segmentation annotations, proceeding as follows:

1. We compute flow fields for each descriptor type.
2. We use the flow estimates to warp the annotations for every frame $I_j, j > 0$ over the first frame (I_0).
3. We compute the overlap between the annotations for frame I_0 and the warped annotations for every frame $I_j, j > 0$ using the Dice coefficient [8] as a metric.

We consider Dense SIFT (DSIFT) [45], SLS, SID, and SSID with ‘Eigen’, ‘SoftMask’ and ‘Edge’ embeddings. We use SLS both in its original form [12] and a PCA variant developed afterwards: we refer to them as SLS-‘paper’ and SLS-PCA. A SLS descriptor is size 8256, whereas its PCA variant is size 528. The code for both was made publicly available by the authors.

For SID construction we use the dense implementation of [14]. We take $K = 28$ rays, $N = 32$ points per ray and $H' = 4$ oriented derivatives, preserving the polarity as in [40], so that the effective number of orientations is $H = 8$. We exploit the symmetry of the Fourier Transform Modulus to discard two quadrants, as well as the DC component, which is affected by additive lighting changes. The size of the descriptor is 3328 for SID and 3360 for SID-Rot. We refer to the publicly available code for further details[2].

For the SID-based descriptors we consider only their rotation-sensitive version, SID-Rot, as the objects in the MOSEG sequences do not contain significant rotations—discarding them in such a case entails a loss of information and a decrease in performance. We use the same parameters for both SID and SSID unless stated otherwise. We use this experiment to determine the values for the λ parameter of SSID of Eq. 1.10: $\lambda = 0.7$ for ‘Eigen’ and $\lambda = 37.5$ for ‘SoftMask’; and Eq. 1.11: $\lambda = 27.5$ for ‘Edge’.

Fig. 1.8 plots the results for every descriptor. Each bin shows the average overlap for all frame pairs under consideration. The results are accumulated, so that the first bin includes all frame pairs ($j \geq 10$), the second bin includes frame pairs with a displacement of 20 or more frames ($j \geq 20$), and so on. We do so to prioritize large displacements; the sequences have varying lengths, so that the samples are skewed towards smaller displacements. As expected, SSID outperforms SID, in particular for large displacements, which are generally correlated with large j. The best overall results are obtained by SSID-Rot with ‘SoftMask’ embeddings, followed by SSID-Rot with ‘Eigen’ embeddings—the ‘SoftMask’ variant does better, despite its

[2] https://github.com/etrulls/softseg-descriptors-release

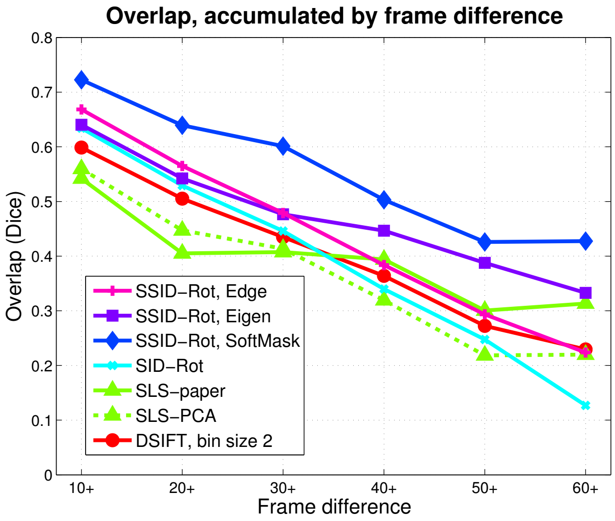


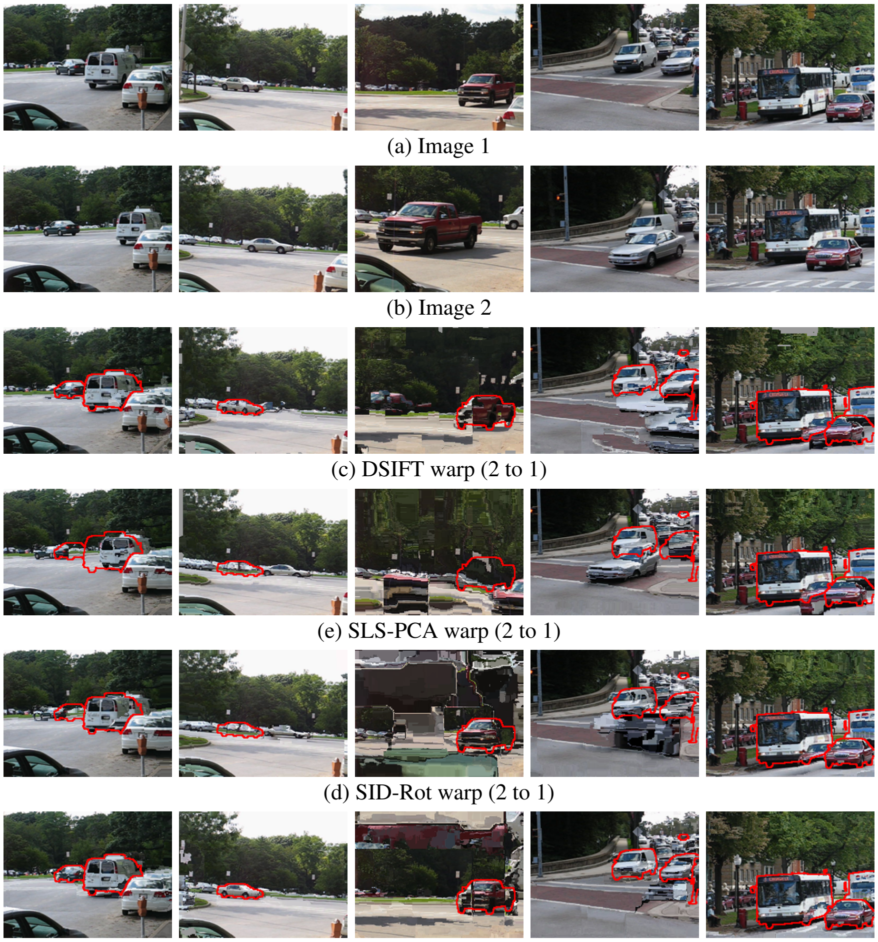
Fig. 1.8: Overlap results over the MOSEG dataset, for all the dense descriptors considered. Each bin shows the average overlap for all frame pairs under consideration. The results are accumulated, so that the first bin ('10+') includes all frame pairs, and subsequent bins ('j+') include frame pairs with a difference of j or more frames. For DSIFT we show only the results corresponding to the best scale.

reduced computational cost. The 'Edge' boundaries also provide a boost over the segmentation-agnostic SID—while they do not perform as well as the 'SoftMask' embeddings, they reduce the cost of extracting the segmentation cues even more drastically.

Additionally, we use the flow fields to warp each image I_j, over I_0. Some large displacement warps are pictured in Fig. 1.9—again, SSID outperforms the other descriptors.

1.4.2 Segmentation-aware SIFT

The application of soft segmentation masks over SID is particularly interesting because it alleviates its main shortcoming: fine sampling over large image areas to achieve invariance. But its success suggests that this approach can be applied to other standard grid-based descriptors—namely SIFT. We extend the formulation to SIFT's 4×4 grid, using the 'SoftMask' embeddings which give us consistently better results with SSID. Fig. 1.10 shows the increase in performance over four different scales. The gains are systematic, but as expected the optimal λ is strongly correlated with the spatial size of the descriptor grid. Fig. 1.11 displays the performance gains; note that the variability could be potentially accounted by the low



(a) Image 1

(b) Image 2

(c) DSIFT warp (2 to 1)

(e) SLS-PCA warp (2 to 1)

(d) SID-Rot warp (2 to 1)

(e) SSID-Rot ‘SoftMask’ warp (2 to 1)

Fig. 1.9: Large displacement motion with SIFT-flow, for some of the descriptors considered in this work. We warp image ‘2’ to ‘1’, using the estimated flow fields. The ground truth segmentation masks are overlaid in red—a good registration should bring the object in alignment with the segmentation mask. We observe that segmentation-aware variant SSID does best—particularly over its baseline, SID. Similar improvements were observed for SDSIFT over DSIFT.

number of samples (31 image pairs). This merits further study, in particular with regards to its application to the multiple scales considered in the construction of SLS descriptors.

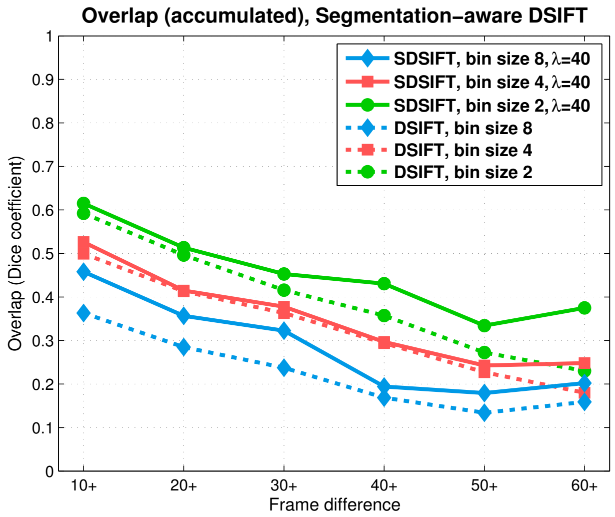


Fig. 1.10: Overlap results over the MOSEG dataset for segmentation-aware DSIFT and its baseline, at different scales.

1.4.3 Wide-baseline stereo

For a second experiment, we consider stereo reconstruction—i.e. the problem of computing a 3D representation of a scene given images extracted from different viewpoints. Stereo is one of the classical computer vision problems, and has been studied for several decades. While *narrow-baseline* stereo (usually defined as two cameras separated by a short distance, pointing in the same direction) is well-undertood, the same cannot be said for its *wide-baseline* counterpart.

Narrow-baseline stereo is often addressed with simple similarity measures such as pixel differencing, or block-wise operations such as the sum of square differences (SSD) or normalized cross-correlation (NCC). As the viewpoint increases, perspective distortion and occlusions become a problem and we cannot rely on these simple metrics—feature descriptors are more robust. Wide-baseline stereo has often been addressed as a multi-step process, using sparse matches as anchors or seeds [37, 48], which can result in gross reconstruction errors if the first matching stage is inaccurate. Dense correspondences are a preferable method.

Modern dense stereo algorithms use local features to estimate the similarity between points, and then impose global shape constraints to enforce spatial consistency. This problem is naturally formulated in terms of discrete energy minimization. For our experiments we use a set-up similar to [40]:

1. We discretize 3D space into $L = 50$ depth bins, from the reference frame of the camera furthest to the right.

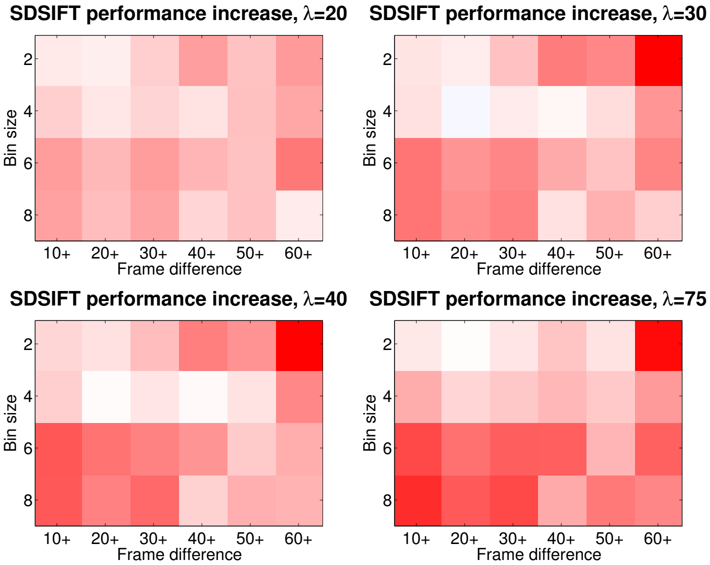


Fig. 1.11: Increase in average overlap for the segmentation-aware SIFT over its baseline, for several SIFT scales, difference in frames j (accumulated), and λ values. White signals no difference in overlap, with shades of red marking an increase (the largest increase in overlap is 0.14). For clarification, note the correspondence between the bottom left figure ($\lambda = 40$) and Fig. 1.10. As expected, high λ values produce more aggressive segmentation masks and more discriminating descriptors, but the optimal λ varies with the SIFT scale.

2. Given a calibrated stereo system, we compute the distance between every pixel in one image and all the possible matching candidates over the other image, subject to epipolar constraints, within the scene range.
3. We store the distance for the best match at every depth bin.
4. We feed the costs (distances) $N_w \times N_h \times L$, where N_w and N_h are the width and height of the image, to a global regularization algorithm, to enforce piecewise smoothness. Each pixel is assigned a label (depth bin) in $\mathscr{L} \in \{1, \ldots, L\}$.

For the last step we use Tree-Reweighted Message Passing [16] with Potts pairwise costs; i.e. a constant penalty if adjacent pixels are assigned different depth labels, and no penalty otherwise. We add an additional label with a constant cost, to model occlusions.

We use the wide baseline dataset of [38], which contains two multi-view sets of high-resolution images with ground truth depth maps. We consider the ‘fountain’ set, as it contains much wider baselines in terms of angular variation than the ‘herzjesu’ set, which exhibits mostly fronto-parallel displacements. As in [40], we use a much smaller resolution, in our case 460 × 308.

First, we evaluate the accuracy of each descriptor. We compute depth maps using the algorithm we just described, and evaluate the error on every visible pixel, using the ground truth visibility maps from [38], without accounting for occlusions. We

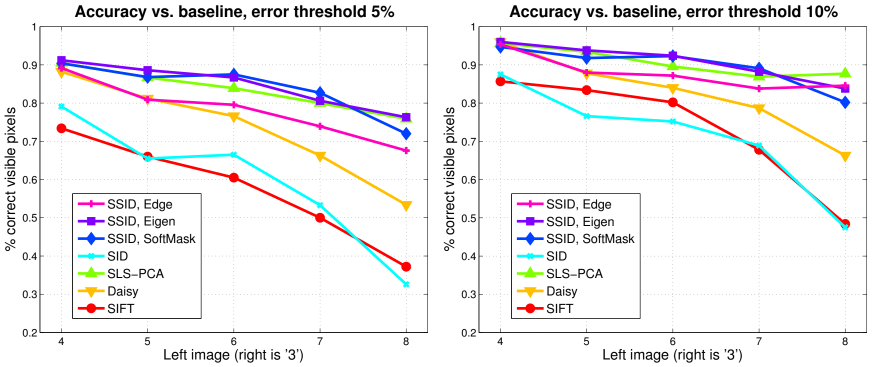


Fig. 1.12: Accuracy at different baselines, for visible pixels only, for two error thresholds (expressed as a fraction of the scene range). Occlusions are not taken into account.

consider DSIFT, SLS and Daisy, as well as SID and SSID. For DSIFT, SLS and Daisy we align the descriptors with the epipolar lines, to enforce rotation invariance, as in [40]. For SID and SSID we consider only the *fully invariant* descriptors, and omit this step. We use SLS-PCA rather than SLS-'paper', which has much lower dimensionality: matching descriptors is the costliest step in dense stereo and is correlated to descriptor size. We show the results on Fig. 1.12. Our SID-based segmentation-aware descriptors outperform the others, except for SLS—but our approach does not require rotating the patch.

Most of the performance gains on wide-baseline stereo reported in [40] stem not from the Daisy descriptor, but from their handling of occlusions. For this Tola et al. introduce a novel approach to latent occlusion estimation for iterative stereo reconstruction. Their technique exploits a set of binary masks, half-disks at different orientations, that disable image measurements from occluded areas. These are similar to our segmentation masks $\mathbf{w}[k,n]$, but binary (i.e. $\mathbf{w}' \in \{0,1\}$) rather than soft ($\mathbf{w} \in [0,1]$) and with a predetermined spatial structure. The most appropriate mask is determined on a per-pixel basis, using the current depth estimates around the pixel to prioritize masks that disable regions with heterogenous label distributions. Subsequent iterations apply the highest-scoring masks to the descriptors as in Eq. 1.7, dropping the measurements likely affected by occlusions from the similarity measure. A downside of this approach is that errors in the first iteration, which does not account for occlusions, can be hard to recover from.

The previous experiment did not take occlusions into account. In a second experiment, we pitch this state-of-the-art iterative technique against our segmentation-aware, single-shot approach. We let the Daisy stereo algorithm run for 5 iterations, and show the results in Fig. 1.13. The performance of SSID with 'Eigen' embeddings is comparable of superior to that of Daisy for most baselines—we achieve this in a single step, and without relying on calibration data to enforce rotation invariance. Additionally, note that we set the λ parameter of Eq. Eq. 1.10 on the motion experiments, and do not adjust them for a different problem: stereo. Fig. 1.14 shows

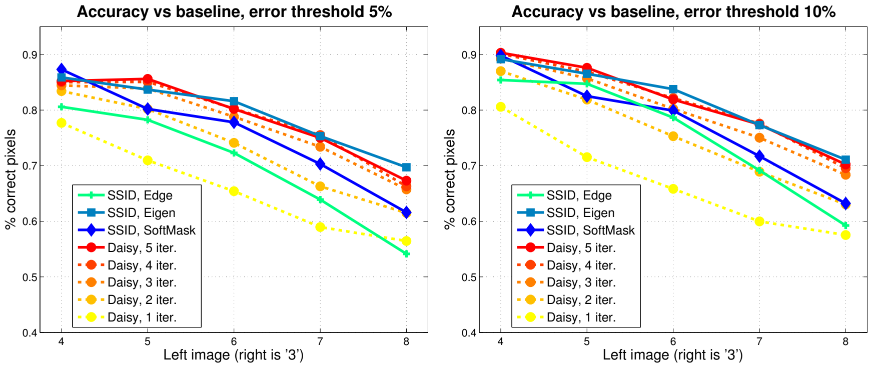


Fig. 1.13: Accuracy of the iterative approach to occlusion estimation of [40] and our segmentation-based, single-shot approach, at different baselines, for two error thresholds (expressed as a fraction of the scene range).

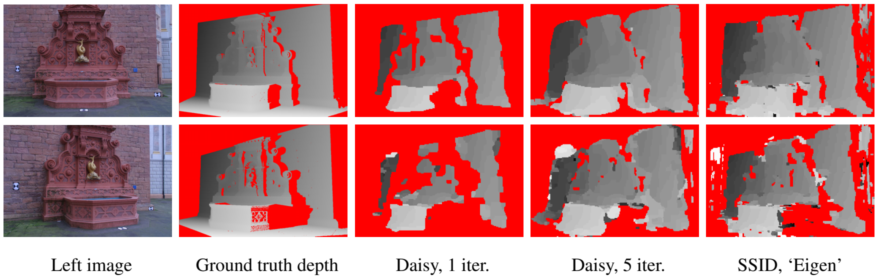


Fig. 1.14: **First column:** We compute depths maps for image pairs $\{5,3\}$ (top) and $\{7,3\}$ (bottom) of [38], over the 'right' viewpoint. Occluded pixels are marked in red. We show images 5 and 7—3 is pictured in Fig. 1.7). **Second column:** Ground truth depth maps. **Third and fourth columns:** iterations #1 and #5 for Daisy with latent occlusion estimation. **Fifth column:** Single-shot reconstruction for SSID with 'Eigen' embeddings.

the depth estimates at two different baselines (image pairs $\{5,3\}$ and $\{7,3\}$, reconstructed over 3)—the reference frame (3) is shown in Fig. 1.7.

1.4.4 Computational requirements

The cost of computing dense SIFT descriptors [45] for an image of size 320×240 is under 1 second (MATLAB/C code). SLS (MATLAB) requires ~21 minutes. SID (a non-optimized MATLAB/C hybrid) requires ~71 seconds. SSID requires ~81 seconds, in addition to the extraction of the masks. Note that for all the experiments in this chapter we compute the 'Eigen'/'SoftMask' embeddings at the original resolution (e.g. 640×480) before downscaling the images. The 'SoftMask' embeddings (MATLAB) require ~7 seconds per image, and the 'Eigen' embeddings

(MATLAB/C hybrid) ~280 seconds. Structured forest boundaries can be computed at multiple frames per second. The computational cost of matching two images with the SIFT-flow framework depends on the size of the descriptors, varying from ~14 seconds for SIFT (the smallest) to ~80 seconds for SID/SSID, and ~10 minutes for SLS-'paper' (the largest).

1.5 Summary and future work

In this work we propose a method to address background variability at the descriptor level, incorporating segmentation data into their construction. Our method is general, simple, and carries a low overhead. We use it to obtain segmentation-aware descriptors with increased invariance properties, which are of the same form as their original counterparts, and can thus be plugged into any descriptor-based application with minimal adjustments. We apply our method to SIFT and to SID descriptors, obtaining with the latter dense descriptors that are simultaneously invariant to scale, rotation and background variability.

We demonstrate that our approach can deal with background changes in large-displacement motion, and with occlusions in wide-baseline stereo. For stereo, we obtain results with SID comparable to the mask-based, state-of-the-art latent occlusion estimation of Daisy [40]—we do so without relying on calibration data to enforce rotation invariance; and in a single step, rather than with iterative refinements. While similar in spirit (both 'gate' the features to achieve invariance against background variability), our method is also applicable to the case where a *single image* is available.

Regarding future work, we can identify at least two directions for extending our work. Firstly, the segmentation-aware SID suffers from high dimensionality, but is likely very redundant. This shortcoming could be addressed with spectral compression and also with metric learning [39]. The latter proved able to both drastically reduce dimensionality problems, and also to increase the discriminative power of descriptors at the same time. Secondly, both of the applications where we assess our segmentation-aware descriptors involve shots of the same scene which differ in time, or viewpoint, but contain identical object instances. Our more recent work in [44] extends the segmentation-aware feature extraction technique to Histogram-of-Gradient (HOG) features and Deformable Part Models by relying on superpixels, but we believe this is only a starting point for leveraging segmentation in feature extraction for recognition—the introduction of segmentation in convolutional network classifiers is one of our current research directions.

Acknowledgements This work was partly funded by Rob-TaskCoop (DPI2010-17112), PAU+ (DPI2011-27510), MIPRCV (Consolider-Ingenio 2010, CSD2007-00018), EU projects ARCAS (FP7-ICT-2011-287617), MOBOT (FP7-ICT-2011-600796) and RECONFIG (FP7-ICT-600825), the ERA-net CHISTERA project VISEN (PCIN-2013-047), and grant ANR-10-JCJC-0205 (HiCoRe).

References

1. M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, 2003.
2. A. Berg and J. Malik, "Geometric blur for template matching," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001.
3. G. Borgefors, "Distance transformations in digital images," *Computer Vision, Graphics, and Image Processing*, 1986.
4. A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990.
5. T. Brox and J. Malik, "Berkeley motion segmentation dataset." http://lmb.informatik.uni-freiburg.de/resources/datasets/moseg.en.html, 2010.
6. D. Casasent and D. Psaltis, "Position, rotation, and scale invariant optical correlation," *Applied Optics*, 1976.
7. R. Deriche, "Using Canny's Criteria to derive a recursively implemented optimal edge detector," *International Journal of Computer Vision*, 1987.
8. L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, 1945.
9. P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," *Proceedings of the International Conference on Computer Vision*, 2013.
10. W. T. Freeman and E. H. Adelson, "The Design and Use of Steerable Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991.
11. B. Fulkerson, A. Vedaldi, and S. Soatto, "Localizing objects with smart dictionaries," in *European Conference on Computer Vision*, 2008.
12. T. Hassner, V. Mayzels, and L. Zelnik-Manor, "On SIFTS and their scales," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.
13. I. Kokkinos and A. Yuille, "Scale invariance without scale selection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
14. I. Kokkinos, M. Bronstein, R. Littman, and A. Bronstein, "Intrinsic shape context descriptors for deformable shapes," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.
15. I. Kokkinos, M. Bronstein, and A. Yuille, "Dense scale-invariant descriptors for images and surfaces," *INRIA Research Report 7914*, 2012.
16. V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.
17. M. Leordeanu, R. Sukthankar, and C. Sminchisescu, "Efficient closed-form solution to generalized boundary detection," *Proceedings of the European Conference on Computer Vision*, 2012.
18. C. Liu, J. Yuen, and A. Torralba, "SIFT flow: dense correspondence across difference scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.
19. K. Liu, H. Skibbe, T. Schmidt, T. Blein, K. Palme, T. Brox, and O. Ronneberger, "Rotation-invariant HOG descriptors using fourier analysis in polar and spherical coordinates," *International Journal of Computer Vision*, 2014.
20. D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 2004.
21. M. Maire, P. Arbeláez, C. Fowlkes, and J. Malik, "Using contours to detect and localize junctions in natural images," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
22. M. Maire, S. X. Yu, and P. Perona, "Object detection and segmentation from joint embedding of parts and pixels," *Proceedings of the International Conference on Computer Vision*, 2011.
23. S. Mallat, "Zero-crossings of a wavelet transform," *IEEE Transactions on Information Theory*, 1991.
24. 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*, 2005.

25. E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," in *Proceedings of the European Conference on Computer Vision*, 2006.
26. P. Perona and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990.
27. M. Porat and Y. Zeevi, "The Generalized Gabor Scheme of Image Representation in Biological and Machine Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1988.
28. X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proceedings of the International Conference on Computer Vision*, 2003.
29. L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, 1992.
30. C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997.
31. U. Schmidt and S. Roth, "Learning rotation-aware features: From invariant priors to equivariant descriptors," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.
32. E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
33. J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997.
34. K. Simonyan, A. Vedaldi, and A. Zisserman, "Descriptor learning using convex optimisation," *Proceedings of the European Conference on Computer Vision*, 2012.
35. K. Simonyan, A. Vedaldi, and A. Zisserman, "Learning local feature descriptors using convex optimisation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.
36. A. N. Stein and M. Hebert, "Incorporating background invariance into feature-based object recognition," in *7th IEEE Workshop on Applications of Computer Vision, IEEE Workshop on Motion and Video Computing*, 2005.
37. C. Strecha, T. Tuytelaars, and L. V. Gool, "Dense matching of multiple wide-baseline views," *Proceedings of the International Conference on Computer Vision*, 2003.
38. C. Strecha, W. von Hansen, L. V. Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
39. C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua, "LDA-hash: Improved matching with smaller descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.
40. E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.
41. C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the International Conference on Computer Vision*, 1998.
42. R. Tron and R. Vidal, "Hopkins 155 dataset." http://www.vision.jhu.edu/data.htm, 2007.
43. E. Trulls, I. Kokkinos, A. Sanfeliu, and F. Moreno-Noguer, "Dense segmentation-aware descriptors," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
44. E. Trulls, S. Tsogkas, I. Kokkinos, A. Sanfeliu, and F. Moreno-Noguer, "Segmentation-aware deformable part models," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
45. A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms." http://www.vlfeat.org, 2008.
46. S. Winder, G. Hua, and M. Brown, "Picking the best daisy," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
47. G. Wolberg and S. Zokai, "Robust image registration using log-polar transform," in *Proceedings of the IEEE International Conference on Image Processing*, 2000.
48. J. Yao and W. Cham, "3D modeling and rendering from multiple wide baseline images," *Signal Processing: Image Communication*, 2006.