

Research Article

EL: Local Image Descriptor Based on Extreme Responses to Partial Derivatives of 2D Gaussian Function

Jasna Maver  and Danijel Skočaj 

Faculty of Computer and Information Science, University of Ljubljana, SI-1000 Ljubljana, Slovenia

Correspondence should be addressed to Jasna Maver; jasna.maver@ff.uni-lj.si

Received 26 March 2019; Accepted 10 August 2019; Published 17 September 2019

Academic Editor: Rafal Zdunek

Copyright © 2019 Jasna Maver and Danijel Skočaj. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a two-part local image descriptor EL (Edges and Lines), based on the strongest image responses to the first- and second-order partial derivatives of the two-dimensional Gaussian function. Using the steering theorems, the proposed method finds the filter orientations giving the strongest image responses. The orientations are quantized, and the magnitudes of the image responses are histogrammed. Iterative adaptive thresholding of histogram values is then applied to normalize the histogram, thereby making the descriptor robust to nonlinear illumination changes. The two-part descriptor is empirically evaluated on the HPatches benchmark for three different tasks, namely, patch verification, image matching, and patch retrieval. The proposed EL descriptor outperforms the traditional descriptors such as SIFT and RootSIFT on all three evaluation tasks and the deep-learning-based descriptors DeepCompare, DeepDesc, and TFeat on the tasks of image matching and patch retrieval.

1. Introduction

Local image descriptors represent an important area of research in computer vision. Reliable local feature matching is required in numerous applications, for example, in emerging mobile visual search (MVS) [1], panorama stitching [2], image mosaicing [3], texture classification [4], partial-duplicate web image retrieval [5], wide-base stereo [6], and object recognition [7, 8]. Computer vision researchers have proposed many types of descriptors. We can divide them into handcrafted descriptors (SIFT [9], GLOH [10], SURF [11], BRIEF [12], KAZE [13], AG [14], Max-SIFT [15], and FBRK [16]) and those based on learning (BestDaisy [17], DeepCompare [18], DeepDesc [19], TFeat [20]).

Various benchmarks and different measures and protocols are available for local image descriptor evaluation [10, 21–25]. Recently, Balntas et al. [26] introduced HPatches, a new public benchmark for evaluation of local image descriptors. It includes a large body of patches obtained from image sequences of different scenes, captured under different lighting conditions and with large changes in viewpoint. The benchmark offers an open-source implementation of protocols for evaluating local image descriptors

on three different tasks: patch verification, image matching, and patch retrieval. In the same paper, the authors also show that a simple normalization of the handcrafted RootSIFT descriptor [27] can boost its performance to the level of deep-learning-based descriptors. The RootSIFT descriptor achieved the best result for the task of image matching and the second best for the task of patch retrieval. These results encouraged us to study the use of first- and second-order partial derivatives of the two-dimensional Gaussian function in defining a local image descriptor.

Local image descriptors based on higher-order image differentials, for example, the local jet [28], differential invariants [29], and steerable filters [30] have been studied before as low-dimensional point descriptors and gave poor results on tests [10]. Using higher-order image differentials in a different way, as histogrammed feature elements, proved to be much more successful. In [31], the authors propose a very simple algorithm based on the responses of a second-order bank of six Gaussian derivatives, which classifies an image location into one of seven Basic Image Features (BIFs): near-flat location, slope-like points, blob-like points (dark and light), line (dark and light), and saddle-like points. Jaccard et al. [32] applied BIFs for phase-contrast

microscopy image segmentation. In [33], the authors extend BIFs to oBIFs, by adding local image orientation to slope, line, and saddle-like points. Slope-like points are assigned a gradient orientation and line and saddle-like points an orientation in the direction perpendicular to the largest eigenvalue of the Hessian. Experiments demonstrate that a larger feature alphabet can lead to better performance and simpler encoding of visual words. In [17], the authors break up the descriptor extraction process into a number of modules and put these together in different combinations. The best descriptors are those with log-polar pooling regions and feature vectors constructed from rectified outputs of steerable quadrature filters. Unfortunately, these descriptors are of large dimensions; therefore, in [34], the authors add modules for dimension and dynamic range reduction.

In our approach, we use the first- and second-order partial derivatives of the two-dimensional Gaussian function, i.e., the edge and line detection filters. The proposed method forms a descriptor by histogramming and pooling magnitudes of the strongest responses to the steerable filters and uses an iterative adaptive thresholding to normalize the histogram values. The proposed descriptor achieves high mAP scores on the tasks of image matching and patch retrieval, and as such, it represents an attractive alternative to popular local descriptors.

The proposed approach is described in detail in the following sections. In the next section, we first describe how the optimal filter orientations are found and the corresponding magnitudes are computed. In the following section, we describe how the descriptor is formed. This is followed by the presentation of the experimental results and the final concluding section.

The proposed descriptor is publicly available at <https://github.com/REVAMJ/ELdescriptor>.

2. Extreme Responses to Edge and Line Filters

The proposed EL image descriptor is based on the first- and second-order partial derivative of the two-dimensional Gaussian function. The two partials have a nice property that they are orthogonal to each other.

2.1. Theory of Steerable Filters. Our algorithm is rooted in the theory of steerable filters, described in [30]. Let

$$g(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\left(\frac{x^2}{2\sigma^2} + \frac{y^2}{2\sigma^2}\right)\right) \quad (1)$$

be the two-dimensional Gaussian function and

$$g_x^{0^\circ} = \frac{\partial}{\partial x} g(x, y) = -\frac{x}{\sigma^2} g(x, y) \quad (2)$$

be its first partial derivative in the x direction (Figure 1). The same function rotated by 90° counterclockwise is

$$g_x^{90^\circ} = \frac{\partial}{\partial y} g(x, y) = -\frac{y}{\sigma^2} g(x, y). \quad (3)$$

Let $(\dots)^\theta$ represent the rotation operator such that for any function $f(x, y)$, $f^\theta(x, y)$ is $f(x, y)$ rotated through an

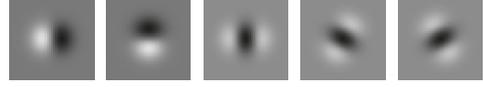


FIGURE 1: Basis filters $G_x^{0^\circ}$ and $G_x^{90^\circ}$ and $G_{xx}^{0^\circ}$, $G_{xx}^{60^\circ}$, and $G_{xx}^{120^\circ}$.

angle θ about the origin. According to the theory of steerable filters, g_x^θ can be synthesized by taking a linear combination of two basis filters $g_x^{0^\circ}$ and $g_x^{90^\circ}$:

$$g_x^\theta = \cos(\theta)g_x^{0^\circ} + \sin(\theta)g_x^{90^\circ}. \quad (4)$$

Let $G_x^{0^\circ} = g_x^{0^\circ} * I$ and $G_x^{90^\circ} = g_x^{90^\circ} * I$ where $*$ is the convolution operation and I is an intensity image. An image response to g_x^θ can simply be computed as

$$G_x^\theta = \cos(\theta)G_x^{0^\circ} + \sin(\theta)G_x^{90^\circ}. \quad (5)$$

The second partial derivative of $g(x, y)$ is equal to

$$g_{xx}^{0^\circ} = \frac{\partial^2}{\partial x^2} g(x, y) = \left(-\frac{1}{\sigma^2} + \frac{x^2}{\sigma^4}\right) g(x, y). \quad (6)$$

Computation of filter (6) in an arbitrary orientation requires two additional basis filters. We have chosen $g_{xx}^{60^\circ}$ and $g_{xx}^{120^\circ}$ (Figure 1). Let $G_{xx}^{0^\circ} = g_{xx}^{0^\circ} * I$, $G_{xx}^{60^\circ} = g_{xx}^{60^\circ} * I$, and $G_{xx}^{120^\circ} = g_{xx}^{120^\circ} * I$, then according to [30], an image response to g_{xx}^θ is computed as

$$G_{xx}^\theta = k_1 G_{xx}^{0^\circ} + k_2 G_{xx}^{60^\circ} + k_3 G_{xx}^{120^\circ}, \quad (7)$$

with interpolation functions k_j , $j = 1, 2, 3$, equal to $k_j = (1/3)[1 + 2\cos(2(\theta - \theta_j))]$ and with θ_1 , θ_2 , and θ_3 being equal to 0° , 60° , and 120° , respectively.

2.2. Filter Best Orientation. At some image location, the filter best orientation is the one that gives the strongest image response. The proposed descriptor will be composed of such responses. For the first-order partial, we compute $\partial G_x^\theta / \partial \theta = 0$ and obtain two filter orientations, one minimum and the other maximum:

$$\begin{aligned} \theta_{E_{\min}} &= \text{atan2}\left(-G_x^{90^\circ}, -G_x^{0^\circ}\right), \\ \theta_{E_{\max}} &= \text{atan2}\left(G_x^{90^\circ}, G_x^{0^\circ}\right). \end{aligned} \quad (8)$$

Three such examples are shown in Figure 2. The minimal and maximal image responses are equal in magnitude (see Figure 2(b)); therefore, it is sufficient to record just one of them; let $\theta_E = \theta_{E_{\max}}$ and $G_E = G_x^{\theta_E}$.

In the case of the second-order partial (7), computation of $\partial G_{xx}^\theta / \partial \theta = 0$ gives two extrema for the function basic period. The minimum is

$$\theta_{L_{\min}} = \frac{1}{2} \text{atan2}\left(\sqrt{3}\left(G_{xx}^{120^\circ} - G_{xx}^{60^\circ}\right), G_{xx}^{120^\circ} - G_{xx}^{60^\circ} - 2G_{xx}^{0^\circ}\right), \quad (9)$$

and the maximum is

$$\theta_{L_{\max}} = \frac{1}{2} \text{atan2}\left(-\sqrt{3}\left(G_{xx}^{120^\circ} - G_{xx}^{60^\circ}\right), -\left(G_{xx}^{120^\circ} - G_{xx}^{60^\circ} - 2G_{xx}^{0^\circ}\right)\right). \quad (10)$$

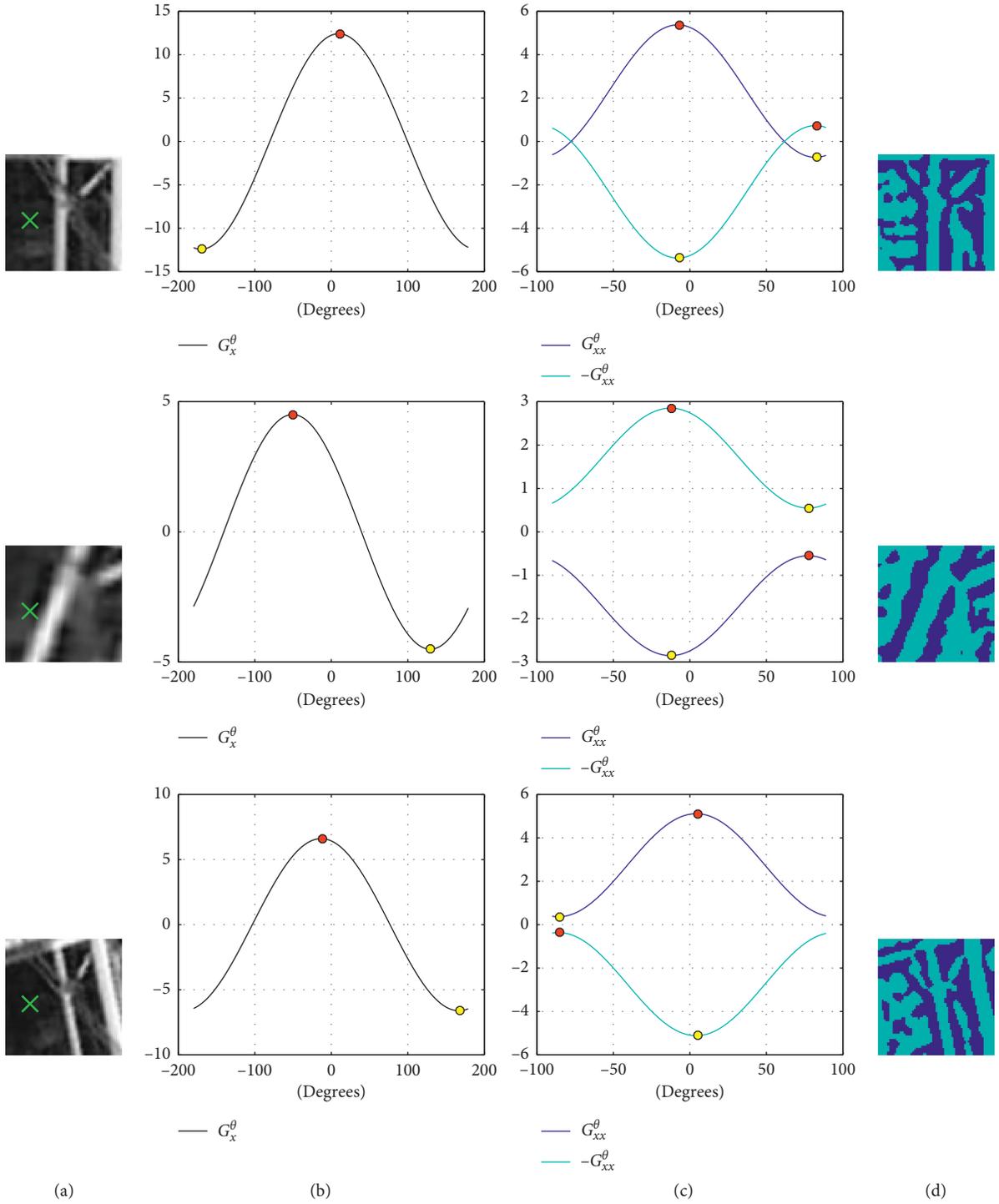


FIGURE 2: Extreme image responses. Maximal image responses are denoted by red dots and minimal by yellow dots. (a) The green cross marks the sample location at which image responses are computed. (b) Image response G_x^θ . The maximal and minimal image responses are equal in magnitude. (c) The blue curve represents G_{xx}^θ . The maximal and minimal image responses are different in magnitude and can be of opposite signs, both negative, or both positive. The cyan curve represents $-G_{xx}^\theta$. The image response $-G_{xx}^{\theta_{L_{\min}}}$ is denoted by a red dot. (d) Color marking of the patches in column (a); cyan color indicates the sample locations where $-G_{xx}^{\theta_{L_{\min}}} > G_{xx}^{\theta_{L_{\max}}}$ while blue represents the remaining locations.

In Appendix, we explain how they are computed. The minimal and maximal image responses are different in magnitude (see Figure 2(c)). Both can be positive or negative, or of opposite signs. If our goal is to record both

values, $G_{xx}^{\theta_{L_{\max}}}$ and $G_{xx}^{\theta_{L_{\min}}}$, then we have to distinguish between positive $G_{xx}^{\theta_{L_{\max}}}$, negative $G_{xx}^{\theta_{L_{\max}}}$, positive $G_{xx}^{\theta_{L_{\min}}}$, and negative $G_{xx}^{\theta_{L_{\min}}}$. Each of the four options requires its own representation (for example, in the case of orientation

binning, a histogram), which results in a long descriptor. However, we can choose to discard a piece of information. At the minimum $\theta_{L_{\min}}$, we compute, rather, the image response to the negative basis filters $-g_{xx}^{0^\circ}$, $-g_{xx}^{60^\circ}$, and $-g_{xx}^{120^\circ}$, which gives $-G_{xx}^{\theta_{L_{\min}}}$. We can interpret $G_{xx}^{\theta_{L_{\max}}}$ as a dark-line detection filter while $-g_{xx}^{\theta_{L_{\min}}}$ is a light-line detection filter. Figure 2(c) shows image responses obtained by the positive and negative basis filters. The procedure is as follows. From equations (9) and (10), we compute $\theta_{L_{\min}}$ and $\theta_{L_{\max}}$ and then from equation (7) the image responses $-G_{xx}^{\theta_{L_{\min}}}$ and $G_{xx}^{\theta_{L_{\max}}}$. The two image responses are compared (see Figure 2(d)) and only the larger is considered further, together with the corresponding filter orientation:

$$G_L = \max\left(-G_{xx}^{\theta_{L_{\min}}}, G_{xx}^{\theta_{L_{\max}}}\right),$$

$$\theta_L = \begin{cases} \theta_{L_{\min}}, & \text{if } -G_{xx}^{\theta_{L_{\min}}} > G_{xx}^{\theta_{L_{\max}}}, \\ \theta_{L_{\max}}, & \text{otherwise.} \end{cases} \quad (11)$$

Notice that G_L is always positive.

3. Descriptor Representation

The proposed descriptor is composed of two parts. The first part uses θ_E and G_E , and the second part uses θ_L and G_L . For simplicity, we name the new descriptor EL (Edges and Lines), because the filters used actually detect edges and lines. Descriptor construction requires the following three steps: orientation binning, orientation pooling, and histogram normalization.

3.1. Orientation Binning. The orientation binning for (θ_E, G_E) is performed in the same way as in SIFT. We quantize orientation into eight histogram bins corresponding to angles $\theta = -180^\circ, -135^\circ, -90^\circ, -45^\circ, 0^\circ, 45^\circ, 90^\circ$, and 135° . At each sample location, we construct a histogram vector of length eight. A filter response (θ_E, G_E) contributes to the two orientation bins adjacent to θ_E , whereby the value G_E is distributed linearly between the two bins. Thus, if θ_E lies between the two bins with angles θ_i and θ_j , then bin θ_i receives a contribution $G_E|\theta_j - \theta_E|/|\theta_j - \theta_i|$. Here, $|\theta_j - \theta_E|$ means the angular distance between the two angles (with wrap-around).

The orientation binning of θ_L is slightly different from the orientation binning of θ_E . Due to the symmetry of the filter $g_{xx}^{0^\circ} = g_{xx}^{180^\circ}$, we quantize orientation only into four orientation bins corresponding to angles $\theta = -90^\circ, -45^\circ, 0^\circ$, and 45° . At each sample location, we construct two histogram vectors of length four. One vector is intended for θ_L equal to $\theta_{L_{\min}}$ and one for θ_L equal to $\theta_{L_{\max}}$. Values of the vector components are determined by distributing G_L to the two bins adjacent to θ_L in the same way as describe above for G_E .

3.2. Orientation Pooling. The vectors from the previous stage are summed together spatially weighted with Gaussian weights according to their distance from the pooling centers. As recommended by Winder and Brown [17], we use 17 pooling centers and three different Gaussian weighting functions, as illustrated in Figure 3.

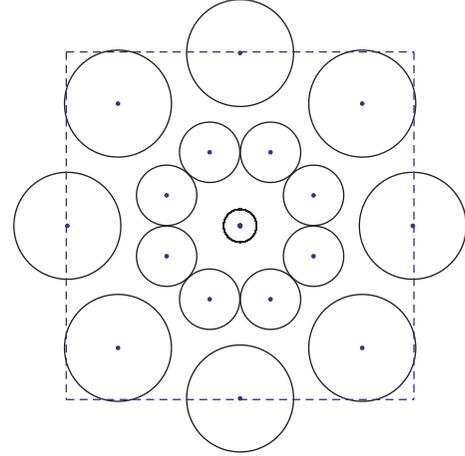


FIGURE 3: Polar arrangement of seventeen Gaussian summation regions. Circles indicate 1 standard deviation. For a 65×65 patch size, the standard deviations are $\sigma_0 = 3$, $\sigma_1 = 5.5$, and $\sigma_2 = 9.75$. The centers of the Gaussian summation regions lie on the rings with radii $r_0 = 0$, $r_1 = 14.5$, and $r_2 = 31.5$. All measures are in pixels.

After orientation summation, each pooling center is represented by one vector of length eight, representing image responses to the first-order partial derivative, and two vectors of length four, representing image responses to the second-order partial derivative. Vectors from all pooling centers are then concatenated into a common vector, i.e., a local image descriptor $D = [d_1, d_2, \dots, d_n]$ with $n = 272$.

3.3. Descriptor Normalization. The descriptor is normalized to reduce the effects of linear and nonlinear illumination changes. The proposed algorithm used by EL starts with an adaptive iterative thresholding of descriptor components. It repeats the following three steps ten times:

- (1) The average value of descriptor components is calculated, $\bar{d} = \sum_{i=1}^n d_i/n$
- (2) The threshold is calculated, $T = t_C \cdot \bar{d}$
- (3) Each descriptor component is thresholded to be no larger than T

The constant $t_C = 2.6$ in step 2 is determined experimentally. Then, we follow the approach used by RootSIFT [27], which uses a square root (Hellinger) kernel instead of the standard Euclidean distance to measure the similarity between SIFT descriptors.

- (4) The descriptor is normalized to have unit L_1 norm
- (5) Each descriptor component is represented by its square root

The graph in Figure 4 demonstrates the effect of the proposed iterative adaptive thresholding. The truncated descriptor components of the corresponding patches are significantly more similar than those in the case of normalization used by RootSIFT, indicated by the circles on the ordinate, or by the single-step adaptive thresholding, indicated by the results in iteration no. 1.

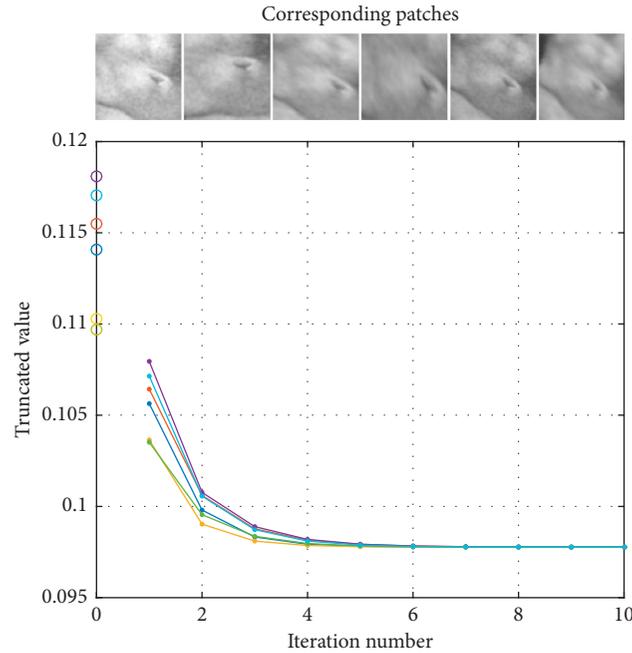


FIGURE 4: The final values of the truncated descriptor components (after step 5) for different numbers of adaptive thresholding iterations (steps 1 to 3) computed for six corresponding patches. The color circles on the ordinate represent the final values of the truncated descriptor components obtained when using RootSIFT normalization.



FIGURE 5: Examples of image pairs from full-image sequences in the HPatches dataset [26]: *i_fenis*, *i_dome*, *i_leuven*, *v_there*, *v_graffiti*, and *v_sunseason*.

4. Results and Discussion

The proposed approach is evaluated on the HPatches dataset (<https://github.com/hpatches/hpatches-dataset>) described in [26], which provides more than 2.5 million preextracted patches of size (65×65) pixels from 116 image sequences captured from different scenes. Changes in the images are due to changing scene lighting conditions and varying camera viewpoints (Figure 5). For each image sequence, patches are detected in the reference image and projected on the target images using the ground truth homographies. Detections are perturbed by increasing amounts of geometric noise, resulting in three patch sets of increased difficulty: easy, hard, and tough.

The authors also define the evaluation protocol and present its open-source implementation for fair comparison of local image descriptors on three different tasks: patch verification, image matching, and patch retrieval. In this

work, we strictly follow the proposed protocol and use the provided implementation for evaluation of our approach and comparison with the related work.

The proposed descriptor is computed for the Gaussian function $g(x, y)$ with $\sigma = 2.4$ pixels (equation (1)), threshold constant $t_C = 2.6$ (see Section 3.3), and descriptor footprint as shown in Figure 3.

4.1. Evaluation of the Proposed Descriptor. First, we evaluate the proposed descriptor denoted by EL and its individual parts denoted by E and L. Figure 6 shows the results in terms of the mean Average Precision (mAP(%)) for three different tasks (patch verification, image matching, and patch retrieval) on three different sets (easy, hard, and tough). Results are also shown for the postprocessed variants of descriptors +EL, +E, and +L obtained by applying ZCA whitening with clipped eigenvalues, followed by power law normalization and L_2 normalization [26].

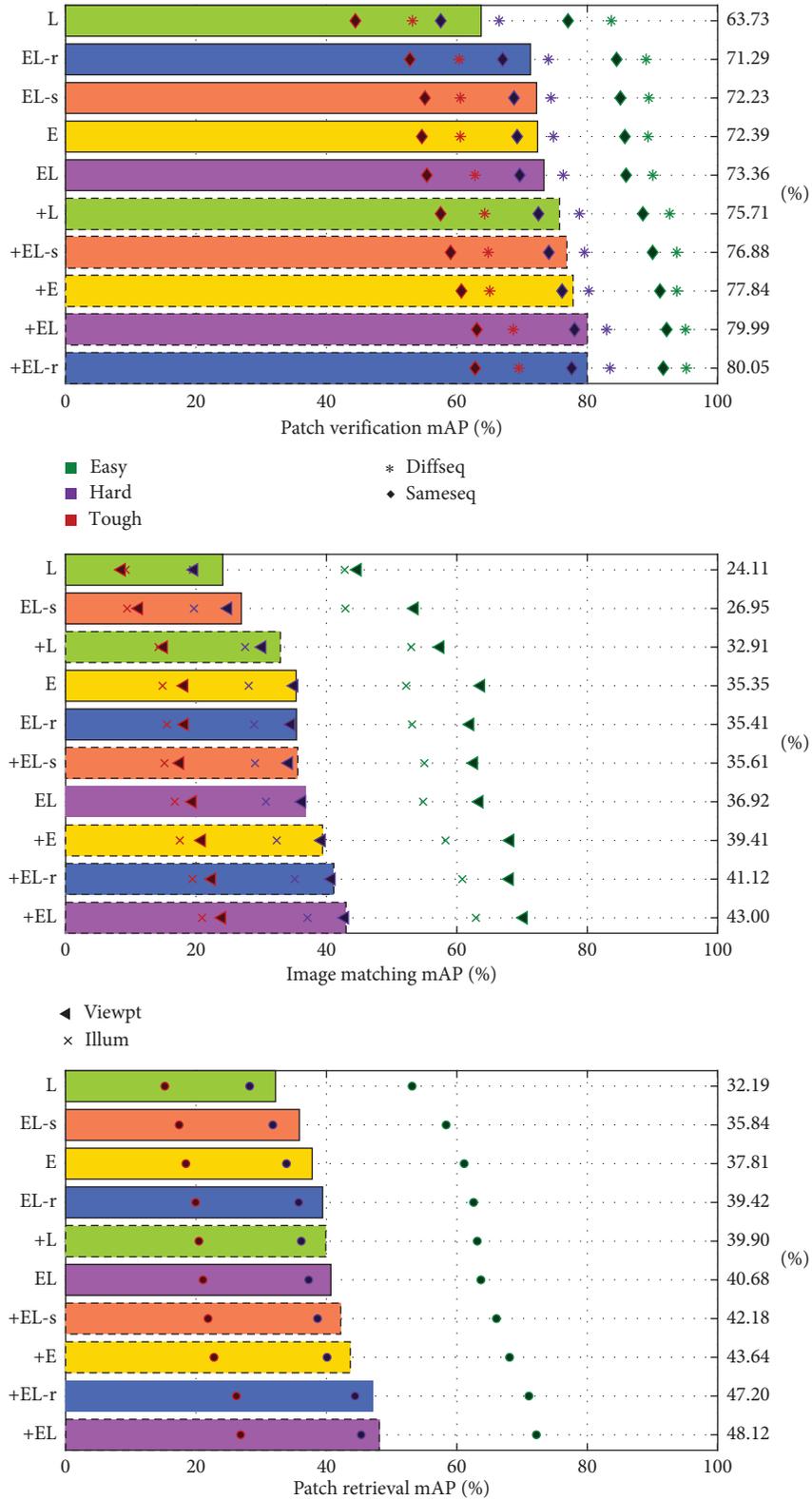


FIGURE 6: Results reported in mAP(%) on three various tasks: patch verification, image matching, and patch retrieval obtained by different variants of EL, E, and L descriptors. Suffixes “s” and “r” in the descriptor names refer to descriptor normalization used by SIFT and RootSIFT. Dashed bar borders and + indicate ZCA projected and normalized features.

We can observe that the combined EL descriptor produces higher scores than both single-component descriptors E and L. Descriptor L gives lower mAP scores than E. We see at least

two reasons for this. In the case of a blob, the filter g_{xx}^θ , used by L, gives an equal response for all filter orientations θ . This means that the filter best orientation is not well defined. In the

TABLE 1: Verification, matching, and retrieval results obtained for the EL, SIFT, and RootSIFT descriptor, and the deep-learning-based descriptors DC-S and DC-S2S [18], DDESC [19], and TF-M and TF-R [20]. All results are reported in mAP(%). The scores are given in pairs. The first value represents the score obtained with the basic method and the second with its ZCA variant [26]. Best result in a column is given in bold.

Descriptor	Verification		Matching		Retrieval	
	Basic, ZCA		Basic, ZCA		Basic, ZCA	
SIFT	65.12, 74.35		25.47, 32.76		31.98, 40.36	
RootSIFT	58.53, 76.70		27.22, 36.77		33.65, 43.84	
DC-S	70.04, 81.63		24.92, 31.65		34.84, 39.68	
DC-S2S	78.23, 83.03		27.69, 32.34		34.76, 38.23	
DDESC	79.51, 81.65		28.05, 35.44		39.83, 44.55	
TF-M	81.90, 82.69		30.61, 34.29		39.40, 40.02	
TF-R	81.92, 83.24		32.64, 34.37		37.69, 40.23	
EL	73.36, 79.99		36.92, 43.00		40.68, 48.12	

TABLE 2: Basic properties of the evaluated descriptors. The speed is measured in thousands of descriptors extracted per second.

Descr.	SIFT	RootSIFT	DC-S	DC-S2S	DDESC	TF-M	TF-R	EL	E	L
Dims	128	128	256	512	128	128	128	272	136	136
Patch Sz	65	65	64	64	64	32	32	65	65	65
CPU speed	2	2	0.3	0.2	0.1	0.6	0.6	0.4	1	0.6

case of a saddle, the minimal and maximal responses of g_{xx}^{θ} have the same magnitude. Due to different image deformations, the algorithm might choose in one situation the minimum while in the other the maximum. These situations increase errors in descriptor components. The differences in scores between the two-part descriptor +EL and one-part descriptor +E for the task of patch verification, image matching, and patch retrieval are 2.15, 3.59, and 4.52 percent point (pp), respectively.

We also wanted to verify the contribution of the iterative adaptive thresholding in the normalization step. We compared the proposed approach with the fixed thresholding, as used by the SIFT and RootSIFT descriptors. We therefore replaced our normalization approach, described by steps 1–5 in Descriptor Normalization, with normalization used by SIFT and RootSIFT (using the fixed threshold $T = 0.12$). We denote the obtained variants of the descriptor as EL-s and EL-r, respectively. We can notice that for the task of image matching by applying the Hellinger kernel, used by EL-r, we can improve mAP for 8.02 pp and by using the proposed iterative adaptive thresholding, used by EL, for additional 1.51 pp. For the postprocessed versions, the improvements are 5.51 and 1.88 pp. The iterative adaptive thresholding improves scores also for the task of patch retrieval. We can conclude that iterative adaptive thresholding used by EL is beneficial for the task of image matching and patch retrieval.

4.2. Comparison to Related Work. We compare the performance of the proposed descriptor with the previously published results on HPatches [26]. Table 1 shows scores obtained with two established handcrafted descriptors SIFT [9] and RootSIFT [27], and the deep-learning-based descriptors DC-S and DC-S2S [18], DDESC [19], and TF-M and TF-R [20].

For all three tasks, EL and its postprocessed variant improve the performance compared to SIFT and RootSIFT.

For the tasks of image matching and patch retrieval, EL also achieves better scores than deep-learning-based descriptors. The EL descriptor is particularly suitable for the task of image matching. Notice that the score 36.92% achieved by EL is higher even than the scores obtained by postprocessed variants of all other descriptors. Its postprocessed variant +EL achieves even better score, defeating all other descriptors for more than 6 pp. Similar to SIFT and RootSIFT, the EL descriptor is less appropriate for the task of patch verification.

We can expect that an approach based on pooling across different scales, in addition to spatial locations, i.e., the approach proposed by [34, 35], would improve the obtained scores even further. However, here we limit our evaluation to one scale only.

Table 2 shows the dimensionality, size of the measurement region in pixels, and extraction time of each descriptor. Note that EL, E, and L are implemented in Matlab; therefore, their time efficiency should be interpreted with caution; a more efficient implementation and code optimization could further speed up the descriptor extraction process.

5. Conclusions

We propose a two-part descriptor, named EL (Edges and Lines), based on the maximal image responses to the first- and second-order partial derivatives of the two-dimensional Gaussian function. The maximal image responses are calculated by using the steering theorems [30]. In this way, we complement the understanding of oBIFs [33]. The two parts of the proposed descriptor, E and L, are of equal size; each one contains 136 values.

To increase the descriptor robustness to nonlinear illumination changes and to increase the impact of less contrasting regions, the fixed thresholding of descriptor

components, as used by SIFT and RootSIFT, is replaced by an iterative adaptive thresholding.

The proposed descriptor was evaluated on HPatches benchmark for three different tasks, namely, patch verification, image matching, and patch retrieval. The post-processed variant of the EL descriptor obtained by applying ZCA whitening with clipped eigenvalues, followed by power law normalization and L_2 normalization, outperforms the postprocessed variant of the RootSIFT descriptor for all three tested tasks and the tested deep-learning-based descriptors for the tasks of image matching and patch retrieval.

One of our goals was also to explore the contribution of the second-order partial derivatives of the two-dimensional Gaussian function. Experimental results show improvements in scores for all three tasks. The largest improvement was obtained for the task of patch retrieval. Here, mAP is improved for 4.52 percent point. This is an important improvement, and we therefore recommend including both parts in a local image descriptor.

Overall, the results are very favorable especially for the tasks of image matching and patch retrieval, which are commonly required in many computer vision applications. What is also worth noting is that the proposed approach is very clear and is based on solid mathematical foundations of the theory of steerable filters. The EL descriptor is publicly available at <https://github.com/REVAMJ/ELdescriptor>.

Appendix

Equation (7) uses three interpolation functions $k_j = (1/3)[1 + 2 \cos(2(\theta - \theta_j))]$, $j = 1, 2, 3$ with θ_1, θ_2 , and θ_3 being equal to $0^\circ, 60^\circ$, and 120° , respectively. The three interpolation functions can be expressed in the following forms:

$$k_1 = \frac{1}{3} [1 + 2 \cos(2\theta)], \quad (\text{A.1})$$

$$\begin{aligned} k_2 &= \frac{1}{3} [1 + 2 \cos(2(\theta - 60^\circ))] \\ &= \frac{1}{3} [1 + 2 \cos(2\theta) \cos(120^\circ) + \sin(2\theta) \sin(120^\circ)] \quad (\text{A.2}) \\ &= \frac{1}{3} \left[1 - \cos(2\theta) + \sqrt{3} \sin(2\theta) \right], \end{aligned}$$

$$\begin{aligned} k_3 &= \frac{1}{3} [1 + 2 \cos(2(\theta - 120^\circ))] \\ &= \frac{1}{3} [1 + 2 \cos(2\theta) \cos(240^\circ) + \sin(2\theta) \sin(240^\circ)] \quad (\text{A.3}) \\ &= \frac{1}{3} \left[1 - \cos(2\theta) - \sqrt{3} \sin(2\theta) \right]. \end{aligned}$$

Using (A.1), (A.2), and (A.3) in equation (7), we obtain

$$\begin{aligned} G_{xx}^\theta &= \frac{1}{3} [1 + 2 \cos(2\theta)] G_{xx}^{0^\circ} \\ &+ \frac{1}{3} \left[1 - \cos(2\theta) + \sqrt{3} \sin(2\theta) \right] G_{xx}^{60^\circ} \quad (\text{A.4}) \\ &+ \frac{1}{3} \left[1 - \cos(2\theta) - \sqrt{3} \sin(2\theta) \right] G_{xx}^{120^\circ}. \end{aligned}$$

A filter orientation, which gives the strongest image response, is found by computing the first derivative of (A.4):

$$\begin{aligned} G_{xx}^{\theta'} &= \frac{2}{3} [-2 \sin(2\theta)] G_{xx}^{0^\circ} + \frac{2}{3} \left[\sin(2\theta) + \sqrt{3} \cos(2\theta) \right] G_{xx}^{60^\circ} \\ &+ \frac{2}{3} \left[\sin(2\theta) - \sqrt{3} \cos(2\theta) \right] G_{xx}^{120^\circ} \\ &= \frac{2}{3} \sin(2\theta) [-2G_{xx}^{0^\circ} + G_{xx}^{60^\circ} + G_{xx}^{120^\circ}] \\ &- \frac{2}{3} \sqrt{3} \cos(2\theta) [G_{xx}^{120^\circ} - G_{xx}^{60^\circ}] = \frac{2}{3} A \sin(2\theta) \\ &- \frac{2}{3} B \cos(2\theta), \quad (\text{A.5}) \end{aligned}$$

with A and B being equal to

$$\begin{aligned} A &= -2G_{xx}^{0^\circ} + G_{xx}^{60^\circ} + G_{xx}^{120^\circ}, \quad (\text{A.6}) \\ B &= \sqrt{3}G_{xx}^{120^\circ} - \sqrt{3}G_{xx}^{60^\circ}. \end{aligned}$$

To find the extrema, we solve $G_{xx}^{\theta'} = 0$, which gives us two solutions for the function base period:

$$\tan(2\theta_1) = \frac{B}{A}, \quad (\text{A.7})$$

$$2\theta_1 = \text{atan2}(B, A), \quad (\text{A.8})$$

$$\tan(2\theta_2) = \frac{-B}{-A}, \quad (\text{A.9})$$

$$2\theta_2 = \text{atan2}(-B, -A). \quad (\text{A.10})$$

To determine whether the extremum is a minimum or maximum, we examine the second derivative. For the minimum, the valid condition is $G_{xx}^{\theta''} > 0$, while for the maximum, $G_{xx}^{\theta''} < 0$. Let us compute $G_{xx}^{\theta''}$:

$$\begin{aligned} G_{xx}^{\theta''} &= \left(\frac{2}{3} \sin(2\theta)A - \frac{2}{3} \cos(2\theta)B \right)' \\ &= \frac{4}{3} \cos(2\theta)A + \frac{4}{3} \sin(2\theta)B. \quad (\text{A.11}) \end{aligned}$$

For the minimum, we obtain the condition $4 \cos(2\theta)A + 4 \sin(2\theta)B > 0$ or equivalently

$$\tan(2\theta) > -\frac{A}{B}; (B > 0 \text{ and } \cos(2\theta) > 0), \quad (\text{A.12})$$

$$\text{or } (B < 0 \text{ and } \cos(2\theta) < 0),$$

$$\tan(2\theta) < -\frac{A}{B}; (B > 0 \text{ and } \cos(2\theta) < 0), \quad (\text{A.13})$$

$$\text{or } (B < 0 \text{ and } \cos(2\theta) > 0),$$

and for the maximum

$$\tan(2\theta) < -\frac{A}{B}; (B > 0 \text{ and } \cos(2\theta) > 0), \quad (\text{A.14})$$

$$\text{or } (B < 0 \text{ and } \cos(2\theta) < 0),$$

$$\tan(2\theta) > -\frac{A}{B}; (B > 0 \text{ and } \cos(2\theta) < 0), \quad (\text{A.15})$$

$$\text{or } (B < 0 \text{ and } \cos(2\theta) > 0),$$

Before we continue, we note that $\cos(2\theta_1)$, with $2\theta_1$ given by equation (A.8), is positive when A is positive and $\cos(2\theta_1)$ is negative when A is negative. On the contrary, $\cos(2\theta_2)$, with $2\theta_2$ given by equation (A.10), is positive when A is negative and $\cos(2\theta_2)$ is negative when A is positive. Solution (A.8) satisfies equations (A.12), and (A.13); therefore, θ_1 represents a minimum while the solution (A.10) satisfies equations (A.14) and (A.15); therefore, θ_2 is a maximum.

Data Availability

The HPatches dataset used to support the findings of this study is publically available at <https://github.com/hpatches/hpatches-dataset> and described in the paper: HPatches: A benchmark and evaluation of handcrafted and learned local descriptors (DOI:10.1109/CVPR.2017.410). The code we wrote is also publically available at <https://github.com/REVAMJ/ELdescriptor>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Slovenian Research Agency (grant number P2-0214).

References

- [1] S. Bianco, D. Mazzini, D. P. Pau, and R. Schettini, "Local detectors and compact descriptors for visual search: a quantitative comparison," *Digital Signal Processing*, vol. 44, pp. 1–13, 2015.
- [2] M. Brawn and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73, 2007.
- [3] D. Ghosh and N. Kaabouch, "A survey on image mosaicing techniques," *Journal of Visual Communication and Image Representation*, vol. 34, pp. 1–11, 2016.
- [4] X. Yu, Y. Zhang, and H. Wang, "A novel local human visual perceptual texture description with key feature selection for texture classification," *Mathematical Problems in Engineering*, vol. 2019, Article ID 3756048, p. 20, 2019.
- [5] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian, "Spatial coding for large scale partial duplicate web image search," in *Proceedings of the 18th ACM International Conference on Multimedia*, pp. 511–520, New York, NY, USA, October 2010.
- [6] Y. Dou, K. Hao, Y. Ding, and M. Mao, "A mean-shift-based feature descriptor for wide baseline stereo matching," *Mathematical Problems in Engineering*, vol. 2015, Article ID 398756, p. 14, 2015.
- [7] K. Mele, D. Šuc, and J. Maver, "Local probabilistic descriptors for image categorisation," *IET Computer Vision*, vol. 3, no. 1, pp. 8–23, 2009.
- [8] L. Xie, J. Wang, W. Lin, B. Zhang, and Q. Tian, "Towards reversal-invariant image representation," *International Journal of Computer Vision*, vol. 123, no. 2, pp. 226–250, 2017.
- [9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [10] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.
- [11] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [12] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: computing a local binary descriptor very fast," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1281–1298, 2012.
- [13] P. Alcantarilla, A. F. Bartoli, and A. J. Davison, "KAZE features," in *Proceedings of the 12th ECCV*, pp. 214–227, Florence, Italy, October 2012.
- [14] R. Mandeljc and J. Maver, "AGs: local descriptors derived from the dependent effects model," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 503–514, 2019.
- [15] L. Xie, Q. Tian, and B. Zhang, "Max-sift: flipping invariant descriptors for web logo search," in *Proceedings of the 18th ACM International Conference on Multimedia*, pp. 5716–5720, Mountain View, CA, USA, June 2014.
- [16] L. Yang and Z. Lu, "A new scheme for keypoint detection and description," *Mathematical Problems in Engineering*, vol. 2015, Article ID 310704, p. 10, 2015.
- [17] S. Winder and M. Brown, "Learning local image descriptors," in *Proceedings of the IEEE Conference CVPR*, pp. 1–8, Rio de Janeiro, Brazil, July 2007.
- [18] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proceedings of the IEEE Conference CVPR*, pp. 4353–4361, Boston, MA, USA, June 2015.
- [19] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Proceedings of the IEEE ICCV*, pp. 118–126, Santiago, Chile, December 2015.
- [20] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks," in *Proceedings of the British Machine Vision Conference*, vol. 11, pp. 1–119, York, UK, September 2016.

- [21] H. Aanaes, A. L. Dahl, and K. S. Pedersen, "Interesting interest points," *International Journal of Computer Vision*, vol. 97, pp. 18–35, 2012.
- [22] J. Heinly, E. Dunn, and J.-M. Frahm, "Comparative evaluation of binary features," in *Computer Vision –ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., vol. 7573 of Lecture Notes in Computer Science, Springer, Florence, Italy, 2012.
- [23] S. Madeo and M. Bober, "Fast, compact, and discriminative: evaluation of binary descriptors for mobile applications," *IEEE Transactions on Multimedia*, vol. 19, no. 2, pp. 221–235, 2017.
- [24] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3D objects," *International Journal of Computer Vision*, vol. 73, no. 3, pp. 263–284, 2007.
- [25] J. L. Schoenberger, H. Hardmeier, T. Sattler, and M. Pollefeys, "Comparative evaluation of hand-crafted and learned local features," in *Proceedings of the Conference Computer Vision and Pattern Recognition*, Honolulu, Hawaii, July 2017.
- [26] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "HPatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proceedings of the IEEE Conference CVPR*, Honolulu, Hawaii, July 2017.
- [27] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proceedings of the IEEE Conference CVPR*, pp. 2911–2918, Providence, RI, USA, June 2012.
- [28] J. Koenderink and A. van Doorn, "Representation of local geometry in the visual system," *Biological Cybernetics*, vol. 55, pp. 376–375, 1987.
- [29] L. M. J. Florack, B. M. Ter Haar Romeny, J. J. Koenderink, and M. A. Viergever, "General intensity transformations and differential invariants," *Journal of Mathematical Imaging and Vision*, vol. 4, no. 2, pp. 171–187, 1994.
- [30] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, 1991.
- [31] L. D. Griffin and M. Lillholm, "Symmetry sensitivities of derivative-of-Gaussian filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1072–1083, 2010.
- [32] N. Jaccard, N. Szita, and L. D. Griffin, "Segmentation of phase contrast microscopy images based on multi-scale local basic image features histograms," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 5, no. 5, pp. 359–367, 2017.
- [33] M. Lillholm and L. Griffin, "Novel image feature alphabets for object recognition," in *Proceedings of the 19th IEEE Conference ICPR*, pp. 1–4, Tampa, FL, USA, December 2008.
- [34] S. Winder, G. Hua, and M. Brown, "Picking the best daisy," in *Proceedings of the IEEE Conference CVPR*, pp. 178–185, Miami, FL, USA, August 2009.
- [35] J. Dong and S. Soatto, "Domain-size pooling in local descriptors: Dsp-sift," in *Proceedings of the IEEE Conference CVPR*, pp. 5097–5106, Boston, MA, USA, June 2015.

