Extracting Deformation-Aware Local Features by Learning to Deform

Anonymous Author(s)
Affiliation
Address
email

Abstract

Despite the advances in extracting local features achieved by handcrafted and learning-based descriptors, they are still limited by the lack of invariance to non-rigid transformations. In this paper, we present a new approach to compute features from still images that are robust to non-rigid deformations to circumvent the problem of matching deformable surfaces and objects. Our deformation-aware descriptor leverages a polar sampling and a spatial transformer warping to provide invariance to rotation, scale and image deformations. We train the model architecture end-to-end by applying isometric non-rigid deformations to objects in a simulated environment as guidance to provide highly discriminative local features. The experiments show that our method outperforms state-of-the-art handcrafted, learning-based image, and RGB-D descriptors in different datasets with both real and realistic synthetic deformable objects in still images. The source code of our work will be made publicly available soon.

1 Introduction

High-quality matching of images taken from cameras in different poses and conditions remains one of the key challenges in several tasks. Tracking points as a camera or an object of interest is moving is crucial for finding and reconstructing objects [28, 29], camera localization and mapping [17, 35], registration [41, 36], and image retrieval [26, 38], to name a few. Extracting invariant and compact representations to describe the vicinity of a pixel has been one of the key forces in enabling high-quality matching. Despite the importance of invariance to varying illumination, viewpoint, and the distance to the object of interest, real-world scenes impose additional challenges. After all, we live in a non-rigid world, where beyond rigid surfaces, it is also populated by objects that might have different shapes due to deformations.

Over the past few decades, many descriptors have been proposed [18, 3, 6, 34, 40, 45, 42, 39, 22, 12]. These approaches can be roughly grouped based on the type of input information, such as intensity and/or depth images, and can be further divided into handcrafted and learning-based methods. While learning-based approaches, such as LogPolar [12], hold state of the art regarding affine transformations and local illumination changes, recent handcrafted techniques achieve better performance in the presence of non-rigid transformations by using depth data [25] or increasing the computational cost to compute the description [23]. Since depth data is not available in many contexts and applications, and the low processing time is highly recommended for low-level processing such as the estimation of local features, devising a descriptor that correctly and reliably establishes invariant features from corresponding points is of central importance to high-quality matching of images in real-world scenarios.

In this paper, we present a new end-to-end method to extract local features from still images that are robust to non-rigid deformations. Conversely to recent approaches that exploit a single network

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

branch to extract features from local patches or depth data, our approach is composed of an additional guidance branch component (spatial transformer non-rigid warper) that jointly learns context about the global deformation affecting the image.

Figure 1 depicts the rationale behind our method. For learning the contextual deformation information, we provide several views of deformed objects by applying isometric non-rigid transformations to the objects in a simulated environment as guidance to extract highly discriminative local deformation-aware features. Unlike non-rigid aware deformations descriptors like GeoBit [25], our approach does not require depth data to produce descriptors robust to non-rigid deformations and it is notably faster than DaLI [23] while holding the best performance.

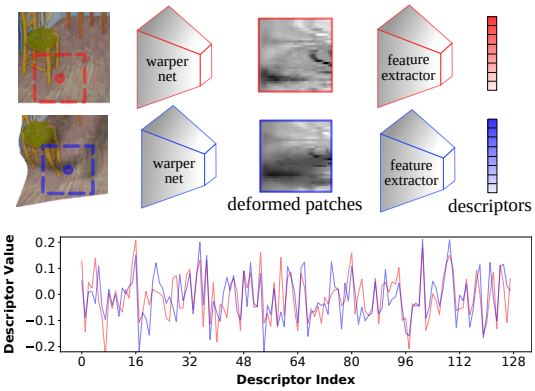


Figure 1: **Learning to deform and extract features**. Our network learns meaningful deformations during training that lead to improved matching performance, as shown in the experiments. Notice that the descriptor's signatures before and after a local deformation (red and blue curves) are highly correlated in this real matching example.

Although many descriptors still disregard non-rigid transformations, in recent years, that assumption has been dispelled, partly due to the increasing popularity in the use of RGB-D sensors to capture properties that characterize 3D objects such as geodesic distances. On the flip side, RGB-D sensors are still much less prevalent than RGB cameras. In our nonrigid-world with flexible humans, animals, tissues, and leaves where high-resolution RGB cameras are ubiquitous, the ability to create discriminative and deformation-invariant descriptors using a single picture plays a central role in achieving high-quality matching in real-world scenarios. In this work, we demonstrate that our end-to-end architecture outperforms state-of-the-art handcrafted and learning-based image and RGB-D descriptors in matching scores on different benchmark datasets of real deformable objects, as well as with realistic application scenarios on content-based object retrieval and tracking.

2 Related work

Handcrafted and learning-based feature extraction. Traditionally, image descriptors are aimed to providing rich and invariant representations of objects observed from different viewing conditions. Two notable handcrafted approaches to describe keypoints on still images are SIFT [18] and ORB [34]. These methods use local gradients and dominant orientation estimation to extract visual features with scale and rotation invariance. Different extensions of these descriptors have been proposed to images in some particular manifolds such as BRISKS [13] and SphORB [47] for spherical images. Recently the advent of convolutional neural networks boosted the development of several end-to-end learned descriptors [45, 42, 39, 22, 12, 19, 30]. These descriptors often outperform handcrafted counterpart on classical image matching benchmarks such as UBC Phototour [5] and HPatches [2]. Yet, the vast majority of existing local descriptors are approximately invariant to affine image transformations, often disregarding images of deformable surfaces, which are the main feature of our descriptor.

Deformation-aware reconstruction and matching. Although most works on local descriptors are designed focusing on affine image transformations, some works considered non-rigid surfaces and deformable objects. In DeepDeform [4], for instance, the authors propose to estimate non-rigid point-wise RGB-D correspondences by a data-driven approach that is then used by a non-rigid 3D reconstruction framework. Although their method works well for fast motions and non-rigid surface tracking, it has a heavy computational burden for establishing correspondences for typical amounts of keypoints from commonly used detectors. Also, the method does not generate a descriptor that can be used for other related vision tasks. Simo-Serra *et al*. proposed the DaLI descriptor [37], where image patches are converted to a 3D surface in order to encode features robust to non-rigid deformations and illumination changes. Despite achieving high matching performance, DaLI suffers from high computational and storage requirements (descriptor dimensionality). Recently, DeformNet [29] is

designed to infer non-rigid shapes from still images. The network architecture is composed of a 2D detection branch and a depth branch. The detection and depth branches are subsequently merged by the shape branch, which uses a pinhole camera model to re-project the shape to 3D. In the same context, GeoBit [25] takes advantage of geodesics from objects surface to compute isometric-invariant features. The method uses RGB-D data combined with the Heatflow [7] method to encode visual features in a binary string that are invariant to deformations. However, GeoBit requires accurate depth information and only works with RGB-D images. Our method, in its turn, only requires monocular images, is faster to compute, and provides a better matching performance, as demonstrated in our experiments.

Spatial attention mechanisms and matching. Spatial transformer networks, introduced by the seminal work of Jaderberg *et al*. [15], allowed differentiable warping operations to be used directly with existing end-to-end network architectures. The core idea is the learning of an attention mechanism (named Localization network) that is parameterized to represent the image transformation of interest, *e.g*., affine, homography or a thin-plate-spline warp. They have been applied to several tasks including image classification [15], semantic alignment [33], geometric matching [32] and descriptors [12]. Spatial attention mechanisms have also been extended into more generic settings via deformable kernels [9], to learn kernel offsets in network layers operators to define deformable convolution networks. More recently, the end-to-end joint detector and descriptor ASLFeat [20] used deformable convolutions to increase the network's expressiveness, which shares some concepts with our method. However, the deformable kernels employed by ASLFeat considers high-level feature maps, and it does not model deformations explicitly. Its strategy results in earlier layers not being aware of low level image deformations, usually present in geometric transformations. Moreover, the method is not robust to rotation changes. Our method, for its turn, leverages two carefully designed spatial transformer networks to sample and guide the model to provide invariance to deformations and, to the best of our knowledge, is the first end-to-end learning method to explicitly tackle non-rigid deformations with a carefully designed spatial attention mechanism.

3 Deformation-Aware Network Architecture

Our DEformation-Aware Local descriptor (DEAL) has been designed to jointly learn to rectify and extract discriminative and invariant features from local regions. The proposed architecture is able to handle deformations and perspective distortions of sampled patches in the vicinity of the keypoint. It also attenuates noise in keypoint attributes which improves the local description performance and robustness to image transformations. We designed our network as a hybrid approach, in the sense that it first extracts dense features and the final output is a local descriptor for each keypoint, unlike both trends of describing local patches like HardNet [22] or extracting dense features such as SuperPoint [10] and R2D2 [31]. Our method combines the best of these two trends: i) the robustness to occlusion, strong illumination, and perspective changes of local patch-based descriptors that work directly with low-level image information; and ii) the capability of dense methods in encoding local semantics from the image to disambiguate hard pairs. Additionally, our model decouples the problem of learning distinctive descriptors robust to deformations in two complementary tasks, *i.e*., first estimating a warp with a spatial attention mechanism and then describing the sampled local patches using the estimated non-rigid warp. Figure 2 outlines a schematic representation of the model.

3.1 Mid-level feature extraction

For the extraction of mid-level image features that encode local deformation information, we employ the ResNet-34 [14] and truncate it at the sixth convolutional block. The motivations to process features from ResNet instead of directly sampling local patches over the image are twofold: First, the mid-level feature map provided by ResNet features allows the network to learn to encode local deformations from contextual cues such as local texture statistics, illumination and viewpoint changes; Secondly, the receptive field of our local feature descriptor is not tied to a fixed support region size and sampling pattern like traditional patch-based local descriptors.

Strong perspective and deformation changes drastically change the optimal support region sizes for the same keypoint on different images; thus, we present an architecture that is aware of these local deformations by design, since the dense feature map allows such information to flow to the non-rigid warper module. The ResNet-34 head outputs a feature map $\mathbf{X} \in \mathbb{R}^{H/8 \times W/8 \times 128}$ by performing

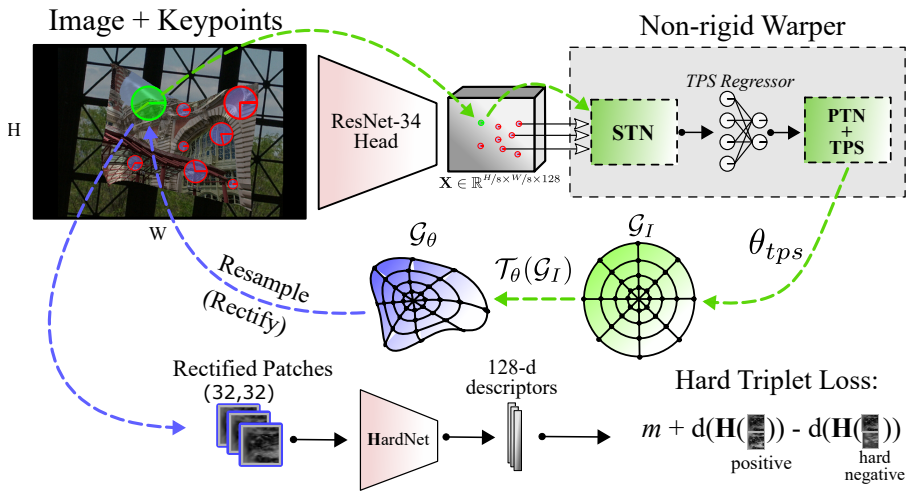


Figure 2: **Architecture for computing descriptors of deforming objects.** The non-rigid warper component is a Thin Plate Splines (TPS) that rectifies image patches using the parameters estimated from ResNet34 features by the regressor network. The rectified patches feed the network that extracts discriminant descriptors. The network is trained end-to-end by optimizing a hard triplet loss.

convolutions until the spatial resolution achieves $1/8$ of the original input resolution. The feature map $\mathbf{X}$ is used as input for the the non-rigid warper component, which computes the TPS parameters with the spatial transformer to model a non-rigid warp.

3.2 Non-rigid warper

The Non-Rigid Warper (NRW) is composed of two spatial transformers and one localization network. The diagram of the interconnection of these components is presented in Figure 2. The first spatial transformer samples the feature map $\mathbf{X}$ at keypoint locations to extract local tensors of shape $5 \times 5 \times 128$, a local feature map centered at the keypoint coordinates used as input to the localization network $\mathcal{S}$, the Thin-Plate Splines (TPS) [11] parameter regressor. After estimating the TPS parameters θ_{tps}, we initialize the identity grid $\mathcal{G}_I$ from a polar grid centered at the keypoint and with a radius proportional to the keypoint's size attribute. The output of the non-rigid TPS warping is the parameters required to obtain the image rectified patches.

Spatial transformer. We designed a Spatial Transformer Network (STN) [15] to apply spatial transformations to the output 3D tensor $\mathbf{X}$ from ResNet in an end-to-end differentiable manner. Given a non-rigid transformation function $\mathcal{T}_\theta$, where θ is the parametrization of the transformation, and the identity mesh grid $\mathcal{M}_I$, we generate transformed mesh grid coordinates as: $\mathcal{M}_\theta = \mathcal{T}_\theta(\mathcal{M}_I)$. Once the meshgrid is transformed, for each pixel spatial coordinate $\mathbf{p}$, $\mathbf{p} \in \mathbb{R}^2$ in $\mathcal{M}_\theta^{(i,j)}, \forall i \in [1, H], \forall j \in [1, W]$, we bilinearly interpolate values in $\mathbf{X}$ for each channel c to obtain the warped features $\mathbf{Y}$ [15]:

$$\mathbf{Y}_{\mathbf{p}}^c = \sum_{m=0}^{1} \sum_{n=0}^{1} \mathbf{X}_{\lfloor \mathbf{p}+(m,n) \rfloor}^c \mid m - \mathbf{p}'_{(1)} \mid \times \mid n - \mathbf{p}'_{(2)} \mid, \tag{1}$$

where $\mathbf{p}'$ is the decimal part of the pixel coordinate $\mathbf{p}$.

The mesh grid $\mathcal{M}_\theta \in \mathbb{R}^{5\times5\times2}$ encodes the support region's spatial coordinate map of each keypoint, obtained by the STN using the keypoint parameters. $\mathcal{M}_\theta$ is used to interpolate the feature map $\mathbf{X}$ to sample the local mid-level descriptors that encode local deformations. The local feature maps are forwarded to the localization network that estimates the parameters of the local deformations.

Localization network. We model the non-rigid object deformations with Thin-Plate Splines (TPS) [11]. TPS warps representing 2D coordinate mappings $\mathbb{R}^2 \mapsto \mathbb{R}^2$ are often used to model non-rigid deformations. They produce differentiable smooth interpolations, which can be easily coupled in neural networks using attention mechanisms. The TPS formulation elegantly enables one to model the affine component of the transformation through an affine matrix component $\mathbf{A} \in \mathbb{R}^{2\times3}$,

and a non-affine component encoded by the weight coefficients $\mathbf{w}_k \in \mathbb{R}^2$ separately representing distortions. Given a 2D point $\mathbf{q} \in \mathbb{R}^3$, weight coefficients and control points $\mathbf{c}_k, \in \mathbb{R}^2$ both in homogeneous coordinates, the corresponding mapping $\mathbf{q}'$ can be computed with:

$$\mathbf{q}' = \mathbf{A}\mathbf{q} + \sum_{k=1}^{n} \rho(\|\mathbf{q} - \mathbf{c}_k\|^2)\mathbf{w}_k, \tag{2}$$

where $\rho = 1/2r^2 \log r^2$ is the TPS radial basis function.

To estimate the coefficient weights of the TPS, we employ a regressor network $\mathcal{S}$, implemented as a Multi-layer Perceptron (MLP) that takes as input the interpolated feature map $\mathbf{Y}$ of each keypoint to compute the TPS parameters θ_{tps}. We use 64 control points in our implementation, which provides a good trade-off between non-rigid modeling capacity and computational cost according to our experiments. The estimated θ_{tps} in conjunction with the identity polar grid $\mathcal{G}_I$ (initialized using the Polar Transformer) are used to sample a rectified image patch.

Polar Transformer Network (PTN). According to Ebel *et al*. [12], polar representations of a patch around keypoints increase the robustness of the descriptor extractor to image transformations, while also achieving rotation equivariance. Considering these findings, we adopt this representation in the sampled rectified patches. For that, a fixed regular polar transformation is applied to the image, parametrized by the keypoints' attributes to generate an identity polar transformation map $\mathcal{G}_I \in \mathbb{R}^{32\times32\times2}$ encoding the initial keypoint rotation, scale and size. We leverage the polar representation, first constructing the polar grid that is initially used as the identity transformation. The identity grid $\mathcal{G}_I$ is transformed to the warped grid $\mathcal{G}_\theta$ using the estimated θ_{tps}, which samples the final rectified patches. Finally, the patches are forwarded to the feature extraction network that computes a distinctive and compact descriptor for the rectified polar patch. We employed HardNet [22] as the backbone network in our implementation.

3.3 Local descriptor extraction and loss

Our architecture can be trained end-to-end since all employed operations are differentiable, including the non-rigid warper component. The hardest in-batch triplet loss [22] is computed on a mini-batch of pairs of descriptors extracted by the HardNet architecture $\mathbf{H}(.)$ after a forward pass, which enforces distinctiveness of the extracted descriptors. The HardNet takes as input 32×32 grayscale patches and outputs a 128-dimensional L2 normalized feature vector for each keypoint.

Let $\mathbf{A} \in \mathbb{R}^{N\times D}$ and $\mathbf{B} \in \mathbb{R}^{N\times D}$ be a matrix of N vertically stacked D-dimensional descriptor vectors of corresponding patches extracted by HardNet. Assuming that the descriptors are L2 normalized, we can calculate the matrix of distances $\mathbf{D}_{N\times N} = 2(1 - \sqrt{\mathbf{A}\mathbf{B}^T})$. The hardest negative example h_i for each row $\mathbf{D}'_i$, $\mathbf{D}' = \mathbf{D} + \alpha\mathbf{I}_{N\times N}$, where α is a constant ≥ 2 needed to supress the corresponding pairs from the diagonal of the distance matrix, is computed as $\delta_h^{(i)} = \min(\mathbf{D}'_i)$. The hardest in-batch margin ranking loss is then calculated as:

$$\mathcal{L}_H\left(\delta_+^{(.)}, \delta_h^{(.)}\right) = \frac{1}{N}\sum_{i=1}^{N} \max(0, \mu + \delta_+^{(i)} - \delta_h^{(i)}), \tag{3}$$

where μ is the margin, $\delta_+ = \|\mathbf{H}(p) - \mathbf{H}(p')\|_2$ is the distance between the corresponding patches, and $\delta_h = \|\mathbf{H}(p) - \mathbf{H}(h)\|_2$ is the distance to the hardest negative sample in the batch.

4 Experiments and Results

We evaluate our descriptor in different publicly available datasets containing deformable objects in diverse viewing conditions such as illumination, viewpoint, and deformation. For that, we have selected the two datasets recently proposed by GeoBit [25] and DeSurT [43].

They contain color images of 11 deforming real-world objects, where keypoints are independently detected with SIFT and ground-truth correspondences are done following the protocol of [16].

Baselines and Metrics. Since this paper tackles the problem of matching images of deformable objects, we compare our results against two deformation-invariant descriptors: DaLI [37] and GeoBit [25]. Aside from these descriptors, we also include two the well-known binary descriptors for 2D images ORB [34] and FREAK [1]; a floating-point gradient based descriptor: DAISY [40]; one descriptor that combines texture and shape: BRAND [24]; and two learning methods, TFeat [42] and Log-Polar [12], which is an improvement of HardNet [22]. We also demonstrate that our method is able to surpass the two recently proposed detect-and-describe methods R2D2 [31] and ASLFeat [20], using our descriptor to extract features in their detected keypoints locations.

The performance assessment is using the matching score (MS) metric defined by Mikolajczyk *et al*. [21]. Considering a pair of images i and j observing the same scene, the metric is defined as: $MS = \#correct / \min(\#keypoints_i, \#keypoints_j)$, where $\#correct$ is the number of correct keypoint matches at a 3 pixels threshold (used in all experiments in this paper) obtained by matching the descriptors using nearest neighbor search, and $\#keypoints_{\{i,j\}}$ is the number of detected keypoints on image i and j. We also use the Mean Matching Accuracy (MMA @ 3 pixels) metric as in [31].

4.1 Implementation details

Training dataset. Although most of the reported results are from real-world images, our network was trained with simulated data only. We use a simulated environment to generate plausible non-rigid deformations (isometric transformations) of surfaces. The dataset is composed of $20{,}000$ (960×720 resolution) image pairs with ground-truth correspondences. In order to generate realistic deformed images, we performed texture mapping of the surfaces with real images extracted from large-scale Structure-from-Motion datasets [44]. The correspondences between frames are generated by first independently detecting up to $2{,}048$ SIFT keypoints for each frame and then corresponding them using the simulation data under a threshold of 3 pixels. Each image pair contains, on average, approximately $1K$ ground-truth correspondences, summing up to a total of about $20M$ keypoint correspondences. Independently detecting the keypoints in the frames is a key step that improves generalization, since the intrinsic properties of the keypoint detector is present.

Network and training setup. We implement[1] our network using PyTorch [27] and optimize it via Adam with initial learning rate of $5e^{-5}$, scaling it by 0.9 every $3,800$ steps. The network is trained end-to-end, setting the TPS regressor's weights of the last layer to zero and the initial warp the identity matrix. We used a batch size of 8 image pairs containing up to 128 keypoint correspondences for each pair in our setup. The keypoint correspondences are randomly sampled from a uniform distribution with fixed seed during training, and we train the network for 10 epochs. The model is trained using a siamese scheme, where descriptors are extracted for the first and second set of corresponding keypoints (positive examples) using two networks with shared weights. The negative examples are calculated using a hard mining strategy in the batch as described in [22] and used in the triplet loss. We also apply Average Pooling in the angle-axis of the polar patches at the end of HardNet feature maps to achieve rotation invariance. Our network implementation has $3.7M$ trainable parameters and takes about 5.5 hours to train on a GeForce GTX 1080 Ti GPU.

Sensibility analysis. To evaluate the sensibility and the effect of hyperparameters in our network, we performed the following experiments: (i) margin ranking parameter sensitivity in the triplet loss; (ii) use of anchor swap in the triplet loss [42]; (iii) larger support region for the non-rigid warper (NRW) component; and (iv) use of dropout in the fully connected (FC) layers. Table 1 shows the MMA average achieved by testing different hyperparameters on the *Bag* sequence [25], which we used as a validation set and removed it from the benchmark experiments. The training steps and initialization are kept the same for all tested variations, and only one tested hyperparameter is changed while fixing all others for the sake of reproducibility and consistency. The baseline model uses the margin $\mu = 0.5$, no anchor swap, STN output of 3×3 and Dropout with probability $p = 0.1$ in the FC layers. Based on the results, we update the final

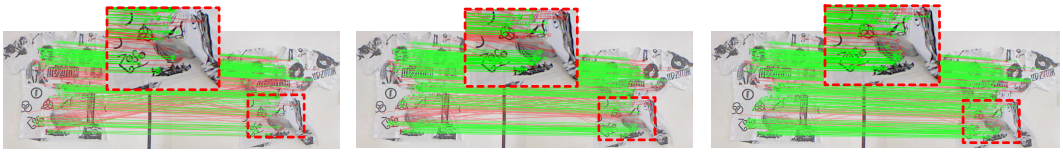
Table 1: Effect of hyperparameters in the proposed network.

Hyperparameter	**MMA** $\uparrow$
Baseline	0.603
Margin 0.25	0.592
Margin 0.75	0.608
Anchor Swp.	0.592
STN out. 5×5	0.613
No Dropout FC	0.601

[1]The training data, network implementation and the simulation framework will be released to the community after the double-blind review process.

Table 2: **Comparison with state-of-the-art descriptors**. Results indicate that the proposed descriptor exhibit improved robustness against deformations in all considered datasets. The ones marked with * are computed on RGB-D images. Best in bold and second-best underlined. The mean was calculated with full-precision values before rounding.

Descriptors	**Datasets:** 833 **pairs total** – MS / MMA @ 3 pixels ↑				**Mean**
	Kinect 1 [25]	*Kinect 2* [25]	*DeSurT* [43]	*Simulation* [25]	
BRAND* [24]	0.17 / 0.34	0.22 / 0.49	0.14 / 0.33	0.04 / 0.09	0.16 / 0.34
ORB [34]	0.19 / 0.38	0.25 / 0.55	0.18 / 0.40	0.14 / 0.30	0.20 / 0.43
DAISY [40]	0.23 / 0.47	0.29 / 0.62	0.16 / 0.37	0.19 / 0.39	0.22 / 0.48
FREAK [1]	0.24 / 0.49	0.33 / 0.72	0.16 / 0.38	0.15 / 0.31	0.23 / 0.51
DaLI [37]	0.25 / 0.51	0.35 / 0.76	0.21 / 0.48	0.10 / 0.22	0.25 / 0.54
TFeat [42]	0.25 / 0.50	0.28 / 0.61	0.21 / 0.48	0.29 / 0.63	0.26 / 0.56
Log-Polar [12]	0.28 / 0.58	0.30 / 0.65	0.23 / 0.54	0.22 / 0.49	0.26 / 0.57
GeoBit* [25]	0.31 / 0.65	0.35 / 0.77	0.20 / 0.47	0.32 / 0.71	0.30 / 0.66
Ours	**0.33 / 0.68**	**0.38 / 0.85**	**0.27 / 0.63**	**0.36 / 0.80**	**0.34 / 0.75**



a) DaLI - 126 inliers b) GeoBit - 179 inliers c) Ours - 208 inliers

Figure 3: **Matching example.** Our descriptor better handles the strong deformation (highlighted in red), among the best competitors. Correct matches are drawn as green, and wrong ones as red lines.

model (used in all experiments) to use STN output of 5×5 and keep other parameters from baseline unchanged, since they do not give significant improvements or decrease the performance. We did not include larger STN outputs since we noticed marginal performance gains, while the model increases its computational requirements.

4.2 Results on real image sequences of deformable objects

Table 2 shows the matching scores for all descriptors in our experiments. For the sake of a fair comparison, the keypoints are shared among all descriptors in this experiment, including rotation and scale attributes of SIFT keypoints. One can see that our method outperforms all descriptors, including GeoBit, that uses additional depth information to extract deformation-invariant features. GeoBit, as expected, achieved the second-best results overall, followed by the learning-based methods. DaLI performs the best among the hand-crafted descriptors, due to its robustness against deformations. BRAND, for its turn, provides the worst scores overall because of its strong assumptions of geometric rigidity of the object in both image and 3D space. Figure 3 shows a matching example from Kinect 1 dataset (*Shirt 1* sequence), depicting a strong deformation. It is noticeable that our method improved the matching, especially in the most deformed regions (red dashed square).

Furthermore, we analyze the performance of our descriptor against detect-and-describe methods R2D2 [31] and ASLFeat [20]. For this end, we had to adapt our method to work with different keypoint detections provided by these methods. Since they detect and describe their keypoints jointly, we cannot directly compare their performance against the descriptor-only baselines from Table 2 in this regime. Even our descriptor being trained using different keypoints, it was able to improve ASLFeat and R2D2 performances. Computing DEAL on R2D2 keypoints increases the MMA of R2D2 from 0.394 to 0.467, and when using ASLFeat detected keypoints with our descriptor, we are able to improve ASLFeat's MMA from 0.563 to 0.618. This result indicates that our approach also generalizes and can be used with other detection techniques to increase robustness to deformations.

Rotation and scale invariance. To evaluate the robustness of our descriptor to rotation and scale, we select sequences with strong rotation and scale transformations, where the camera suffers in-plane

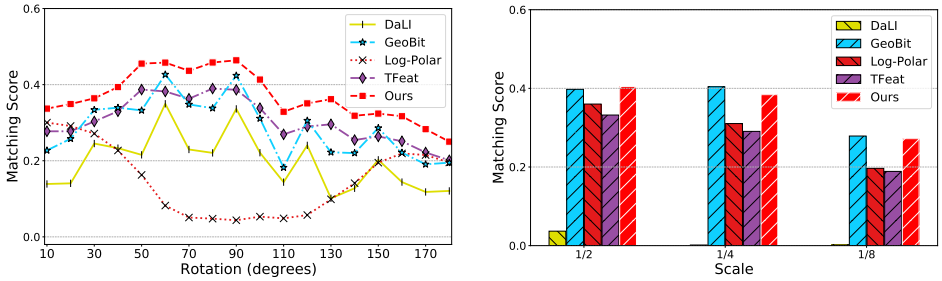


Figure 4: **Rotation and scale invariance.** Matching score curves obtained by matching SIFT keypoints showing results for rotation and scale for each target frame relative to the reference using the *Simulation* dataset. This experiment evaluates the robustness of the descriptors to both deformation, scale, rotation, and illumination in the most challenging sequence.

rotations from 0° to 180° with 10° degrees steps for rotation tests. For the scale tests, the camera was moved backward in the Z direction, producing downscaling of the object. From Figure 4, one can notice that our method holds the best invariance to image in-plane rotations, and its very close to GeoBit in the scale sequences. Note that GeoBit is an RGB-D descriptor, *i.e.*, it requires depth information from the scene that is used to rectify deformations and to obtain the true scale of the object in the image, while our method only requires the monocular image.

4.3 Ablation studies and processing time

To evaluate the contribution of the components of our architecture and support our implementation decisions, we performed the ablation analysis of different parts of the proposed method.

Contribution of the TPS warper. We consider two different setups: (i) using only the PTN network with fixed parameters; or (ii) with the NRW component for modeling the deformation. In this analysis, we use the scale simulation sequences, that were purposefully chosen because they exhibit mostly strong deformations across multiple image pairs. We observed that enabling the non-rigid warper increases the robustness against strong non-rigid deformations. NRW significantly improves the MMA as can be seen in Table 3 (best in bold). Both networks were trained until convergence with the same training procedure and HardNet initialization.

Table 3: Ablation.

Method	MMA ↑
PTN	0.75
NRW	**0.79**

Use of pre-trained ResNet features. We also verified if transfer learning from a pre-trained ResNet model on ImageNet is feasible for the task of patch rectification. However, the training of the network did not converge well when using the pre-trained weights. We argue that the non-rigid warper module is focused on learning warping parameters, which is a considerably distinct task compared to classification. Thus, we initialize the weights of the entire network and train it from scratch.

Processing time. We executed the three deformation-invariant descriptors (DaLi, GeoBit, and DEAL) on a set of 250 descriptors (from 640×480 resolution images), running on a Intel (R) Core (TM) i7-7700 CPU @ 3.60 GHz and a GTX 1080 Ti GPU. Our descriptor was significantly faster than DaLI and GeoBit. While our method (GPU) spent 0.03 seconds to compute the descriptors, DaLI (CPU-only) and GeoBit (CPU-only) spent 112.95 and 33.72 seconds, respectively.

4.4 Applications

To further demonstrate the effectiveness of our descriptor in potential real-world applications, we performed experiments in two complementary real-world applications: object retrieval and tracking of deformable objects (more results can be found in the supplementary material).

Object Retrieval. In the retrieval application, we used a Bag-of-Visual-Words approach [8]. For each descriptor, we first construct a visual dictionary used to compute a global descriptor for each image. Given a query image, we compute the global descriptor and use K-Nearest Neighbor search

Table 4: **Tracking average LPIPS and RANSAC inliers rate.** Approaches with * are computed on their own detected keypoints, while ** uses RGB-D images. Best in bold and second-best underlined.

	DaLI	DELF*	GeoBit**	Log-Polar	TFeat	Ours
Inliers RANSAC ↑	0.32	0.25	**0.46**	0.36	0.32	**0.46**
LPIPS ↓	0.44	0.33	**0.23**	<u>0.24</u>	<u>0.24</u>	<u>0.24</u>

to obtain the top K closest objects. The metric used is the retrieval accuracy, *i.e.*, the number of correct objects retrieved in the top K images. Figure 5 shows the results with the best performing descriptors according to the matching experiments. Our method achieved the best accuracy with $K \leq 10$. In this experiment, we also consider DELF [38], the state-of-the-art descriptor designed and trained for image retrieval. Since DELF does not allow the description on commonly adopted keypoint detectors, we perform its evaluation using its own detected keypoints. The results indicate that our deformation-aware network was able to provide good results in a different but related task.

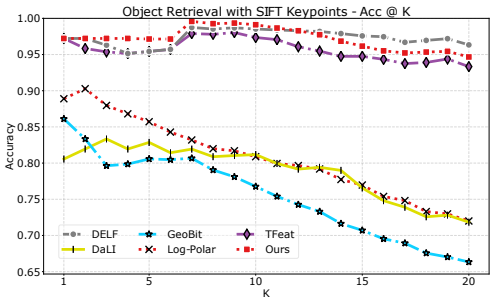


Figure 5: **Object retrieval.** Our descriptor outperforms all descriptors with $K \leq 10$, including the state-of-the-art retrieval descriptor DELF.

Non-rigid tracking results. This task consisted of tracking a region-of-interest of a template image over time using the correspondences from the computed descriptors. We approximate the deformation using TPS warping, which was combined with a RANSAC filtering to remove outlier correspondences for all descriptors, following the protocol proposed in [41]. Table 4 shows the average values of the Perceptual Patch Similarity distance (LPIPS) [46] and the number of inliers found by RANSAC for the main competitors. We also included DELF using their own detected keypoints on all sequences. As can be seen, even using RGB data only, our descriptor showed similar results to GeoBit. Representative qualitative results are shown in Figure 6 for the sequences *Shirt3* and *kanagawa_rot*. While the correspondences using the our descriptor (last row) allowed the tracking of frames even in extreme conditions of rotation and deformation, DaLI was particularly affected by the object changes.

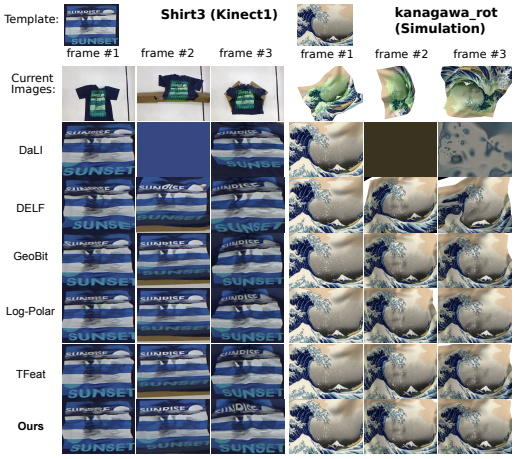


Figure 6: **Tracking results.** Qualitative tracking results for two sequences. Please notice the performance of the proposed approach (last row) for these challenging frames.

5 Conclusions

This paper presents a new local descriptor robust to non-rigid deformations, scale, and rotation transforms. The local features are extracted by a deformation-aware architecture trained in an end-to-end manner following a deformation guidance strategy. Besides providing highly discriminative and invariant feature vectors, our approach requires only still images and is time-efficient. We demonstrate the effectiveness and robustness of our descriptor by extensive experiments comprising the comparison with state-of-the-art descriptors, ablation studies, and its use on two real applications. A limitation of our work is that its performance is compromised when dealing with surfaces having sharp deformations, discontinuities, and reflections. Please check our supplementary material for more visual results, including examples of failure cases of all methods when affected by extreme deformations.

References

[1] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517. Ieee, 2012.

[2] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, 2017.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008.

[4] A. Božič, M. Zollhöfer, C. Theobalt, and M. Nießner. DeepDeform: Learning non-rigid rgb-d reconstruction with semi-supervised data. 2020.

[5] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73, 2007.

[6] 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*, 34(7):1281–1298, 2012.

[7] K. Crane, C. Weischedel, and M. Wardetzky. Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Trans. Graph. (TOG)*, 32(5):152:1–152:11, oct 2013.

[8] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[9] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[10] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018.

[11] G. Donato and S. Belongie. Approximate thin plate spline mappings. In *European conference on computer vision*, pages 21–31. Springer, 2002.

[12] P. Ebel, A. Mishchuk, K. M. Yi, P. Fua, and E. Trulls. Beyond cartesian representations for local descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 253–262, 2019.

[13] H. Guan and W. A. Smith. Brisks: Binary features for spherical images on a geodesic grid. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4516–4524, 2017.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[15] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. *arXiv preprint arXiv:1506.02025*, 2015.

[16] Y. Jin, D. Mishkin, A. Mishchuk, J. Matas, P. Fua, K. M. Yi, and E. Trulls. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision*, pages 1–31, 2020.

[17] S. Kumar, A. Cherian, Y. Dai, and H. Li. Scalable dense non-rigid structure-from-motion: A grassmannian perspective. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, pages 91–110, 2004.

[19] Z. Luo, T. Shen, L. Zhou, S. Zhu, R. Zhang, Y. Yao, T. Fang, and L. Quan. Geodesc: Learning local descriptors by integrating geometry constraints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 168–183, 2018.

[20] Z. Luo, L. Zhou, X. Bai, H. Chen, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan. Aslfeat: Learning local features of accurate shape and localization. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 6589–6598, 2020.

[21] 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*, 65(1-2):43–72, 2005.

[22] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems*, pages 4826–4837, 2017.

[23] F. Moreno-Noguer. Deformation and illumination invariant feature point descriptor. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1593–1600, 2011.

[24] E. R. Nascimento, G. L. Oliveira, M. F. M. Campos, A. W. Vieira, and W. R. Schwartz. BRAND: A Robust Appearance and Depth Descriptor for RGB-D Images. In *Proc. IROS*, 2012.

[25] E. R. Nascimento, G. Potje, R. Martins, F. Chamone, M. Campos, and R. Bajcsy. Geobit: A geodesic-based binary descriptor invariant to non-rigid deformations for rgb-d images. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[26] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-scale image retrieval with attentive deep local features. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3476–3485, 2017.

[27] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NeurIPS Workshops*, 2017.

[28] J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration and realistic augmentation. *International Journal of Computer Vision*, 76(2):109–122, 2008.

[29] A. Pumarola, A. Agudo, L. Porzi, A. Sanfeliu, V. Lepetit, and F. Moreno-Noguer. Geometry-Aware Network for Non-Rigid Shape Prediction from a Single View. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[30] J. Revaud, C. De Souza, M. Humenberger, and P. Weinzaepfel. R2D2: Reliable and repeatable detector and descriptor. In H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 12405–12415. Curran Associates, Inc., 2019.

[31] J. Revaud, P. Weinzaepfel, C. De Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger. R2d2: Repeatable and reliable detector and descriptor. *arXiv preprint arXiv:1906.06195*, 2019.

[32] I. Rocco, R. Arandjelovic, and J. Sivic. Convolutional neural network architecture for geometric matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[33] I. Rocco, R. Arandjelović, and J. Sivic. End-to-end weakly-supervised semantic alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[34] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: an efficient alternative to SIFT or SURF. In *Proc. ICCV*, Barcelona, 2011.

[35] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[36] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[37] E. Simo-Serra, C. Torras, and F. Moreno-Noguer. DaLI: deformation and light invariant descriptor. *International Journal of Computer Vision*, 115(2), 2015.

[38] M. Teichmann, A. Araujo, M. Zhu, and J. Sim. Detect-to-retrieve: Efficient regional aggregation for image search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5109–5118, 2019.

[39] Y. Tian, B. Fan, and F. Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 661–669, 2017.

[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*, 32(5):815–830, 2010.

[41] Q.-H. Tran, T.-J. Chin, G. Carneiro, M. S. Brown, and D. Suter. In defence of ransac for outlier rejection in deformable registration. In *ECCV*, 2012.

[42] D. P. Vassileios Balntas, Edgar Riba and K. Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016.

[43] T. Wang, H. Ling, C. Lang, S. Feng, and X. Hou. Deformable surface tracking by graph matching. In *IEEE International Conference on Computer Vision*, 2019.

[44] K. Wilson and N. Snavely. Robust global translations with 1dsfm. In *European Conference on Computer Vision*, pages 61–75. Springer, 2014.

[45] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned invariant feature transform. In *ECCV*, 2016.

[46] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[47] Q. Zhao, W. Feng, L. Wan, and J. Zhang. Sphorb: A fast and robust binary feature on the sphere. *International journal of computer vision*, 113(2):143–159, 2015.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Please check Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code, data and instructions will be released after the double-blind review process.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please, check Section 4.1 – Implementation details.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We followed the protocols used in [16], [25] and [31], where keypoints are independently detected for each frame and metrics are based on the matching scores [21].
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please check Section 4.1 for the complete implementation details.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We are using two existing recently available databases of non-rigid objects, as well as the code of several existing baselines as presented in the beginning of Section – Experiments and Results. The authors and creators are cited accordingly for all assets.
   (b) Did you mention the license of the assets? [No] Licenses will be mentioned in the released code and assets after the double-blind review process.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]