Learning to Detect Good Keypoints to Match Non-Rigid Objects in RGB Images

Welerson Melo[1], Guilherme Potje[1], Felipe Cadar[1], Renato Martins[2], and Erickson R. Nascimento[1]
[1]Universidade Federal de Minas Gerais [2]Université Bourgogne Franche-Comté
{welerson.melo,guipotje,cadar}@dcc.ufmg.br, renato.martins@u-bourgogne.fr, erickson@dcc.ufmg.br

***Abstract*—We present a novel learned keypoint detection method designed to maximize the number of correct matches for the task of non-rigid image correspondence. Our training framework uses true correspondences, obtained by matching annotated image pairs with a predefined descriptor extractor, as a ground-truth to train a convolutional neural network (CNN). We optimize the model architecture by applying known geometric transformations to images as the supervisory signal. Experiments show that our method outperforms the state-of-the-art keypoint detector on real images of non-rigid objects by 20 p.p. on Mean Matching Accuracy and also improves the matching performance of several descriptors when coupled with our detection method. We also employ the proposed method in one challenging real-world application: object retrieval, where our detector exhibits performance on par with the best available keypoint detectors. The source code and trained model are publicly available at https://github.com/verlab/LearningToDetect_SIBGRAPI_2022**

I. Introduction

In the last decades, several methods have been proposed to efficiently detect keypoints. Seminal works such as Harris Corner [1] and SIFT [2] allowed significant advancements in many applications tasks such as Content-Based Image Retrieval, Structure-from-Motion (SfM) [3], [4] and registration [5]. Keypoint detectors aim to find discriminative visual patterns that are repeatable on different images of the same scene. An effective detector should be invariant to different illumination conditions and equivariant to viewpoint and scale changes. Furthermore, beyond rigid transformations, objects may have different shapes over time due to deformations; therefore, robustness to non-rigid deformations is also a key factor to consider while locating points for visual correspondence.

Recently, feature detection methods have been moving towards learned-based systems [6]–[12] delivering results that significantly outperform the handcrafted counterpart [2], [13]. However, learning to detect keypoints is not a well-defined task. A learned keypoint detector generally learns to find patterns that are repeatable across images; however, this strategy does not guarantee that the detected points are good to be matched. For example, points on edges and repetitive patterns that usually occur in man-made structures are challenging to be matched due to high texture ambiguity. Methods that first detect and then describe keypoints may harm the matching performance in regions where accurate matching is not possible [7]. Moreover, performing matching simultaneously with the detection and description has often high computational complexity [10]. Given two images A and B with feature sets F_A and F_B, matching them has a time complexity of $O(|F_A| \cdot |F_B|)$. As each image pixel may potentially become a feature, the problem quickly becomes intractable, needing carefully designed training schemes and massive computational resources. Regarding the non-rigid transformation, very few works have been proposed to address the non-rigid deformation invariance task. Recent explorations [14]–[16] propose to tackle non-rigid deformation but relying on depth information. Despite the advances achieved, RGB cameras are still by far the most common type of imaging sensor.

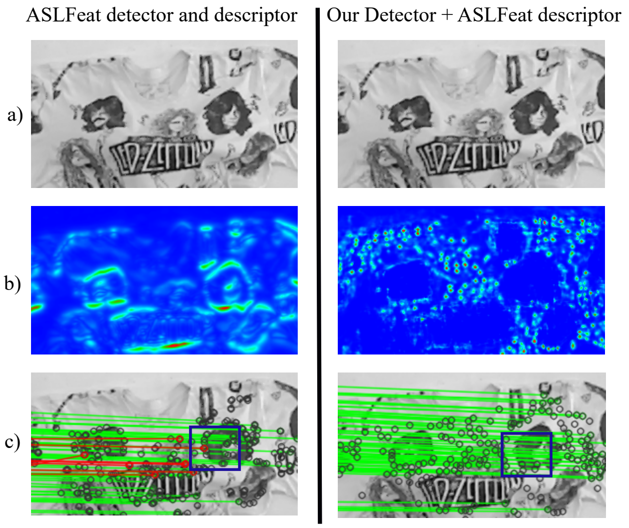
ASLFeat detector and descriptor
Our Detector + ASLFeat descriptor
a)
b)
c)

Fig. 1. **Results on detecting good keypoints.** a) Input image with non-rigid deformations; b) Score maps of ASLFet and ours; c) Correct matches (green) and incorrect ones (red), as well as circles representing the detected keypoints. One can notice that not all peaks in the score map are keypoints because we chose the top 1,024 according to the score value. Notably, our method provides more reliable points to be matched.

In this paper, we present a learned detection strategy designed to tackle non-rigid deformations on still images (Figure 1 shows some qualitative results). We address the keypoint detection problem efficiently in a well-defined manner exploiting the assumption that good features to be detected are also salient points that are likely to yield correct matches. We propose a novel learned detection methodology that predicts ground-truth matching maps based on an existing detector-descriptor configuration. The network is trained to detect good features according to the map derived from true descriptor matches and it can be easily coupled with any combination of

978-1-6654-5385-1/22/$31.00 ©2022 IEEE

existing detector-descriptor pairs.

We evaluate our detector on three different benchmark datasets (*i.e.*, Kinect1, Kinect2, and DeSurT) of real deformable objects, as well as with application scenarios on content-based object retrieval, validating that our method can reach state-of-the-art performance not only in matching evaluation scores but also in a practical related computer vision task. Figure 1 illustrates the matching quality of detected keypoints of our detector and the recent ASLFeat detector [9].

The contribution of our work is two-fold: (i) a novel keypoint detection training framework aimed to improve the matching performance of existing descriptors, and (ii) the first learned keypoint detector optimized to cope with non-rigid deformations that works only using standard RGB images.

II. Related work

A. Handcrafted and learning-based feature detection

Most of the existing keypoint detectors are based on handcrafted approaches to select feature patterns such as blob, corners or edges [1], [2], [13], [17]. However, in the past few years, the use of deep learning for feature extraction and image matching pipeline became the new gold standard approach. On the flip side, these learning-based approaches are mostly applied to feature description tasks [18], [19] and jointly detection-description [5], [7]–[10], [20], [21]. Few works have focused on learning to detect keypoints by increasing detection probability for repeatable areas between image pairs [22], [23]. In these works, the resulting keypoint detector has high repeatability, although matching performance is hindered since the detected keypoints are often ambiguous [6]. Barroso-Laguna *et al.* [6] use fixed handcrafted filters alongside the learned ones, aiming to improve representation and generalization of low-level features. However, these methods can provide poor detections if the handcrafted filters have some bias that could not help to detect good keypoints, *e.g.*, clustering keypoints along edges and corners. To solve the these problems, the LLF detector [11] trains a detector to retrieve low-level features and keypoint locations by adding a layer that learns to select a keypoint with matching reliability. A drawback of this method is that it does not consider real matching scenarios, which in practice does not improve the matching accuracy of the detected keypoints.

Recently, several works follow the describe-then-detect scheme, where the keypoints are detected from a dense descriptor map. Revaud *et al.* [7] claim that detection and description are inseparably tangled, thus keypoints should be detected based on repeatability and reliability. Suwanwimolkul *et al.* [11] observed that in methods such as D2-Net [8] and ASLFeat [9], there is no guarantee that the selected keypoints are associated with matching the learned descriptors, since the keypoint selection is rather handcrafted. Therefore, the matched keypoints do not always have high accuracy. Figure 1 shows these problems of handcrafted keypoint selection in the highlighted squared region. Conversely, in our proposed approach, the peaks are generated directly from the network output (score map in Figure 1-b), and the detector is trained to improve matching accuracy with real image pairs.

Fewer works consider the descriptor matching in the training pipeline such as our approach. GLAM [24] detects keypoints based on matching quality, however for retinal images, a very specific domain. SEKD [21] proposes a non-domain specific detector and descriptor by first detecting keypoints based on repeatability, and then filtering the reliable keypoints based on the matching. This approach can yield subpar results when a large set of good keypoints for matching is not found in the repeatability optimization stage. DISK [10] considers detection and description in a probabilistic relaxation and applies a reinforcement learning strategy to optimize detection and description jointly. As drawback, the method requires careful hyperparameter tuning to converge. Tonioni *et al.* [25] trained a decision tree to learn to select 3D keypoints based on good matches. The authors argue that good features to be detected are those likely to yield correct matches. We apply a similar strategy on 2D keypoints. Similarly to GLAM [24], we use the results of matching descriptor with a weight strategy for the Matching Heatmap, applied to a general domain.

B. Non-rigid deformation feature detection and matching

To circumvent the problem of non-rigid transform and keypoint description, descriptors such as the DEAL method [19] proposes a deformation-aware local feature description strategy that learns to describe non-rigid patches without depth information. The DaLi descriptor [26] encodes features robust to non-rigid deformations and illumination changes. In the same context, GeoBit descriptor [15] uses geodesics from objects surface to compute isometric-invariant features working with RGB-D images. Unlike the proposed approach, these methods are solely concerned with the description and overlook the detection stage. Recently, UKPGAN [14] proposes a 3D keypoint detector where keypoints are detected for the task of 3D reconstruction. However, their method is suitable only in context of 3D keypoints, conversely to our proposed detector for 2D images. To the best of the authors' knowledge, no work deals with 2D feature detection on images with non-rigid deformations. In this work, we propose a methodology that can be used to obtain keypoints robust to non-rigid deformations relying only on visual information.

III. Methodology

Our detector is designed to learn to extract keypoints from images being affected by non-rigid deformations. The training setup enforces that keypoints are detected in repeatable locations having confident matching probability for a given descriptor. The learned detector is designed with a 4-level deep U-net [27] with a final sigmoid activation function. It is also composed of 3×3 convolution blocks with batch normalization and ReLU activations. U-net is our method of choice due to its past successes in dense regression and semantic segmentation tasks. Figure 2 depicts the network architecture with the Siamese scheme and the final loss $\mathcal{L}$ calculation. In the next

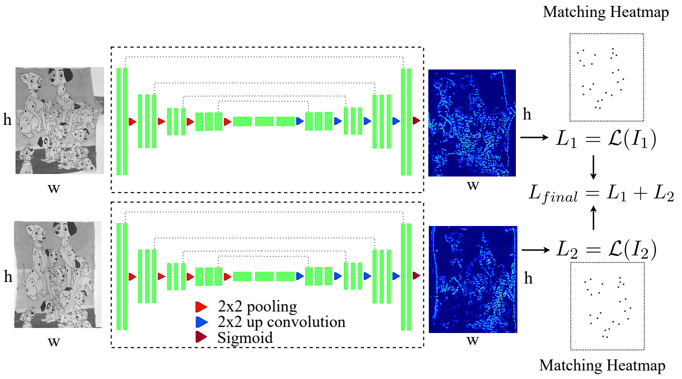

Fig. 2. **Overview of the network architecture used as backbone for keypoint detection.** The Siamese network is optimized to detect reliable keypoints to be matched for a given descriptor. Each green block represents a 3×3 2D convolution layer, followed by batch normalization and ReLU activation function. The two branches share the weights.

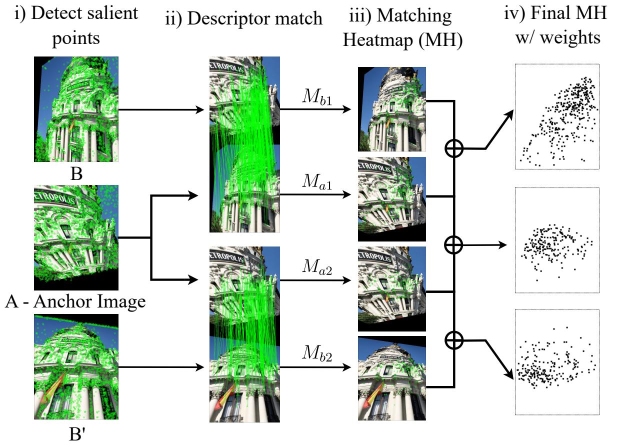

Fig. 3. **Overview of the detector training framework.** Our framework is composed four steps: i) First, we detect the keypoints using a base detector for images A, B and B' (an anchor, and two transformed versions of the anchor with random homography and non-rigid image transformations); ii) Then, we extract the descriptors on the detected keypoints and then find correspondences with nearest neighbor search;iii) Using the correct matches, we build a Matching Heatmap (MH) from the location of correct matches for each input image; iv) MH weighting based on keypoint quality, *i.e.*, true matching repeatability.

sections, we explain the training framework, the loss design, and implementation details.

A. *Keypoint detection learning framework*

Unlike learned keypoints as Key.net [6], and describe-to-detect extraction methods such as R2D2 [7], and ASLFeat [9], in our learning strategy, the key idea is to leverage an existing detector-descriptor pair to bootstrap the learning process that is focused on high confident matches. Let $A \in \mathbb{R}^{H \times W}$ be an image from our training dataset, defined as the anchor image. We generate images B and B' by applying two different deformations composed of a random homography and a thin-plate spline warp (TPS) [28] (g and g' respectively) on the anchor image A. The TPS warps representing 2D coordinates are often used to model non-rigid deformations, giving us an apparatus to work with this type of transformation. Now, for images A, B and B', we detect k salient keypoints according to a base detector. In our experiments, we use $k = 0.02 \times H \times W$, which results in keypoints covering a good portion of all image regions.

Once we have selected the salient pixels, we extract descriptors for each keypoint location and then match the descriptors of image A with the descriptors of image B, and descriptors of image A with descriptors of image B'. Please notice that the positions of the correct matches can be found using g and g' in this setup. For each image, the keypoint position (x, y) of a descriptor that passes the mutual nearest and ratio matching tests (and that is also a correct match) is added to the set C_i, where i is the index of the keypoint for image A, B, or B'. We train our model to detect keypoints using the location of correct matches of descriptors as ground-truth for the training. We name the generated map using true matches by *Matching Heatmap* (MH). This process is summarised in Figure 3.

Let M_{a1}, M_{a2} be the MH of A, and M_{b1}, M_{b2} the MH of B and B' respectively (as can be seen in Figure 3), with values ranging in $[0, 1]$, where the value 0 means low matching confidence regions and 1 means high matching confidence regions. The MH has the same resolution of the input image and then we set the MH value as 1 in the position (x, y) if it is in the set C_i. In the last step, we combine the MH from all pairwise matches in a way that map locations have more weight where descriptors were correctly matched on both match attempts, *i.e.*, matches of image A with B and A with B'. As a result, we have a final MH for image A as: $M_a = (M_{a1} + M_{a2})/2$. For images B and B', we apply a similar idea, except that now it has three degrees of weights. Considering image B, we have descriptors that are correct in both image pairs; descriptors that are correct in the match of B and A, and descriptors that are correct in the match of B' and A. The latter is also represented on MH of B, but with a small weight. The same idea is applied to B'. That way, we have the global MHs for B: $M_b = (g(M_a) + M_{b1})/2$, and for B': $M_{b'} = (g'(M_a) + M_{b2})/2$. To make the global MHs easier to be learned by the CNN model, we apply a 3×3 Gaussian kernel in all invidual MHs.

Finally, the matching map has the information of confident locations to be matched for each image. Notice that by choosing only correct matches, we are consequently selecting repeatable keypoints, meaning that our model implicitly learns to be repeatable. To further enforce repeatability of detected points, we also employ a Siamese scheme [29] to maximize similarities of the score map and the MH of the anchor image and its variations, at same time. These strategies improves the repeatability of the detector under geometric transformations. Our method is agnostic to the choice of the base detector and descriptor. In the experiments section, we will show the capability of the proposed detection approach to improve the matching capability of two recent descriptors.

B. *Loss function*

Due to the imbalance between the number of positive and negative pixels in the MH, the full map in the training stage tends to bias the model towards predicting a map with very low scores on average. To solve this problem, we randomly sample a fixed number of negative examples at each pass of an image in training. Considering that n is the amount of positive examples in an image, we uniformly sample n negatives examples and back-propagate $2n$ examples. We formulate this strategy as a binary pixel-wise mask F having value 1 on the chosen pixels, and 0 otherwise. Given an image I, its relative MH M, and the model output score map S, we define $S' = S \times F$. As we aim to maximize the similarity across the MH and the score map image, the cosine similarity ($cossim$) between S' and M is adopted:

$$\mathcal{L}_{cossim}(I) = 1 - cossim\left(S', M\right). \tag{1}$$

When $cossim(S', M)$ is maximized, the MH and the score map tend to be close. Although cosine similarity has good convergence properties, it disregards the magnitude of the values between the score maps and therefore we also consider the L2 loss:

$$\mathcal{L}_{simple}(I) = \frac{1}{2n} \sum_{i=1}^{H \cdot W} \left(S'_i - M_i\right)^2. \tag{2}$$

We further exploit the fact that the regressed map needs to peak at the position of the keypoints. Thus, for even faster convergence, we employ a third loss term in order to force local peakiness of the score map. Considering a set of non overlapping patches $\mathcal{P} = \{p\}$ that contains all $N \times N$ patches within the image I where there is at least one non-zero pixel on the equivalent location of the patch on M, the peakiness loss term of the score map is defined as:

$$\mathcal{L}_{peak}(I) = 1 - \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \left(\max_{(i,j) \in p} S_{i,j} - \operatorname*{mean}_{(i,j) \in p} S_{i,j} \right). \tag{3}$$

The final loss $\mathcal{L}$ is given by the weighted sum of the *cossim*, L2 and peak losses:

$$\mathcal{L}(I) = \lambda_1 \mathcal{L}_{cossim}(I) + \lambda_2 \mathcal{L}_{simple}(I) + \lambda_3 \mathcal{L}_{peak}(I). \tag{4}$$

C. Implementation details

The weights λ_1, λ_2, and λ_3 were empirically found by performing grid-search on a range of sensible values, and we kept the ones that best enhanced the convergence of the score maps. The weights used in the experiments are $\lambda_1 = 3.0$, $\lambda_2 = 1.0$ and $\lambda_3 = 0.3$. Even though most of the results are from real images, our network is trained using only synthetic warps. We use part of [19] simulated data to apply the non-rigid deformations and homography as explained on Subsection III-A. The dataset comprises 400×300 resolution images. In the train step, a random image from the dataset is chosen as the anchor image A (see Subsection III-A). In total, $10K$ pairs of images with different and random transformation were used in the training pipeline. We optimize the network via Adam with initial learning rate of 0.006, scaling it by 0.9 every 500 steps for 7 epochs. We used a batch size of 12 images containing at least 32 peaks in its MH. With approximately 150 positive examples per MH, our model was trained on about $1.5M$ positive examples. In the testing, we used non-maximum suppression (NMS) with a window size of 5×5 pixels. We also post-process the keypoints with an edge elimination step such as SIFT [2] (with a threshold of 10). The top-k keypoints regarding detection scores are kept, while filtering those whose scores are lower than 0.2.

IV. Experiments and results

Our approach expects a base detector and a base descriptor to be used for generating the data to our training framework. Given the wide variety of detectors and descriptors available, we chose ASLFeat [9] because in addition to being the state-of-the-art for the detection-description task, its architecture has deformable convolutional kernels. The deformable kernels target to learn dynamic receptive fields to accommodate the ability of modelling geometric variations, which is a nice feature in our context since we are dealing with non-rigid transformations. We also selected the DEAL [19] descriptor that has invariance to non-rigid deformations.

We evaluate our detector 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 [15], [16] and one by DeSurT [30]. They contain color images of 11 deforming real-world objects.

A. Metrics and baselines

Since the main goal of our feature detection is to maximize the number of correct feature matches, the main performance assessment is done with the Mean Matching Accuracy (MMA) as in Revaud *et al.* [7], which computes the ratio of correct matches and possible matches; and the Matching Score (MS) [7]. We also use the keypoint Repeatability Rate (RR), the most used keypoint metric, defined as the ratio of possible matches and the minimum number of keypoints in the shared view with a 3 pixels error threshold.

We compare our results against five detectors. We consider two handcrafted detectors: SIFT [2] and AKAZE [31], which provides stable keypoints and are still considered good baselines according to a recent study [32]; FAST [33], a basic corner detector; Keynet [6], a cutting edge learning-based detector; and one state-of-the-art jointly learned detection and description method, ASLFeat [9].

B. Results on sequences of deformable objects

Table I shows the results of the experiments using the keypoint detectors for matching image pairs. We detect 1,024 points for each detector on each image. Our main experiment aimed to analyze how the detected keypoints influence the quality of the matching. For this purpose, we chose two descriptors: DEAL [19] and ASLFeat [9]. One can see that, on average, our keypoints paired with DEAL descriptors outperforms all detector-DEAL combinations in both MS and MMA metrics. It is also clear that for the ASLFeat descriptor, our detector reaches the best MMA for all datasets. It is worth mentioning that our method increases the avg. MMA

TABLE I

Detector matching performance comparison. Best in bold and second-best underlined. The higher the value, the better. Results show that our detector provides keypoints on images with non-rigid deformations that enhances the matching results.

Dataset 770 pairs total - MS / MMA@3 pixels									
Detector + ASLFeat	Kinect1	Kinect2	DeSurT	**Mean**	**Detector + DEAL**	Kinect1	Kinect2	DeSurT	**Mean**
SIFT	0.35 / 0.77	0.37 / 0.85	0.26 / 0.63	0.33 / 0.75	SIFT	0.33 / 0.68	0.38 / **0.85**	0.27 / **0.63**	0.33 / 0.72
FAST	0.43 / 0.69	**0.53** / 0.85	**0.33** / 0.56	**0.43** / 0.70	FAST	0.36 / 0.58	**0.51** / 0.81	**0.29** / 0.49	0.39 / 0.63
AKAZE	0.39 / 0.66	0.49 / 0.76	0.26 / 0.48	0.40 / 0.66	AKAZE	0.38 / 0.65	0.47 / 0.74	0.23 / 0.42	0.36 / 0.60
Keynet	0.31 / 0.65	0.35 / 0.62	0.24 / 0.51	0.30 / 0.59	Keynet	0.27 / 0.58	0.34 / 0.59	0.22 / 0.45	0.28 / 0.54
ASLFeat	0.31 / 0.58	0.39 / 0.69	0.28 / 0.53	0.33 / 0.60	ASLFeat	0.31 / 0.66	0.40 / 0.73	0.25 / 0.54	0.32 / 0.64
Ours	**0.49 / 0.86**	0.48 / **0.89**	0.31 / **0.66**	**0.43 / 0.80**	Ours	**0.45 / 0.79**	0.46 / **0.85**	0.28 / 0.59	**0.40 / 0.74**

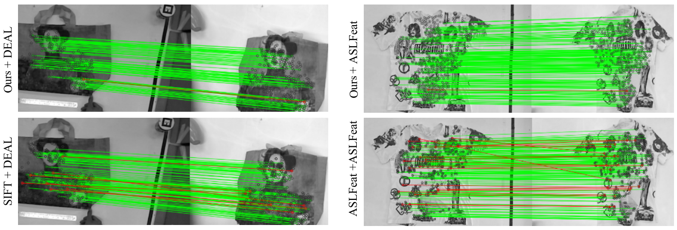

Fig. 4. **Qualitative results on a real non-rigid matching example.** The green lines show correct correspondences, while red lines depict wrong correspondences. Our keypoint detector improves matching accuracy of DEAL and ASLFeat descriptors with respect to SIFT and ASLFeat detectors, while maintaining a similar number of total matches.

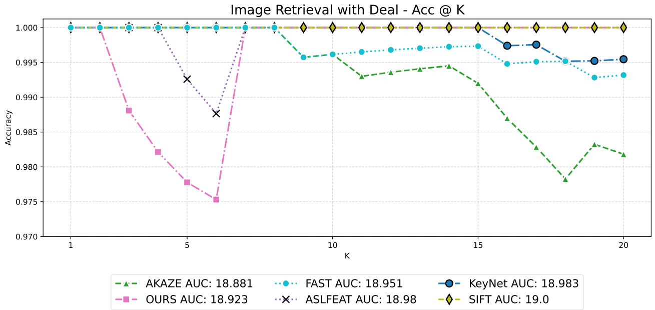

Fig. 5. **Non-rigid object retrieval application.** The graph shows the retrieval accuracy@K for $K = 20$ using a nonrigid descriptor and various detectors.

scores from 0.60 to 0.80 (20 p.p.) when replacing ASLFeat's detector with our detector, and has a significant distance of 5 p.p. from the second best MMA (SIFT-ASLFeat). For the DEAL descriptor, our detector achieves most of the best and second-best MS and MMA scores, increasing, on average, about 7 p.p. and 2 p.p. for MS and MMA, respectively, in comparison with SIFT detector used to train the DEAL descriptor. Figure 4 shows a matching example of our detector combined with different descriptors. Our method is able to deliver well-distributed matches in the image as well as the SIFT and ASLFeat detectors, but with improved accuracy.

We also analyze the detectors' repeatability with the RR scores. Of all methods, FAST has the best RR with 0.59 in average. As second best, AKAZE, ASLFeat, and our method reach a RR of 0.50. The detector with the worst RR metric was SIFT with 0.43. These scores show that our detector could also reach a competitive RR, while increasing the MMA and MS matching metrics. It is worth to notice that the ASLFeat detector, even if presenting a high RR has the smaller MS and MMA, as can be seen in Table I. One can also note that a high RR does not imply good matching scores.

C. *Results on the application of object retrieval*

To further demonstrate the effectiveness of our detector in potential applications, we performed experiments in one important related real-world task: content-based object retrieval. The goal is to retrieve the top K images corresponding to a given query. To represent each image, we used a Bag-of-Visual-Words approach. For each keypoint, we first construct a visual dictionary with the DEAL [19] descriptor, which is used to compute a global descriptor for each image. Given a query image, we calculate the global descriptor and use the K-Nearest Neighbor search to obtain the top K closest objects. We use the same datasets of Table I. We use retrieval accuracy (the number of correct objects retrieved in the top K images) to evaluate the performance of the detectors. Since the queries and database of the application are deformable, we choose only to use a descriptor that models isometric deformations. Figure 5 shows the retrieval accuracy for $K = 20$, where our detector performed similarly to the other methods. For $K > 6$, our detector performed similar to SIFT, with is the method used to train the nonrigid descriptor. The results indicate that our detector can perform well even on a non-matching task.

D. *Ablation and sensitivity analysis*

As part of ablation studies, we train our model using two configurations: (i) a Siamese network scheme and (ii) using a standard network training scheme, *i.e.*, using a single branch. The experiments show that a Siamese scheme (i) helps the model to learn repeatable keypoints and improve MS. RR increased from 0.46 to 0.50, and MS from 0.39 to 0.43 by using the Siamese scheme, and MMA decreased from 0.81 to 0.80, however inliers significantly increased from 150 to 170.

To evaluate the contribution of the components of our proposed loss (Equation 4), and support our implementations decisions, we evaluate three different setups: (i) using cosine

TABLE II
ABLATION ANALYSIS. THE HIGHER THE BETTER.

Loss combination	**RR**	**MS**	**MMA@3**	**Inliers**
$\mathcal{L}_{cossim}$	0.42	0.37	**0.81**	121
$\mathcal{L}_{cossim} + \mathcal{L}_{simple}$	0.48	0.39	0.80	155
$\mathcal{L}$ (complete)	**0.50**	**0.43**	0.80	**170**

similarity term only; (ii) full loss of Equation 4 with equal weights to cosine similarity and L2 losses, *i.e.*, $\lambda_1 = 1.0$, $\lambda_2 = 1.0$, and $\lambda_3 = 0.3$; and (iii) the *complete loss* of Equation 4 with optimal weights. The results in Table II show that setup (iii) is the best one. MMA@3 with a value of 1 p.p. higher for the setup (i) can be explained by the smaller number of inliers. *Complete loss* setup has a higher number of inliers and MS, maintaining a high MMA@3. To support the weighting step choice in our training framework, we also report the strategy of giving equal weights according to repeatable matching, where all MHs peaks has a constant value of 1.0. We obtained values of 0.45 and 0.37 for the RR and MS on equal weights strategy, which is significantly lower than what we achieve using the proposed weighted Matching Heatmap strategy (0.50 and 0.43 for RR and MS, respectively).

V. CONCLUSION

We introduce a novel approach to detect keypoints on images affected by non-rigid deformations, emphasizing improved matching scores. To that end, we proposed a training framework for training a CNN with non-rigidly deformed images exploring the hypothesis that a detector can learn the likelihood of correct matching for a given descriptor. The experimental results show that our method achieved the state-of-art detection and matching performance on non-rigid deformation datasets. Through extensive investigation, we observed that the repeatability of the detector alone is not enough to make a good detector. We also show the efficiency of our detector in a real-world application, demonstrating that learning to detect good keypoints is a promising research direction for performance improvement in real-world tasks. A limitation of our work is that the framework still depends on a base keypoint detector, and may be biased toward specific local characteristics of the base detector. As a future work direction, we plan to investigate how to remove the base detector from the pipeline and learning to detect directly from the descriptor.

ACKNOWLEDGMENTS

The authors would like to thank CAPES, CNPq, FAPEMIG, Google and Petrobras for funding different parts of this work. We also thank NVIDIA for the donation of a Titan XP GPU.

REFERENCES

[1] C. Harris, M. Stephens *et al.*, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, 2004.

[3] J. Heinly, J. L. Schonberger, E. Dunn, and J.-M. Frahm, "Reconstructing the world in six days (as captured by the yahoo 100 million image dataset)," in *CVPR*, 2015.

[4] G. Potje, G. Resende, M. Campos, and E. R. Nascimento, "Towards an efficient 3D model estimation methodology for aerial and ground images," *Machine Vision and Applications*, Sep 2017.

[5] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *CVPR*, 2020.

[6] A. B. Laguna and K. Mikolajczyk, "Key.net: Keypoint detection by handcrafted and learned CNN filters revisited," *TPAMI*, 2022.

[7] J. Revaud, C. De Souza, M. Humenberger, and P. Weinzaepfel, "R2D2: Reliable and repeatable detector and descriptor," *NeurIPS*, 2019.

[8] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-net: A trainable CNN for joint description and detection of local features," in *CVPR*, 2019.

[9] 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 *CVPR*, 2020.

[10] M. Tyszkiewicz, P. Fua, and E. Trulls, "DISK: learning local features with policy gradient," *NeurIPS*, 2020.

[11] S. Suwanwimolkul, S. Komorita, and K. Tasaka, "Learning of low-level feature keypoints for accurate and robust detection," in *WACV*, 2021.

[12] L. O. Vasconcelos, E. R. Nascimento, and M. F. Campos, "KVD: Scale invariant keypoints by combining visual and depth data," *Pattern Recognition Letters*, vol. 86, pp. 83–89, 2017.

[13] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *ECCV*, 2006.

[14] Y. You, W. Liu, Y.-L. Li, W. Wang, and C. Lu, "UKPGAN: Unsupervised keypoint ganeration," *arXiv preprint arXiv:2011.11974*, 2020.

[15] E. R. Nascimento, G. Potje, R. Martins, F. Cadar, M. F. Campos, and R. Bajcsy, "GEOBIT: a geodesic-based binary descriptor invariant to non-rigid deformations for RGB-D images," in *ICCV*, 2019.

[16] G. Potje, R. Martins, F. Cadar, and E. R. Nascimento, "Learning geodesic-aware local features from RGB-D images," *CVIU*, vol. 219, p. 103409, 2022.

[17] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *ICCV*, 2011.

[18] Z. Luo, T. Shen, L. Zhou, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan, "ContextDesc: local descriptor augmentation with cross-modality context," in *CVPR*, 2019.

[19] G. Potje, R. Martins, F. Chamone, and E. Nascimento, "Extracting deformation-aware local features by learning to deform," *NeurIPS*, 2021.

[20] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: self-supervised interest point detection and description," in *CVPRW*, 2018.

[21] Y. Song, L. Cai, J. Li, Y. Tian, and M. Li, "SEKD: self-evolving keypoint detection and description," *arXiv preprint arXiv:2006.05077*, 2020.

[22] N. Savinov, A. Seki, L. Ladicky, T. Sattler, and M. Pollefeys, "Quad-networks: unsupervised learning to rank for interest point detection," in *CVPR*, 2017.

[23] L. Zhang and S. Rusinkiewicz, "Learning to detect features in texture images," in *CVPR*, 2018.

[24] P. Truong, S. Apostolopoulos, A. Mosinska, S. Stucky, C. Ciller, and S. D. Zanet, "GLAMpoints: greedily learned accurate match points," in *ICCV*, 2019.

[25] A. Tonioni, S. Salti, F. Tombari, R. Spezialetti, and L. D. Stefano, "Learning to detect good 3D keypoints," *IJCV*, vol. 126, no. 1, 2018.

[26] F. Moreno-Noguer, "Deformation and illumination invariant feature point descriptor," in *CVPR*, 2011.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015.

[28] G. Donato and S. Belongie, "Approximate thin plate spline mappings," in *ECCV*, 2002.

[29] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," *NeurIPS*, 1993.

[30] T. Wang, H. Ling, C. Lang, S. Feng, and X. Hou, "Deformable surface tracking by graph matching," in *ICCV*, 2019.

[31] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE features," in *ECCV*, 2012.

[32] 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," *IJCV*, vol. 129, no. 2, 2021.

[33] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *ECCV*, 2006.