UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Welerson Augusto Lino de Jesus Melo

LEARNING TO DETECT GOOD KEYPOINTS TO MATCH NON-RIGID OBJECTS IN RGB IMAGES

Belo Horizonte
2023

Welerson Augusto Lino de Jesus Melo

LEARNING TO DETECT GOOD KEYPOINTS TO MATCH NON-RIGID OBJECTS IN RGB IMAGES

Final Version

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Erickson Rangel do Nascimento
Co-Advisor: Renato José Martins

Belo Horizonte
2023

Melo, Welerson Augusto Lino de Jesus.

M528l Learning to detect good keypoints to match non-rigid objects in RGB images [recurso eletrônico] / Welerson Augusto Lino de Jesus Melo - 2023.

1 recurso online (53 f. il., color.) : pdf.

Orientador: Erickson Rangel do Nascimento
Coorientador: Renato José Martins
Dissertação (Mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciências da Computação.
Referências: f. 49-53

1. Computação – Teses. 2. Visão por computador – Teses. 3. Aprendizado profundo - Teses. 4. Detecção de Objetos – Teses. I. Nascimento, Erickson Rangel do. II. Martins Renato José. III. Universidade ederal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Computação. IV.Título.

CDU 519.6*84(043)

Ficha catalográfica elaborada pela bibliotecária Irénquer Vismeg Lucas Cruz
CRB 6/819 - Universidade Federal de Minas Gerais - ICEx

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

LEARNING TO DETECT GOOD KEYPOINTS TO MATCH NON-RIGID OBJECTS IN RGB IMAGES

WELERSON AUGUSTO LINO DE JESUS MELO

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

Prof. Erickson Rangel do Nascimento - Orientador
Departamento de Ciência da Computação - UFMG

Prof. Renato José Martins - Coorientador
Escola de Engenharia - Universidade da Borgonha

Prof. William Robson Schwartz
Departamento de Ciência da Computação - UFMG

Prof. Thiago Luange Gomes
Departamento de Informática - UFV

Prof. André Filgueiras Araújo
Google Research

Belo Horizonte, 23 de fevereiro de 2023.

Documento assinado eletronicamente por **Renato José Martins**, **Usuário Externo**, em 15/09/2023, às 13:52, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Thiago Luange Gomes**, **Usuário Externo**, em 15/09/2023, às 17:04, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Erickson Rangel do Nascimento**, **Professor do Magistério Superior**, em 18/09/2023, às 08:56, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **André Filgueiras de Araújo**, **Usuário Externo**, em 18/09/2023, às 09:45, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **William Robson Schwartz**, **Professor do Magistério Superior**, em 19/09/2023, às 09:12, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2601780** e o código CRC **2248C5F8**.

Acknowledgments

Eu gostaria de agradecer a todos que de alguma forma contribuiu para o desenvolvimento deste trabalho, especialmente as seguintes pessoas e entidades:

- Meu orientador, Erickson R. Nascimento, que me guiou durante esta jornada. Ele sempre foi muito prestativo e contribuiu de diversas formas para o desenvolvimento deste trabalho.
- Ao meu coorientador, Renato J. Martins, que contribuiu no desenvolvimento do trabalho, assim como no meu pessoal.
- Aos meus colegas do VeRLab, em especial ao Guilher Potje, por toda a assistência e conhecimento passado durante o desenvolvimento do trabalho, o que sem ele não conseguiria proseguir em vários momentos. Assim como ao Felipe Cadar, pela parceria e auxílio durante todo o trajeto, assim como ao fazer com que este trabalho fosse mais completo
- Aos meus pais, Carlos e Vânia, por me auxiliarem em tudo que precisei e me apoiarem sempre nas minhas decisões.
- À minha esposa, Francyelle, por estar ao meu lado sempre me apoiando e auxiliando em todos os momentos e decisões.
- À CAPES, CNPq, FAPEMIG, Google e Petrobras pelo financiamento de diferentes partes deste trabalho. Também agradecemos à NVIDIA pela doação de uma GPU Titan XP.
- À Deus, pelas oportunidades.

Resumo

Detecção, descrição e correspondência de pontos de interesse são componentes essenciais de muitas aplicações de visão computacional. Ao longo dos anos, vários algoritmos foram propostos para resolver tarefas de detecção e descrição de pontos de interesse. Com a revolução do aprendizado profundo, os métodos baseados em algoritmos de aprendizado para detecção e descrição de pontos de interesse superaram os métodos artesanais. A fim de melhorar a correspondência, propomos a detecção e descrição de pontos de interesse aprendidos em conjunto. No entanto, esses métodos pretendem melhorar as correspondências de forma indireta por meio da similaridade dos descritores. Devido a isso, alguns métodos propõem incluir correspondências no pipeline de treinamento, porém não com correspondências verdadeiras dos descritores que estão treinando, culminando em um baixo número de correspondências corretas. Além disso, os métodos para detectar pontos de interesse não se preocupam com a deformação não rígida dos objetos; portanto, a robustez a deformações não rígidas também é um fator chave a ser considerado ao localizar pontos para correspondência visual. Neste trabalho, mostramos que um alto número de correspondências corretas pode ser alcançado aprendendo como detectar bons pontos de interesse independentemente do método descritor. E apresentamos um novo método de aprendizado de máquina para a detecção de ponto-chave projetado para maximizar o número de correspondências corretas para a tarefa de correspondência de imagem não rígida. Nossa estratégia de treinamento usa correspondências verdadeiras, obtidas combinando pares de imagens anotadas com um extrator de descritor predefinido, como ground-truth para treinar uma rede neural convolucional (CNN) de maneira semi-supervisionada. Otimizamos a arquitetura do modelo aplicando transformações geométricas conhecidas às imagens como sinal de supervisão. Experimentos mostram que nosso método supera os detectores de ponto-chave existentes em imagens reais de objetos não rígidos em 20 p.p. na Mean Matching Accuracy e também melhora o desempenho da correspondência de vários descritores quando acoplados ao nosso método de detecção. Também empregamos o método proposto em uma aplicação desafiadora: recuperação de objetos, ao qual o nosso detector apresenta desempenho no mesmo nível dos melhores detectores de ponto-chave disponíveis.

Palavras-chave: Objetos deformáveis, correspondência visual, matching de pontos-chaves.

Abstract

Keypoint detection, description, and matching are essential component of many computer vision applications. Throughout the years numerous algorithms were proposed to solve keypoint detection and description tasks. With the deep learning "revolution", learned keypoint detection and description methods surpassed hand-crafted ones. In order to improve matching, joint-learned keypoint detection, and description were proposed. However, these methods intend to improve matching indirectly through the similarity of the descriptors. Because of that, some methods propose to include matching in the training pipeline, but not with true matches of the descriptors they are training, culminating in a low number of correct matches. In addition, methods to detect keypoints are not concerned with non-rigid deformation of objects; therefore, robustness to non-rigid deformations is also a key factor to consider while locating points for visual correspondence. In this work, we claim that a high number of correct matches can be achieved by learning how to detect good keypoints independently of the descriptor method. 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) in a semi-supervised fashion. 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 application: object retrieval, where our detector exhibits performance on par with the best available keypoint detectors.

Keywords: Deformable Objects, Visual Correspondence, Matching

List of Figures

1.1 Example of object deformation over time. 13

2.1 ASLFeat method. 18

4.1 Results on detecting good keypoints. 27
4.2 Overview of the network architecture used as a backbone for keypoint detection. 28
4.3 Overview of the detector training framework. 29

5.1 Image sample of the used dataset. 35
5.2 Experimental Detector. 37
5.3 Qualitative results on a real non-rigid matching of dataset Kinect1/Bag 41
5.4 Qualitative results on a real non-rigid matching of dataset Kinect1/Blanket . . 42
5.5 Qualitative results on a real non-rigid matching of dataset Kinect1/Shirt1 . . 43
5.6 Non-rigid object retrieval application. 46

List of Tables

5.1 **Experiment on Network with fine-tuning.** The higher the better. Bold is the best for the column. 37
5.2 **Ablation of siamese trainning architecture.** The higher the better. Bold is the best for the column. 38
5.3 **Sensibility Analysis on loss.** The higher the better. Bold is the best for the column. 39
5.4 **Detector + ASLFeat descriptor matching performance comparison.** Best in bold and second-best underlined. The higher the value, the better. . . 39
5.5 **Detector + DEAL descriptor matching performance comparison.** Best in bold and second-best underlined. The higher the value, the better. . . 40

Contents

1 Introduction **12**

1.1 Objective and Contributions . . . 14

1.2 Thesis Organization . . . 15

2 Theoretical Background **16**

2.1 Keypoint detection, description, and matching . . . 16

2.1.1 SIFT detector and descriptor . . . 16

2.1.2 Convolution Neural Network for keypoint detection and description 17

2.1.3 ASLFeat detector and descriptor . . . 17

2.1.4 Brute force matching and ratio test . . . 18

3 Related Work **19**

3.1 Handcrafted methods . . . 19

3.1.1 Handcrafted detectors . . . 19

3.1.2 Handcrafted descriptors . . . 20

3.2 Learned-based methods . . . 20

3.2.1 Learned-based detectors . . . 21

3.2.2 Learned-based descriptors . . . 22

3.2.3 Jointly learned detector and descriptors . . . 23

3.3 Keypoint detection and description for non-rigid deformations . . . 24

4 Methodology **26**

4.1 Network design . . . 26

4.2 Keypoint detection learning framework . . . 29

4.3 Loss function . . . 31

5 State of the art comparison **33**

5.1 Implementation details . . . 33

5.2 Datasets . . . 34

5.3 Metrics and baselines . . . 34

5.4 Ablation and sensitivity analysis . . . 36

5.4.1 Network fine-tuning . . . 36

5.4.2 Ablation of siamese training architecture . . . 38

5.4.3 Ablation on loss function . . . 38

5.4.4 Sensitivity analysis on keypoint weights of training framework . . . 39
5.5 Experiments . . . 39
5.5.1 Quantitative Results . . . 44
5.6 Results on the application of object retrieval . . . 45

6 Conclusion **47**
6.1 Future Works . . . 47

Bibliography **49**

Chapter 1

Introduction

High-quality matching of features of images from cameras in different poses and conditions is an important challenge in several computer vision tasks. Applications tasks such as Content-Based Image Retrieval, Structure-from-Motion (SfM) [15, 40], tracking points as a camera or an object of interest [54], and registration [50, 37], are examples of tasks which high-quality feature matching is crucial. The first step for feature matching is to select a set of locations in the images with some properties, such as corners or blobs, to be further described and matched; these kind of localized features are called interest points or simply keypoints. DeTone et al. [9] define keypoints in RGB images as 2D locations in an image that are stable and repeatable from different lighting conditions and viewpoints. Seminal works such as Harris Corner [14], SIFT [20], and SURF [5] allowed significant advancements in many applications tasks using the above constraints. It is clear that an effective keypoint detector should be repeatable and invariant to different illumination conditions and equivariant to viewpoint and scale changes.

However, knowing that the final purpose of the keypoint is the high quality of the matching, a keypoint detector should also consider matching in its point selection methodology because repeatability does not imply good points to be matched at the end. For example, points on edges and repetitive patterns that usually occur in man-made structures are challenging to be matched due to the high texture ambiguity. Add knowledge of the match on keypoint detection is a big challenge for the hand-crafted detectors due to the fact that we need first to detect to then describe the points and so match them. In addition, there is little knowledge about the real behavior of the descriptors in different contexts, e.g., some descriptors can be less robust on lighting changes. Descriptors are intended to describe a local region, thus the described points are highly discriminative to be found in another image of the same scene. However, a local region detected by a specific detector can be a good region to a descriptor, increasing the number of correct matches, and, at the same time, can be a bad region to another descriptor, degrading matching performance. To surpass that problem, recent works use deep learning techniques for the tasks of detecting and describing keypoints such that keypoint detection and description are in the same learning pipeline; that is called jointly learned approaches [33, 12, 22, 52, 42, 44]. Deep jointly learned methods deliver results

Figure 1.1: Example of object deformation over time.

that significantly outperform the handcrafted counterpart [20, 5].

Jointly learned detection and description methods link detection and description, improving matching performance. However, However, joint learning of detection and description do not consider matching on the pipeline. Performing the matching simultaneously with the detection and description often has high computational complexity [52]. 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 pixel in the image may potentially become a feature, the problem quickly becomes intractable, needing carefully designed training schemes and massive computational resources [43].

Furthermore, beyond rigid transformations, objects may have different shapes over time due to deformations, as it can be observed in Figure 1.1. Therefore, robustness to non-rigid deformations is also a key factor to consider while locating points for visual correspondence. Regarding the non-rigid transformation, very few works have been proposed to address the non-rigid deformation invariance task. The recent explorations, such as the work of Yu [56] propose to tackle non-rigid deformation but rely on depth information. Despite the advances achieved, RGB cameras are still by far the most common type of imaging sensor. Some recent works have treated the description problem in non-rigid deformation images [8, 29, 32]. However, as far as we know, no work has proposed

a detection method to deal with non-rigid deformations for RGB images. Given all this background, can we aggregate knowledge of real matches of a given descriptor in detection without massive computational resources? And can this detector be robust to non-rigid deformations in images? In this thesis, we present a learned detection strategy designed to tackle non-rigid deformations on still images (please see Figure 4.1 for 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 aggregate knowledge of real matches by proposing a novel learned detection methodology that predicts ground-truth matching maps based on an existing detector-descriptor configuration. The network is trained to learn to detect good features according to the map derived from true descriptor matches. It is worth mentioning that our approach can be easily coupled with any combination of existing detector-descriptor pairs. Because of the above characteristic, our method can be used in scenarios in which descriptors cannot be changed and we aim to improve the matching by changing the detector method. We evaluate our detector on three different benchmark datasets (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 4.1 illustrates the behavior of our detector in comparison with the recent ASLFeat detector [22] and the final matching quality of detected keypoints.

1.1 Objective and Contributions

Our goal is to develop a keypoint detector method capable of surpassing the mentioned problems of aggregating knowledge of real matches of a given descriptor on the detector without massive computational resources and, at the same time, creating a detector robust to non-rigid deformations in images.

The main contribution of our work is two-fold:

- Propose and implement a novel keypoint detection training framework aimed to improve the matching performance of existing descriptors;
- Propose and implement the first learned keypoint detector optimized to cope with non-rigid deformations that work only using standard RGB images.

Parts of the results in this thesis were presented and published at the main track

of SIBGRAPI 2022 [23]

1.2 Thesis Organization

This thesis is organized in the following chapters. In Chapter 2, we review the recent state-of-the-art of detection techniques present in the literature. In Chapter 3, we present the proposed method and the implementation details. In Chapter 4, we present experiment details and the evolution of our method. Sequentially, in Chapter 5, we present the results by testing the proposed approach and state-of-the-art detectors and comparing it to recent detectors. Finally, in Chapter 6, we discuss the results and research perspectives.

Chapter 2

Theoretical Background

In this chapter, we explore and detail important concepts and techniques used in keypoint detection and matching of objects with non-rigid deformations, as well as more details on important detection algorithms for our proposed method.

2.1 Keypoint detection, description, and matching

The first step for feature matching is to select a set of locations in the images with some property, such as corners or blobs, to be further described and matched; these kinds of localized features are called interest points or simply keypoints. DeTone et al. [9] define keypoints in RGB images as 2D locations in an image that are stable and repeatable from different lighting conditions and viewpoints. The selected local regions of the image should be described. The descriptor algorithm summarizes the region in a vector that represents uniquely that location with the aim of finding that specific point in another image. Because of that, descriptors should be distinctive and unique. It is common to find detector and descriptor algorithms that are complementary, that is, they are made to be used together. However, they can be used separately, that is the case of the SIFT detector and descriptor, an example of a handcrafted method to detect and describe keypoints.

2.1.1 SIFT detector and descriptor

Distinctive Image Features from Scale-Invariant Keypoints, known as SIFT, is the most cited method of feature detection and description. SIFT is one of the handcrafted methods that implement a mathematical strategy to detect patterns on the image, which are the keypoints. We use the SIFT detector to introduce the handcrafted general strategy

and background.

The SIFT algorithm is composed of four main steps: scale-space extrema detection, keypoint localization, orientation assignment, and feature descriptor generation. The first step, scale-space extrema detection, in the SIFT algorithm is to create a scale-space representation of the input image by convolving the image with a Gaussian kernel at different scales. This is done to detect features at different scales. The scale-space is then searched for extrema, which are points that have maximum or minimum values in both the spatial and scale dimensions. Once the extrema are detected, potential keypoints are identified by comparing them to their neighboring points. The extrema that are not sufficiently stable and repeatable across scales are discarded as they are likely to be noise or background features. After keypoints are identified, their orientation is assigned by taking the gradient magnitude and orientation of the pixel values around the keypoint. The gradient orientation histogram is then generated and the peak orientation is selected as the keypoint's orientation. Finally, a feature descriptor is generated for each keypoint by computing the gradient magnitudes and orientations around the keypoint at the selected scale and orientation. These gradient values are then transformed into a descriptor vector, which is normalized to make it invariant to changes in illumination and contrast.

2.1.2 Convolution Neural Network for keypoint detection and description

Convolutional Neural Networks (CNNs) have been successfully applied to a wide range of computer vision tasks, including object detection, image segmentation, and keypoint detection and description. It is not different for keypoint detection and description. We use ASLFeat [22] as an example of the usage of CNN in keypoint detection description pipeline.

2.1.3 ASLFeat detector and descriptor

ASLFeat uses CNN to extract features from the input image and generate a final score map to select the keypoints location and select the equivalent descriptor of that location.

Figure 2.1 shows ASLFeat CNN architecture for the detection and description of

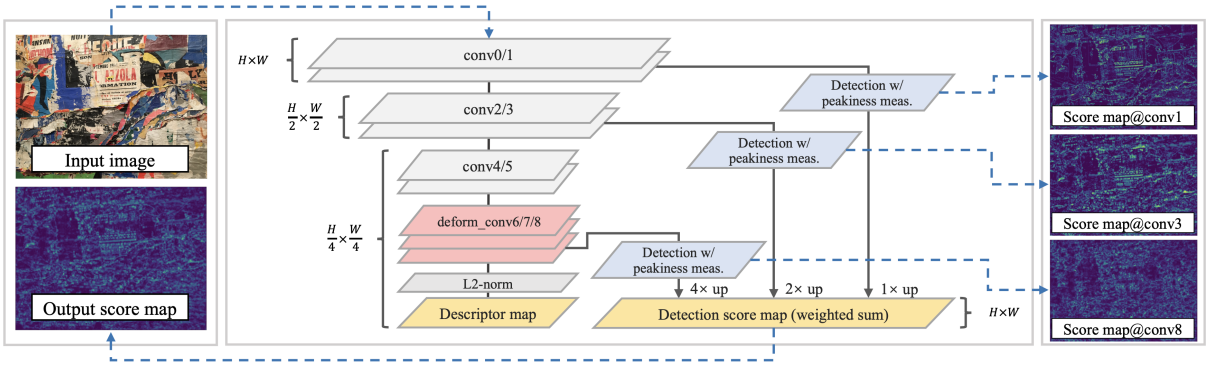

Figure 2.1: ASLFeat [22] detection and description CNN architecture.

keypoints by learning. In the ASLFeat method, the convolutional layers are used to extract features, differently of handcrafted methods that use a mathematical strategy to extract the local features. ASLFeat uses 9 layers to extract features to generate a descriptor map at the end of the layers. ASLFeat uses three different layers to extract the keypoints using a peakness strategy. The peakness strategy selects peaks in the partial score map of the layer. The final score map is generated from a weighted sum of the 3 partial score maps generated. To the convolutional layers extract relevant features to generate the descriptor in the last layer and the keypoints as mentioned above, the training of the network uses two images of the same scene and enforces equal final score maps, as well as equal descriptors for each equivalent pixel across the images. To be invariant to illumination, viewpoint, and other transformations, input images are transformed before passing through the network, forcing the learning of the algorithm to that transformation variation.

Strategies similar to the one use of the ASLFeat are used by other CNN-based keypoint detectors and descriptors.

2.1.4 Brute force matching and ratio test

Given a set of image descriptors computed from two different images, these image descriptors can be mutually matched by for each point finding the point in the other image domain that minimizes the Euclidean distance between the descriptors represented as D-dimensional vectors. To suppress matches that could be regarded as possibly ambiguous, Lowe et al. [20] only accepted matches for which the ratio between the distances to the nearest and the next nearest points is less than 0.8. That strategy is called Ratio Test for matching of features.

Chapter 3

Related Work

This chapter first reviews core feature detection and description techniques that relate to ours. Then, we discuss recent methods for detecting and describing local features designed for handling non-rigid deformations.

3.1 Handcrafted methods

In this section, we present some handcrafted detectors and descriptors through the years and the state-of-the-art of this category.

Such as in many areas of computer science, computer vision has been affected by the machine learning trend. Handcrafted nomenclature was not used till the advance of machine learning in the sub-area of feature extraction. Handcrafted approaches are designed algorithms without a training phase on data to adjust their parameters. They are with data-driven approaches where part of the parameters of the algorithm (or the entire set of paramaters) are set after a tranining phase on some data samples. Classical detectors and descriptors are, most of them, called handcrafted detectors and descriptors. It does not mean that there are no handcrafted parts in the most recent works or in learned methods.

3.1.1 Handcrafted detectors

For many years, handcrafted detectors have dominated the field of detection, with the Harris Corner detector [14] being one of the most widely recognized and utilized methods. Also known as traditional detectors, handcrafted detectors aim to localize geometric structures through engineered algorithms. The Harris and Hessian [6] detectors

use first and second-order derivatives to find corners or blobs in images. Those detectors have been further extended to handle affine transformation and also detect features in multi-scale [26, 24]. The main limitation of these methods lay in the transformation of the geometric structure, such as scale.

Lowe's work, Distinctive Image Features from Scale-Invariant Keypoints, known as SIFT [20], is the most cited detector paper of feature detection. The SIFT algorithm looks for blobs over multiple scale levels. Later, SURF [5] accelerated the detection process based on SIFT scale space by using integral images and an approximation of the Hessian matrix. Multi-scale improvements also were proposed in KAZE [2] and its extension, A-KAZE [1], where the Hessian detector was applied to a non-linear diffusion scale space in contrast to widely used Gaussian pyramid. And most recently, Zhang and Sun have proposed an improvement in corner detection by applying first-order and second-order intensity variation along with multiple directions, the SOAGDD algorithm [58], reaching the state-of-the-art of handcrafted detectors. Handcrafted detectors have the limitation of not being easily adaptable to be robust to non-rigid deformations.

3.1.2 Handcrafted descriptors

Handcrafted descriptors are mostly based on local statistics (e.g., gradients). They use a fixed configuration for region pooling, for example, SIFT descriptor [20] and its variant such SURF descriptor [5]. GLOH descriptor [25] uses a polar arrangement of summing regions, while DAISY [48] employs a set of multi-size circular regions grouped into rings. This approach has the advantage of proving invariant representation of the local patches. Most of the handcrafted local descriptors has a strong rotation equivariance, which most learned models lack [52]. On the other hand, handcrafted descriptors suffer from the same problem of handcrafted keypoint detectors of not to be easily adaptable to be robust to non-rigid deformations since its not easy to model a statistical algorithm to deal with deformations in RGB images.

3.2 Learned-based methods

Learned-based algorithms started gradually to be used for feature extraction tasks. Over the years, deep learning algorithms started to be used by the computer vision com-

munity; they were initially used for high-level tasks. With the growth of data, computational power availability, and the research on deep learning, the deep approaches started to be used for dozen of computer vision low-level tasks, such as feature extraction [45]. Feature extraction and image matching pipelines using learned-based methods became the new gold standard approach. On the flip side, these learning-based approaches are mostly applied to feature description tasks and jointly detection-description.

3.2.1 Learned-based detectors

In the feature detection field, Features from Accelerated Segment Test (FAST) [35] was one of the first attempts to use machine learning to derive a corner keypoint detector. FAST algorithm uses a Decision Tree classifier to speed up the detection task, i.e., the computational efficiency is its most important advantage. Nevertheless, it reaches consistent repeatability results, as could be confirmed in our experiments (see Chapter 5.5). Based on the FAST detector and aiming to have an open-source alternative to SIFT and SURF detector and descriptor, the Oriented FAST and Rotated BRIEF (ORB) [36] method was proposed. ORB performance is similar to the SIFT on the task of feature detection while is almost two orders of magnitude faster. The main idea is that ORB adds an orientation component and multi-scale to the FAST keypoints.

With the success of deep-learned methods in general object detection and feature descriptors, the research community was motivated to explore similar techniques for feature detectors. Thus, CNNs started to be used for the keypoint detection task. TILDE [53] trained multiple piece-wise linear regression models to identify interest points that are robust under severe weather and illumination changes. Lenc and Vedaldi [19] introduced a new formulation to train CNN based on feature co-variant constraints and added predefined detector anchors, showing improved stability in training. QuadNet [39] has focused on learning keypoint detection for repeatability by increasing the keypoint in repeatable areas between image pairs. As QuadNet has employed the ranking loss, Zhang et al. [57] added the grid-wise peakiness for the sparse detection. Laguna and Mikolajczyk [18] have stated that QuadNet [39] and Zhang et al. [57] repeatability is high, but their matched keypoints have low accuracy since the ground truth of keypoints location is not well defined. KCNN [10] and KeyNet [18] resorted to using handcrafted keypoints in training due to the consistent representation of handcrafted keypoints to low-level features. Nevertheless, these methods can provide poor detection if the handcrafted have some bias that could not help to detect good keypoints, e.g., clustering keypoints along edges and corners and fail where handcrafted methods fail [44]. Our method uses a stan-

dard detector on the training process. However, the training framework filters the bad keypoints.

The recent work of Suwanwimolkul et al. [44] enforces the importance of considering low-level features in the detection of good and accurate keypoints. They claim that improving the low-level feature can improve matched keypoint location and descriptor matching as well. Low-Level Feature (LLF) detector [44] is based on R2D2 [33] keypoint detection. LLF in combination with others descriptor algorithms has increased the matching mean score by increasing the keypoint detection accuracy and repeatability. A drawback of this method is that it does not consider real matching scenarios, which, in practice, do not improve the matching accuracy of the detected keypoints. It is worth noticing that some handcrafted approaches, such as SIFT, is still a good baseline method due to its stability across different types of scenes and applications.

3.2.2 Learned-based descriptors

Learning-based feature descriptors can be divided into two groups: the one that applies its method to a sparse set of keypoint detected by a standard keypoint detector and the one that densely describes the image. For the first group, we have the work of Simo-Serra et al. [41] as one of the first works to use deep networks to describe keypoints. After that, we have the work of Balntas et al. [4], Mishchuk et al. [27] and Contextdesc [21]. Contextdesc descriptor, for example, receives a set of keypoints and use visual context encoder that integrates high-level visual understandings from regional image representation and a geometric context encoder that consumes unordered points and exploits geometric cues from 2D keypoint distribution. For the second group, the one that describes densely over the image, i.e., descriptor algorithms that are able to generate a descriptor from each pixel of an image, one of the first works we can find is the work of Savinov et al. [38], Noh et al. [30], and Fathy et al. [13]. Fathy et al., for example, propose a CNN-driven scheme for coarse-to-fine hierarchical matching, as an effective and principled replacement for conventional pyramid approaches to learn more effective dense descriptors in the context of geometric matching tasks.

Learned-based descriptors are usually trained using a metric learning loss that seeks to maximize the similarity of descriptors corresponding to the same patches and minimize it otherwise [33]. This is the case of some works such as Contextdesc [21], the PN-Net [3], L2-Net [46], and Sosnet [47]. Contrastive loss and triplet loss were widely used to train these networks to optimize the global objective based on local comparison with local patches.

3.2.3 Jointly learned detector and descriptors

The detection and description tasks were traditionaly tackled as two separate tasks. We could use a handcrafted detector with a learned descriptor, a learned keypoint with a handcrafted descriptor, or both learned approaches. Because of the correlation between detection and descriptor matching, the most recent works in the feature extraction pipeline have proposed the joint learning of detection and description tasks, which means that the detection and description are in the same network and are trained as a unique task. Some recent works [12, 22] have claimed that detection and description are inseparably tangled. Thus, keypoints should be detected based on the repeatability and reliability of descriptors.

LIFT algorithm [55] was the first to propose the jointly learned approach. In the following, using a large-scale dataset of annotated landmark images [30] trained DELF, an approach targeted for image retrieval that learns local features as a by-product of a classification loss coupled with an attention mechanism. R2D2 [33] uses deep learning to jointly enhance detection and description via learning *discriminability* and give less importance to repeatability for improving the description and matching. In the R2D2 work, the reliability is trained based on the Average-Precision metric while simultaneously optimizing for the descriptor. SuperPoint [9] method presents a network that first extracts salient points and then a transformation between pairs of images. Superpoint network was trained with annotated corners. For other path, D2-Net [12] proposes the *describe-and-detect* approach. The algorithm first computes a set of CNN feature maps; the maps are used to compute the descriptors and detect keypoints. This approach tries to make detection better for matching tasks. D2-Net uses local maxima of feature maps to extract the keypoint. The results of D2-Net were surpassed by other works as the R2D2. Recent works [22, 44] have stated that R2D2, as other jointly learned approaches, has a lack of low-level information and keypoint accuracy since it down-sample the image and does not use a robust method to recover keypoint localization.

Following the idea of D2-Net, the ASLFeat algorithm [22] uses the *describe-and-detect* approach modifying its network and reusing some R2D2 ideas to perform strong shape-awareness geometric invariance and improve keypoint localization. For that, ASLFeat adds for the first time in the feature extraction pipeline the Deformable Convolutional Network (DCN) and detects keypoint in three different layers of the network to, in the end, joint detection score map and make the detection partially scale-invariant, and, make the localization accuracy of detected keypoint in the original image more robust to the down-sample performed by the network. In addition, the network was retrained end-to-end. Also claiming to enhance the low-level feature and keypoint accuracy, the recent LLF method [44] has compared the newest detectors and descriptors in the feature matching

problem.

In general, in jointly learned detection keypoints are selected using thresholds such as the detection score of D2-Net [12] or the local peakiness score of ASLFeat [22]. Therefore, the matched keypoints do not always have high accuracy. Figure 4.1 shows these problems of handcrafted keypoint selection in the ASLFeat methods. The highlighted squared region is an edge region where ASFeat detector applies an edge threshold to select only good points on edges. However, we can see that there are a large number of points clustered over the edges and an incorrect match. Conversely, in our proposed approach, the peaks are generated directly from the network output. And as can be seen in the score map of Figure 4.1-b, the score map generated tries to peak a point, attenuating the high score edges effect. In addition, in our proposed approach, the detector is trained to improve matching accuracy with real image pairs.

Fewer works consider descriptor matching in the training pipeline, such as our approach. GLAM [51] detects keypoints based on matching quality; however, for a very specific domain of retinal images. SEKD [42] 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 [52] considers detection and description in a probabilistic relaxation and applies a reinforcement learning strategy to optimize detection and description jointly. As a drawback, the method requires careful hyperparameter tuning to converge. Tonioni [49] 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 [51], we use the results of matching descriptors, however, with a weighting strategy for the Matching Heatmap, applied to a general domain.

3.3 Keypoint detection and description for non-rigid deformations

Rigid deformations on objects are the one which the position and orientation of points in the object relative to an internal reference frame are not changed; for example, rotation and translation. Non-rigid deformations on objects are the deformations which position and orientation of points within the object are changed relative to both an internal and external reference frame. In our work, we consider isometric non-rigid deformations,

which deformations preserve distances across points in the surface of the object.

To circumvent the problem of non-rigid deformations on keypoint description, descriptors such as the DEAL method [32] propose a deformation-aware local feature description strategy that learns to describe non-rigid patches without depth information. DaLi descriptor [28] encodes features robust to non-rigid deformations and illumination changes. In the same context, Nascimento et al. propose GeoBit descriptor [29], a descriptor that uses geodesics from object surfaces to compute isometric-invariant features working with RGB-D images, while in Geopatch [31] Potje et al. propose a description strategies with the key idea of learning feature representations on undistorted local image patches using surface geodesics working with RGB-D images. Note that none of the above works deal 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, which is a novel accomplishment to the best of our knowledge.

Chapter 4

Methodology

In this chapter, we detail the main steps of our methodology. It is a novel pipeline to detect keypoints with the property of being robust to non-rigid deformation and improve matching tasks. We first designed a network to learn to extract keypoints from images being affected by non-rigid deformations. Secondly, we proposed a training framework to enforces keypoints to be detected in repeatable locations having confident matching probability for a given descriptor.

In the first task, a network learns to extract keypoints from images being affected by non-rigid deformations. We use the same idea of rigid transformation for training a network. However, in addition to homographic changes, we add non-rigid deformations on the training dataset, as we show in the next sections. For the second task, a training framework enforces keypoints to be detected in repeatable locations having a confident matching probability for a given descriptor. We first use the assumption that a model can learn to extract the features of a good descriptor looking at the descriptors and selecting good descriptors based on its matching. As this strategy is not easy to be learned, we use the local information on the input image of the good descriptors, i.e., the region around the keypoint of that descriptor, as a feature to be learned. With that, the model can learn the likelihood of a region generating a good descriptor to be matched for improving the detector.

4.1 Network design

Our designed network receives an image as input and outputs a score map whose peaks are the location of keypoints good to be matched. Works such Superpoint [9], D2-net [12], and ASLFeat [22] networks generate score map from a downsampled tensor of the input image. As the score map needs to be in the same resolution as the input image, the generated score map is upsampled to be in the same resolution as the input image. Our main goal is to output peaks in the score map. In the upsample process, the score

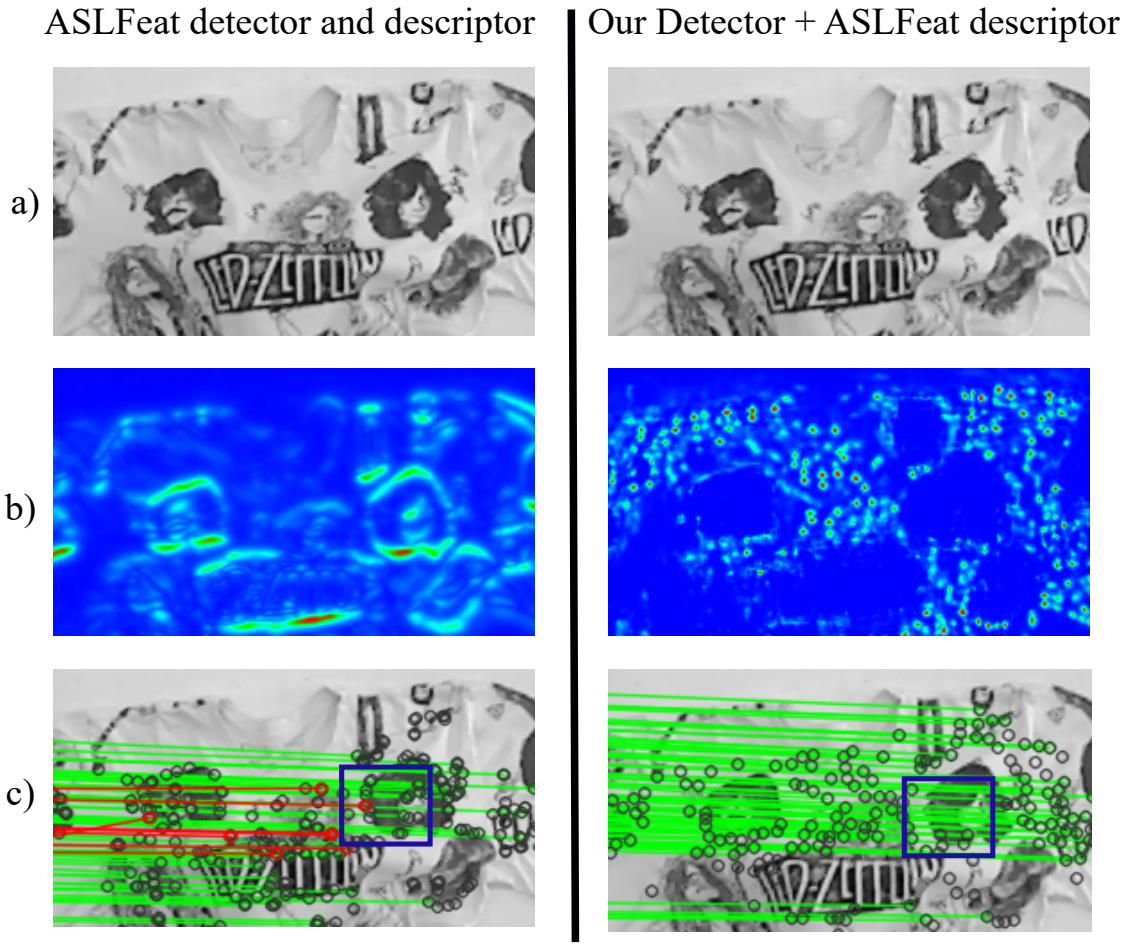

Figure 4.1: **Results on detecting good keypoints.** a) Input image with non-rigid deformations; b) Score maps of ASLFeat and ours; c) Correct matches (green) and incorrect ones (red), as well as circles representing the detected keypoints. One can notice that not all peaks in the score map are keypoints because we chose the top 1,024 according to the score value. Notably, our method provides more reliable points to be matched.

map peaks can be shifted, interfering with detector accuracy and consequently degrading matching quality. Because of that, we adopt a 4-level deep Unet [34] with a final sigmoid activation function as our network architecture. That way, the network outputs a signal to each pixel of the input image.

In the past, Unet [34] was used successfully in dense regression and semantic segmentation tasks. The Unet architecture (Figure 4.2) is composed of 3×3 convolution blocks with batch normalization and ReLU activation. High-resolution features from the contracting path are combined with the upsampled output; a successive convolution layer can then learn to assemble a more precise output based on this information. In the upsampling part Unet has also a large number of feature channels, which allows the network to propagate context information to higher resolution layers. As a consequence, the ex-

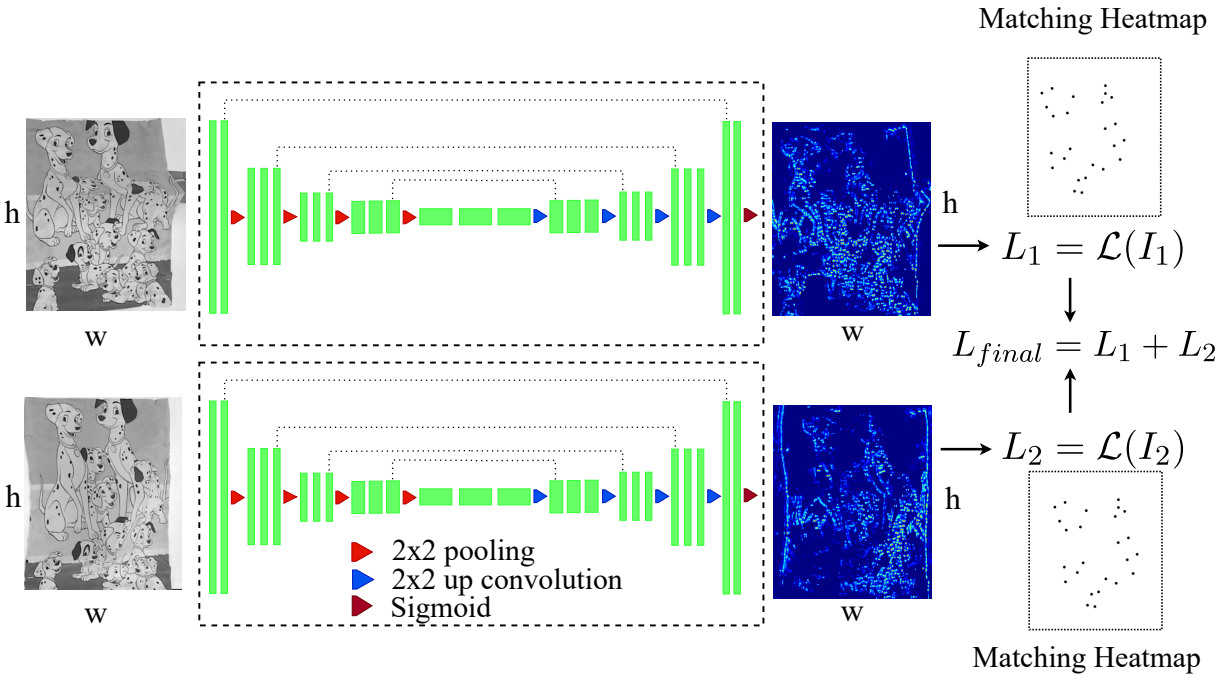

Figure 4.2: **Overview of the network architecture used as a backbone for keypoint detection.** The Siamese network is optimized to detect reliable keypoints to be matched for a given descriptor. Each green block represents a 3×3 2D convolution, followed by batch normalization and ReLU activation function. The two branches share the weights.

pansive path is symmetric to the contracting path and yields a U-shaped architecture that names the architecture.

In the training process, we want the network to learn a score map by imitating a Matching Heatmap that works as a ground-truth. As in the training process we have two images of the same scene, and the same peaks in the two equivalent Matching Heatmaps. Because of that, to improve repeatability of equivalent score maps, we use a Siamese scheme [17]. Siamese networks were first introduced by Bromley and LeCun [7] to solve signature verification as an image matching problem. A Siamese neural network consists of twin networks that accept distinct inputs but are joined by an energy function at the top. This function computes some metrics between the highest level feature representation on each side. Figure 4.2 illustrates that process. The weights of the two branches are shared.

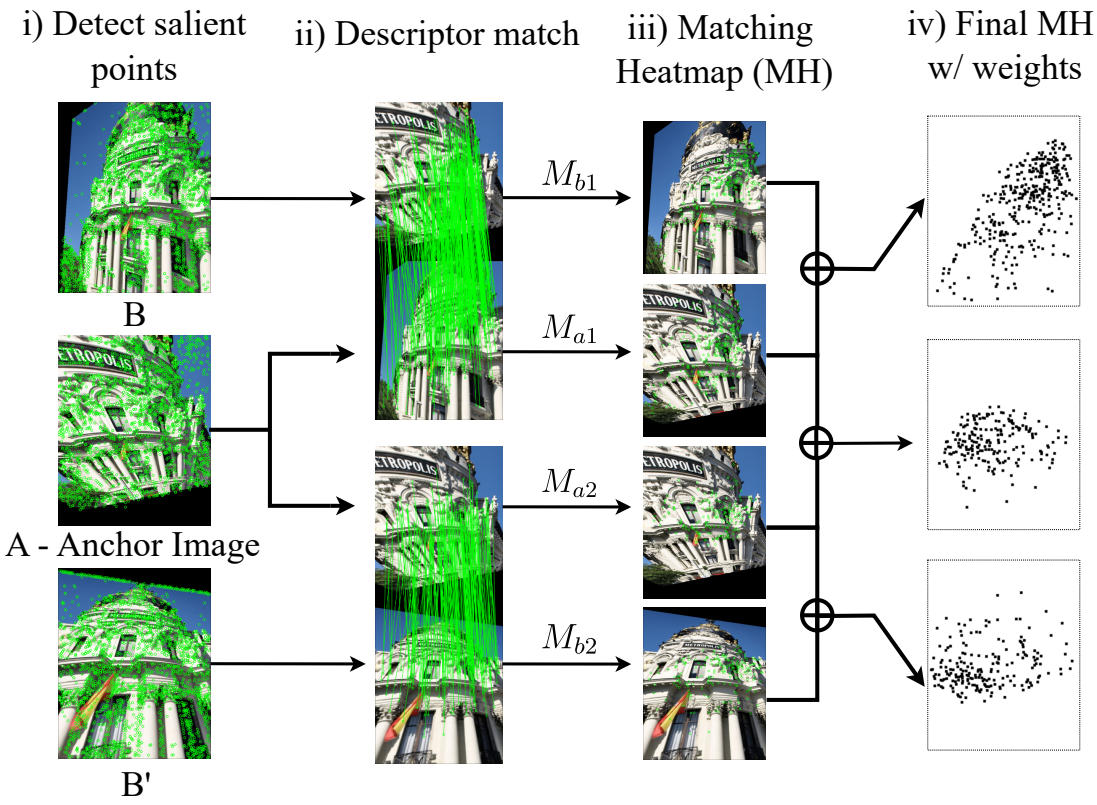

Figure 4.3: **Overview of the detector training framework.** Our framework is composed of four steps: i) First, we detect the keypoints using a base detector for images A, B and B' (an anchor and two transformed versions of the anchor with random homography and non-rigid image transformations); ii) Then, we extract the descriptors on the detected keypoints and then find correspondences with the 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.

4.2 Keypoint detection learning framework

Learned keypoints as Key.net [18], and describe-to-detect extraction methods such as R2D2 [33], and ASLFeat [22] focus their training framework on the repeatability of keypoints, and not consider non-rigid deformations. Unlike these works, in our learning strategy, the key idea is to leverage an existing detector-descriptor pair to bootstrap the learning process that is focused on highly confident matches in images with non-rigid deformations. To that aim, we use correct matches of images with non-rigid deformation of the same scene as a *Matching Heatmap* that works as a guide to the learning process.

Let $A \in R^{H \times W}$ 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) [11] (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. The homography warps give us some invariance to viewpoints' rigid changes. In the sequence, 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 in a semi-supervised manner 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 4.3.

Let M_{a1}, M_{a2} be the MH of A (relative to the matching with B and B' consecutively), and M_{b1}, M_{b2} the MH of B and B' (relative to the matching of both with A), 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 as 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 weight. 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 individual MHs. Finally, the matching map (Figure 4.3) has the information of confident locations to be matched for each image.

In addition, the above strategy increases the number of positive samples since points that were matched only in B are now added with a small weight in B', and vice versa. 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 the repeatability of detected points, we also employ a Siamese scheme [7] to maximize similarities of the score map and the MH of the anchor image and its variations at the same time. This strategy also 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.

4.3 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 the 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:

$$\mathcal{L}_{cossim}(I) = 1 - cossim\left(S', M\right). \tag{4.1}$$

When $cossim(S', M)$ is maximized, the MH and the score map tend to be close. R2D2 [33] uses a similar strategy applying the cosine similarity as its loss function. 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:

$$\mathcal{L}_{simple}(I) = \frac{1}{2n} \sum_{i=1}^{H \cdot W} \left(S'_i - M_i\right)^2. \tag{4.2}$$

We further exploit the fact that the regressed map needs to peak at the position of the keypoints; that fact, in practice, it also allows a better performance of the detector in repeatability and matching scores. Thus, for even faster convergence, we employ a third loss term in order to force the local peakiness of the score map, based on ASLFeat [22] peakiness strategy. 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:

$$\mathcal{L}_{peak}(I) = 1 - \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \left(\max_{(i,j) \in p} S_{i,j} - (i,j) \in pmeanS_{i,j} \right). \tag{4.3}$$

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

$$\mathcal{L}(I) = \lambda_1 \mathcal{L}_{cossim}(I) + \lambda_2 \mathcal{L}_{simple}(I) + \lambda_3 \mathcal{L}_{peak}(I). \tag{4.4}$$

Because of the Siamese scheme, the final loss function becomes the sum of the loss $\mathcal{L}$ applied to the images in both branches. That way, we have:

$$\mathcal{L}_{final} = L_1 + L_1, \tag{4.5}$$

where $L_1 = \mathcal{L}(I_1)$, $L_2 = \mathcal{L}(I_2)$, and I_1 and I_2 are the input images of the Siamese network.

Chapter 5

State of the art comparison

In this chapter, we show the dataset used in our experiments, the evaluation methodology and baseline used, and the ablation study and sensibility analysis to support our decisions and select the best parameters for the final and definitive method.

Our approach expects a base detector and a base descriptor to generate the data for training our framework. Given the wide variety of detectors and descriptors available, we chose ASLFeat [22] because, in addition to being the state-of-the-art detection-description task, its architecture has deformable convolutional kernels. The deformable kernels target learning dynamic receptive fields to accommodate the ability to model geometric variations, which is a important feature in our context since we are dealing with non-rigid transformations. We also selected the DEAL [32] descriptor, which is robust to non-rigid deformations.

5.1 Implementation details

The weights λ_1, λ_2, and λ_3 were empirically found by performing a 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 [32] simulated data to apply the non-rigid deformations and homography as explained on Section 4.2. The dataset comprises 400×300 resolution images. In the training step, a random image from the dataset is chosen as the anchor image A (see Section 4.2). In total, $10,000$ pairs of images with different and random transformations were used in the training pipeline. We optimize the network via Adam with an initial learning rate of 0.006, scaling it by 0.9 every 500 step 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 order to balance examples at each iteration, we randomly select negative examples. And

for each input image, we cut a convex region formed by the positive examples to avoid selecting negative examples on good regions of the image, however not detected due to occlusions, which can, that way, confound the model.

In the testing, we used non-maximum suppression (NMS) with a window size of 5×5 pixels. Just to ensure that we will not have points on edges, we also post-process the keypoints with an edge elimination step as SIFT edge elimination method (with a threshold of 10). The top-k keypoints regarding detection scores are kept, while filtering those whose scores are lower than 0.2.

5.2 Datasets

We evaluate our detector in different publicly available datasets containing deformable objects in diverse viewing conditions such as illumination, viewpoint, and non-rigid deformation. For that, we selected the dataset recently proposed by Nascimento et al. [31], and Potje et al. [29], i.e., Kinect1 and Kinect2, respectively; and one proposed by DeSurT [54]. Each dataset has folders with a base image and target images with some geometric and non-rigid transformation computing a total of 770 images. They contain color images of 11 deforming real-world objects and ground-truth correspondences are done following the protocol of [16]. Kinect1 and Kinect2 datasets have images with non-rigid deformations and small variation on rotation and translation. DeSurt has images with non-rigid deformations, small rotation and perspective changes.

Figure 5.1 shows some examples of images from these datasets and their deformation.

5.3 Metrics and baselines

Since the main goal of our feature detection is to maximize the number of correct feature matches, the performance assessment uses the Mean Matching Accuracy (MMA) in combination with the Matching Score (MS). The MMA metric is computed as in Revaud [33], where the matching accuracy is the average percentage of correct matches in an image pair considering multiple pixel error threshold. In our experiments, we use an error threshold of three pixels, which we call MMA@3. The MS can be defined as the

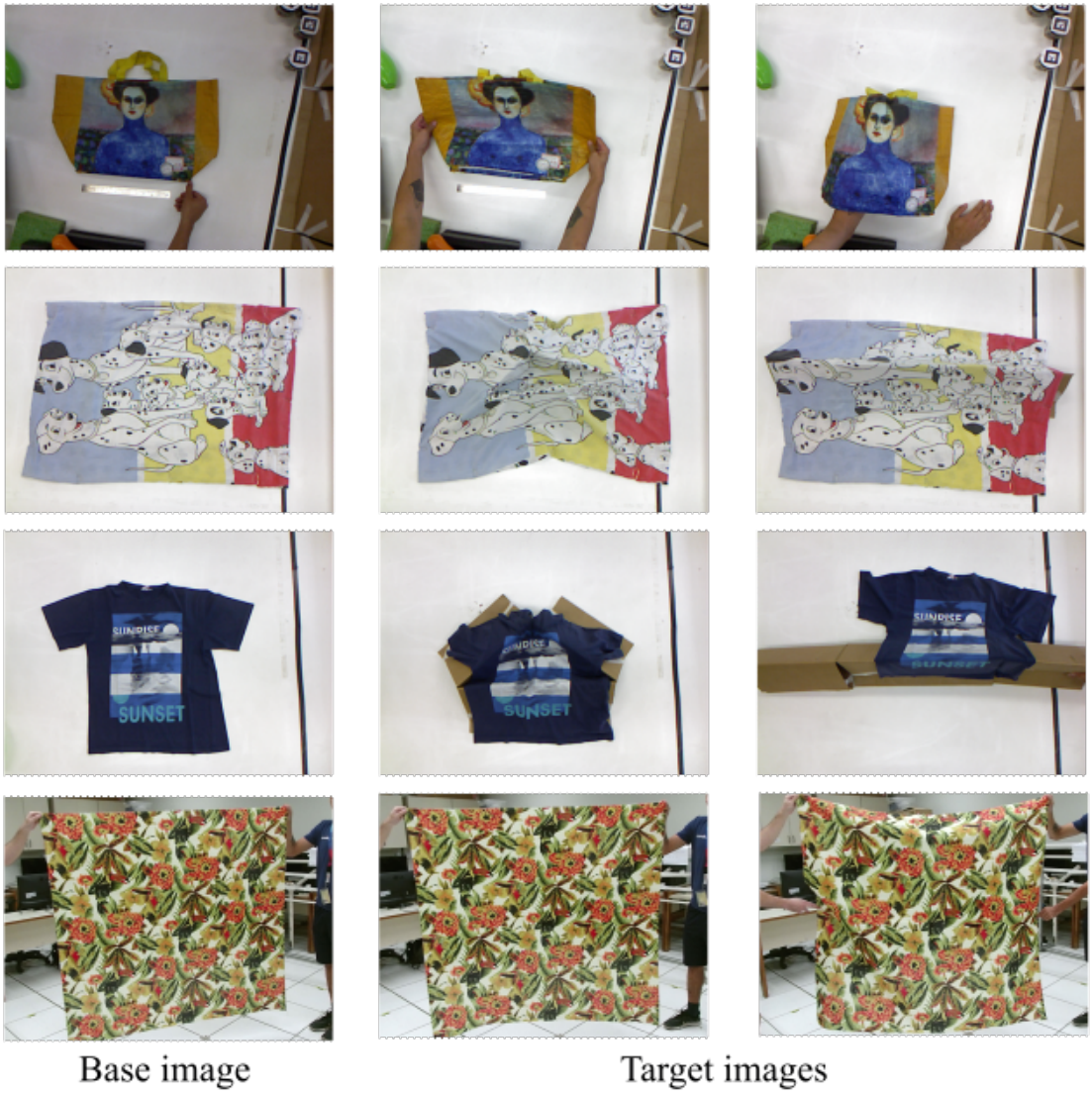

Figure 5.1: **Image sample of the used dataset**. The base image (left-most image) and two target images with different non-rigid deformation (images of column two and column three) from dataset Kinect1 and Kinect2

average ratio between ground-truth correspondences that can be recovered by the whole pipeline and the total number of estimated features within the shared viewpoint region when matching points from the first image to the second and the second image to the first one.

Since keypoint repeatability is also the most used metric for detector evaluation and has an indirect influence on MS, we use the keypoint Repeatability Rate (RR) to compare our detector with the existing ones. RR is defined as the ratio of possible matches and the minimum number of keypoints in the shared view with a pixels error threshold e. In our experiments, we use $e = 3$, the same value used in most of the papers that evaluate detectors.

We compare our results against five detectors. We consider two handcrafted detectors: SIFT [20] and AKAZE [2], that provide stable keypoints and are still considered good baselines according to a recent study [16]; FAST [35], a basic corner detector; Keynet [18], a cutting edge learned based detector; and one state-of-the-art jointly learned detection and description method, ASLFeat [22].

5.4 Ablation and sensitivity analysis

In this section, we present some ablation study and sensibility analysis that was used to support our decisions and select the best parameters for the final and definitive model of keypoint detection. For that, we consider the RR, MS, MMA@3, and the number of inliers as metrics to compare different configurations.

5.4.1 Network fine-tuning

In this study, we evaluate if the network achieves better results from two training strategies: (i) fine-tuning a pre-trained network to detect and describe; and (ii) from scratch training.

For fine-tuning, we used part of the pre-trained ASLFeat architecture as the basis of the new network, adding layers to learn to detect good keypoints for matching and robust to non-rigid deformations. The architecture of ASLFeat and the new architecture used for training (called Our Experimental Detector) can be seen in Figure 5.2. The idea is that we can use the pre-trained features by ASLFeat and back-propagate the signal only in the new layers (Figure 5.2 b-ii). For strategy (ii), which can be seen in Figure 4.2, we perform the training from scratch using the Unet architecture from the input image.

To test the two strategies, we train the models till convergence and use the best result of parameters configuration. We test on the three aforementioned non-rigid datasets: Kinect1 [29], Kinect2 [29], and DeSurT [54].

Table 5.1 shows the large advantage of training the network from scratch. Analyzing the training data, we could notice that it was harder to train the network with fine-tuning. The base layers is trained just for repeatability. Because of that, points with a high signal for repeatability but a bad point for matching make the training process harder and confuse the model in the learning process.

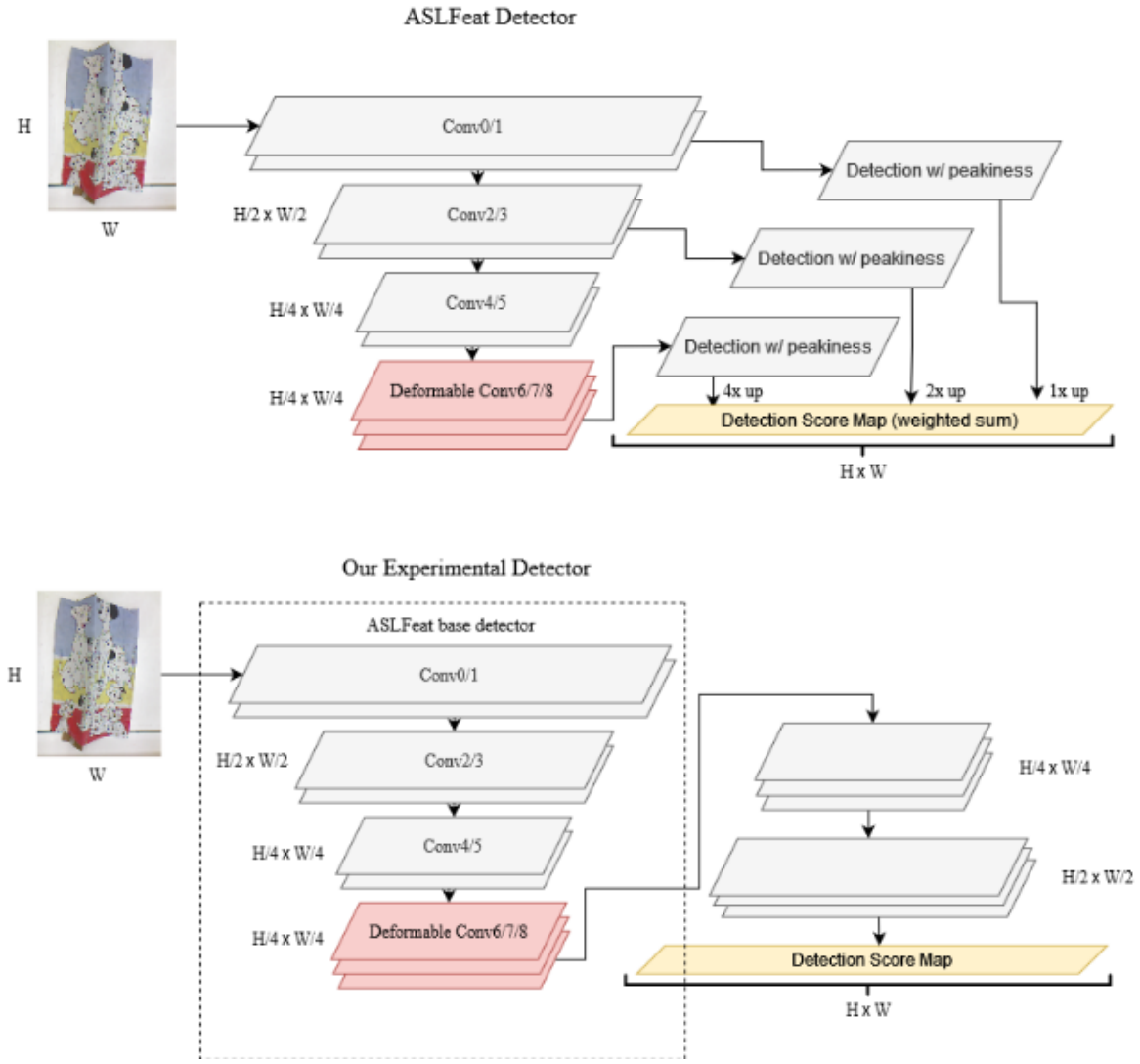

Figure 5.2: **Our Experimental Detector**. On the top of figure (a) one can see the ASLFeat detector architecture; on the bottom of figure (b) we have our experimental architecture with two main parts: a piece of ASLFeat architecture (b-i) (into the dotted square), and our new proposed layers to learn good matches (b-ii).

Table 5.1: **Experiment on Network with fine-tuning.** The higher the better. Bold is the best for the column.

Networking Training	RR	MS	MMA@3	Inliers
Fine-tuning	0.26	0.21	0.77	121
Unet from scratch	**0.50**	**0.43**	**0.80**	**170**

5.4.2 Ablation of siamese training architecture

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, as can be seen in Table 5.2. RR increased from 0.46 to 0.50, and MS from 0.39 to 0.43 by using the Siamese scheme, and MMA@3 decreased from 0.81 to 0.80. However, inliers significantly increase from 150 to 170.

Table 5.2: **Ablation of siamese trainning architecture.** The higher the better. Bold is the best for the column.

Networking Training	RR	MS	MMA@3	Inliers
Siamese Network Scheme	**0.50**	**0.43**	0.80	**170**
Standard Scheme	0.46	0.39	**0.81**	150

5.4.3 Ablation on loss function

To evaluate the contribution of the components of our proposed loss function (Equation 4.4), and support our implementations decisions, we evaluate three different setups: (i) using cosine similarity term only; (ii) full loss of Equation 4.4 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 4.4 with optimal weights. We train the models until convergence and test them in the same datasets of Table 5.4. The final result with mean RR, MS, MMA@3, and inliers of all datasets can be seen in Table 5.3, with the best results in bold. The results in Table 5.3 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, being that way, the chosen configuration.

Table 5.3: **Sensibility Analysis on loss.** The higher the better. Bold is the best for the column.

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**

5.4.4 Sensitivity analysis on keypoint weights of training framework

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 have 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 achieved using the proposed weighted Matching Heatmap strategy (0.50 and 0.43 for RR and MS, respectively).

5.5 Experiments

In this section, we show the obtained results of our detector in three different datasets for three metrics and one real world application.

Table 5.4: **Detector + ASLFeat descriptor matching performance comparison.** Best in bold and second-best underlined. The higher the value, the better.

Dataset 770 pairs total - MS / MMA@3 pixels				
Detector + ASLFeat	Kinect1	Kinect2	DeSurT	**Mean**
SIFT	0.35 / <u>0.77</u>	0.37 / <u>0.85</u>	0.26 / <u>0.63</u>	0.33 / <u>0.75</u>
FAST	<u>0.43</u> / 0.69	**0.53** / <u>0.85</u>	**0.33** / 0.56	**0.43** / 0.70
AKAZE	0.39 / 0.66	<u>0.49</u> / 0.76	0.26 / 0.48	<u>0.40</u> / 0.66
Keynet	0.31 / 0.65	0.35 / 0.62	0.24 / 0.51	0.30 / 0.59
ASLFeat	0.31 / 0.58	0.39 / 0.69	0.28 / 0.53	0.33 / 0.60
Ours	**0.49** / **0.86**	0.48 / **0.89**	<u>0.31</u> / **0.66**	**0.43** / **0.80**

Table 5.5: **Detector + DEAL descriptor matching performance comparison.** Best in bold and second-best underlined. The higher the value, the better.

Dataset 770 pairs total - MS / MMA@3 pixels				
Detector + DEAL	Kinect1	Kinect2	DeSurT	**Mean**
SIFT	0.33 / <u>0.68</u>	0.38 / **0.85**	0.27 / **0.63**	0.33 / <u>0.72</u>
FAST	0.36 / 0.58	**0.51** / <u>0.81</u>	**0.29** / 0.49	<u>0.39</u> / 0.63
AKAZE	<u>0.38</u> / 0.65	<u>0.47</u> / 0.74	0.23 / 0.42	0.36 / 0.60
Keynet	0.27 / 0.58	0.34 / 0.59	0.22 / 0.45	0.28 / 0.54
ASLFeat	0.31 / 0.66	0.40 / 0.73	0.25 / 0.54	0.32 / 0.64
Ours	**0.45** / **0.79**	0.46 / **0.85**	<u>0.28</u> / <u>0.59</u>	**0.40** / **0.74**

Tables 5.4 and 5.5 represent experiments results. The tables show MS and MMA with 3 pixel error threshold for each dataset for several combinations of *Detector + Descriptor*. For that two experiments, we detect a fixed amount of keypoints, 1,024 keypoints, for each detector on each image. With that experiments, we aim to analyze how the detected keypoints influence the quality of the matching. For this purpose, we chose two descriptors: ASLFeat [22] and DEAL [32]. ASLFeat [22], a state-of-the-art detector & descriptor that employs deformable convolutions. And also DEAL [32], a deformation-aware descriptor invariant to non-rigid transformations. With that, we can test our detector with a descriptor that was not trained to describe non-rigid objects and was trained in a *describe-and-detect* manner; that is the case of ASLFeat. And with a descriptor that is not trained in a *describe-and-detect* manner, but is invariant to non-rigid transformations; that is the case of DEAL. For results on Table 5.4, we train our detector with ASLFeat keypoints and describe them with ASLFeat descriptor. For results on Table 5.5, we train our detector with ASLFeat keypoints and describe them with DEAL descriptor.

In Table 5.4, one can see that our detector reaches the best MMA for all datasets. In comparison with ASLFeat detector, our method increases the avg. MMA scores from 0.60 to 0.80 (20 p.p.) when replacing ASLFeat's detector to our detector, and has a significant distance of 5 p.p. from the second best MMA (SIFT-ASLFeat). For MS, we achieve the best mean as well as the FAST detector, however, we achieve 10 p.p. in MMA mean.

One can see, in Table 5.5 that, on average, our keypoints paired with DEAL descriptors outperforms all detector-DEAL combinations in both MS and MMA metrics. 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 that was used to train the DEAL descriptor.

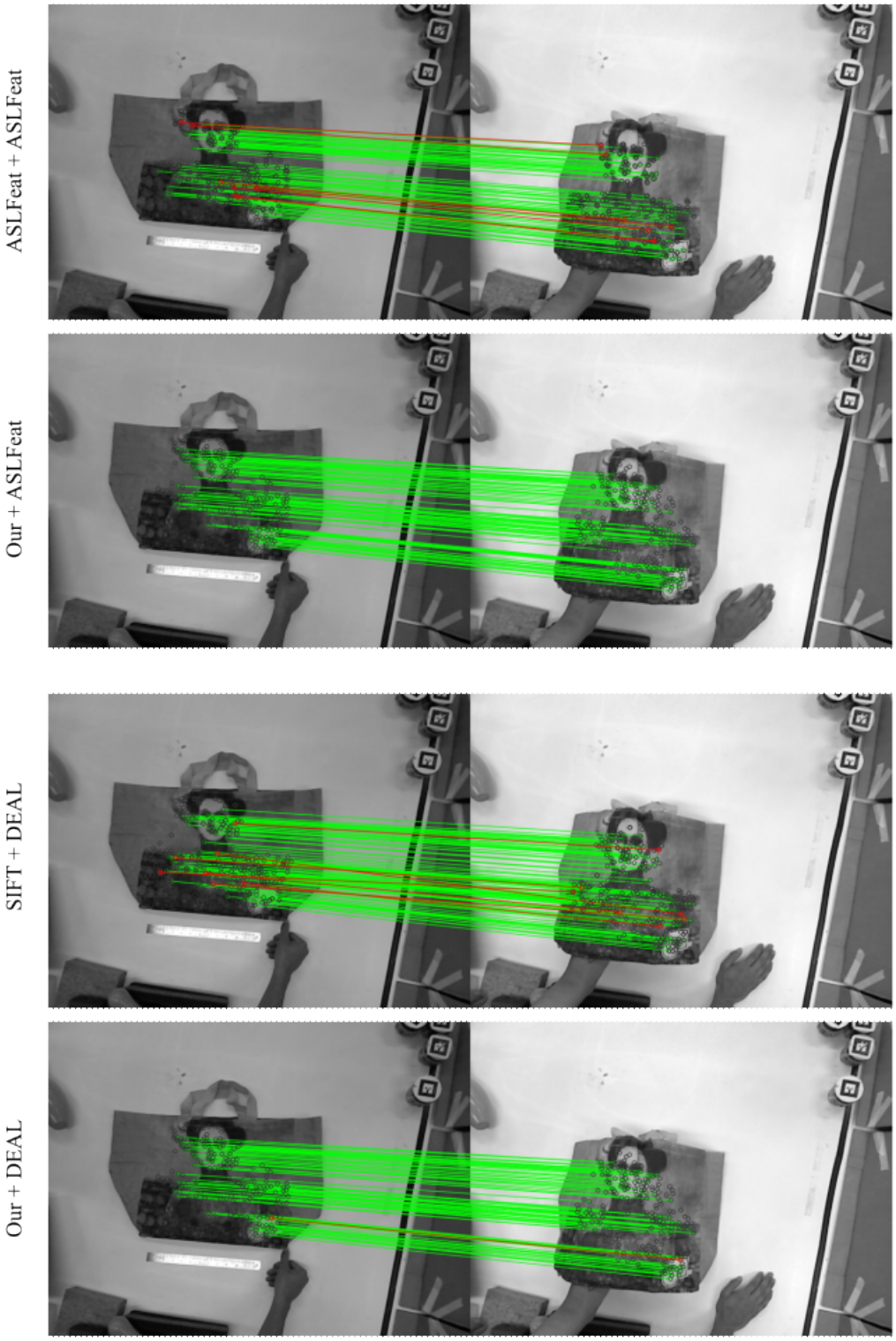

Figure 5.3: **Qualitative results on a real non-rigid matching of dataset Kinect1/Bag**. The green lines show correct correspondences, while the red lines depict wrong correspondences.

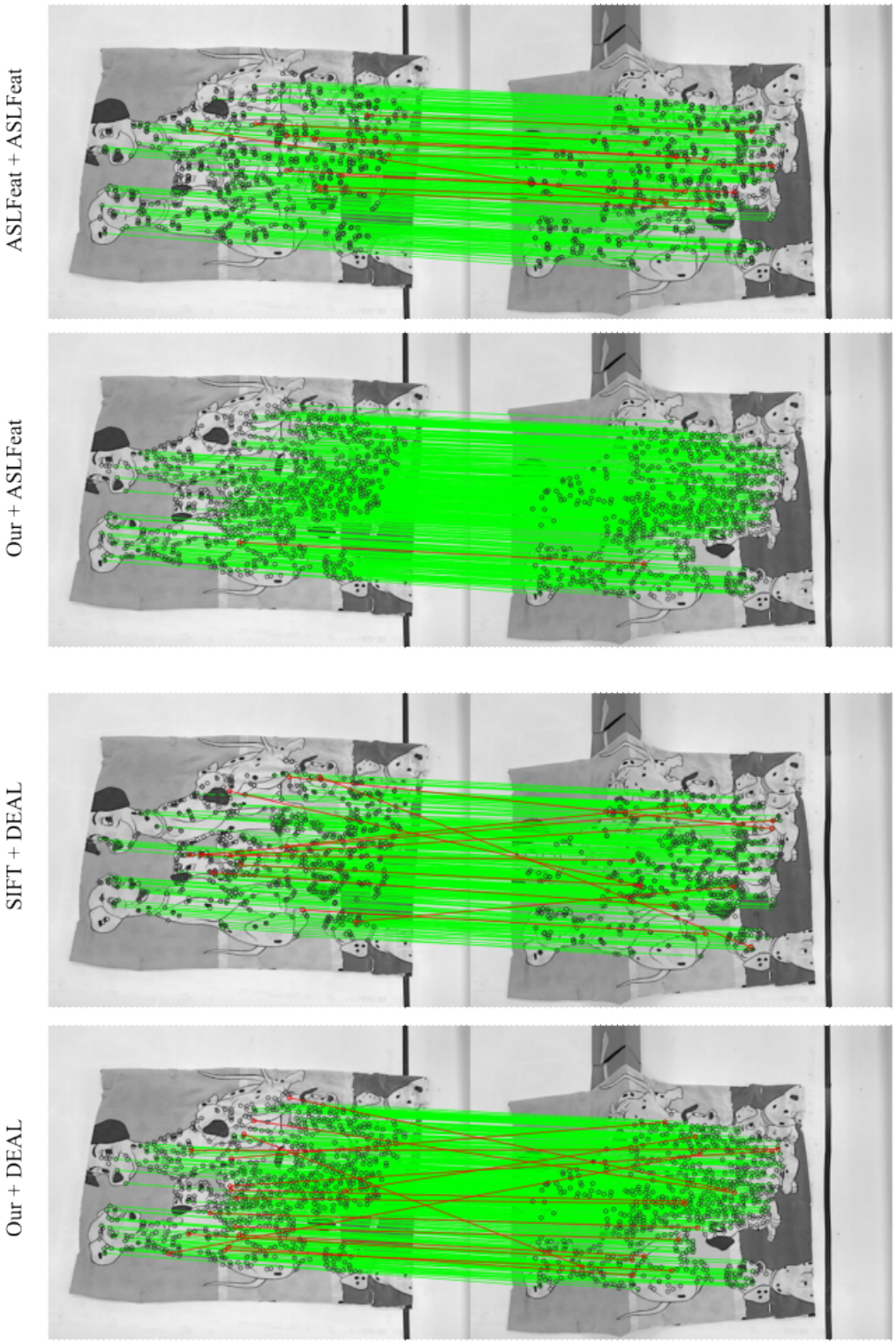

Figure 5.4: **Qualitative results on a real non-rigid matching of dataset Kinect1/Blanket**. The green lines show correct correspondences, while the red lines depict wrong correspondences.

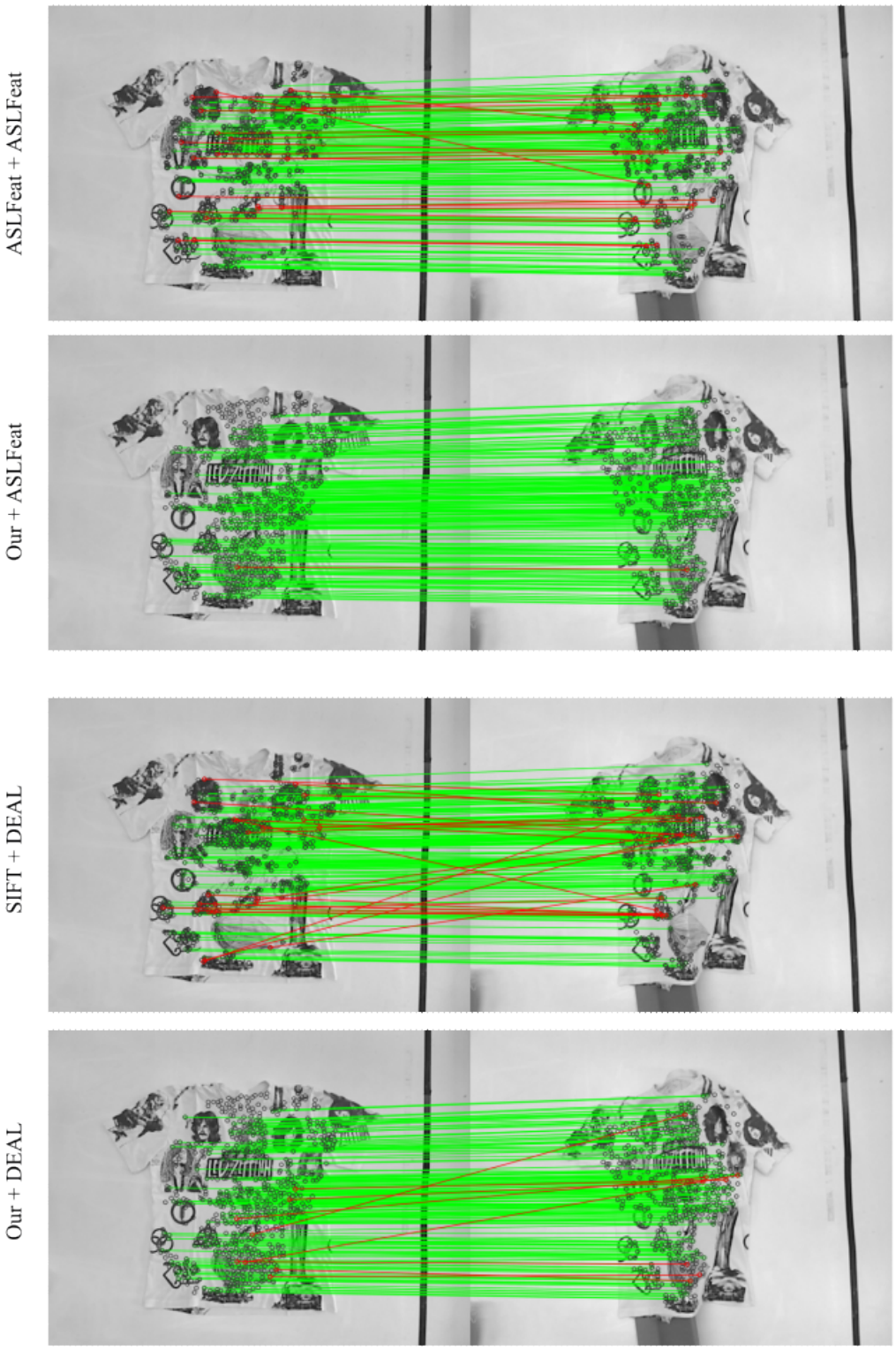

Figure 5.5: **Qualitative results on a real non-rigid matching of dataset Kinect1/Shirt1**. The green lines show correct correspondences, while the red lines depict wrong correspondences.

Since our main goal is the quality of matching, RR is not the main metric to evaluate. However, we also analyze the detectors' repeatability with the RR score. From all methods, FAST has the best RR with 0.59 on average. As second best, AKAZE, ASLFeat, and our method reach a RR of 0.50. The worst detector in 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 noticing that ASLFeat detector, even if presenting a high RR has the smaller MS and MMA, as can be seen in Table 5.4. One can also note that a high RR does not imply good matching scores.

One can see that handcrafted detectors perform similarly or worst than the learned detectors of literature in matching evaluation. That happens because learned detectors are trained in specific contexts with images from some dataset with some characteristics, and that way, could not generalize well for different domains. Since there is no detector trained for non-rigid deformations, the detectors can not generalize well for that type of image. Handcrafted detectors are more general and could generalize well for different domains, having a more stable result independent of the domain.

5.5.1 Quantitative Results

Figures 5.3, 5.4, and 5.5 show matching examples of our detector combined with different descriptors in comparison with the detector that the descriptor was trained with. Our method is able to deliver well-distributed matches in the image as well as SIFT and ASLFeat detectors, but with improved accuracy. In Figure 5.3, one can notice that Our-ASLFeat combination detected keypoints that was 100% correctly matched, and our keypoints with DEAL descriptors have only one keypoint that was wrong matched; while SIFT-DEAL and ASLFeat-ASLFeat combination have several wrong matches. And can be noticed that the amount of keypoint detected, as well as the spatial distribution of keypoints in the image detected, are similar for all the three detectors. Figure 5.4 shows a similar result for Our-ASLFeat, however Our-DEAL combination present a similar number of wrong matches. Figure 5.5 shows a very similar result for Our-ASLFeat in comparison with Figure 5.4. And Our-DEAL combination has a better visual result than SIFT-DEAL combination.

The visual results are in agreement with the results in the tables above. From the Table 5.4, one can see a great superior result of Our-ASLFeat in comparison with ASLFeat-ASLFeat combination, while from the Table 5.5, Our-DEAL combination has a better performance than SIFT-DEAL combination, but the difference between them is not that big. These results can be explained by the fact that in both cases, our detector

was trained with ASLFeat keypoints, and the ASLFeat keypoints was trained to couple with ASLFeat descriptors, fact that improved final matching performance.

5.6 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 [32] 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 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 5.6 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 [20], with is the method used to train the non-rigid descriptor. The results indicate that our detector can perform well even on a non-matching task. Because of influence of ASLFeat [22] keypoints used in the training methodology, one can see that for values of $K \leq 6$ our method fail. However, one can see that methods that are good on matching task as shown in Tables 5.4 and 5.5, such as AKAZE and FAST, do not perform well for $K > 10$, while our method maintain the maximum accuracy.

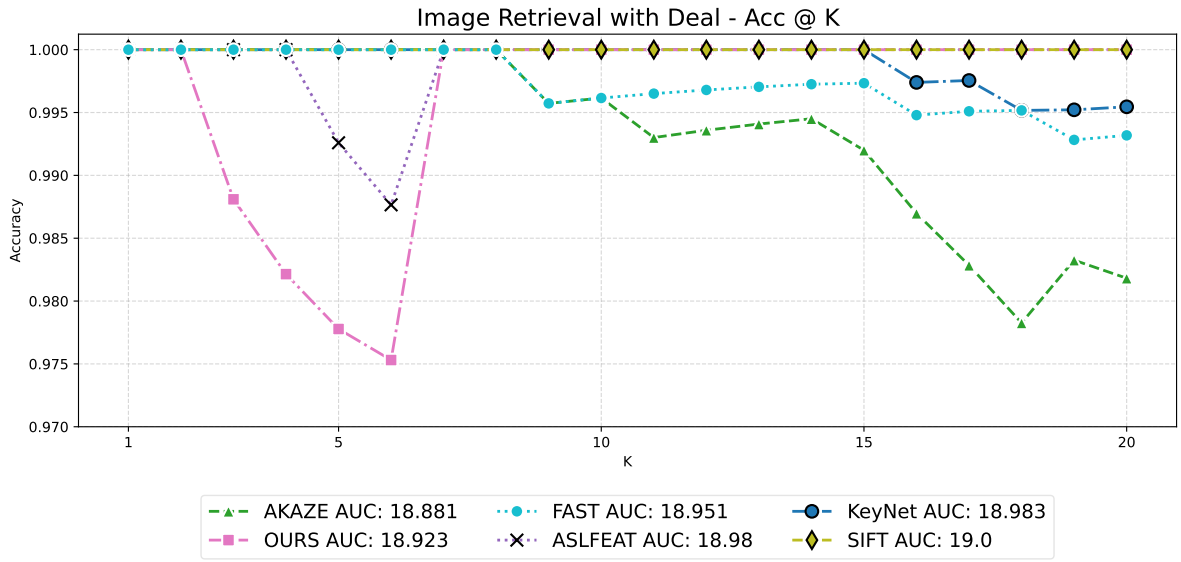

Figure 5.6: **Non-rigid object retrieval application.** The chart shows the retrieval accuracy@K for $K = 20$ using a non-rigid descriptor and various detectors.

Chapter 6

Conclusion

In this work, we proposed a novel approach to detect keypoints on images affected by non-rigid deformations, emphasizing improved matching scores, which contributed as the first detector trained to be robust to non-rigid transformation.

Our main contribution is a semi-supervised 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. We explore several solutions to the Matching Heatmap strategy on the detector training, choosing the one that uses weights for different matching repeatability.

The experimental results show that our method achieved state-of-the-art detection and matching performance on non-rigid deformation datasets. In general, we could see that our detector remained stable both in relation to MS and MMA, as well as between the datasets and between the different descriptors. Even when dealing with descriptors with different proposals and training forms, and one of them was not trained to describe with invariance to non-rigid transformations, our detector achieved good results, learning to be robust, during the detection, of this type of transformation, which resulted in a better quality matching. 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 non-rigid object retrieval, a real-world application, demonstrating that learning to detect good keypoints is a promising research direction for performance improvement in real-world tasks.

6.1 Future Works

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. Removing the base detector from the pipeline, and learning to detect directly from the descriptor, is a possible improvement. We can do that by using dense descriptor method

and matching all the descriptors of two images in a way that the model can learn from the good matches which local regions are good to be detected. However, one of the difficulties of that strategy is the ambiguity on similar regions of an image such as textureless areas where the matching method can find a matching and the local patch there is no relevant information to train the model. We believe that investigating the usage of semantics descriptors could be a good path to surpass the above problem of textureless areas.

Another drawback of our method is the light sensibility. Because of the bad results of matching on dark regions of the image, our method tends to detect a few points in regions with low illumination, which can be bad in some contexts and applications. However, a future experiment with lighting conditions changes should be done to confirm that hypothesis.

A path to future work is to apply our learning approach of the Matching Heatmap strategy to train in the rigid domain using, for example, homography, and testing on traditional datasets such as HPatches. That way, we could see the behavior of the proposed method on the rigid transformation domain and investigate how we can improve matching for transformations such as rotation and scale, with task-specific training, i.e., training focused on rotation or on scale transformation.

Bibliography

[1] Pablo F Alcantarilla and T Solutions. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *TPAMI*, 34(7):1281–1298, 2013.

[2] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. KAZE features. In *ECCV*, 2012.

[3] Vassileios Balntas, Edward Johns, Lilian Tang, and Krystian Mikolajczyk. Pn-net: Conjoined triple deep network for learning local image descriptors. *arXiv preprint arXiv:1601.05030*, 2016.

[4] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Bmvc*, volume 1, page 3, 2016.

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006.

[6] Paul Beaudet. Rotationally invariant image operators. In *International Conference on Pattern Recognition (ICPR)*, 1978.

[7] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993.

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017.

[9] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: self-supervised interest point detection and description. In *CVPR Workshops*, 2018.

[10] Paolo Di Febbo, Carlo Dal Mutto, Kinh Tieu, and Stefano Mattoccia. KCNN: extremely-efficient hardware keypoint detection with a compact convolutional neural network. In *CVPR Workshops*, 2018.

[11] Gianluca Donato and Serge Belongie. Approximate thin plate spline mappings. In *ECCV*, 2002.

[12] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable CNN for joint description and detection of local features. In *CVPR*, 2019.

[13] Mohammed E Fathy, Quoc-Huy Tran, M Zeeshan Zia, Paul Vernaza, and Manmohan Chandraker. Hierarchical metric learning and matching for 2d and 3d geometric correspondences. In *ECCV*, pages 803–819, 2018.

[14] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[15] Jared Heinly, Johannes L Schonberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In *CVPR*, 2015.

[16] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *IJCV*, 129(2), 2021.

[17] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML*, volume 2, page 0. Lille, 2015.

[18] Axel Barroso Laguna and Krystian Mikolajczyk. Key. net: Keypoint detection by handcrafted and learned cnn filters revisited. *TPAMI*, 2022.

[19] Karel Lenc and Andrea Vedaldi. Learning covariant feature detectors. In *ECCV*, pages 100–117. Springer, 2016.

[20] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2), 2004.

[21] Zixin Luo, Tianwei Shen, Lei Zhou, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. ContextDesc: local descriptor augmentation with cross-modality context. In *CVPR*, 2019.

[22] Zixin Luo, Lei Zhou, Xuyang Bai, Hongkai Chen, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. ASLFeat: learning local features of accurate shape and localization. In *CVPR*, 2020.

[23] Welerson Melo, Guilherme Potje, Felipe Cadar, Renato Martins, and Erickson R Nascimento. Learning to detect good keypoints to match non-rigid objects in rgb images. In *2022 35th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, volume 1, pages 61–66. IEEE, 2022.

[24] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.

[25] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *TPAMI*, 27(10):1615–1630, 2005.

[26] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and L Van Gool. A comparison of affine region detectors. *IJCV*, 65(1):43–72, 2005.

[27] Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. *arXiv preprint arXiv:1705.10872*, 2017.

[28] Francesc Moreno-Noguer. Deformation and illumination invariant feature point descriptor. In *CVPR*, 2011.

[29] Erickson R Nascimento, Guilherme Potje, Renato Martins, Felipe Cadar, Mario FM Campos, and Ruzena Bajcsy. GEOBIT: a geodesic-based binary descriptor invariant to non-rigid deformations for RGB-D images. In *ICCV*, 2019.

[30] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *ICCV*, pages 3456–3465, 2017.

[31] Guilherme Potje, Renato Martins, Felipe Cadar, and Erickson R Nascimento. Learning geodesic-aware local features from rgb-d images. *Computer Vision and Image Understanding*, 219:103409, 2022.

[32] Guilherme Potje, Renato Martins, Felipe Chamone, and Erickson Nascimento. Extracting deformation-aware local features by learning to deform. *NeurIPS*, 2021.

[33] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. *NeurIPS*, 2019.

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015.

[35] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *ECCV*, 2006.

[36] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, 2011.

[37] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020.

[38] Nikolay Savinov, Lubor Ladicky, and Marc Pollefeys. Matching neural paths: transfer from recognition to correspondence search. *Advances in Neural Information Processing Systems*, 30, 2017.

[39] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-networks: unsupervised learning to rank for interest point detection. In *CVPR*, 2017.

[40] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[41] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *ICCV*, pages 118–126, 2015.

[42] Yafei Song, Ling Cai, Jia Li, Yonghong Tian, and Mingyang Li. SEKD: self-evolving keypoint detection and description. *arXiv preprint arXiv:2006.05077*, 2020.

[43] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: detector-free local feature matching with transformers. In *CVPR*, 2021.

[44] Suwichaya Suwanwimolkul, Satoshi Komorita, and Kazuyuki Tasaka. Learning of low-level feature keypoints for accurate and robust detection. In *WACV*, 2021.

[45] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[46] Yurun Tian, Bin Fan, and Fuchao Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *CVPR*, pages 661–669, 2017.

[47] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In *CVPR*, pages 11016–11025, 2019.

[48] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *TPAMI*, 32(5):815–830, 2009.

[49] Alessio Tonioni, Samuele Salti, Federico Tombari, Riccardo Spezialetti, and Luigi Di Stefano. Learning to detect good 3d keypoints. *IJCV*, 126(1), 2018.

[50] Quoc-Huy Tran, Tat-Jun Chin, Gustavo Carneiro, Michael S Brown, and David Suter. In defence of ransac for outlier rejection in deformable registration. In *ECCV*, 2012.

[51] Prune Truong, Stefanos Apostolopoulos, Agata Mosinska, Samuel Stucky, Carlos Ciller, and Sandro De Zanet. GLAMpoints: greedily learned accurate match points. In *ICCV*, 2019.

[52] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: learning local features with policy gradient. *NeurIPS*, 2020.

[53] Yannick Verdie, Kwang Yi, Pascal Fua, and Vincent Lepetit. Tilde: A temporally invariant learned detector. In *CVPR*, pages 5279–5288, 2015.

[54] Tao Wang, Haibin Ling, Congyan Lang, Songhe Feng, and Xiaohui Hou. Deformable surface tracking by graph matching. In *ICCV*, 2019.

[55] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *ECCV*, pages 467–483. Springer, 2016.

[56] Yang You, Wenhai Liu, Yong-Lu Li, Weiming Wang, and Cewu Lu. Ukpgan: Unsupervised keypoint ganeration. *arXiv preprint arXiv:2011.11974*, 2020.

[57] Linguang Zhang and Szymon Rusinkiewicz. Learning to detect features in texture images. In *CVPR*, 2018.

[58] Weichuan Zhang and Changming Sun. Corner detection using second-order generalized gaussian directional derivative representations. *TPAMI*, 43(4):1213–1224, 2019.