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

Guilherme Augusto Potje

From Robustness to Efficiency: Deformation-Aware and Efficient Local Feature Extraction for Images

Belo Horizonte
2024

Guilherme Augusto Potje

From Robustness to Efficiency: Deformation-Aware and Efficient Local Feature Extraction for Images

Final Version

Dissertation 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 Doctor in Computer Science.

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

Belo Horizonte
2024

Potje, Guilherme Augusto.

P863f From robustness to efficiency: [recurso eletrônico] deformation -aware and efficient local feature extraction for images / Guilherme Augusto Potje – 2024.
1 recurso online (159 f. il, color.) : pdf.

Orientador: Erickson Rangel do Nascimento.
Coorientador: Renato José Martins.

Tese (Doutorado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.
Referências: f. 144 -159

1. Computação – Teses. 2. Visão por computador – Teses. Aprendizado profundo - Teses. 3. Processamento de imagens – Teses. 4. Mapeamento digital - Teses. I. Nascimento, Erickson Rangel do. II. Martins, Renato José. III. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação. IV. Título.

CDU 519.6*84(043)

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

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

FOLHA DE APROVAÇÃO

From Robustness to Efficiency: Deformation-Aware and Efficient Local Feature Extraction for Images

GUILHERME AUGUSTO POTJE

Tese 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
Département d'Informatique - Université de Bourgogne

Assinado de forma digital por Claudio Rosito Jung
Dados: 2024.12.16 11:21:28 -03'00'

PROF. CLÁUDIO ROSITO JUNG
Departamento de Informática Aplicada - UFRGS

PROF. MARIO FERNANDO MONTENEGRO CAMPOS
Departamento de Ciência da Computação - UFMG

DR. ANDRE ARAUJO
Deepmind - Google

PROF. VINCENT LEPETIT
Laboratoire d'Informatique Gaspard-Monge - Ecole des Ponts ParisTech

Belo Horizonte, 13 de dezembro de 2024.

Acknowledgments

First, I want to thank my beloved Isabella and my parents, Marco and Elizabeth, for their support and understanding all the way. They motivated, supported, and were always there to help me. Without their help, this work would not have been possible.

I also want to express my deepest gratitude to Professors Erickson and Renato, from whom I have learned one of the most precious skills: to conduct quality research through hard work and collaboration. Their mentorship helped me grow both personally and professionally.

I would also like to thank all VerLab team members. Their discussions and support during the development of this work played a significant role in its progress. A special thanks to Cadar, who worked closely with me. We built a solid foundation by working through the research topics together.

I also want to thank Andre Araujo, who contributed to this work by offering insightful and valuable new perspectives, and for participating in the committee.

Finally, I want to thank the other esteemed members of this dissertation committee: Professors Claudio Jung, Mario Campos, Thiago Oliveira, and Vincent Lepetit. Their unquestionable knowledge, thoughtful suggestions, questions, and recommendations were key to improving the quality of this work.

I would also like to thank the PPGCC program for providing the academic resources, and especially Sonia, who was always helpful with the paperwork, as well as everyone who, in some way, contributed to this journey.

Last but not least, I want to thank CAPES, CNPq, FAPEMIG, Google, and ANR (ANR-23-CE23-0003-01) for financially supporting this work.

Resumo

A correspondência visual e a percepção geométrica são aspectos fundamentais de sistemas de visão para a sobrevivência no reino animal. Não é surpreendente que várias tarefas relevantes, como navegação autônoma, reconstrução 3D e registro de imagens, que dependem de características de imagem de baixo nível, continuem a servir como base para tarefas mais avançadas de Visão Computacional. Nesse contexto, descritores locais de imagem fornecem representações compactas e eficientes, operando sobre um conjunto esparso de pontos confiáveis e bem localizados no mundo físico. No entanto, os métodos existentes na literatura oferecem, no máximo, invariância aproximada a transformações afins de imagem, ignorando superfícies deformáveis. Nesta tese, avançamos nessa direção ao estudar e desenvolver novas técnicas para o cálculo de características locais em imagens, considerando aspectos de invariância a deformações e eficiência computacional.

Inicialmente, exploramos a ciência de deformações no estágio de descrição, baseando nossa hipótese em restrições geodésicas das superfícies em torno de de interesse, utilizando imagens RGB-D para modelagem de superfície. Em seguida, expandimos essas ideias para o paradigma do aprendizado profundo, propondo novos componentes que dotam descritores modernos e detectores de pontos de interesse a capacidade de modelar as deformações explicitamente, eliminando a necessidade de informações de profundidade. Além dos métodos propostos, contribuímos com um novo conjunto de dados real, composto por imagens RGB-D de objetos sujeitos a deformações não rígidas (como camisas, roupas, pinturas e bolsas) com correspondências anotadas a nível de pixel, e um simulador capaz de gerar, de forma eficiente e abundante, dados sintéticos realistas para a avaliação de métodos de correspondência não-rígida. Ao longo de diversos experimentos, demonstramos a importância de considerar a deformação na construção dos descritores, não apenas em métricas de correspondência, mas também em três aplicações práticas: recuperação de imagem, rastreamento não rígido e registro de superfícies 3D. Simultaneamente, abordamos um dos maiores desafios dos atuais métodos de descrição baseados em aprendizado profundo: o custo de processamento. Como contribuição final, apresentamos um detector de pontos de interesse e um extrator de características locais que oferece o melhor desempenho em termos de equilíbrio entre eficiência computacional e acurácia frente ao estado da arte.

Palavras-chave: descritores locais; mapeamento geodésico; aprendizado de descritores; imagens RGB-D; correspondência não-rígida; ciência de deformação.

Abstract

Visual correspondence and geometric perception are critical for living beings in the animal kingdom, where vision is essential for survival. Similarly to a biological vision system, modern solutions for autonomous navigation, 3D reconstruction and image registration rely on local image cues for solving higher-level Computer Vision problems. In this context, local image descriptors efficiently provide a compact representation and estimates of point-wise correspondences between images, as they work with a sparse set of reliable and well-localized points in the physical world. Most of the existing handcrafted and learning-based methods are still at best approximately invariant to affine image transformations, disregarding deformable surfaces. In this dissertation, we take one step further by studying and developing novel techniques to compute local features from images, from the perspective of invariance and also speed.

First, we explore deformation-awareness in the description stage, grounding our hypothesis based on geodesic constraints of the surfaces around keypoints, relying on RGB-D images for surface modeling. Then, we further expand the ideas to the deep learning paradigm, where novel components are proposed to endow modern learned local feature descriptors and keypoint detectors with deformation awareness, removing the requirement of depth information. In addition to the methods, to evaluate the current state-of-the-art on image correspondence based on sparse keypoints, we release to the community a new real-world dataset of RGB-D images of several different objects (shirts, cloths, paintings, bags) subjected to non-rigid deformations, alongside annotated ground-truth correspondences, and a physics simulation software capable of producing abundant, plausible synthetic ground-truth for both correspondences and geometry of deforming surfaces inexpensively. Throughout several experiments, we demonstrate the importance deformation-awareness brings to the performance of descriptors not only on low-level matching metrics but also on three real-world applications: image retrieval, non-rigid tracking and 3D surface registration. Concurrently, we address a major challenge in current deep learning-based local features: processing cost. As a final contribution, we introduce a general-purpose keypoint detector and local feature extractor that provides state-of-the-art results in terms of the trade-off between computation and accuracy.

Keywords: local image descriptors; geodesic mapping; descriptor learning; RGB-D images; non-rigid correspondence; deformable description; deformation-awareness.

List of Figures

List of Tables

Frequently Used Acronyms

ARAP	As-Rigid-As-Possible	30
CNN	Convolutional Neural Network	19
DeSurT	Deformable Surface Tracking	64
HOG	Histogram of Oriented Gradients	26
IR	Infrared	47
MLP	Multi-Layer Perceptron	90
MS	Matching Score	66
MMA	Mean Matching Accuracy	93
RANSAC	Random Sample Consensus	35
SfM	Structure-from-Motion	16
SIFT	Scale-Invariant Feature Transform	16
STN	Spatial Transformer Network	22
TPS	Thin-Plate Splines	34
VSLAM	Visual Simultaneous Localization and Mapping	16

Frequently Used Symbols

Symbol	*Domain*	*Description*
H	$\mathbb{N}_+$	Image height (in pixels)
W	$\mathbb{N}_+$	Image width (in pixels)
N	$\mathbb{N}_+$	Arbitrary array length
$\mathbf{K}$	$\mathbb{R}^{3\times3}$	Instrinsic parameters of the pinhole camera model
$\mathcal{I}$	$\mathbb{R}^{H\times W}$	Grayscale image represented as a discrete matrix of intensity values
$\mathcal{D}$	$\mathbb{R}_+^{H\times W}$	Depth image represented as a discrete matrix of distance values
$\mathcal{K}$	$\mathbb{R}^{N\times2}$	Array of keypoint coordinates in image space
μ	$\mathbb{R}$	Margin value used in contrastive losses
$\mathbf{M}$	$\mathbb{R}^{H\times W}$	Learned keypoint heatmap

Contents

Chapter 1

Introduction

For over two decades, the paradigm of keypoint detection and local feature description became the holy grail of image matching, serving as a fundamental building block to countless higher-level Computer Vision and Robotics tasks as Visual Simultaneous Localization and Mapping (VSLAM) [18], Structure-from-Motion (SfM) [109, 126, 160], large-scale image retrieval [96], image stitching [12, 170], to name a few. Since the seminal work of Lowe that introduced the SIFT [78], one of the most iconic methods for computing local image features and still used today as a strong baseline, many other local image descriptors have been proposed. Local descriptors can be roughly grouped based on the type of input information, such as intensity and/or depth images, and can be further divided into handcrafted and learning-based methods. Low processing time is highly desired for the estimation of local features, which led to a series of works on binary feature representations [15, 120, 41, 91], and therefore devising a descriptor that correctly and efficiently establishes invariant features from corresponding points is of central importance for high-quality matching of images in real-world scenarios.

Feature invariance to geometric and photometric changes, is one of the most important properties of local features [84]. Ideally, a highly invariant feature enables correct matching under challenging image transformations. However, most of the existing handcrafted and learning-based local descriptors are still at best approximately invariant to affine image transformations, often assuming that affine invariance is enough for perspective transformations and deformable surfaces, which works reasonably for several applications having small to moderate viewpoint changes, but fail in more challenging situations. We show thorough extensive experiments and real-world applications that explicitly modelling invariance in description can significantly improve the robustness of local feature methods under challenging image matching scenarios.

1.1 Motivations

The ability to extract discriminative and invariant descriptors from images plays a central role in solving computer vision problems in the wild, where high-resolution RGB cameras are ubiquitous. It is noticeable that the community invests significant efforts into novel approaches to improve image matching for rigid scenes [5, 120, 15, 70, 1, 142, 87, 31], but disregards the fact that many objects in the real world can deform in more complex ways, often due to the difficulty of modelling and understanding non-rigid shapes. Many applications in industry, medicine, and agriculture require tracking, retrieval, and monitoring of arbitrary deformable objects and surfaces, where a general-purpose image descriptor algorithm is needed to achieve accurate results. Even though image-based methods tend to exploit much of the rich information engrafted in images wisely, these techniques are restricted to a fixed sampling pattern in the image domain. Thus, the performance of standard affine descriptors tends to degrade when non-rigid deformations arise in the scene. Moreover, scale and rotation invariance in these descriptors are usually achieved by an a priori estimation of the scale and orientation parameters [120, 78], which is often noisy and ambiguous.

Geometric information such as depth images, on the other hand, has become increasingly popular and used to compute distinctive feature descriptors. Their information is less sensitive to lack of texture and allows a better description of rigid surfaces. Therefore, multimodal approaches [94, 91, 169, 144] have shown that the integration of both the texture information from intensity images with geometry cues from depth maps can improve the description of sparse keypoints. Despite the successful application of multimodal methods for rigid image registration, they are still at best approximately invariant to affine transformations and local illumination changes by construction. When strong affine or non-rigid deformations are present, their performance may drastically decrease. Since in real-world applications, objects may assume different forms when being bent or folded, other types of transformations are worth considering. Isometry, for instance, is a non-rigid transformation that usually appears in the world around us. An isometric deformation is a length-preserving mapping, where the geodesic distance does not change in the manifold after deforming the surface. In other words, the geodesic distance is an intrinsic property and plays an important role in characterizing invariance in real-world objects. This rationale motivated us to investigate strategies for enhancing local features by a mapping which is isometrically invariant as shown in Figure 1.1, which demonstrates the concepts applied in the context of local visual feature extraction in this work.

However, several problems associated with depth sensors, such as depth noise, miscalibration, missing data (holes), are usually present in the raw data measurements, requiring additional pre-processing steps and careful implementation to avoid error propaga-

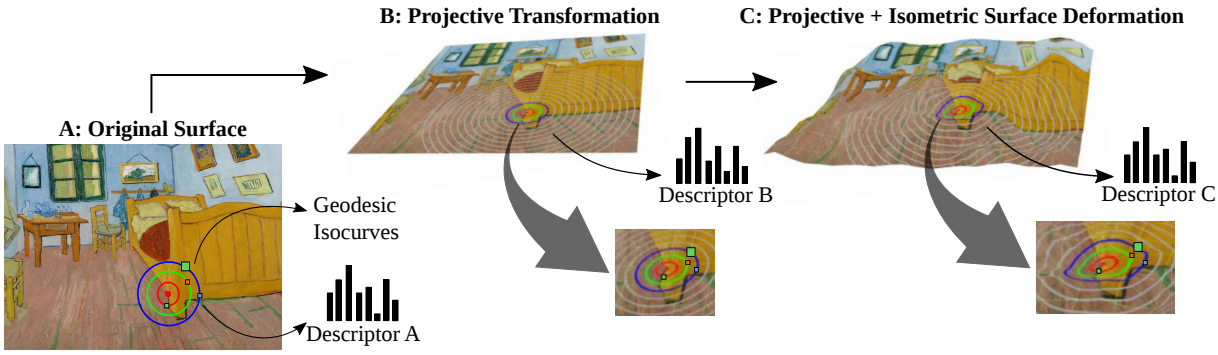


Figure 1.1: **Example of extracting geodesic-aware features to consider isometric deformations**. Due to the invariance property of the geodesic distance, the pixel intensities lying on the same goeodesic distance (shown as isocurves) from the keypoint center do not change after the deformations and can be sampled to compute an invariant visual descriptor. Most available descriptors are limited to affine image deformations (descriptor B). Despite the affine and isometric deformations, descriptors A, B, and C are still similar (pixels indicated by squares in red, blue, and green lie on the same isocurve after the deformation) using the geodesic-aware approaches proposed in this dissertation.

tion. Furthermore, in numerous contexts and applications, depth data remains inaccessible, and optimizing for minimal processing time is essential, particularly for low-level tasks like local feature estimation. Consequently, the development of descriptors that accurately and reliably identifies invariant features from corresponding points for common visible light cameras is also critical for high-quality image matching in practical scenarios.

The emergence of contemporary deep learning architectures has led to significant advancements, with various studies showcasing impressive learned invariance to factors such as illumination and perspective transformations using common RGB images [166, 151, 140, 87, 36, 31, 80]. Despite these achievements, our extensive experiments reveal that current methods exhibit diminished invariance to non-rigid transformations, even when re-trained in non-rigid data. This limitation underscores the need for further advancements in deep learning models to adequately address the challenging task of non-rigid description. Therefore, this dissertation introduces, besides depth-based approaches, innovative strategies to enhance modern deep learning frameworks with explicit deformation-awareness using single RGB images. These strategies combine the strengths of learned invariance from data with appropriately integrated inductive biases. Figure 1.2 presents a conceptual overview of the techniques proposed in this dissertation for estimating invariant descriptors from deforming surfaces from a monocular RGB frame.

Simultaneously, we identified a critical oversight in current deformation-aware methodologies: the omission of the keypoint detection phase, which restricts their efficacy under challenging deformations. While traditionally treated as distinct, the interdependence of keypoint detection and description tasks has been illuminated by recent joint detection and description approaches [113, 138], suggesting that the quality of keypoint

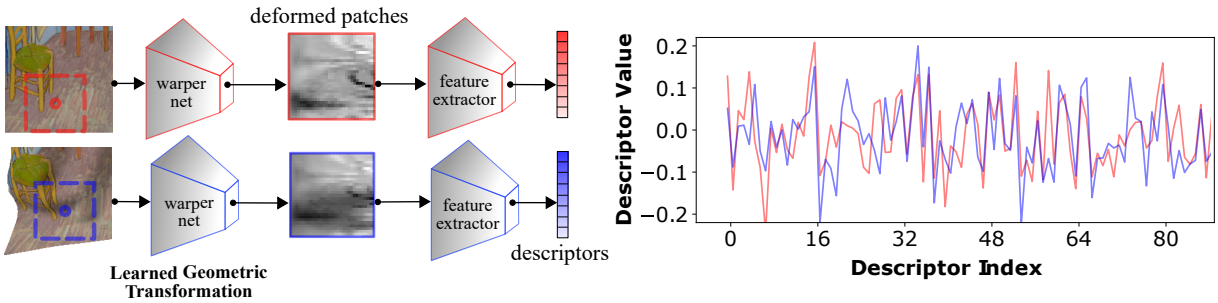


Figure 1.2: **Learning to deform**. We show it is possible for a CNN to learn meaningful geometric transformations during training that lead to improved matching performance. Notice that the descriptor's signatures (Descriptor Value plot) of corresponding keypoints, with and without a local deformation (red and blue curves), are highly correlated when extracted with our deformation-aware approach.

detection directly influences descriptor effectiveness, and vice versa, enabling the identification of keypoints tailored for specific objectives. We thus devised a novel framework that concurrently learns keypoints and descriptors that are jointly optimized for matching non-rigid surfaces, demonstrating enhanced robustness to deformations, varied viewpoints, and illumination changes. We provide empirical evidence underscoring the key role of the detection phase and explicit deformation awareness in achieving superior matching performance under deformation transformations.

Finally, we observed that the current literature in local feature matching prioritizes accuracy while often overlooking compute efficiency, resulting in bottleneck issues in real-world deployment. In mobile robotics and augmented reality applications, it is crucial for models to operate on hardware-constrained environments. To enable efficient and accessible feature extraction using deep convolutional networks, we thus introduced to the community a novel Convolutional Neural Network (CNN) designed as a general-purpose, lightweight image matching solution, emphasizing the compute-accuracy trade-off in an image matching pipeline. The proposed solution is capable of achieving 25 frames per second (FPS) inference for VGA images on a standard CPU laptop (Intel i5).

1.2 The Visual Correspondence Problem

The existing body of research in image matching has predominantly centered on rigid correspondences, as evidenced by the abundant literature on the subject [78, 120, 31, 91]. This focus is partly due to the maturity of projective geometry [46], a well-established research area characterized by more clearly defined and constrained problems. Another contributing factor is the large number of datasets and ground-truth data available

for rigid correspondence [132, 2, 126, 112]. In contrast, the correspondence problem for non-rigid surfaces has attained less attention [63, 129], largely due to its inherent challenges. These challenges include the high degree of freedom in transformations and the difficulties in acquiring accurate and automated ground-truth data for evaluation benchmarks and also for training Machine Learning models. This dissertation revisits the local feature extraction paradigm for visual correspondence of deformable surfaces in the visible spectrum, particularly in light of advancements in modern depth sensors and contemporary deep learning paradigms. Within this framework, we define our problem statement as follows:

Problem Definition. *Given an intensity image pair* $(\mathcal{I}_1, \mathcal{I}_2)$*, which may have additional information such as scene depth* $(\mathcal{D}_1, \mathcal{D}_2)$ *or precomputed keypoints* $(\mathcal{K}_1, \mathcal{K}_2)$*, we seek to compute an invariant and distinctive local image representation that allow one to properly correspond two image points associated to the same physical 3D point in the world. An ideal descriptor would correctly match all co-visible points from image pairs under different imaging conditions, including illumination changes, partial occlusions, projective, and non-rigid transformations.*

Therefore, this dissertation is focused on addressing the problem of detecting and describing local features from objects that may deform over time. We aim to close the existing gap in local image descriptors by proposing novel strategies that account for image deformations. We hypothesize that explicitly modeling invariance through a geometric mapping function accounting for the local deformations present in objects and scenes allow concrete improvements on the reliability of image correspondence under previously disregarded conditions such as non-linear distortions. By investigating the literature, we formulate our dissertation statement as follows:

Dissertation Statement. *To compute local image representations invariant to non-rigid transformations, we are required to reason about the local geometric structure of the manifold surface that composes it. By exploiting invariant properties such as geodesic distances on the manifold, or implicitly learning local geometric cues from data, we can improve the invariance of the representations regarding non-rigid transformations beyond the affine model with reduced loss of feature distinctiveness.*

To address the visual correspondence problem of deforming surfaces, we faced three major research challenges: *i)* extracting geometric properties from depth estimated from low-cost devices is problematic, due to the high noise present in the data; *ii)* non-rigid deformations introduce complex appearance changes in scene illumination and texture, undermining the capabilites of current state-of-the-art local image descriptors in several tasks, thus requiring novel components that account for image deformations; *iii)* obtaining

ground-truth for both evaluating and studying non-rigid deformations is a difficult task, since there is no global transformation that provides a dense correspondence map from a reference image to a target image. Unlike rigid scenes – where a single homography transformation (planar case) or epipolar geometry & scene depth (non-planar case) can be estimated reliably with available methods and technologies – deformable scenes offer fewer constraints for defining a correspondence field between images. Therefore, the development of geometric-aware methods capable of extracting invariant features from images, as well as the introduction of high-quality benchmark data of non-rigid scenes, were essential milestones for this dissertation.

1.3 Contributions

In this dissertation, we take a step towards local feature extraction invariance to non-rigid deformations in RGB-D and RGB still images. We first introduced a geodesic-aware descriptor called *GeoBit* [93], a handcrafted binary descriptor that encodes deformation-invariant features efficiently using a vector of binary tests. However, a major drawback of GeoBit is that it uses a computationally expensive method to estimate the geodesic properties of the surface, which is undesired, specially on resource-constrained computers. To address this problem, we subsequently designed a novel, more efficient strategy for sampling pixels over the geodesic distance field, which in addition enables the use of vanilla CNNs to train a learning-based descriptor named *GeoPatch* [107], that is invariant to isometric deformations by construction.

The main limiting factor of geodesic-aware methods is that they heavily rely on depth data to extract geodesic distances from the surfaces. Thus, addressing the depth requirement constraint of our prior approaches, we have devised an end-to-end trainable architecture that embeds explicit modeling of geometric transformations directly into a CNN backbone, designated as *DEAL* (Deformation-aware Local Features) [108]. This approach uses solely RGB images, significantly broadening its practical applicability. Through our experiments, we demonstrate the viability of learning geometric priors from synthetic datasets, leading to state-of-the-art performance on real-world imagery. Building on this foundation, we further develop *DALF* (also Deformation-aware Local Features) [105], a joint keypoint detector & descriptor that incorporates DEAL's geometric-aware module into a novel non-rigid keypoint detection learning framework, enabling concurrent training for both detection and description of local features aware of deformation.

Then, going beyond invariance to non-rigid deformations, we tackle another major bottleneck in current modern deep local feature extraction: processing cost. We revisit

fundamental design choices in a CNNs for detecting, extracting, and matching local features. Our general-purpose learning-based method called *XFeat* [106] satisfies a critical need for fast and robust data-driven algorithms suitable for resource-limited devices.

The main contributions [1] of this dissertation can be summarized as follows:

1. We present a geodesic-aware strategy to extract isometric invariant image patches from noisy RGB-D data efficiently. We demonstrate the improvement in feature representation by designing a binary descriptor (GeoBit), and a new learning-based descriptor (GeoPatch) that outperforms other methods in accuracy and is competitive in terms of processing time;

2. We devise the first learned method to explicitly model non-rigid geometric transformations in CNNs for local feature matching. We demonstrate that training solely with synthetic deformation warps can still yield significant improvements in real images, achieving state-of-the-art performance across different vision tasks;

3. We propose a novel joint keypoint detection and description framework that is deformation-aware and jointly optimized end-to-end. The technique is based on a reinforcement learning algorithm for unified training, using only synthetic warps as supervision combined with Spatial Transformer Networks (STNs) that capture deformations by learning context priors. Extensive experimental evaluation provide solid evidence that our approach advances the state-of-the-art in three different tasks: (i) image matching; (ii) image retrieval and (iii) 3D non-rigid registration;

4. We provide to the community a new RGB-D benchmark comprising 11 real-world objects and a large simulated dataset of RGB-D images. All real-world objects, captured with Kinect™, versions 1 and 2, were subjected to a variety of non-rigid deformations, with ground-truth matches obtained by manual annotation and an accurate motion capture system. Keypoints with pixel-level correspondence and dense flow-field obtained via a deformation model is provided for all sequences. The simulated dataset also provides ground-truth correspondences of deforming surfaces in challenging realistic conditions, such as varying non-rigid deformation, non-linear illumination changes, different perspectives, and diverse textures;

5. A new compact CNN for sparse and semi-dense local feature extraction is proposed to handle the critical need for efficient learning-based feature detection and extraction. Our new method is at least 5× faster than the most lightweight methods currently available, while delivering comparable results in several indoor and outdoor evaluation benchmarks.

[1] We have released all source code, datasets, and the simulation software, allowing the creation of new datasets and encouraging future research in invariant & efficient local image representations. The project page with all information is available at https://www.verlab.dcc.ufmg.br/descriptors.

1.3.1 Relevant Publications

The work presented in this dissertation has resulted in scientific publications in the following Computer Vision and Machine Learning venues:

- Nascimento, E., Potje, G., Martins, R., Cadar, F., Campos, M., & Bajcsy, R. "GEO-BIT: A Geodesic-Based Binary Descriptor Invariant to Non-Rigid Deformations for RGB-D Images". *International Conference on Computer Vision (ICCV)*. 2019.

- Potje, G., Martins, R., Chamone, F., & Nascimento, E. "Extracting Deformation-Aware Local Features by Learning to Deform". *Conference on Neural Information Processing Systems (NeurIPS)*. 2021.

- Potje, G., Martins, R., Cadar, F., & Nascimento, E. R. "Learning Geodesic-Aware Local Features from RGB-D Images". *Computer Vision and Image Understanding (CVIU)*. 2022.

- Potje, G., Cadar, F., Araujo, A., Martins, R., & Nascimento, E. "Enhancing Deformable Local Features by Jointly Learning to Detect and Describe Keypoints". *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023.

- Potje, G., Cadar, F., Araujo, A., Martins, R., & Nascimento, E. "XFeat: Accelerated Features for Lightweight Image Matching". *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024.

1.4 Dissertation Overview

The remainder of this document is organized as follows. In Chapter 2 we introduce the reader to relevant theoretical background. Chapter 3 discusses related works in local feature representations, multi-modal representations and deformation-aware image matching. In Chapter 4, we first introduce the reader to our solution based on RGB-D images, and then describe our proposed real and synthetic datasets for evaluating the

performance of image correspondence under deformations, alongside a thorough evaluation of the proposed method in the image retrieval and non-rigid surface tracking tasks. We also discuss in the experimental section the computational aspects, robustness to illumination changes, sensitivity to deformation levels, and sensitivity to different input conditions for our proposed methods. In Chapter 5, we further extend the ideas to RGB images by proposing novel strategies coupled with learning-based techniques, and provide an extensive evaluation of existing methods for local feature correspondence under surface deformations, with additional real-world applications as non-rigid 3D registration and image-based deformable object retrieval. Chapter 6 describes our proposed efficient convolutional network design and learning framework for deep-learning based lightweight image matching, and includes a detailed analysis of compute-accuracy tradeoff considering state-of-the-art local feature extraction methods. In Chapter 7, we present an overview of the dissertation's results with respect to the initially defined objectives, highlighting future research directions based on the current state-of-the-art in local features and the remaining limitations.

Chapter 2

Theoretical Background

In this chapter, we introduce the concept of local image representations, a fundamental component in computer vision pipelines, serving as a foundational step in various applications such as image registration, visual recognition and localization, 3D reconstruction, and robot perception. Subsequently, we discuss key principles of geometric transformations and image registration, which recur as a central theme throughout this dissertation.

2.1 Local Image Features

Local image features are widely adopted in computer vision solutions and has an extensive and established literature. Carefully engineered solutions such as SIFT [78] and ORB [120] are still adopted today, due to efficient and reliable implementations available in traditional libraries [153, 8], and well-understood literature behind their algorithms. Handcrafted sparse keypoints are usually the choice for camera pose estimation and visual odometry tasks, as they provide repeatable points that can be matched across many image pairs in a tractable way. The drawback of handcrafted solutions though are mainly their dependence on low-level image structures such as well-defined blobs and corners, and rich local texture. Furthermore, significant changes in camera viewpoint and scene illumination drastically reduce their performance. With recent advancements in deep learning, remarkable improvements regarding invariance against geometric and photometric changes enabled substantial improvements in a myriad of computer vision tasks. Figure 2.1 illustrates an example of a fundamental application of local features: image matching.

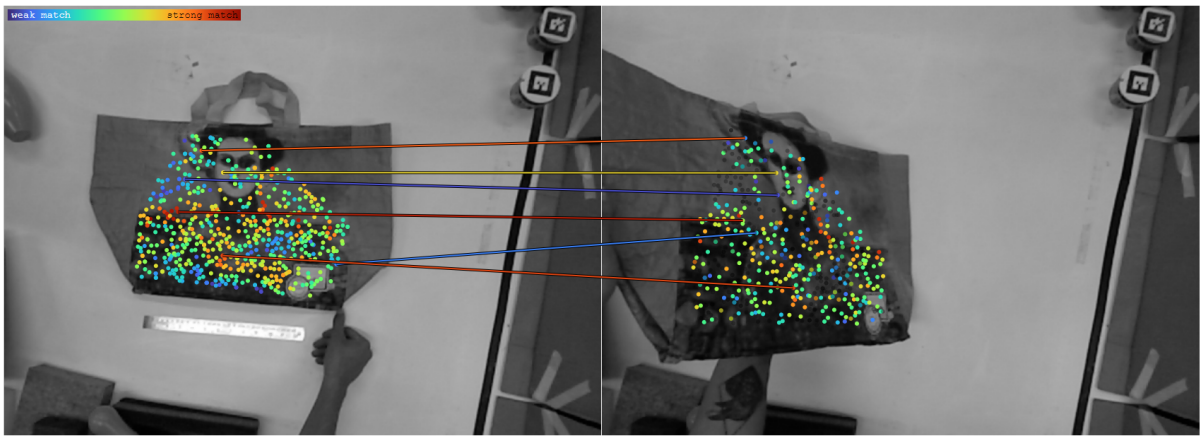


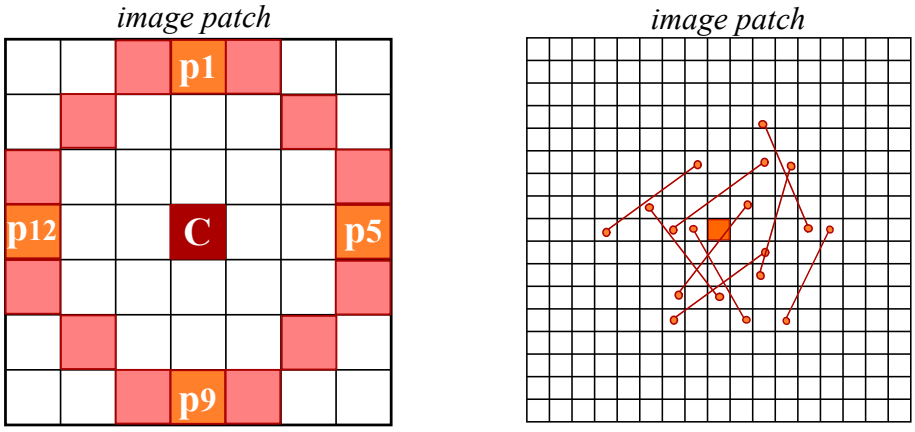
Figure 2.1: **Local feature matching example**. The colored circles represent detected keypoints. For each keypoint, a local representation is encoded by a vector of numbers, which is stored for matching via distance comparison. In the visualization, confident matches are shown in red, while non-confident matches are shown in blue, based on the nearest neighbor distance.

2.1.1 Handcrafted solutions

The term 'handcrafted' is common in computer vision, referring to those carefully engineered methods using domain knowledge and expertise. In local feature detection and extraction, algorithms designed to extract salient image points and compute local representations consider aspects such as rotation invariance and the repeatability of pixels under varying imaging conditions. One of the simplest and most efficient solution for sparse keypoint detection and description is the FAST algorithm [118] for detecting salient pixels in an image and BRIEF [16] descriptor extractor, which extracts a binary vector from a local image patch through a series of pixel intensity comparison tests, and are still used today due to their simplicity and effectiveness in less challenging scenarios. Figure 2.2 shows the simplicity and elegance of this approach.

More advanced strategies such as Difference of Gaussian keypoints [78] enabled robust and accurate keypoint detection at sub-pixel level by performing scale-space construction of the image and interpolating the detections between scales. Descriptors such as Histogram of Oriented Gradients (HOG) and SIFT descriptor, perform gradient analysis in a normalized local image patch, which allow increased robustness to illumination and affine image transformations.

Other strategies relying on corners such as the Harris corner detector [44] are also widely adopted, which provide stable interest points under large viewpoint changes. These seminal approaches have enabled the practical implementation of several computer vision solutions in 3D reconstruction [160], real-time camera pose estimation and localization and mapping [18].



(a) **FAST** Corner Detector (b) **BRIEF** binary encoding

Figure 2.2: **FAST detection plus BRIEF extraction**. FAST corners (a) are extremely fast to detect due to the efficient detector design. FAST uses a Bresenham circle of radius three as the support region for each pixel. First, four support points (in orange) are tested to verify the candidate corner. If the test passes, the algorithm proceeds to check if a number of contiguous pixel intensities in the circle are higher or lower than the center pixel's intensity (C) plus a margin value. Once keypoints are extracted, BRIEF (b) can be used to extract a binary vector, *e.g.*, $[0, 0, 1, 0, 1, 1, 0]$, according to intensity comparison tests in the form $t_1 < t_2$, shown as pairs in the figure – (b), which are exceptionally fast to compute on modern hardware leveraging SIMD (Single Instruction, Multiple Data) instructions. Vector distances can then be carried out using bitwise XOR operator.

2.1.2 Learning-based solutions

Early learning-based solutions on local image features traditionally relied on classic machine learning techniques mixed with handcrafted-designed strategies. An example is the FAST [118] corner detector, relying on decision trees to quickly decide if an image patch is a potential keypoint. Optimizing the parameters for existing handcrafted solutions [11] was also an effective strategy to improve the matching performance. Other strategies such as SIFT-PCA [14] included to learn a compact representation of gradient patches using PCA, or using randomized trees [69] to optimize the keypoint recognition rate of specific objects during training.

With the advent of deep convolutional networks (e.g., VGG, ResNet, UNet) [131, 47, 117], numerous architectures for local feature extraction have been proposed. TFeat [151], L2Net [140], and HardNet [87] initially processed local image patches derived from traditional keypoint detectors such as SIFT keypoints and were trained via contrastive learning [42, 151]. These approaches demonstrated significant improvements over handcrafted gradient-based local feature extraction methods, such as HOG, due to their ability to learn discriminative features that are more robust to changes in viewpoint and illumination. However, their reliance on handcrafted keypoint solutions led to a saturation in

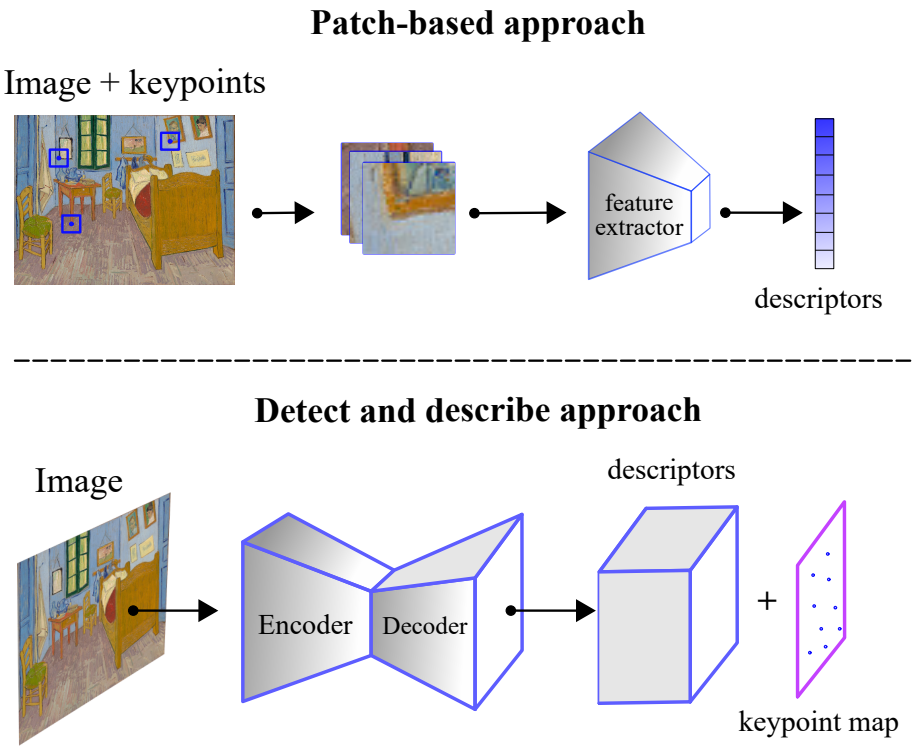


Figure 2.3: **Local feature extraction paradigms**. Patch-based approaches rely on pre-detected keypoints to extract normalized patches, which are then processed by a CNN, thus depending on the quality of these keypoints. In contrast, the detect and describe paradigm processes the entire image through a CNN, jointly extracting a dense feature map and a keypoint map. This approach can be more efficient for handling a high number of keypoints commonly used in computer vision tasks and is often more robust to challenging imaging conditions, such as strong illumination changes, since they can learn keypoints suitable for the descriptors and vice-versa.

performance.

Another paradigm that emerged is the detect-and-describe approach using a single convolutional network backbone. D2Net [35], SuperPoint [31], and R2D2 [113] are seminal works that proposed self-supervised joint keypoint detection and description learning. Since traditional supervision with labeled points is not well defined [113, 37] due to the inherent lack of labels for keypoints other than existing handcrafted solutions, these works address the problem by focusing on the repeatability and reliability properties of the detected points (e.g., R2D2). Figure 2.3 depicts the two current paradigms for computing local features using CNNs.

2.2 Geometric Transformations

Geometric transformations are mathematical operations that acts on sets of points within a given space. These transformations are crucial in various fields, including computer vision, graphics, and robotics, as they enable the manipulation and analysis of geometric shapes and images. This section discusses two core types of geometric transformations used in this dissertation: projective and isometric. Projective transformations deal with mappings that preserve the collinearity of points, while isometric transformations focus on preserving distances on a surface manifold.

2.2.1 Projective

According to Hartley & Zisserman [46], geometry is the study of properties that remain invariant under groups of transformations. In this context, 2D projective geometry, a mathematical framework that extends the classic Euclidean geometry, focuses on the properties of the projective plane under projective transformations, also known as projectivities. Projectivities are invertible mappings that preserve the collinearity of points, meaning they map lines to lines within the projective plane. These transformations can be represented by a non-singular 3×3 matrix, denoted as $\mathbf{H}$, commonly referred to as the homography matrix. The homography matrix acts on homogeneous 3-vectors as $\mathbf{x}' = \mathbf{H}\mathbf{x}$. A projective transformation on a point $\mathbf{x}$ can thus be described as:

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}. \tag{2.1}$$

This matrix encapsulates the relationship between two planes in projective space. Additionally, the homography encodes relative camera motion between two images in three specific cases: (i) pure camera rotation, (ii) scenes where observed points lie at infinity, and (iii) observed points lying on a plane. If the intrinsic camera calibration is known, the extrinsic parameters can be recovered. The homography plays a fundamental role in numerous computer vision applications, including image stitching, 3D reconstruction, and camera calibration.

2.2.2 Isometry

In geometry, a space is defined as an n-dimensional manifold if every point within it has a neighborhood that is homeomorphic to an open subset of $\mathbb{R}^n$. In simpler terms, this means that each small region around any point in the manifold looks like a piece of n-dimensional space, though the overall shape can be much more complex [9]. The Earth can be seen as a familiar example of a 2D manifold: We can localize ourselves in a map having latitude and longitude coordinates, thus being locally flat. But we know that the planet is a 3D body and is globally curved in 3D.

In the context of this dissertation, the surfaces of objects can be represented as continuous two-dimensional manifolds within the three-dimensional Euclidean space. These are called embedded surfaces because they are subspaces within a larger $\mathbb{R}^3$ space.

The length of a path in a manifold is known as a geodesic, which can be thought of as the shortest route between two points on a curved surface. Let's imagine we are on a walk in the park, and the entrance is point A. There is a hill in the middle of the park, and to cross the park to the end point B, we have two options: getting around the hill or climbing the hill. In our example, it may be much faster to just climb the hill instead of walking around it, because of the geodesic path connecting those two points on the manifold surface (the park terrain in this case). This brings us to the definition of isometric deformation of a 3D shape: a transformation that preserves geodesic distances, meaning the shape can bend or twist without stretching or compressing any part of it.

Since in practice we work with discrete representations of 3D surfaces, for example, point clouds or triangular meshes, computing a transformation that minimizes geodesic distance of a 3D shape is often treated as an optimization problem having local constraints. Assuming the widely adopted triangular mesh representation, given a mesh $\boldsymbol{\mathcal{M}} = (\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{E}})$, a good approximation of an isometric transformation is the As-Rigid-As-Possible (ARAP) method [133], which finds a set of rotation matrices R for each edge in the mesh that minimizes the following equation:

$$E(\boldsymbol{\mathcal{V}}') = \sum_{(i,j)\in\boldsymbol{\mathcal{E}}} \left\| (\boldsymbol{\mathcal{V}}'_i - \boldsymbol{\mathcal{V}}'_j) - R_{ij}(\boldsymbol{\mathcal{V}}_i - \boldsymbol{\mathcal{V}}_j) \right\|^2, \tag{2.2}$$

where $\boldsymbol{\mathcal{V}}'$ is the transformed mesh. A practical usage of the ARAP formulation is to register several non-rigid 3D scans into a single, canonical model [95]. Notice that the minimization of this cost function relies on local constraints (the edges of the mesh), and locally enforces the preservation of the distances between adjacent vertices in the mesh, which satisfies the definition of an isometric transformation of 3D shapes.

In the context of simulating realistic isometric deformations in 3D, one established, physics-informed approach is to represent a 3D surface as a grid of $M \times N$ particles having

mass in 3D space. For flat surfaces, such as a piece of cloth for example, we can assume the particles initially lie on a planar surface in 3D. Each particle has a position $\mathbf{p} \in \mathbb{R}^3$ and is connected to its immediate neighbors, as demonstrated in Figure 2.4. Newton's second law is applied to accumulate the acceleration vector $\mathbf{a}$ of the particle at a given instant when a force vector $\mathbf{f}$, like wind and gravity, is applied:

$$\mathbf{a}(t+1) = \mathbf{a}(t) + \mathbf{f}(t)/m, \tag{2.3}$$

where the scalar m is the mass of the particle.

Then, Verlet integration, a numerical method for approximating the integration of Newton's laws of motion, is applied to translate the acceleration into velocity. Let $\mathbf{p}_{i-1}$ be the position of the particle in the last timestep $i-1$. Given the current's particle position $\mathbf{p}_i$ we can compute the next position $\mathbf{p}_{i+1}$ with the following equation:

$$\mathbf{p}_{i+1} = \mathbf{p}_i + (\mathbf{p}_i - \mathbf{p}_{i-1})(1-\delta) + \mathbf{a}\Delta t^2 \tag{2.4}$$

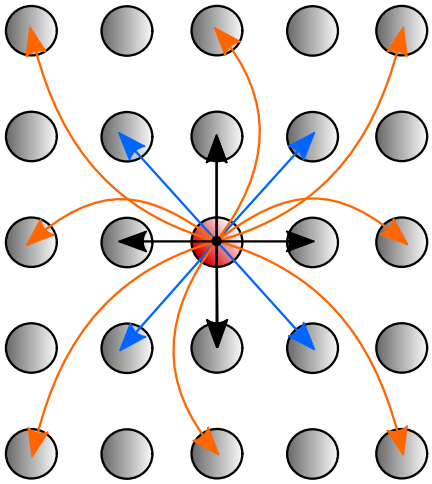


Figure 2.4: **Producing isometric deformations of surfaces**. Each edge represent a distance constraint that has to be satisfied. The constraint satisfaction step ensures that the geodesic distance of each particle connected by a constraint remains similar to the initial distance, enforcing isometric deformations of the surface.

where δ is a damping factor accounting for air resistance, and Δt is the constant time interval between simulation steps ($\delta = 0.01$ and $\Delta t^2 = 0.175$ in our simulation framework used in this dissertation). These parameters were selected through visual inspection and grid search, aiming to achieve more realistic deformation and cloth motion. After each step, the acceleration term $\mathbf{a}$ is reset to the null vector, as it has been integrated into the velocity. After applying the motion model, a constraint satisfaction optimization is performed for each particle. The constraint satisfaction step seeks to enforce constant distance of neighboring particles according to their connectivity in the grid. Thus, for each constraint connecting particle j to particle k, given the target distance constraint $\eta_{j,k}$, we compute a correction vector $\Delta\mathbf{e}_{j,k}$ as follows:

$$\Delta\mathbf{e}_{j,k} = (\mathbf{p}_j - \mathbf{p}_k)\frac{\eta_{j,k}}{\|\mathbf{p}_j - \mathbf{p}_k\|}, \tag{2.5}$$

and apply it to both particles $\mathbf{p}_j = \mathbf{p}_j + 0.5\Delta\mathbf{e}_{j,k}$ and $\mathbf{p}_k = \mathbf{p}_k - 0.5\Delta\mathbf{e}_{j,k}$ to enforce the target distance $\eta_{i,j}$ between them. In practice this iterative process is repeated a few times, which is enough to converge. This approach can be used to produce realistic and diverse surface deformations, and was implemented as a simulation environment in this dissertation [107]. Some generated deformation examples can be seen on Figure 2.5.

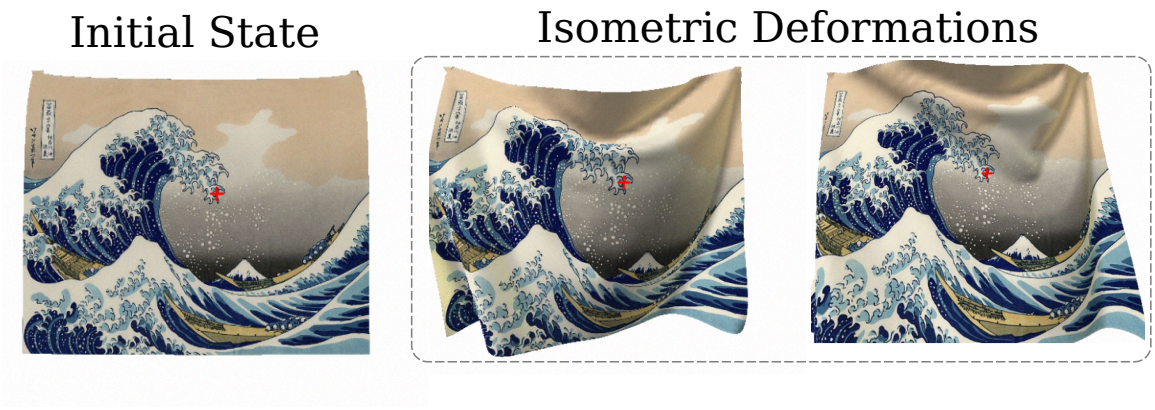


Figure 2.5: **Simulating isometric deformations of surfaces**. The initial state presents a grid of particles in a planar configuration. After several iterations based on Newton's laws of motion and the constraint satisfaction optimization, the surface undergoes realistic isometric deformation. A red cross marks a tracked point throughout the simulation, using known mesh coordinates.

2.3 Image Registration

Image registration refers to the process of finding a parametric model that relates images observing the same scene. This step usually follows local feature matching, as most models require point correspondences as input [137]. In this section, we will discuss the registration techniques used in the dissertation for enabling synthetic data generation, dataset label refinement, and the development of several practical applications used in our experimental validations.

2.3.1 Planar

Given two images observing the same planar scene, we can relate them by an homography matrix $\mathbf{H} \in \mathbb{R}^{3\times3}$. To find an homography relating two images, four non-colinear points on the same 3D plane are enough [46]. The direct linear transform (DLT) algorithm can be used to find the entries of $\mathbf{H}$ by solving a linear system in the form $\mathbf{Ax} = 0$, and refinement techniques, such as the gold-standard least squares minimization (assuming Gaussian noise in image point coordinates) can be used to further refine the solution using a geometric error as in the following:

$$\arg\min_{\mathbf{H}} \sum_{i} \left|\mathbf{x}'_i - \mathbf{H}\mathbf{x}_i\right|^2, \tag{2.6}$$

where $\mathbf{x}$ and $\mathbf{x}'$ are a set of corresponding points. The optimization is performed in the Euclidean image space $\mathbb{R}^2$ by minimizing the residual reprojection error measured in pixels.

The homography matrix can also be used to implement synthetic image warps, often being used in data augmentation strategies in machine learning for improving a model's robustness to viewpoint changes.

2.3.2 Epipolar

A non-planar 3D scene viewed from two different cameras can be related by a fundamental matrix $\mathbf{F} \in \mathbb{R}^{3\times3}$. The fudamental matrix is a rank 2 matrix, relating corresponding sets of points in two images [46] as follows:

$$\mathbf{x}'^{T}\mathbf{F}\mathbf{x} = 0, \tag{2.7}$$

where $\mathbf{x} \in \mathbb{R}^3$ and $\mathbf{x}' \in \mathbb{R}^3$ are the homogeneous coordinates of corresponding points in the first and second image, respectively. This equation encapsulates the epipolar constraint of a stereo image pair, and can be computed only from point correspondences if $\mathbf{F}$ is unknown. This constraint arises from the coplanarity of the camera centers in a stereo setup. If the cameras internal (intrinsic) parameters $\mathbf{K} \in \mathbb{R}^{3\times3}$ are known, it is possible to compute $\mathbf{E}$ from $\mathbf{F}$:

$$\mathbf{E} = \mathbf{K}_2^{\top}\mathbf{F}\mathbf{K}_1, \tag{2.8}$$

where:

- $\mathbf{F}$ is the Fundamental matrix computed from point correspondences between the stereo image pair;
- $\mathbf{E}$ is the Essential matrix, which encodes the extrinsic parameters, *i.e.*, the relative rotation and translation between the two cameras up to an uknown scale factor;
- $\mathbf{K}_1$ and $\mathbf{K}_2$ are the intrinsic matrices of the first and second cameras, respectively. The intrinsic parameters consist of the principal point (c_x, c_y), focal lengths (f_x, f_y), and the skew factor. In modern cameras, it is commonly assumed that the skew factor is negligible, and pixels, square $(f_x = f_y)$, and the principal point is near the center of the image [126].

The essential matrix $\mathbf{E}$ is related to the relative rotation $\mathbf{R}$ and translation (up to an unknown scale factor) $\mathbf{t}$ of the cameras as follows:

$$\mathbf{E} = [\mathbf{t}]_{\times}\mathbf{R}, \tag{2.9}$$

where $[\mathbf{t}]_{\times}$ is the *skew-symmetric* matrix of the unit translation vector $\mathbf{t}$, allowing us to use the Singular Value Decomposition (SVD) approach to compute $\mathbf{R}$ and $\mathbf{t}$ from $\mathbf{E}$. The detailed steps for obtaining the relative extrinsic parameters are thoroughly discussed in the Multiple View Geometry book [46].

These mathematical frameworks of projective and multiple-view geometry are the core principles in any modern VSLAM system and SfM pipelines such as ORBSLAM-3 [18] and COLMAP [126], fueling cutting-edge technology in navigation from surveying drones to self-driving cars and commercial 3D mapping software.

2.3.3 Non-rigid

Thin-Plate Splines (TPS) represent 2D coordinate mappings $\mathbb{R}^2 \mapsto \mathbb{R}^2$ and are commonly used to model non-rigid deformations in image space. They produce differentiable, smooth interpolations, and can be coupled with both sparse keypoint correspondences, or directly optimized via photometric or other error functions due to its differentiable nature in terms of mathematical operations. The TPS formulation allows for the modeling of the affine transformation component through an affine matrix $\mathbf{A} \in \mathbb{R}^{2\times3}$, while the non-affine component is captured by weight coefficients $\mathbf{w}_k \in \mathbb{R}^2$, representing offsets from the base affine transformation. For a given 2D point $\mathbf{q} \in \mathbb{R}^3$, with weight coefficients and control points $\mathbf{c}_k \in \mathbb{R}^2$, both in homogeneous coordinates, the corresponding mapping $\mathbf{q}'$ is computed as:

$$\mathbf{q}' = \mathbf{A}\mathbf{q} + \sum_{k=1}^{n} \rho(\|\mathbf{q} - \mathbf{c}_k\|^2)\mathbf{w}_k, \tag{2.10}$$

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

The TPS parameters can be found in closed form given a set of corresponding points, and its warping quality will depend on the distribution and coverage of the input set of correspondences.

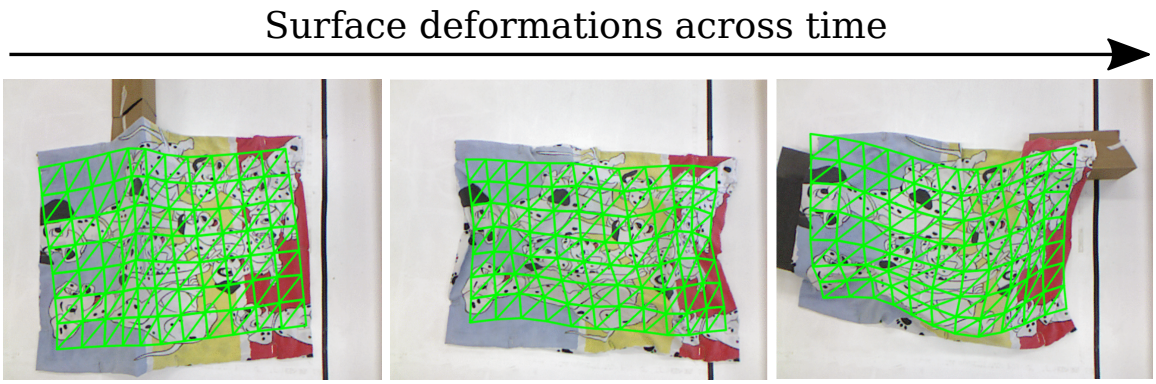


Figure 2.6: **Thin-plate splines registration.** A mesh grid is initially defined on the planar (undeformed) object. In subsequent frames, a dense flow field is estimated by fitting a TPS warp using keypoint correspondences. For visualization, the template grid is warped to match the current deformed object, as shown by the drawn mesh grid.

2.3.4 Robust fitting

In the previous sections, we introduced several geometric transformation models for image registration into a known referential. These transformations require a known set of point correspondences, typically estimated using local feature correspondence methods. However, correspondences derived from algorithms often contain a significant proportion of incorrect matches. This is problematic because model fitting approaches assume all correspondences are correct, leading to corrupted geometric solutions when as few as a single incorrect match is present.

The most widely adopted strategy for robust fitting of geometric solutions is the Random Sample Consensus (RANSAC) algorithm, first introduced in computer vision by Fischler & Bolles [38], and subsequently improved for increased robustness and stability [4, 68]. The core idea of the algorithm is simple yet powerful: randomly sample a minimal set of points, fit a model to this set, and validate this hypothesis against the entire set of points. After several iterations, the set of points with the largest consensus support is chosen. A simple and intuitive example is the fitting of a 2D line on a set of 2D points. The minimal sample in this case is two, since two points can define a line. After computing a line hypothesis, we can compute the distance from all points in the set to the line. The most important parameter is the threshold value τ that classifies points as inliers or outliers, and depends on the intrinsic noise of the data at hand, requiring tuning.

Given the minimum number of points n required to fit a hypothesis, RANSAC iteratively updates the estimated inlier ratio w, which in practice is computed from the inlier ratio of the best model found so far. The probability P that the selected model consists solely of inliers is given by:

$$P = 1 - (1 - w^n)^k. \tag{2.11}$$

For simplicity, in this equation we assume we sample n points from the set independently and with replacement, providing an upper bound in the number of required iterations. Thus, we can effectively estimate the number of iterations k required to achieve an inlier-free model estimation with the desired probability P. The advantage of RANSAC compared to other robust estimators such as the Least Median Squares [119], is that it can deal with a large proportion of outliers, which is often the case when considering the output of image matching methods.

Chapter 3

Related Work

This chapter provides a through review of visual correspondence methods in the literature, including local features, intensity-based descriptors, multi-modal approaches, as well as geometry-aware algorithms. We discuss the advantages and limitations of all the methods, providing the necessary contextualization, and comparing the relevant technical differences to our proposed approaches whenever appropriate. Figure 3.1 presents an overview of relevant methods in the form of a taxonomy tree.

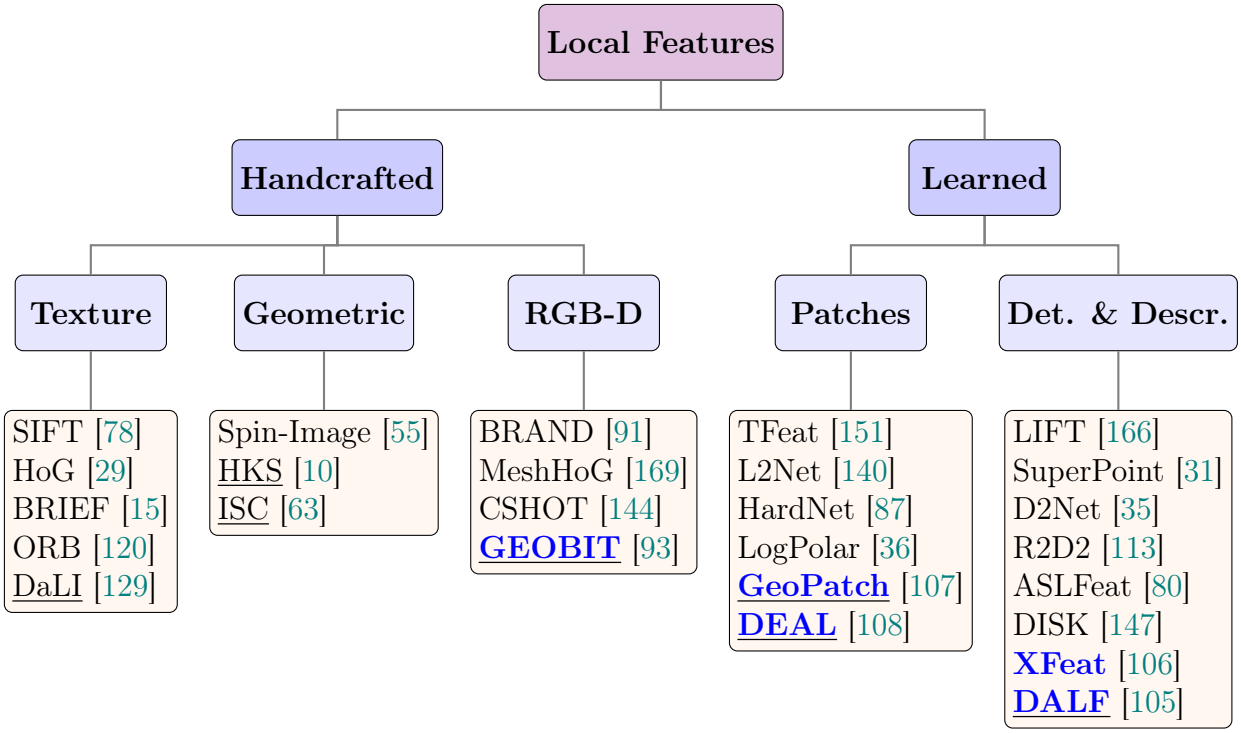


Figure 3.1: **Taxonomy of local image feature extraction methods.** Modern local features are divided into two main categories: handcrafted and learning-based approaches, and can roughly be grouped by the input modality. As most learned methods primarily utilize RGB images, further categorization is omitted for simplicity. Learning-based approaches can be further grouped into patch-based and detect & describe paradigms. The blue bold font highlights the works proposed in this dissertation. Methods that explicitly tackle non-rigid deformations are underlined. Our proposed contributions address gaps in the literature, particularly in descriptors for RGB and RGB-D modalities, where the existing research is limited.

3.1 Local Image Features

Traditionally, local image features techniques focus on providing rich and invariant representations of objects observed from different viewing conditions. Established image keypoint detection methods seek to extract repeatable regions in images, *i.e.*, localized points that are stable under different viewing conditions. The classic Harris detector [45] employs image derivatives that are used to compute cornerness scores, while one the most used handcrafted detector SIFT [78], for instance, detects high-response blob regions using Difference of Gaussians (DoG) image operator. To represent these repeatable points, methods such as SIFT [78] and ORB [120] extract texture patterns by encoding local image gradients into a vector, while also estimating a dominant orientation of the keypoint's neighborhood to provide invariance to rotation transformations.

3.1.1 Handcrafted local features

One of the first approaches from signal processing that was used to compute image patch similarity was the *normalized cross correlation* measure. Afterwards, techniques such as local jets [62], steerable filters [39] and image moments [148] were proposed to compute local image features that could be used for image related tasks, such as edge, contour and orientation detection. The HOG [29] method provided increased robustness to image noise and illumination changes, and was used for over a decade to extract image features to be used in classification algorithms such as Support Vector Machines [24] and Random Forests [48]. SIFT [78] exhibits impressive invariance to affine image transformations, while keeping its distinctiveness, becoming one of the most employed method for camera calibration and image registration until today. Inspired by the idea of Local Binary Patterns (LBP) [98], the use of binary strings to assemble the feature vector has become popular, and several binary descriptors such as BRIEF [15], ORB [120], BRISK [71], BASE [90] and BRAND [91] have been proposed. The main advantage of using binary strings to represent feature vectors is their reduced computational cost for extraction and comparison, and reduced storage requirements compared to floating-point vectors. Although being able to increase invariance to moderate affine image transformations, traditional handcrafted image-based descriptors are designed for rigid scenes. Different extensions of these descriptors have been proposed to images in some particular manifolds such as SphORB [174] and BRISKS [41] for spherical images.

3.1.2 Data-driven approaches

The previously mentioned works are carefully handcrafted designed methods for feature detection and extraction, which was the norm until recently, except for some hybrid methods that involved carefully designed optimization strategies to learn priors from data [11, 14, 69]. The advent of deep learning [131] has changed this scenario. Recent works based on CNNs showed the potential of learning the descriptor extractors from large unstructured image collections derived from structure-from-motion reconstructions [13, 72]. Recent description approaches based on CNNs [151, 140, 87, 79, 36, 130, 158] consume a local patch assuming a pre-defined keypoint detector. These methods achieved state-of-the-art performance using SIFT keypoints in the image matching benchmark [54]. The networks are trained using metric learning [42, 151]. These descriptors often outperform handcrafted counterparts on classical image matching benchmarks such as UBC Phototour [13] and HPatches [2], however, for 3D reconstruction, handcrafted features are still competitive, according to a comprehensive survey made by Jin *et al.* [54]. Following the trend of learning descriptor extractors, the work of Ebel *et al.* [36] demonstrated that training a CNN in a polar sampled patch can provide improved results than traditional Cartesian sampling. They also showed that by using a polar sampling in the vicinity of a keypoint can significantly increase robustness to scale variations. As patch-based methods rely on a pre-defined keypoint detector that may produce keypoints in unreliable or ambiguous regions, difficulties in challenging scenarios such as illumination and perspective changes can negatively affect the performance of patch-based solutions because detection and description steps are decoupled.

DELF [97] and DELG [138] works demonstrated that coupling the detection and description phases using an attentive mechanism for keypoint selection based on higher-level image semantics can substantially boost retrieval performance. Simultaneously, local feature extraction has been shifting towards learning both detection and description of local features jointly. Remarkable examples that learn both keypoints and descriptors are LIFT [167], SuperPoint [31], LF-Net [99] and R2D2 [113], where both the detection and description of keypoints are learned end-to-end from data, since it is advantageous performance-wise to solve both tasks simultaneously in terms of computation and matching accuracy. The architecture design of detect-and-describe approaches are similar, where a CNN backbone extracts a keypoint heatmap, and a dense descriptor map, and the major changes lies in the loss functions for self-supervised keypoint learning and the contrastive learning losses for optimizing the embedding space, meaning that they are designed to deal with approximately affine transformations by construction, and may perform sub-optimally when deformation transformations are occuring in an object or scene.

Key.Net [64] showed that it is possible to improve keypoint detection by combining

handcrafted filters and learned filters by observing that using both handcrafted and learned CNNs filters could outperform purely learned methods, since the handcrafted features can provide anchor filters, decreasing the complexity of the network, thus, increasing generalization. Revaud *et al.* [113] claim that some keypoints can be repeatable but not discriminative, and propose R2D2 to jointly train a detector and descriptor based on the idea of predicting the reliability of the detected keypoints, leading to a reliable keypoint detection that can be described confidently.

Although learning-based methods demonstrate impressive performance in images similar to the training set domain, when more general scenes are considered for matching, such as images having deformation transformations, their performance may significantly decrease, and fine-tuning the models is required to recover some of the lost performance. However, even with fine-tuning, certain methods may degrade performance, as they are not explicitly designed to handle deformation transformations.

3.1.3 Multi-modal methods

The use of multiple sensorial information, such as texture and geometric features, has shown to be an effective approach to improve the performance of several computer vision tasks, and has been gaining even more attention recently in semantic segmentation [73, 88, 43], localization and mapping [20], reconstruction of human geometry and texture [163, 155]. They improve the discrimination power of feature vectors, thanks to the ever increasing availability of depth, and other data modalities such as text [111] more recently.

In the last decade, many works that use multiple cues to improve performance have been proposed. In order to increase the recognition rate, the global descriptor VOSCH [57] combines depth and texture. Another descriptor that uses both depth and texture is MeshHOG [169]. The authors used a texture extracted from 3D models to create scalar functions defined over a 2D manifold. CSHOT [144] an extension of the shape only descriptor SHOT [143], incorporates texture alongside geometry information.

Similarly, Lai *et al.* [65, 66] proposed to use two well-known descriptors for each type of data: SIFT for image and Spin-Image for geometry, and then concatenate both to compose the feature vector. Lightweight descriptors that are able to combine geometrical and texture information were also proposed. Nascimento *et al.* [91, 92] presented the descriptor BRAND, which encodes information as a binary string embedding geometric and texture cues, and provides rotation and scale invariance. Martins *et al.* [82] exploited the complementary properties of color and depth information encoded on RGB-D images to improve the convergence of direct (appearance-based) RGB-D registration. The fusion

of depth and visual data was also exploited by Liang *et al.* [168] to compute perspective invariant feature patches. Detectors of keypoints making use of both visual and geometrical information also have been proposed. Vasconcelos *et al.* [150] presented KVD, a keypoint detector that applies a decision tree to fuse depth and RGB data and enable their approach to work in the absence of visual data.

Several approaches have adopted multi-task and side information training schemes. The work presented by Hoffman *et al.* [49] propose a depth hallucination scheme for improving object detection. During the training, color and depth images are used in the detection, while learning to predict the depth related features from color information. On test time, only the color information is required. In the same direction, in Piasco *et al.* [104] a depth hallucination mechanism is explored for predicting the geometry of scenes observed from different conditions (image retrieval) in the context of long-term visual localization.

Although multi-modal methods for feature extraction incorporate local geometry to enhance representation power, they often assume scene rigidity when encoding geometry and texture, resulting in pronounced performance degradation in the presence of scene deformations. The vast majority of existing local descriptors are approximately invariant to affine image transformations, disregarding images of deformable surfaces in both the design of the methodology and and learning strategy for learned-based methods, which are the main motivations of the design behind our proposed methods in Chapters 4 and 5.

3.1.4 Efficient description & matching

Recent works highlight the growing emphasis on computational efficiency for description and matching using modern neural network architectures. SuperPoint [31] proposed a self-supervised CNN for both keypoint detection and description. However, one major disadvantage of using SuperPoint is that it can still incur significant computational costs when applied to image sizes that are common for image matching. SiLK [40] reevaluates elements of learned feature extraction, proposing an effective yet simple strategy for keypoint and descriptor learning that achieves performance comparable to existing methods. The key aspect that underscores SiLK's competitiveness – its dependence on the original image size for descriptor extraction – is also its main drawback in terms of computational cost, as it substantially slows down inference. ALIKE [176] introduced a lightweight network balancing robustness and speed, with differentiable keypoint detection and a neural reprojection loss. Yet, its reliance on the original image resolution in the final feature map considerably increases memory and compute footprints. ZippyPoint [56]

incorporates quantization and binarization in a CNN. Although it achieved notable speed improvements, it requires custom compilation and specific low-level processor arithmetic operations, restricting its applicability across diverse hardware.

Works considering minimalist CNN architectures may employ both fixed handcrafted and learned filters in convolutional blocks [64]. Beyond feature extraction, recent advancements in feature matching also highlight the necessity for quick inference speeds. LightGlue [76] speeds up learnable feature matching and maintains high accuracy compared to SuperGlue [122]. Nevertheless, LightGlue's transformer-based architecture is still costly for tasks where computational efficiency is critical.

Considering the efficiency aspect, we address the high-efficient and robust image matching problem for ubiquitous deployment: from resource-limited devices such as low-budget boards and embedded systems to smartphones and cloud applications in Chapter 6.

3.2 Deformation-Aware Reconstruction and Matching

Although most works on local descriptors are designed focusing on affine image transformations, incipient works considered non-rigid surfaces and deformable objects. One of the enduring grand challenges in shape analysis is to extract properties that preserve the intrinsic geometry of shapes. Geodesic distance is a well known intrinsic property as far as isometric transformations are concerned. Kokkinos *et al.* [63] built the intrinsic shape context (ISC) descriptor based on properties of the geodesic distance. In the same direction, the geodesic distance descriptors [127] approach presented and evaluated a new basis for geodesic distance representation. The authors also showed how to approximate the geodesic distance efficiently. Despite advances achieved by the works that employ geodesics for shape analysis, their techniques are focused on 3D shapes only, disregarding photometric surface properties.

Carceroni & Kutulakus proposed a non-rigid reconstruction system from videos [19], employing *dynamic surfels* as geometric entities to extract 3D motion, shape and reflectance of an object from a frame sequence. By introducing a dynamic photo-consistency constraint, the approach progressively refine the surfels through coarse-to-fine sampling and parameter refinement using linear and non-linear optimization steps. Since the framework is based on sequence of frames with high spatial overlap, it is not applicable to wide-baseline image matching of deforming surfaces.

Deformation and Light Invariant (DaLI) [129] descriptor is one of the first proposed local feature methods for RGB images to consider deformations. DaLI interprets image

patches as a 3D surface in order to encode features robust to non-rigid deformations and illumination changes. The method demonstrated improved matching performance, however it suffers from high computational and storage requirements (descriptor dimensionality). DeformNet [110] is designed to infer non-rigid shapes from still images. The network architecture is composed of a 2D detection branch and a depth branch. The detection and depth branches are subsequently merged by the shape branch, which uses a pinhole camera model to re-project the shape to 3D. The disadvantage of DeformNet is that it considers a fixed surface grid pattern of a known size, limiting its applicability in more general real-world task settings.

In DeepDeform [7], for instance, the authors propose to estimate non-rigid point-wise RGB-D correspondences by a data-driven approach that is then used by a non-rigid 3D reconstruction framework. The proposed network learns to predict a probability heatmap of a target keypoint's position from patches centered at keypoints. Although their method works well for fast motions and non-rigid surface tracking, its architecture is designed to handle few keypoints, and has a heavy computational burden for establishing correspondences for typical amounts of keypoints from typical image matching tasks. Moreover, the method does not generate a descriptor that can be used for other related vision tasks.

3.3 Spatial Attention Mechanisms and Matching

STNs, introduced by the seminal work of Jaderberg *et al.* [52], allowed differentiable warping operations to be used directly with existing network architectures. The core idea is the learning of an attention mechanism (named Localization Network) that is parameterized to represent the image transformation of interest, *e.g.*, affine, homography or a thin-plate-spline warp. They have been applied to several tasks including image classification [52], semantic alignment [115], geometric matching [114], and local descriptors [36]. LF-Net [99], a detect-and-describe method, first locates interest points using a fully convolutional network, which are then cropped in patches with a STN layer considering an affine transformation encoded by keypoint attributes. The cropped patches are subsequently used to compute local descriptors by the description network. Spatial attention mechanisms have also been extended into more generic settings via deformable kernels [28], to learn kernel offsets in network layers operators to define deformable convolution networks.

More recently, the end-to-end joint detector and descriptor ASLFeat [80] used deformable convolutions to increase the network's expressiveness, which shares some concepts with our methods. However, the deformable kernels employed by ASLFeat

consider high-level feature maps, and it does not model deformations explicitly. Its strategy results in earlier layers not being aware of low level image deformations, usually present in geometric transformations. Moreover, the method is not robust to rotation changes.

3.4 Research Contextualization and Relevance

Our first proposed approach detailed in Chapter 4 – *Geodesic-Aware Local Descriptors*, incorporates several insights from the literature of both local feature extraction and multi-modal representations, aiming to enhance the quality of matching local image features under isometric scene deformations. Specifically, we adopt a multi-modal strategy, leveraging depth data obtained through consumer-grade RGB-D cameras to estimate intrinsic surface properties to provide invariance to isometric deformations, while also being invariant to rotation and scale transformations in image space. Our proposed descriptors GeoBit and GeoPatch contrasts with prior works by explicitly modeling deformation invariance in the descriptor design using intrinsic surface properties (geodesics) for sampling image pixels. Our techniques build geodesic-aware descriptors that take into account both visual and geometrical information to create a distinct and invariant representation of a keypoint neighborhood.

Subsequently, in Chapter 5 — *Data-Driven Deformation-Aware Descriptors*, we introduce DEAL, a spatial transformer network that learns deformation transformations, guiding the feature extractor toward deformation invariance using only a single RGB image as input. To the best of our knowledge, DEAL is the first learning-based approach to specifically address non-rigid deformations in local feature extraction, utilizing a deformation-aware spatial attention mechanism. In the second part of Chapter 5, we extend DEAL by incorporating an end-to-end reinforcement learning strategy to jointly learn keypoint detection and deformation-aware descriptors, referred to as DALF. DALF is the first method in the literature to explicitly consider scene deformations for both keypoint detection and description, where the network components work cooperatively during the learning stage.

Finally, Chapter 6 — *Accelerated Features* presents XFeat, our proposed method for general-purpose efficient feature detection and description based on a lightweight convolutional network architecture. Recent approaches emphasize image matching accuracy and robustness, often at the cost of significantly increasing computational demands. In contrast, we demonstrate that it is possible to drastically reduce computational overhead in both sparse keypoint extraction and pixel-level semi-dense matching, while achieving

performance comparable to or exceeding more computationally intensive methods. Compared to existing learned-based methods, XFeat achieves state-of-the-art balance between computation and accuracy, offering a 5× speedup over the previous fastest method.

Chapter 4

Geodesic-Aware Local Descriptors

In this chapter, we describe the extraction of isometric-invariant visual features from RGB-D images with two description techniques. Our approaches efficiently compute isometric invariant image patches from noisy RGB-D data by modeling the surface as a smooth 2D manifold and then estimating geodesic isocurves defined on a local manifold. In this way, our descriptors exploit the intrinsic surface properties to provide invariance over isometric deformations. We characterize a descriptor designing strategy comprising two main steps: firstly, the geodesic distance computed from depth maps provides an isometric invariant mapping function $f(\cdot)$. This mapping function returns the pixel location and retains invariance over deformations (as illustrated in the example shown in Figure 4.1); secondly, pixel intensities in images are used to extract distinctive features that will compose the descriptor feature vector $\mathbf{d}_f$.

Two novel descriptors will be introduced in this chapter. They share the same geodesic-aware strategy for extracting invariant features from noisy RGB-D data and non-rigid surfaces. GeoBit is based on binary intensity tests that compose a binary string encoding visual information, while GeoPatch uses a shallow CNN that extracts distinctive features from a geodesically rectified patch. The CNN parameters are learned by optimizing a ranking loss term from correspondences obtained from synthetic data.

As stated, the key idea in our geodesic-aware strategy is mapping each pixel's location, preserving the intrinsic geometry of the surface, and further analyzing the vicinity of the keypoint. Since the descriptors encode pixel locations using the geodesic distance, the visual pattern remains invariant in the presence of isometric deformations of the surface.

Our algorithms consider as inputs: a list of N detected keypoints $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{N\times 2}$; an estimate of the intrinsic camera matrix $\mathbf{K} \in \mathbb{R}^{3\times 3}$ and an RGB-D image $\boldsymbol{\mathcal{F}} = \{\boldsymbol{\mathcal{I}},\ \boldsymbol{\mathcal{D}}\}$, composed of an image $\boldsymbol{\mathcal{I}} \in [0,\ 1]^{H\times W}$ as pixel intensities and $\boldsymbol{\mathcal{D}} \in \mathbb{R}_{+}^{H\times W}$ as depth information. Thus, for each pixel $\mathbf{p} \in \mathbb{P}^2$ in Cartesian coordinates, $\boldsymbol{\mathcal{I}}(\mathbf{p})$ provides the pixel intensity and $\boldsymbol{\mathcal{D}}(\mathbf{p})$ the respective depth.

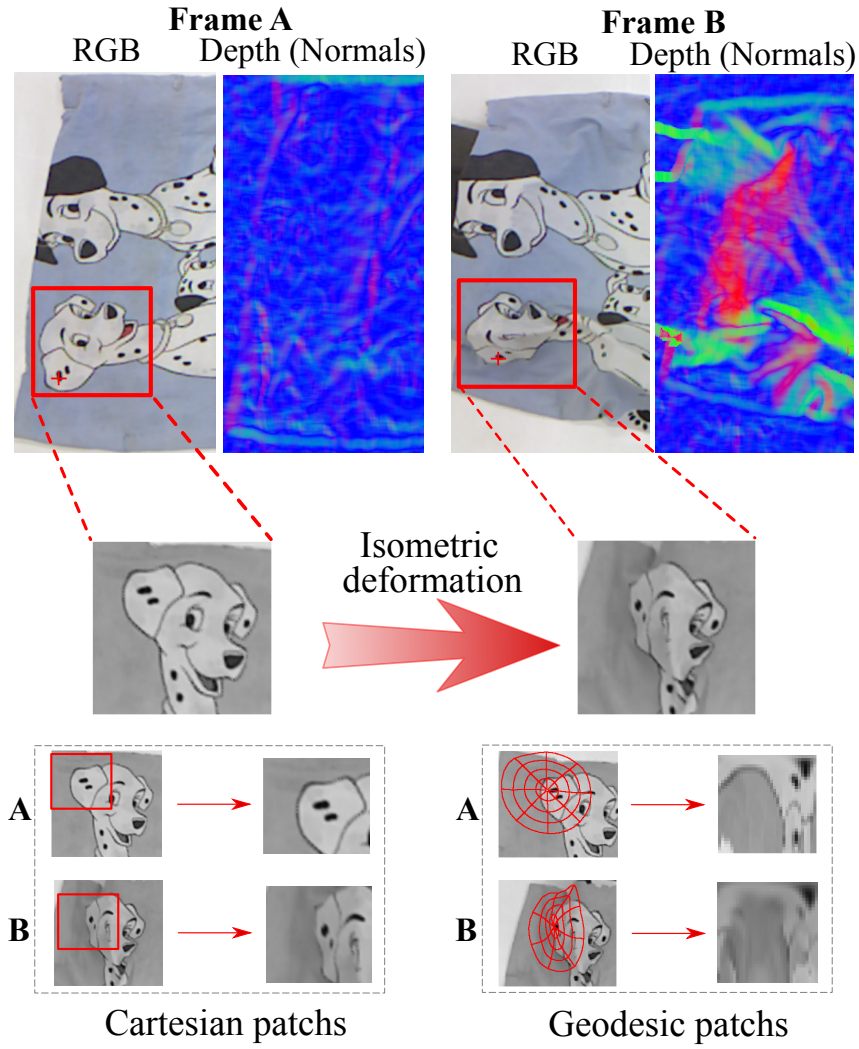


Figure 4.1: **Example of an RGB-D patch rectified by the geodesic mapping function** $f(\cdot)$. The deformation is represented by the color changes in the surface normals from the depth (the normal vectors orientations are encoded by color as shown in the first row). The image intensity is sampled from a geodesic polar grid extracted from the mesh and is shown in red color at the bottom right. The resulting intensities and gradients of the sampled geodesic patch maintains features of the original texture, while the induced deformation around the keypoint distorts the Cartesian patch shown in the bottom left.

4.1 Depth Preprocessing

In general, depth information obtained through Infrared (IR) sensors and stereo systems comes with considerable noise and missing values, requiring treatment before feature extraction. In our pipeline, we apply a preprocessing stage that denoises the depth values and fill missing data when possible. Therefore, we implemented strategies to (i) remove noise and artifacts from the raw depth image, (ii) fill missing roles in continuous surfaces when the missing depth region is small enough for interpolating the depth, and finally (iii) we build a mesh from the filtered depth image using efficient neighboring

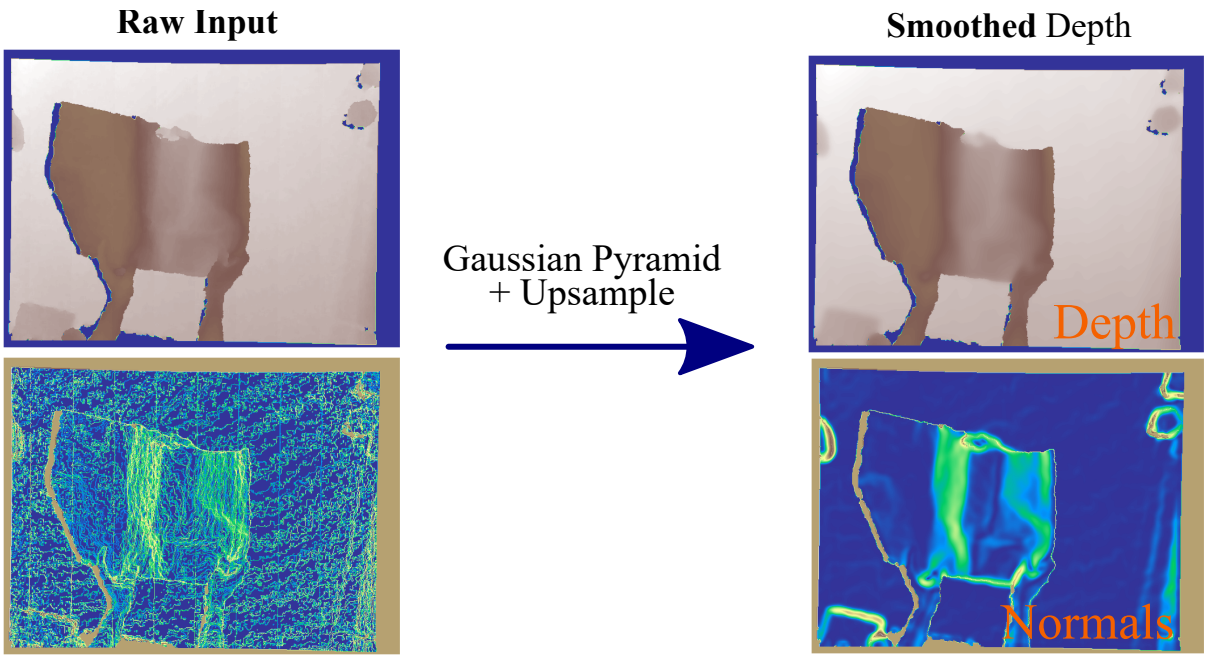


Figure 4.2: **Visualization of the noisy and noise-free geometry.** The left column shows the raw input depth from Kinect1, where discretization artifacts and depth noise are evident, especially when visualizing the normals. These artifacts contribute to reduced algorithm performance. The right column presents the smoothed geometry obtained using the preprocessing steps, which significantly reduces noise and artifacts.

connectivity strategy. All these steps are performed before running the description stage.

4.1.1 Depth denoising

To remove high-frequency noise and discretization artifacts introduced by the discrete depth planes of stereo matching algorithms (e.g., those in Kinect1), we first sub-sample the depth using a Gaussian smoothing pyramid strategy. Two downsampling iterations are applied, which provided the best trade-off between denoising and resolution preservation in our experiments. The number of downsampling iterations is resolution-dependent: for each doubling of the base resolution (VGA), one additional downsampling iteration is applied.

After filtering, the image will have a smaller resolution; we thus upsample the filtered depth map using nearest neighbor interpolation to avoid creating new measurements in missing depth values. Figure 4.2 shows an example of the denoising step, where significant artifacts are effectively removed from the raw depth input.

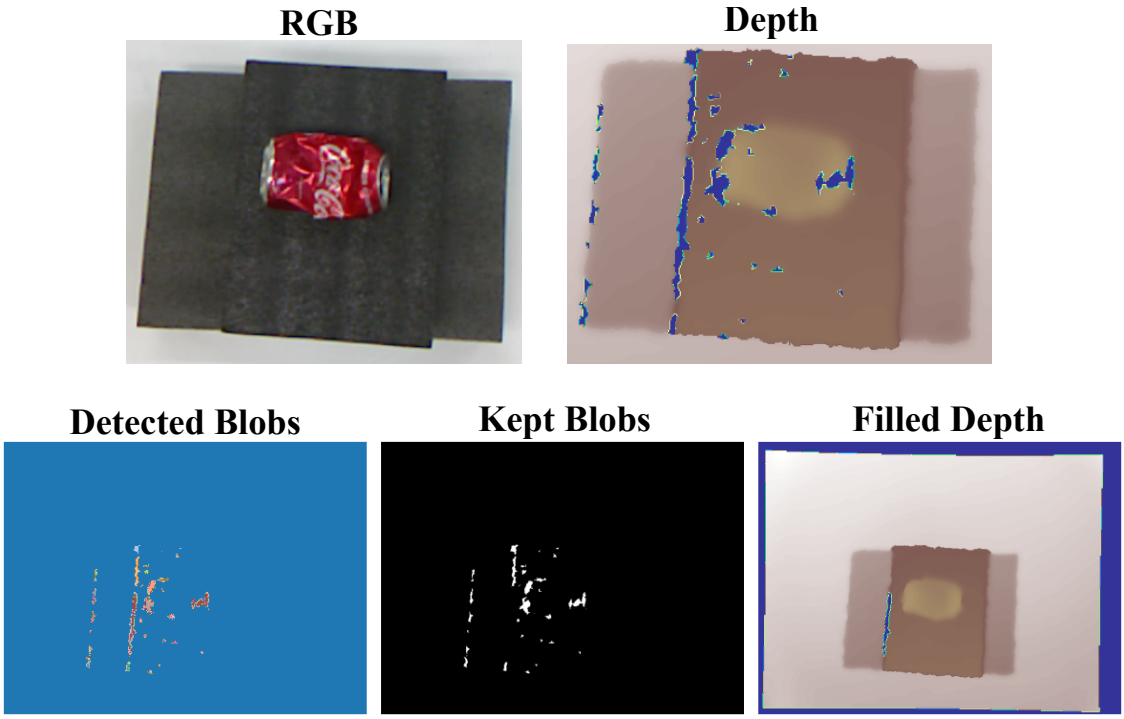


Figure 4.3: **Filling missing depth values.** For some objects, the Kinect sensor cannot provide complete depth maps due to the loss of IR signal. In this case, for our algorithms to be able to still compute descriptors for regions with missing depth, we interpolate depth values using the vicinity information of valid depth measurements.

4.1.2 Hole filling

Depth sensors based on IR illumination are susceptible to reflectance issues due to material properties. For instance, dark or reflective surfaces may interfere with the camera's emitted IR pattern, preventing depth estimation in certain object regions and resulting in holes in the depth map, as shown in Figure 4.3. To address the holes issue in the depth map $\mathcal{D}$, we first segment the depth image into valid and non-valid regions considering the depth values. Then, we use a simple algorithm based on the connectivity of the pixels to segment invididual regions into blobs of missing depth. Finally, we apply Inverse Distance Weighting (IDW) interpolation, using the surrounding known depth values to fill these gaps. If the blob perimeter exceeds 400 pixels (determined empirically on a set of relevant images with holes), we avoid interpolating the missing values due to the excessive amount of missing data, which may introduce additional artifacts.

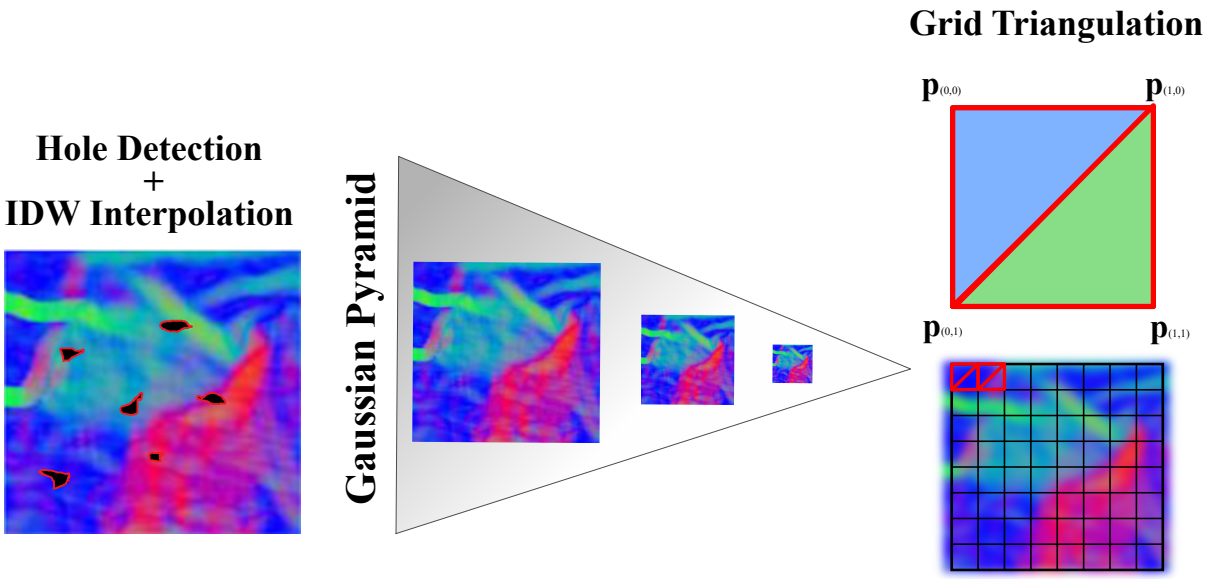


Figure 4.4: **Preprocessing & mesh triangulation.** We obtain the final mesh representation $\mathcal{M}$ by first enhancing the depth map $\mathcal{D}$ with the hole filling and denoising steps. Then, we perform the *grid triangulation* step by an efficient tessellation strategy considering the pixel neighborhood in image space.

4.1.3 Mesh construction

Converting the depth map $\mathcal{D}$ into a simplicial mesh $\mathcal{M}$ is an essential step for applying many advanced computational geometry algorithms. Since the depth map is already structured in a regular grid, we exploit this fact to triangulate the mesh using a fast tessellation strategy based on pixel neighborhoods in image space. Given an image pixel $\mathbf{p}$, we consider its neighboring pixels in the structured image grid to compute the simplicial facets. This process produces a two-manifold, non-self-intersecting triangular mesh by construction, which are key properties for computing geodesic distances. Figure 4.4 illustrates the complete process of depth filtering and fast mesh triangulation. The improvements from applying preprocessing steps to the final mesh representation used by the feature extraction algorithms are shown in Figure 4.5.

4.2 Geodesic Mapping Function

The first stage of our proposed methods consists in extracting the local intrinsic surface properties. These properties are encoded by the geodesic mapping function, and

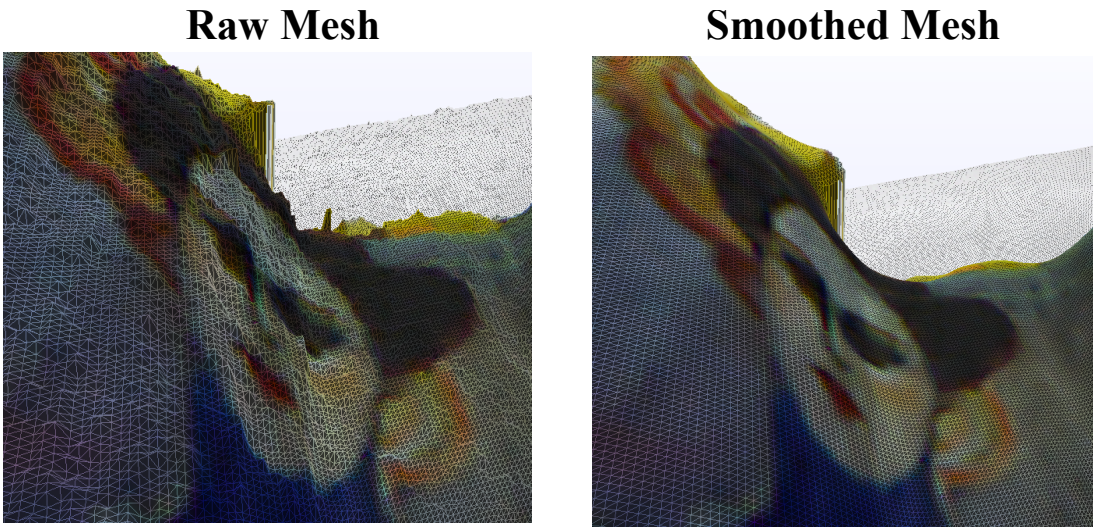


Figure 4.5: **Comparison between raw and filtered depth data used for constructing the final mesh.** When triangulating the raw depth without preprocessing, significant noise is evident (left image). In contrast, the refined mesh in the right image better approximates the surface curvature, enabling more reliable geodesic distance estimation.

can be computed using different strategies. We first have considered the heat flow method [25] for computing geodesic distances, however, we have later observed that one could improve efficiency by approximating geodesics in a local coordinate frame for each keypoint. In the following sections, we describe how to compute the mapping function using the heat flow, in addition to our proposed geodesic paths expansion procedure, which is more efficient for computing local geodesics.

4.2.1 Geodesic Approximation with Heat Flow

In this section, we describe how to compute the geodesic distance between any two points in a 2D manifold using a diffusion strategy, named the Heat Method (or heat flow) proposed by Crane *et al.* [25]. Although other strategies could be used (*e.g.*, the fast marching algorithm [135]), the heat flow approximation brings us the advantage of pre-factoring for efficiency when computing the geodesic distance field from several points in the same mesh.

Let $\mathbf{u} \in \mathbb{R}^{|\boldsymbol{\mathcal{V}}|}$ be a piecewise linear function on a 2D manifold, *i.e.*, a simplicial complex mesh $\boldsymbol{\mathcal{M}}$ that comprises a collection of triangles and vertices $\boldsymbol{\mathcal{V}} = \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$, where each edge is shared by at most two triangles. For each vector on a triangle with unit normal $\mathbf{N}$ and face area A_f, $\mathbf{e}_i^1$ and $\mathbf{e}_i^2$ are the two edge vectors incident to the vertex i, and u_i is the value at the opposing vertex. We denote the function $\phi : \boldsymbol{\mathcal{V}} \times \boldsymbol{\mathcal{V}} \rightarrow \mathbb{R}_+$ as

the geodesic distance approximation between any pair of vertices. In order to approximate the geodesic distance ϕ using the heat flow, we solve the Poisson equation:

$$\mathbf{L}_C\phi = \boldsymbol{\nabla} \cdot \mathcal{X}, \tag{4.1}$$

where $\mathbf{L}_C \in \mathbb{R}^{|V|\times|V|}$ is the cotangent Laplacian matrix, and $\boldsymbol{\nabla} \cdot \mathcal{X}$ contains the integrated divergences computed in the normalized vector field $\mathcal{X}$. In a 2D manifold sampled as a triangular mesh, the following divergence operator approximation holds [25]:

$$\boldsymbol{\nabla} \cdot \mathcal{X} = \frac{1}{2}\sum \cot\theta_1(\mathbf{e}_i^1 \cdot \mathcal{X}_j) + \cot\theta_2(\mathbf{e}_i^2 \cdot \mathcal{X}_j), \tag{4.2}$$

where, for each vertex i, we sum over all adjacent triangles j of vertex i. The angles θ_1 and θ_2 are the opposing angles of vertex i and the vectors $\mathcal{X}_j$ are gathered from $\mathcal{X} = -\nabla\mathbf{u}/\|\nabla\mathbf{u}\|_2$, where the discrete gradient $\nabla\mathbf{u}$ can be computed as:

$$\nabla\mathbf{u} = \frac{1}{2A_f}\sum_i u_i(\mathbf{N} \times \mathbf{e}_i^1). \tag{4.3}$$

Finally, the $\mathbf{u}$ function using the heat flow, for a fixed time t, is given by solving the system $(\mathbf{U} - t\mathbf{L}_C)\mathbf{u} = \boldsymbol{\delta}_i$, where $\mathbf{U}$ is a diagonal matrix encoding the vertex areas and $\boldsymbol{\delta}_i$ is a vector with 1 in the i-th component and 0 in all others. We then define the set $\boldsymbol{\Phi}$ composed of isocurves after discretizing the ϕ function (see Figure 4.6). Since the geodesic distances are deformation-invariant properties, as far as isometric transforms are concerned, all pixels belonging to a specific isocurve will remain in the same isocurve after the surface deformation.

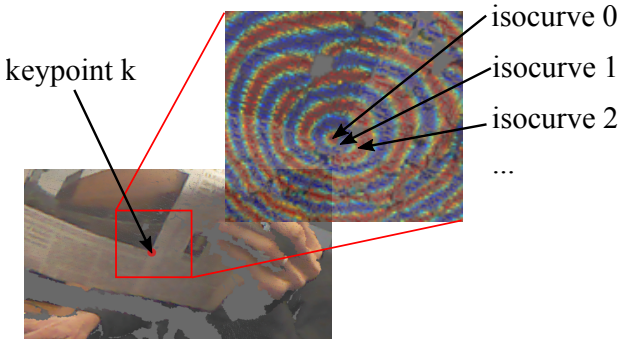


Figure 4.6: **Isocurve generation process with the heat flow method.** After approximating the geodesic distance using heat flow, we discretize the ϕ into isocurves of 4 cm size. Each test pair is localized using the isocurve id and the angle w.r.t. to the patch orientation.

4.2.2 Geodesic Approximation with Geodesic Paths Expansion

By considering a local estimation of the geodesic distances, one drastically increases time performance, and also simplifies the sampling of the scalar field in a triangular mesh by simple linear interpolation. We refer to the resulting image patch as Geodesic Patch (GeoPatch), since it builds an image patch based on the local geodesic distance field. It is worth mentioning that we use the the Geodesic Paths Expansion algorithm as the mapping function $f(\cdot)$ for the final descriptors, since it is much faster than Heatflow for the purpose of estimating the local geodesics around the keypoints, while maintaining the accuracy performance, as we demonstrated in the Experiments chapter.

Given a *manifold* $\boldsymbol{\mathcal{M}} = (\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{E}})$ defined by a triangular mesh, where $\boldsymbol{\mathcal{E}} \in \boldsymbol{\mathcal{V}} \times \boldsymbol{\mathcal{V}}$ are the faces edges and $\boldsymbol{\mathcal{V}} = \{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$ the vertices $\in \mathbb{R}^3$, similar to the ISC descriptor approach [63], we define a local polar coordinate system (u, r) on each keypoint $\mathbf{k}_i \in \boldsymbol{\mathcal{K}}$, where the center is the respective vertex $\mathbf{v}_i$, u is the angular coordinate and r is the radial coordinate in geodesic units. Then, we construct a mapping function $f : \mathbb{R}^3 \to \mathbb{R}^2$ for an image patch that maps points on the manifold embedded in $\mathbb{R}^3$ using the estimated geodesic distances, to the Cartesian coordinate system of an image patch (see Figure 4.7). Since there is a rotation ambiguity, we arbitrarily set the canonical orientation to be axis-aligned with the RGB image; thus, the constructed patch is not invariant to image in-plane rotations yet, which will be handled in later steps. In order to sample the points in this coordinate system, the walking direction axis u is discretized in m uniform bins such as $u_i = 2\pi i/m$, $i \in \{1, \ldots, m\}$. For the distance coordinate r, a constant σ (walking distance step) is chosen in order to sample the distance in $r_j = j\sigma$, $j \in \{1, \ldots, n\}$. Therefore, we construct the image patch by mapping the geodesic polar coordinates to the rectified patch $\mathbf{P}^{m \times n}$ by sampling m orientations and, for each orientation, n points in a geodesic path direction (the blue dots showed in Figure 4.7), which are geodesically equidistant from each other.

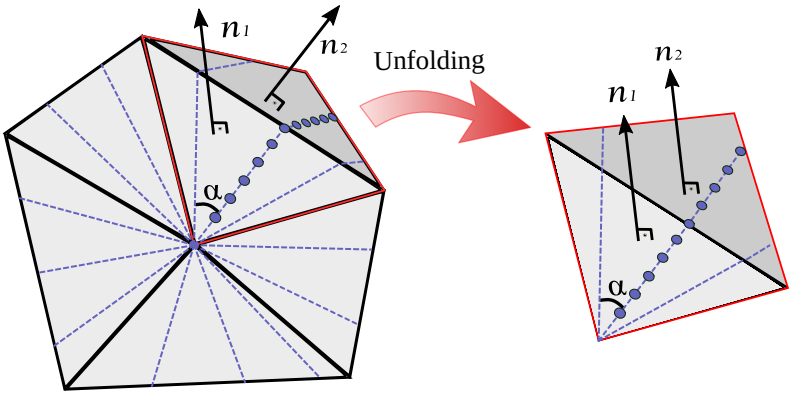


Figure 4.7: **Geodesic paths computation in a local polar coordinate system**. Each walking direction is cast from the center vertex (keypoint). The angle $u_i = i\alpha$ between each ray is defined by discretizing the unit circle by the number of desired angular bins (we used 32 bins in our implementation). The blue dots along a path represent the points $(u_i, r_j), i \in \{1, ..., m\}, j \in \{1, ..., n\}$ that are equi-sampled in the i-th geodesic path.

Algorithm 1: Geodesic Paths Expansion $f(\cdot)$

BuildPatch $(\mathcal{M}, \mathcal{I}, \mathbf{v}_i, m, n, \sigma)$

inputs : A manifold mesh $\mathcal{M}$; the image $\mathcal{I}$; the keypoint vertex $\mathbf{v}_i$; the number of angular and radial bins m, n, and sampling distance σ.

output : A $\mathbf{P}^{m \times n}$ geodesically rectified image patch.

$\mathbf{U} \leftarrow$ `sampleDirs`$(\mathcal{M}, \mathbf{v}_i, m)$;

foreach *Direction* $u_i \in \mathbf{U}$ **do**

- $r_{max} \leftarrow 0$;
- $\mathcal{P} \leftarrow \{\mathbf{v}_i\}$; `/* Intersection set */`
- $v_{current} \leftarrow \mathbf{v}_i$;
- **while** $r_{max} < n\sigma$ **do**
  - $v_{next} \leftarrow$ `intersect`$(\mathcal{M}, v_{current}, u_i)$;
  - $u_i \leftarrow$ `unfold`$(\mathcal{M}, v_{next}, u_i)$;
  - $\mathcal{P} \leftarrow \mathcal{P} \cup \{v_{next}\}$;
  - r_{max} += $||v_{next} - v_{current}||_2$;
  - $v_{current} \leftarrow v_{next}$;
- $\Omega \leftarrow$ `sampleEquidistant`$(\mathcal{P}, n, \sigma)$;
- **foreach** *Sampled 3D point* $\mathbf{v}_j \in \Omega$ **do**
  - $\tilde{\mathbf{p}} \sim \pi(\mathbf{v}_j)$;
  - $\mathbf{P}_{[i,j]} \leftarrow \mathcal{I}(\tilde{\mathbf{p}})$;

return $\mathbf{P}$;

The geodesic path is incrementally created by applying a similar idea to the Fast Marching algorithm [135], which computes the geodesic distance between vertices in a triangular mesh. Since our problem only requires the ray to be cast in a specific known

direction on the manifold, we only need to find edge intersections and perform vector rotations in one direction, leading to a number of computations proportional to the size of the support region. This procedure is simpler and more efficient than Fast Marching. Thus, let $\mathbf{v}_i$ be the coordinate system center, each ray is shot outward the center, and a geodesic path is created considering the initial direction of each bin. The initial directions lie on the plane of their respective faces, bounded by one-ring distant faces. Then, we compute the intersection of the direction and the next triangle edge. In order to continue the path, we rotate the direction $\mathbf{u}$ of the ray assuring that whenever the next face is unfolded to have the same normal as the current face, the result is a straight line between the triangles. This is achieved by applying Rodrigues' rotation formula:

$$\mathbf{u}_r = \mathbf{u}\cos\theta + (\mathbf{k} \times \mathbf{u})\sin\theta + \mathbf{k}(\mathbf{k} \cdot \mathbf{u})(1 - \cos\theta), \tag{4.4}$$

where $\mathbf{u}_r$ is the rotated direction, $\mathbf{k} = \mathbf{n}_1 \times \mathbf{n}_2$ is the orthogonal direction, and θ is the angle between the face normals $\mathbf{n}_1$ and $\mathbf{n}_2$. We call this step *unfolding* and it is illustrated in Figure 4.7. The unfolding is repeated until the maximum geodesic distance is met.

Each sampled 3D point $\mathbf{v} = \begin{bmatrix} X & Y & Z \end{bmatrix}^T$ along the geodesic path is projected onto the RGB image coordinates using the perspective projection function π as follows:

$$\pi(\mathbf{v}) = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \tag{4.5}$$

where f_x and f_y are the focal lengths of the camera, s the skew factor accounting to non-rectangular pixels, and c_x and c_y the coordinates of the principal point. The intensity values from the image are linearly interpolated, filling each pixel of the geodesic patch. The Algorithm 1 shows the steps to construct the geodesic image patch.

An illustration of a real strong deformation example is shown in Figure 4.1, where we compare the geodesic patch to the classic Cartesian sampling approach. The geodesic grid plot indicates that the geodesic distances between the sampled points do not change, granting isometry invariance for the constructed geodesic patch. One can see that the Cartesian sampled patch presents textures such as the eyes of the dog that were not present in the undeformed patch A. Also, the ear of the dog is completely distorted by the non-rigid deformation. Conversely, the geodesic patch kept the features accordingly to the undeformed template, although some parts were blurred by the interpolation method, which is inevitable since the original light rays coming from the surface of the object are sub-sampled by the camera sensor in such extreme deformation case.

4.3 Geodesic-Aware Feature Extraction

After mapping pixels locations from a deformable surface to a rectified image patch $\mathbf{P}$ using the mapping function f, we proceed to extract the visual features $\mathbf{d}_f$. We present two novel efficient geodesic-aware descriptors. The first, called *GeoBit*, is a handcrafted binary descriptor that encodes deformation-invariant features using a vector of binary tests and the second method, named *GeoPatch*, is based on learning the feature extraction with a shallow convolutional neural network.

4.3.1 GeoBit: Binary Descriptor

The GeoBit descriptor exploits visual and geometrical information to encode deformation-invariant features into a binary vector. While the use of texture information results in a high distinctiveness descriptor, the depth information allows us to define the binary tests invariant to non-rigid deformations and scale, as previously discussed in Section 4.2.

4.3.1.1 Binary feature extraction

After computing the geodesic patch of the keypoint's neighborhood, we can compute the visual features based on a predefined set of binary intensity tests over the polar coordinates, as depicted in Figure 4.9. GeoBit performs binary tests in the neighborhood around the keypoint. These tests are based on a set of pixels selected by a distribution function. We tested two different distributions, as suggested in BRIEF [17].

The pattern of each distribution is illustrated in Figure 4.9. Assuming that the origin of the patch coordinate system is located at the keypoint, we selected 512 pairs of pixels using the following distributions: i) An isotropic Gaussian distribution $\mathcal{N}(0, \frac{30^2}{100})$, whose standard deviation is derived from BRIEF's implementation [15]; and ii) a uniform distribution, where we randomly selected $1,024$ different angles and isocurves. In our experiments, the Gaussian distribution yielded a slight performance improvement. This result aligns with the findings of BRIEF [17], where similar improvements were observed with the Gaussian distribution. The intuitive explanation for its superiority is that tests

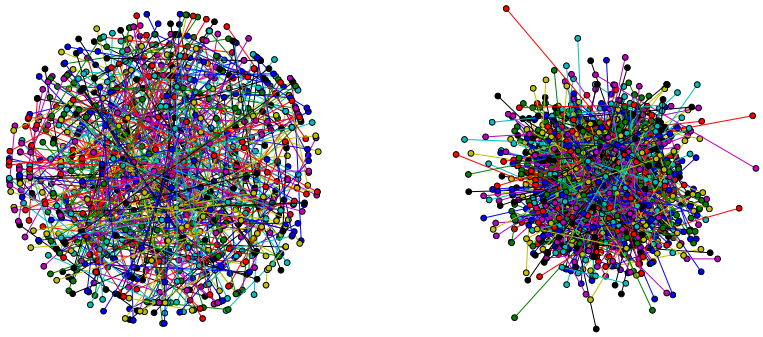


Figure 4.9: **Binary tests patterns using uniform (on the left) and normal distributions (on the right)**. We tested both distributions and found that the Gaussian distribution results in slightly higher recognition rates.

are concentrated near the keypoint's center, where changes due to rotation, perspective, and distortions are less pronounced compared to the patch extremities.

Our binary descriptor implementation has a dimension of 512 bits for each computed orientation, and 16 possible orientations, resulting in a memory footprint of $1,024$ bytes. The direction changes of gradients around the keypoint computed using image intensity comparison tests have small memory storage requirements, and can be matched very efficiently in modern CPUs by leveraging vectorized low-level instructions. Given an image keypoint $\mathbf{k} \in \boldsymbol{\mathcal{K}}$, we extract the rectified geodesic patch $\mathbf{P}$ centered at $\mathbf{k}$ as discussed in Section 4.2. We then sample pixel pairs around the keypoint $\mathbf{k}$ using a fixed pattern with locations given by a distribution (Figure 4.9 shows two tested distribution patterns). We store, for each point in the pattern, the isocurve passing at r_i, and the rotation α w.r.t. the patch orientation, as illustrated in Figure 4.8 with two test pairs of points lying onto two different isocurves. We can then build the set $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \ldots, n\}$, as the fixed set of sampled pairs from $\mathbf{P}$, where $\mathbf{x}_i$ and $\mathbf{y}_i$ encode the isocurve and angle of the i-th pixel of the binary test pair, *e.g.*, $\mathbf{x_i} = (\alpha_i, r_i)^T$. The extracted descriptor from the patch $\mathbf{P}$ associated with the keypoint $\mathbf{k}$ is then represented as the binary string:

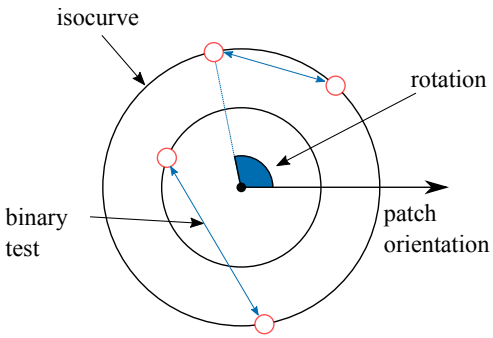


Figure 4.8: **Example of two binary tests to extract the visual features**. For each binary test in the pattern, we store the isocurve c and the rotation α w.r.t. the patch orientation of two points.

$$\mathbf{d}_f = \sum_{1}^{n} 2^{i-1} [\mathbf{P}(\mathbf{x}_i) < \mathbf{P}(\mathbf{y}_i)], \tag{4.6}$$

where $\mathbf{P}(\mathbf{x}_i)$ returns the pixel intensity in the polar coordinates $\mathbf{x}_i$ and $[t]$ is the Iverson bracket that returns 1 if the predicate t is true and 0 otherwise. The comparison in the

bracket captures gradient changes in the keypoint neighborhood.

Similar to DaLI descriptor, in GeoBit, the invariance to rotation is achieved by producing rotated versions of the tests' set in polar coordinates $\mathcal{S}_\theta$ after transforming all tests $\mathcal{S}$ in the first axis by a horizontal shift $\mathbf{t}_\theta$:

$$\mathcal{S}_\theta = \{(\mathbf{t}_\theta(\mathbf{x}_i), \mathbf{t}_\theta(\mathbf{y}_i))|(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{S}\}. \tag{4.7}$$

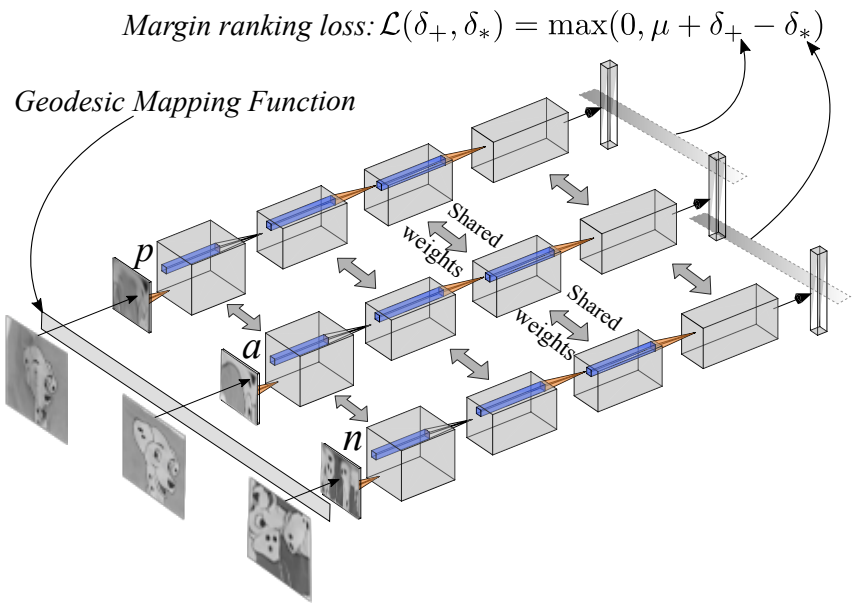
Therefore for each keypoint we compute a set of candidate descriptors (we used 16 candidates in our experiments) with different orientations by rotating the coordinates of the pattern points in set $\mathcal{S}$ using discretized rotations uniformly sampled from $[0, 2\pi]$, *i.e.*, adding $\theta = n\pi/8,\ n \in \{0, \ldots, 15\}$ to the first coordinate $\mathbf{t}_\theta(\mathbf{x}_i) = (u_i + \theta, r_i)$.

In the matching step, we select the feature vector with an orientation that results in the smallest hamming distance between two compared keypoints. This strategy has shown better performance when compared to calculating the canonical orientation for each keypoint using gradient-based approaches, mainly because non-rigid deformations around the keypoints introduce additional noise in the orientation estimation.

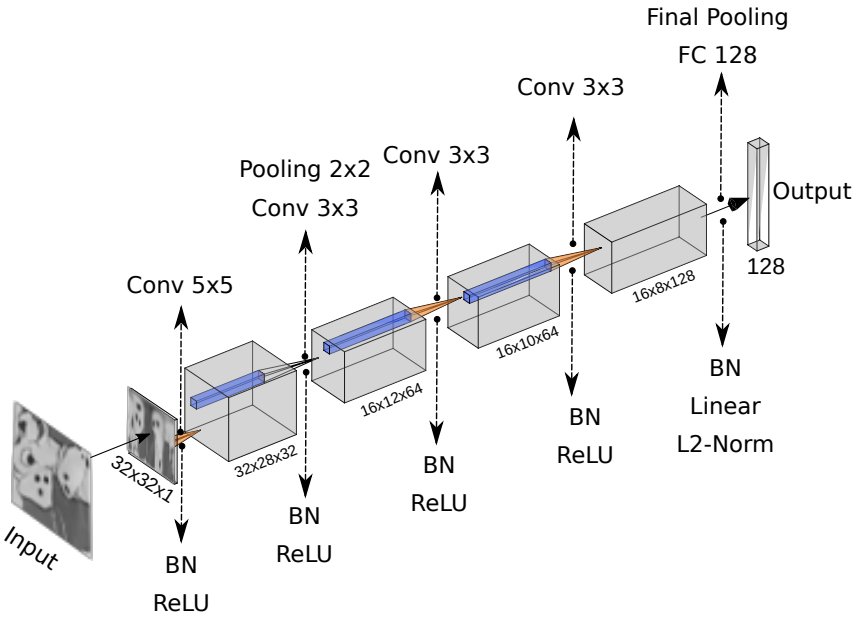
4.3.2 GeoPatch: Geodesic-Guided Feature Learning

In this section, we describe the learning-based geodesic descriptor. The 128-dimensional floating-point feature vector $\mathbf{d}_f$ for a given image patch $\mathbf{P}$ is computed by forward-propagation of the rectified patch $\mathbf{P}$ in the network $\mathbf{G}$, *i.e.*, $\mathbf{d}_f = \mathbf{G}(\mathbf{P})$. We adopted a variation of a shallow convolutional architecture from [151] to build $\mathbf{G}$. Figure 4.10 illustrates the geodesic patch estimation and the network architecture during the training and prediction stages. Although in this work, we chose a compact network for inference efficiency, deeper and more complex networks can be employed as well. Our goal with the GeoPatch descriptor is to demonstrate that we can efficiently use a classic CNN to handle a non-Euclidean geometric task. By using the proposed geodesic image patch to feed a compact CNN, we do not overwhelm the network to handle many different image transformations, which inadvertently leads to overfitting. Moreover, we take advantage of the network's powerful ability to learn feature maps that can be used to extract a compact but distinctive descriptor.

The network $\mathbf{G}$ is trained using N triplets of patches $(\mathbf{a}^{(i)},\ \mathbf{p}^{(i)}, \mathbf{n}^{(i)})$, where $\mathbf{a}$ is an anchor patch, $\mathbf{p}$ is a positive corresponding patch (a projection of the same keypoint in different views), and $\mathbf{n}$ is a non-matching (negative) patch of the keypoint. The triple siamese network architecture used during training is detailed in Figure 4.10.



(a) Triple siamese training setup.



(b) Detailed branch architecture.

Figure 4.10: **Triple siamese architecture used in the GeoPatch descriptor**. During the training, we fed a siamese CNN (a) with the patch triplets $(\mathbf{a}, \mathbf{p}, \mathbf{n})$. The margin ranking loss is computed and the errors are back-propagated. After the training stage, one branch is used to extract a 128-dimensional descriptor $\mathbf{d}_f$. The detailed architecture of the branch is shown in (b).

The triplets are built by sampling indices of patches from a uniform distribution, considering a training dataset with annotated image matches. We use the margin ranking loss function with an anchor swap to train the network. The margin ranking loss with anchor swap considers the L_2 distance δ between the computed descriptors after a mini-

batch forward pass:

$$\mathcal{L}\left(\delta_{+}^{(\cdot)}, \delta_{*}^{(\cdot)}\right) = \frac{1}{N}\sum_{i=1}^{N} \max(0, \mu + \delta_{+}^{(i)} - \delta_{*}^{(i)}), \tag{4.8}$$

where $\delta_{+} = \|\mathbf{G}(p) - \mathbf{G}(a)\|_2$ is the distance between the positive and anchor patches, and

$$\delta_{*} = \min(\|\mathbf{G}(p) - \mathbf{G}(n)\|_2, \|\mathbf{G}(a) - \mathbf{G}(n)\|_2), \tag{4.9}$$

is the hardest negative distance in the triplet, since there is two possible negative distances (while there is only one possible positive distance). Finally, μ is a positive scalar encoding the margin's length. By using the hardest negative distance in the triplet, the gradient updates are larger, which means that the network is learning more with little computation overhead since no extra forward passes are needed, and the three possible distance computations in the triplet are negligible. When minimizing the margin ranking loss, the network seeks to bring closer matching patches in the descriptor space, while at the same time, it tries to push out non-matching ones of the hypersphere defined by the margin μ, as shown in Figure 4.11.

Our network implementation employs circular padding in the horizontal axis before all convolution layers to take advantage of the circularity of the polar grid sampling in the patch construction. Due to the equivariance property of the polar sampling, rotations in the polar space are converted to horizontal shifts in the sampled patches. In this context, pooling operations provide invariance to small translations in the patches, *i.e.*, to rotations of up to 4 pixels in the image, since there are two 2×2 max pooling operations. To achieve full invariance to rotation, we perform max pooling in the horizontal axis of the tensor (angle axis in geodesic patch) before the fully-connected layer. We also tested to augment the data by applying horizontal shifts to the patches during training, however, the first method works slighly better according to our experiments.

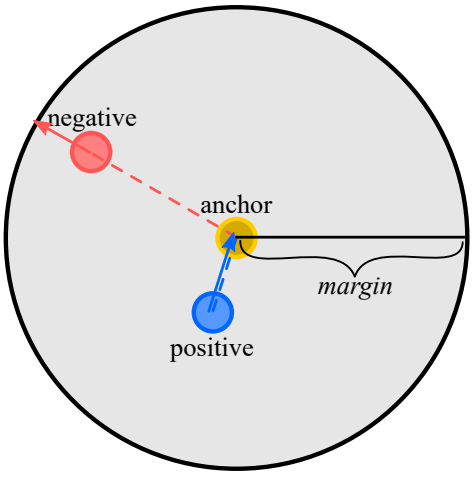


Figure 4.11: **Visual interpretation of the margin ranking loss**. The loss seeks to pull the positive data point closer to the anchor point, while pushing the negative example away. In the ideal case, if we pull the positive sample to be exactly at the anchor point, and the negative sample to be away by the margin, the loss achieves its minimum.

In the last layer of the network, the convolutional filters are flattened and forwarded to a fully connected layer with linear activation function, projecting the features to a 128 dimensional space, which are then L_2 normalized to produce the final descriptor feature vector $\mathbf{d}_f$.

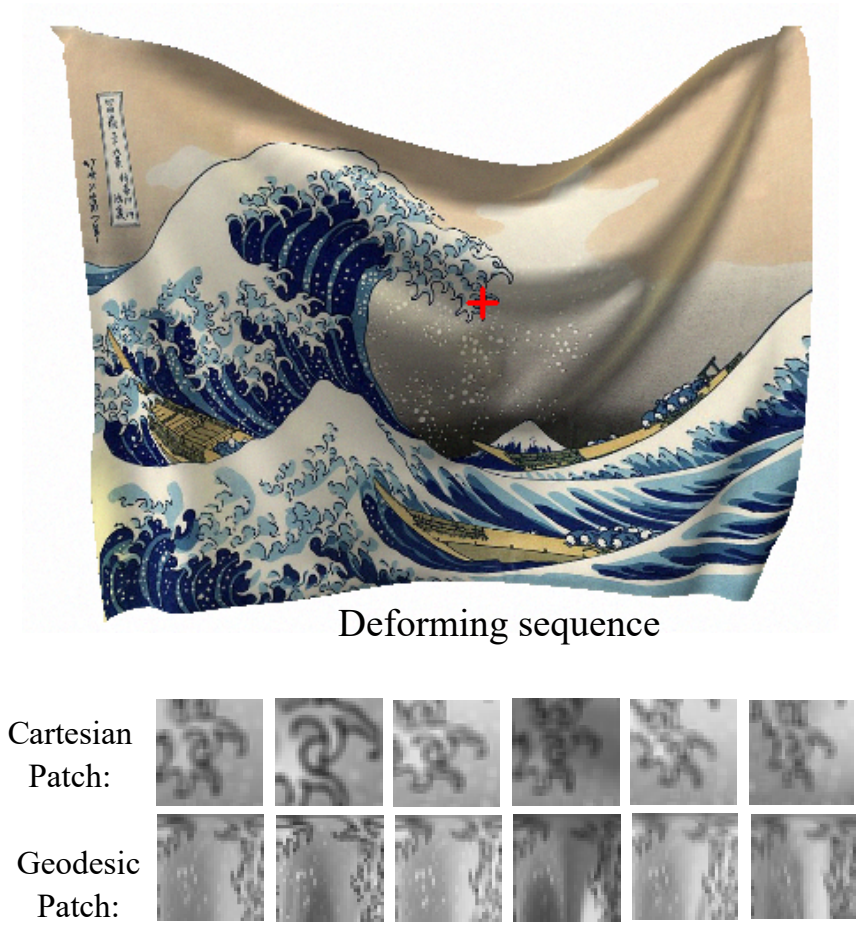


Figure 4.12: **Geodesic versus Cartesian patch sampling**. In this visualization, we track a single keypoint and compute a local patch using two strategies: (i) geodesic and (ii) Cartesian sampling. Notice that the Cartesian sampling approach is sensitive to local deformations, while our proposed geodesic sampling is able to rectify the intensity changes induced by the deformations that are being applied to the object. It is worth mentioning that there are changes induced by non-linear illumination as well, which should be handled by the descriptor extractor in both Cartesian and geodesic sampling.

Although we have adopted a simple network backbone with the margin ranking loss, more recent network architectures can easily achieve deformation invariance by using the proposed geodesic patches, as demonstrated in Figure 4.12, where one can replace the classic Cartesian sampling with our proposed geodesic sampling approach. Since our goal is to demonstrate that the proposed rectification scheme can be used to learn a local feature descriptor invariant to isometries and rotation using a generic patch-based CNN, we opted to adopt as backbone a simpler, lightweight network for designing GeoPatch, which is capable of modeling geodesic polar patches.

4.4 RGB-D Non-Rigid Datasets

The lack of non-rigid image benchmarks in the literature motivated us to introduce a new dataset composed of two datasets of real-world objects acquired with two different RGB-D sensors, and an additional simulated dataset with thousands of RGB-D images of deforming objects, all having ground-truth correspondences. We have developed a physics simulation of particles to create arbitrary non-rigid isometric deformations with ground-truth correspondences, resulting in deforming surfaces that act like a bed-sheet in a clothesline on a windy day. The physics simulation framework is implemented in OpenGL, achieving the necessary efficiency and flexibility to allow low-cost generation of thousands of images of realistic deforming objects suitable for training Deep Neural Networks and matching performance benchmarking. The simulation also provides efficient low-level access to the simulation data, *i.e.*, Z-buffer, camera parameters, and perfect correspondence in the sequences. Details about the simulation algorithm is provided in the Theoretical Background (Section 2.2.2)

4.4.1 Real-World Data

To evaluate the matching capability of existing descriptors on real-world images of deformable surfaces, we built a new dataset of deforming objects [1] composed of 11 deformable objects, which are split into two sets considering the sensor used and the annotation method, as shown in Figure 4.13. The RGB-D images were captured with Kinect™, versions 1 and 2.

In the first set, the Kinect 1 sequence, the images were acquired at a resolution of 640×480 pixels with Kinect version 1, and image correspondences of a set of landmarks were manually annotated. The landmarks are selected by a human annotator taking into consideration both the points' distinctiveness and uniform spatial distribution of points in the image. In the second set, the Kinect 2 sequence, the images are at resolution $1,920 \times 1,080$ acquired with the Kinect version 2, and flat reflective markers tightly fixed in a uniform grid behind the objects' surfaces determined the position of the landmarks. In this sequences, all correspondences between frames were automatically obtained with a high precision motion capture system (OptiTrack™).

The intrinsic camera matrix $\mathbf{K}$ for both Kinect 1 and Kinect 2 is obtained through

[1] Publicly available at https://www.verlab.dcc.ufmg.br/descriptors

Figure 4.13: **Examples of real-world and simulated (sim.) data in our dataset**. The objects are subjected to deformations while frames are acquired with an RGB-D camera. For each object, we provide ground-truth keypoint correspondences that can be used to evaluate the performance of local descriptors.

the device firmware, utilizing the default calibration provided by the manufacturer. We visually inspected the alignment between depth and texture at various depths used in the experiments and observed no visible distortions. Therefore, we assumed the factory calibration parameters to be accurate for these devices.

A robust intensity-based registration approach was developed to estimate a ground-truth deformation model for each image pair. The goal of the registration is to establish a TPS deformation model of densified keypoints. Since the TPS model, in general, tends to have larger errors in regions far from the control points, it is not possible to simply apply a TPS with a sparse set of correspondences and use them as ground-truth directly. In this sense, a first deformation model using TPS is estimated through the sparse set of annotated correspondences, *i.e.*, the landmarks' location, which will be close to global optima with respect to some photometric error cost (we used the Structure Similarity (SSIM) metric in the optimization also considering a multi-resolution Gaussian pyramid of four levels). The control points are densified by regularly sampling points in the source image and the deformation model parameters are then photometrically optimized via gradient descent. As result, we obtain a refined deformation model between each image pair that can be used to establish correspondences of detected keypoints reliably. Ground-truth matches for the Blueflower and Van Gogh sequence are depicted in Figure 4.15.

In addition to the proposed dataset of deformable objects, we enhance the public

Figure 4.14: **Sample image for each of DeSurT's objects**. We used the coarse annotated correspondences from the DeSurT dataset to generate accurate correspondences, with the TPS optimization technique described in this section.

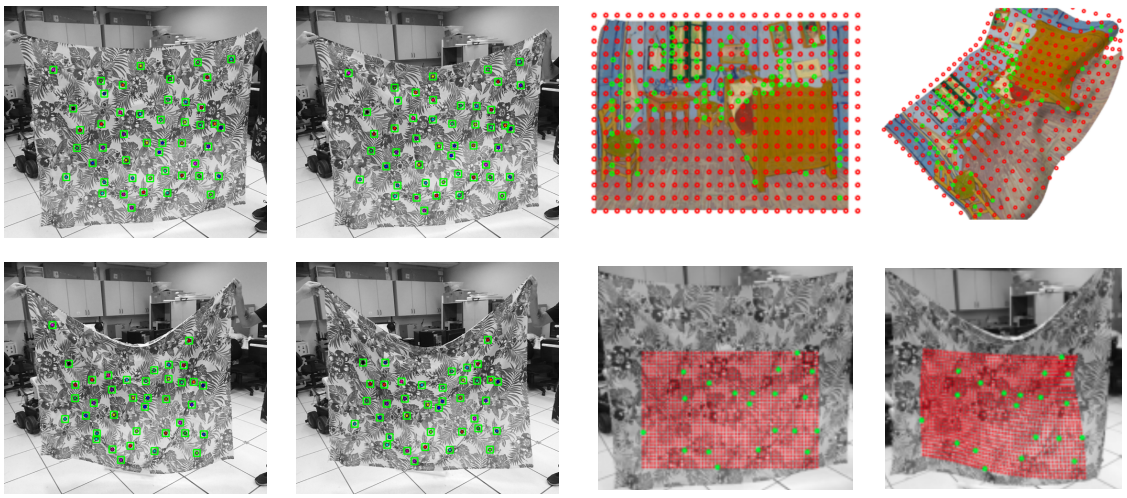


Figure 4.15: **Ground-truth correspondences**. A mocap and manual annotation are used to estimate pixelwise-accurate landmark tracking in the images (green squares on the left plots). The right plots illustrate the estimation of the accurate ground-truth deformation model. Starting from the sparse set of accurate ground-truth annotated correspondences (indicated by the green points) as initialization to the deformation, we refine the deformation by registering the source and template images using a dense set of control points regularly sampled in the reference frame (shown in red). The deformation model is stored and used later to compute the matching score metric for the independently detected keypoints for each image pair.

dataset DeSurT [157], where correspondences are used as landmarks to compute the TPS model for the DeSurT dataset. DeSurT contains challenging RGB-D sequences of deforming objects, as depicted in Figure 4.14. Several objects from DeSurT exhibit textureless and repetitive textures, posing challenges to all local descriptors, complementing our acquired datasets in these aspects.

4.4.2 Synthetic Data

In our simulated environment, objects are represented by a grid of particles having mass in 3D space. Considering a grid of particles having mass and a 3D position, Newton's second law is applied in conjunction with Verlet integration, to act over the particles' position, *i.e.*, when forces like wind and gravity are applied. A constraint satisfaction optimization step is performed over all particles to enforce constant distance of neighboring particles, thus keeping the deformation isometric. Details about the simulation algorithm are provided in Section 2.2.2. Deformations are induced by the forces applied onto them, implemented as wind and gravity. For each simulation round, we generate (i) random wind forces in all directions to generate diverse object deformations in a chaotic fashion, (ii) illumination variation such as intensity, global position, number of light sources, directional lighting, and color changes to enforce realistic non-linear illumination diversity, and (iii) Gaussian noise in image pixels to simulate real camera sensors.

The texture is applied onto the mesh generated by the grid and rendered with diffuse illumination as the cloth moves (which causes non-linear illumination changes). While the simulation is running, pixels are uniformly sampled from the image and the Harris corner score is used to retain approximately 100 corner-like features. These 100 points are used as landmarks to compute the TPS model, which provides the groundtruth correspondence to the initial rough deformation model, which is refined by the registration using the denser sets of keypoints. For each simulation, we make sure to use random inputs for the simulation parameters, including wind force, wind direction, illumination strength, illumination position, and Gaussian noise in the image pixels, ensuring different outcomes in every run, resulting in realistic images and rich deformations.

To obtain plausible textures for the simulations, we built a broad set of images by merging several Structure-from-Motion image collections [160]. We first filtered out images that contain people (for privacy reasons) by using an ensemble of image detectors and then removed the few remaining images containing people by hand. Finally, images that do not have a minimum number of detected keypoints were also removed. The texture dataset was imported in the simulation, and for each texture, we ran the simulation with different input parameters and grabbed 30 snapshots, generating a set of 30 RGB-D images with ground-truth correspondences.

4.5 Experiments

In this section, we describe the set of experiments used to assess our descriptors in matching, tracking, and object retrieval tasks. We evaluate GeoBit and GeoPatch with both simulated and real data and compare the results against different descriptors. We adopt the Matching Score (MS) as metrics of comparison. The simulated datasets provide a tractable but realistic set up to test specific behavior of the descriptors, while the real data demonstrate the applicability of the approach on real scenarios.

4.6 Baselines and Metrics

We compared our results against the well-known binary descriptors for 2D images ORB [120] and FREAK [1]; a floating-point gradient based descriptor: DAISY [142]; a descriptor that combines texture and shape: BRAND [91]; a deformation-invariant descriptor: DaLI [129]; and two learning methods, TFeat [151] and Log-Polar [36], which is an improvement of HardNet [87].

A detailed performance assessment was conducted using the MS metric as defined by [85] and SIFT keypoints detected independently for each frame. After detecting the keypoints using the SIFT detector, we selected the $2,048$ most salient keypoints according to the response attribute. The ground-truth matches are given by the estimated ground-truth deformation model between the two frames.

In our experiments, we matched all pairs of keypoints from two images using exhaustive (brute-force) matching, *i.e.*, comparing each descriptor to all others in the other set to find its nearest neighbor. We labeled as valid matches, two keypoints corresponding to the same physical location (according to the ground-truth) as positive, and as negative otherwise.

The matching score [85] is given by the number of correct matches divided by the smallest number of keypoints detected independently from the two images. Notice that since the keypoints are detected independently, the repeatability rate of the points may no longer be one. We use SIFT keypoints as the standard keypoint detector to compute the matching score for all local descriptors that are keypoint based.

4.7 Parameters and Implementation Details

Similar to classic RGB descriptors, the main parameter of our descriptors is the size of the support region around a keypoint defined by the isocurve thickness. By testing a set of empirically chosen values for all descriptors on the Bag1 dataset, we experimentally found that using a support region of 75 millimeters for the geodesic patch estimation, and keypoint radius of 15 pixels for the RGB descriptors, produces slightly better recognition rates for all methods. Thus, considering these optimal values, the support region size of each method is then fixed in the rest of the experiments.

4.7.1 Convolutional Network Training

To train the CNN, we used Stochastic Gradient Descent (SGD) with learning rate $lr = 0.1$, weight decay of 10^{-4} on all weights including bias, and a batch size of $1,000$ samples. The network was trained for 200 epochs, which was sufficient for convergence on the validation set. The network was entirely trained on simulation data generated by the described approach in Subsection 4.4.2 using internet photo collections as textures. The training set contains 1.6M different deforming patches extracted from 2,000 different image instances obtained from the 1DSfM dataset [160] after filtering people and repeated images. The simulated dataset images used in the experiment comes from an independent simulation instance, using unrelated images as textures, *i.e.*, there is no data from the synthetic data of our dataset in the training stage.

4.8 Experimental Results

This section presents the experimental analysis conducted to compare our method with several baselines available in the literature. In addition to our proposed benchmark datasets, we also consider the DeSurT [157] RGB-D dataset for the task of non-rigid surface correspondence. Furthermore, we ablate our method under different configurations and conduct an in-depth parameter sensitivity analysis. To conclude, we test relevant methods in two different application tasks, namely, object retrieval and non-rigid surface

tracking.

4.8.1 Matching Performance

We note that among all methodologies, our descriptor GeoPatch stands out as the descriptor with the highest averages in both matching score over different deformations. From these results, we can draw the following observations. First, BRAND performance is drastically reduced by deformations since its computation is based on the normals of a support region, which is not an intrinsic property of a surface, hence not being invariant to non-rigid isometric deformations. Second, the photometric information is also impaired by the deformations, which penalizes twice RGB-D descriptors not aware of deformations like BRAND.

The performance of the RGB descriptors is also directly impacted by the deformations, since all of them use a sampling strategy over image pixels defined in the Cartesian space, and any deformation including projective transformations decreases their performance. On the other hand, our methods provide invariance to both projective (rigid) and isometric non-rigid deformations in image space, and their performance is only affected by the quality of the input depth map. This negative effect is observed in the Kinect 1 dataset, where the depth measurements have a higher level of noise when compared to Kinect 2.

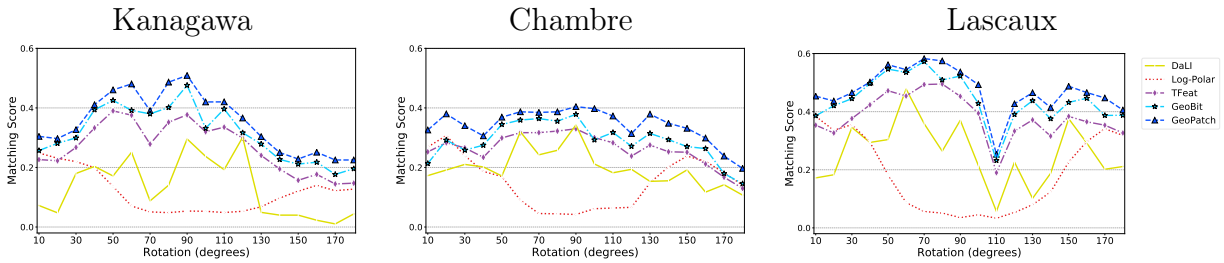
The TFeat and Log-Polar descriptors achieved the second-best results in mean Recognition Rate and AUC values, which indicates that local descriptors can benefit from deep learning approaches. It is worth mentioning that while TFeat and Log-Polar were trained on real large-scale datasets with ground-truth correspondences obtained with structure-from-motion settings, allowing them to better generalize to unseen patches, our learned descriptor was entirely trained on simulation data. Nevertheless, our method was capable of performing well on real benchmarks beyond the classic metrics, as we shall demonstrate in this section with two different applications, namely, image retrieval and non-rigid tracking.

Aside from our datasets and DeSurT, we also evaluated the GeoPatch (the descriptor with the best results) in objects without shape deformation. We used the *3D Keypoint Matching Benchmark* test-set from [171]. This dataset presents patches with strong illumination changes and point-of-view (wide-baselined correspondences). In this evaluation, we followed the evaluation protocol defined by [171]. After 25 epochs of fine-tuning on the validation-set data, GeoPatch achieved a FPR95 error of 19.57% in the test-set containing $10,000$ unseen matches, while 3DMatch descriptor obtained a 35.3%

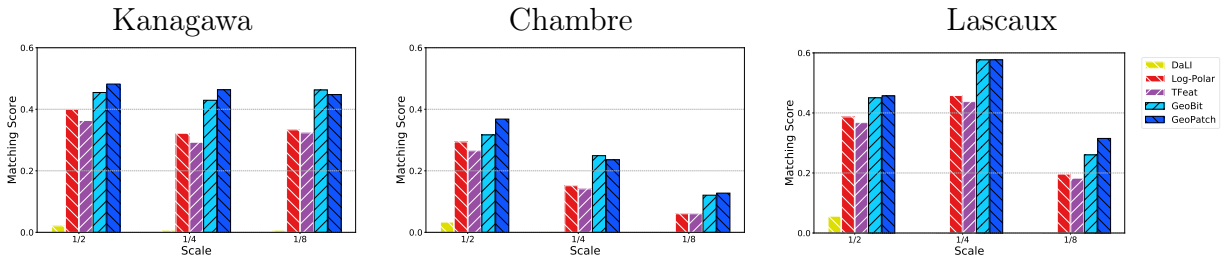
Table 4.1: **Comparison using SIFT keypoints**. Our descriptors are able to provide higher matching scores in the sequences from our dataset and the DeSurT dataset. Best in bold, second-best in italic. All baseline methods are described in the beginning of Subsection 4.6.

Dataset	Object (# pairs)	Avg. Matching Scores								
		BRAND	*DAISY*	*DaLI*	*FREAK*	*Log-Polar*	*ORB*	*TFeat*	*GeoBit*	*GeoPatch*
Kinect 1	Shirt2 (18)	0.24	0.30	0.35	0.33	0.36	0.25	0.32	**0.44**	*0.43*
	Blanket1 (15)	0.15	0.25	0.28	0.22	0.29	0.20	0.26	**0.34**	*0.33*
	Shirt3 (17)	0.21	0.27	0.28	0.28	0.30	0.22	0.27	**0.35**	*0.34*
	Can1 (6)	0.07	0.05	0.09	0.07	*0.16*	0.05	0.13	**0.18**	*0.16*
	Bag1 (4)	0.12	0.08	0.15	0.18	0.22	0.10	0.16	**0.30**	*0.28*
	Shirt1 (14)	0.17	0.26	0.25	0.26	0.30	0.20	0.26	**0.34**	**0.34**
Kinect 2	doramaar_l (29)	0.28	0.35	*0.41*	*0.41*	0.35	0.34	0.35	0.40	**0.42**
	toucan_m (29)	0.26	0.33	**0.38**	*0.36*	0.32	0.28	0.30	0.31	0.35
	blueflower_l (29)	0.27	0.29	0.37	0.36	0.30	0.24	0.27	*0.38*	**0.39**
	blueflower_h (29)	0.20	0.28	**0.37**	0.30	0.30	0.24	0.27	*0.36*	*0.36*
	toucan_h (29)	0.24	0.31	**0.35**	*0.34*	0.33	0.26	0.29	0.28	0.33
	redflower_h (29)	0.12	0.12	*0.17*	0.16	0.13	0.11	0.12	**0.19**	**0.19**
	blueflower_m (29)	0.19	0.27	0.35	0.30	0.30	0.23	0.26	*0.36*	**0.37**
	toucan_l (29)	0.26	0.35	*0.37*	**0.38**	0.36	0.29	0.34	0.29	0.35
	lascaux_l (29)	0.33	0.46	0.52	0.53	0.48	0.44	0.47	*0.55*	**0.56**
	redflower_m (29)	0.14	0.18	0.24	0.21	0.19	0.15	0.18	*0.27*	**0.28**
	redflower_l (29)	0.18	0.21	0.29	0.25	0.22	0.18	0.20	*0.30*	**0.31**
DeSurT	brick (29)	0.20	0.16	*0.31*	0.22	0.28	0.25	0.26	*0.31*	**0.32**
	campus (29)	0.16	0.22	*0.30*	0.17	0.28	0.21	0.25	0.28	**0.32**
	sunset (29)	0.08	0.12	0.11	*0.13*	**0.15**	0.09	0.12	0.12	**0.15**
	cushion1 (29)	0.11	0.17	0.15	0.14	0.20	0.13	0.18	*0.21*	**0.23**
	scene (29)	0.21	0.19	*0.29*	0.20	0.28	0.23	0.27	*0.29*	**0.32**
	cushion2 (29)	0.07	0.06	0.07	0.08	*0.09*	0.06	0.07	**0.11**	*0.09*
	cobble (29)	0.16	0.09	0.21	0.15	0.24	0.21	0.22	*0.26*	**0.28**
	newspaper2 (29)	0.15	0.20	0.21	0.18	*0.24*	0.17	0.21	*0.24*	**0.26**
	newspaper1 (29)	0.16	0.23	0.27	0.19	*0.32*	0.24	0.28	*0.32*	**0.34**
Simulation	kanagawa_rot (18)	0.06	0.22	0.13	0.21	0.12	0.19	0.27	*0.31*	**0.35**
	lascaux_rot (18)	0.08	0.36	0.26	0.31	0.18	0.33	0.38	*0.44*	**0.47**
	kanagawa_scale (3)	0.02	0.11	0.01	0.06	0.35	0.04	0.33	*0.45*	**0.46**
	chambre_scale (3)	0.02	0.06	0.01	0.03	0.17	0.03	0.16	*0.23*	**0.24**
	chambre_rot (18)	0.06	0.25	0.20	0.18	0.15	0.20	0.26	*0.29*	**0.34**
	lascaux_scale (3)	0.02	0.10	0.02	0.08	0.35	0.06	0.33	*0.43*	**0.45**
Mean	*	0.15	0.22	0.24	0.23	0.26	0.20	0.25	*0.31*	**0.33**

FPR95 error in the test-set.



(a) Rotation invariance experiment.



(b) Scale invariance experiment.

Figure 4.16: **Rotation and scale invariance.** Matching score curves obtained by matching SIFT keypoints showing results for (a) rotation and (b) scale for each target frame relative to the reference. This experiment evaluates the robustness of the descriptors to both deformation, scale, rotation, and illumination in the most challenging sequences.

4.8.2 Rotation and Scale Invariance

We also pit our descriptor against other methods in terms of robustness to rotation and scale transformations in a more detailed experiment. For these tests, we used the Simulation dataset, where the camera suffers in-plane rotations ranging from 0° to 180° degrees, using a step size of 10° degrees for rotation. For the scale invariance tests, the camera is moved backward in the Z direction to produce downscaling in image space. The rotation and scale attributes from SIFT keypoints are shared among all descriptors, with exception to DaLI and GeoBit that compare rotated versions of the descriptors, and also GeoPatch, which is invariant to rotation due to the final max pooling in the geodesic patch's angle axis.

Figure 4.16 shows the matching score curves for rotation and scale transforms. The results are given by the score as a function of the rotation angle and scale. It is worth noting that our descriptors outperforms all methods in all frames in both rotation and scale evaluations. We can observe that our learning-based method responded with the lowest variance with respect to rotation, thanks to the pooling scheme presented, which endows the network to be rotation invariant and discriminant at the same time, without a previous orientation estimation step. This response is important since the deformations can introduce additional noise in the orientation estimation step. Under isometric surface

deformations, a geodesic curve is an intrinsic property that is preserved between views. Notably, the scale invariance of our descriptors are obtained from the isometric invariance property of the geodesics contained in the depth information.

We observed a decrease in Log-Polar's performance for the rotation sequences coming from an adverse effect of padding under large image in-plane rotations since all keypoints share the same orientation attribute from SIFT keypoints. Log-Polar uses a very large support region due to its patch sampling scheme. The Log-Polar's original implementation employs a reflection padding on the image borders to ensure all sampled pixels in a keypoint's support region exist. In this case, when there are large rotations between images, the pixels from the padding change inconsistently compared to the patch's real pixels, which can drastically reduce the matching performance of the descriptors computed by Log-Polar for padded keypoints under large in-plane rotation.

4.8.3 Ablation and Parameter Analysis

To verify each component's contribution to our descriptors' performance, we conducted an ablation study followed by an evaluation of the parameter settings for the support region. Table 4.3 shows that the matching score increases when depth data becomes less noisy and the interpolation step slightly increases the quality of the matches. It can be seen in Table 4.3 that our methods work best with the support region size in the range of 75 to 100 millimeters, and adopt the default value of 75 mm for both GeoBit and GeoPatch in our implementation. Moreover, despite the geodesic walk method being able to run several times (up to 60×) in our experiments faster than Heatflow, both Heatflow and the geodesic walk mapping functions provide the same performance in matching quality. Table 4.2 shows that the use of max-pooling in the angle-axis of the output tensor before the fully-connected layer in GeoPatch improves the matching quality for rotation transformations (*chambre_rot* sequence) without strongly diminishing the matching score in the absence of camera rotations (*Shirt3* sequence).

Table 4.2: **Rotation invariance analysis.** We evaluate the impact of the strategies adopted to achieve rotation invariance for GeoPatch in the presence (*chambre_rot* dataset) and absence (*Shirt3* dataset) of rotations. Results are reported in avg. matching scores metric. MaxPool A.A. denotes max-pooling on the angular axis.

Experiment	***Shirt3***	***chambre_rot***
No Rotation	0.355	0.150
MaxPool A.A.	0.335	0.342
Data Augment.	0.324	0.335

Regarding the rotation invariance, we performed a detailed assessment of three strategies. In the first one, we did not consider rotation invariance and used it as a reference implementation. In the second strategy, we performed data augmentation by shifting the patches in the horizontal axis, and the third strategy consisted of pooling the output tensor in the horizontal axis (angle axis in the geodesic patch) before the fully-connected layer. According to the results shown in Table 4.2, we can observe that the pooling strategy works slightly better; thus, we adopted it in our implementation.

Table 4.3: **Ablation and parameter study.** Ablation study for GeoBit and GeoPatch when considering different depth input. Parameter study for GeoBit and GeoPatch when considering different parameters, including algorithmic modules. We report the average matching scores in *Shirt3* and *toucan_medium* datasets.

Experiment	**GeoBit**	**GeoPatch**
Raw Depth	0.263	0.261
Constant Depth	0.321	0.313
Depth Smoothing	0.328	0.342
Heat Flow	0.328	–
Geodesic Paths	0.328	–
Simple Fill	0.324	0.337
Interpolation	0.327	0.341
Support Reg. 50	0.313	0.324
Support Reg. 75	0.328	0.342
Support Reg. 100	0.322	0.346
Support Reg. 125	0.319	0.342

Table 4.3 also shows the ablation and parameter analysis of our method under different conditions of depth. We can observe that using the raw depth without any preprocessing step provides the worst performance in terms of matching scores due to the high-frequency noise present in the data. The constant depth experiment was performed by using all depth as a constant value, considering the object's approximate depth relative to the camera. In this scenario, our descriptors behave like regular RGB descriptors. Finally, the pyramid smoothing achieves the best results, demonstrating that even in noisy conditions, our method is able to extract useful information from depth to rectify the patches.

We also compare our methods to work with not only depth noise but missing data. To test our methods in a worst-case scenario where there are large amounts of holes in the object's surface, we artificially introduced holes in the depth images. An example can be visualized in Figure 4.17. The holes were generated following the observation from [34] where the authors demonstrate the uncertainty of depth values from RGB-D sensors is usually higher in the edges of objects, which are also often edges in the intensity image. Thus, we detect high contrast edges in images and apply morphological operations to generate a hole mask that is applied to the original depth. Two different depth inpaint methods are tested, a simple filling algorithm that iterates over the image and replace invalid depth with the last seen valid depth values, and a more elaborate method that detects the holes and uses inverse distance weighting interpolation scheme to fill the holes

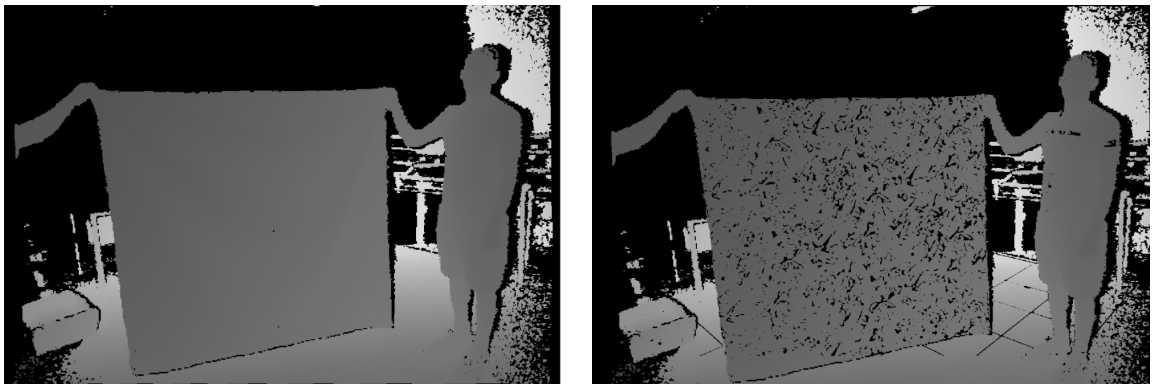


Figure 4.17: **Missing depth completion experiment.** We introduce artificially generated holes in the depth map (right image) to evaluate the robustness of our methods to missing data.

Table 4.4: **Processing timing and descriptor size.** Timing in seconds of each step for the descriptors considering **250** keypoints, and size in bytes of each descriptor.

Method	**Size**	$f(.)$ **map**	**Extraction**	**Matching**	**Total**
Log-Polar	512	–	0.072	0.949	1.021
TFeat	512	–	0.012	1.002	1.014
DaLI	105,600	–	112.95	61.62	174.57
GeoBit	1,024	0.534	0.074	0.531	1.139
GeoPatch	512	0.534	0.009	1.033	1.576

according to the valid pixels around the holes. Results from Table 4.3 indicate that the interpolation scheme provides modest performance gains in the matching scores.

4.8.4 Processing Time

Table 4.4 shows the required time of each step for the compared descriptors in the same hardware. The code was executed on a set of **250** keypoints, images with the resolution of 640×480 running on an Intel (R) Core (TM) i7-7700 CPU @ 3.60 GHz and a GTX 1060 GPU. The matching time refers to the brute-force matching done in CPU.

Both of our proposed descriptors were, on average more than 100 times faster than DaLI, which shows state-of-the-art performance in matching regarding the description of deformable objects. For our method, the $f(.)$ map, which currently runs on the CPU, also includes the computation time of the mesh data structure used to perform the geodesic distance calculations. Regarding our previous strategy to extract geodesic distances, our current method runs several times faster than Heatflow strategy. While using Heatflow

takes 33.263 seconds on the same set of keypoints, our new strategy runs at almost 2 frames per second, achieving a speedup of approximately 60 times for computing the geodesics. The learning approaches run the descriptor extraction on GPU, which results in considerable speed-up in extraction time. GeoBit may be a better choice when no GPU is available since it requires lower computational effort and runs entirely on CPU, and allows fast descriptor matching using the Hamming distance. This difference is clearly visible in the matching time of GeoBit, that matches 16 versions of a single descriptor faster than a single match of all other methods. It is worth noting that GeoBit single descriptor size is 64 bytes, and for fairness, we report the numbers with rotation invariance enabled for all methods in Table 4.4. By reducing the number of rotated versions of GeoBit, it is possible to reduce its memory footprint and increase the time performance even more, at the expense of decreasing its invariance to rotation.

4.9 Applications

Aside from the matching task, we also evaluated our descriptors on two real-world applications: object retrieval and tracking deformable objects. In these applications, we also use the detected SIFT keypoints used to compute matching scores metric. The $2,048$ most salient keypoints according to the response attribute are kept. It is noteworthy that correspondence with detected keypoints is harder than using the annotated ones since several keypoints might not have true correspondences.

In these experiments we include all sequences from DeSurT [157] dataset. There are two main aspects that make this dataset particularly challenging: first, this dataset contains objects with poor texture, including repetitive structures that can be ambiguous to local descriptors; second, it only provides a sparse grid of depth measurements over the object, which requires depth interpolation in order to generate the RGB-D input.

In the applications, we also consider as baselines two recent state-of-the-art methods, namely DELF [139] for image retrieval, and R2D2 [113] for both detection and description of local features for the task of surface tracking. Please also note that their provided implementation did not allow the evaluation on SIFT keypoints as for the other presented methods. This limitation prevented including DELF and R2D2 matching scores on our dataset, where all descriptors are evaluated with the same set of keypoints, either with landmarks or detected SIFT keypoints.

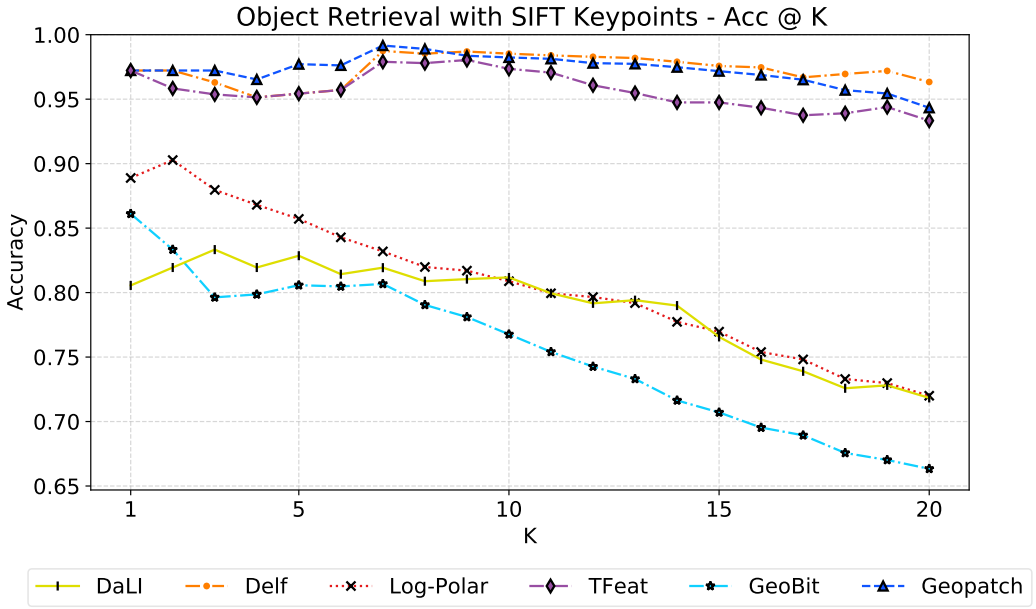


Figure 4.18: **Object retrieval results.** Average accuracy relative to top K results with different descriptors using SIFT keypoints for all descriptors except DELF, and considering images from all datasets.

4.9.1 Object Retrieval

To compare the descriptors in a retrieval task, we implemented an object retrieval method based on the Bag-of-Visual-Words approach [164]. We also include a comparison with the state-of-the-art DELF retrieval pipeline [139]. Worth mentioning is that their provided implementation did not allow the evaluation on SIFT keypoints.

The retrieval application is important to test different aspects of the descriptors, such as the distribution of the descriptor space induced by the descriptor method itself. Having a continuous and well-distributed space is paramount to other useful algorithms that are generally applied to extracted descriptors, such as clustering and approximate nearest neighbor search in large-scale image databases. In order to evaluate these properties, we built a small dictionary of 10 representative centroids by employing the K-medoids clustering [101] for all descriptors. We chose K-medoids to be able to seamlessly use the same method for both binary and floating-point descriptors, and also for a fair comparison among them. The visual dictionary was built on a sample of $10,000$ randomly chosen points considering all the datasets.

Since the original images contain more than the object itself, we used a mask to detect SIFT keypoints only in the object region for all images in the database. This procedure is required because RGB-D descriptors (GeoBit, GeoPatch, and BRAND) also rely on the depth information to successfully extract the descriptors, and in most

images, the background is usually far from the depth sensor and there is no available depth information in such cases. For that reason, the mask is needed to avoid processing keypoints where no depth is available. When processing images in the wild, a simple filter is used to remove keypoints with invalid depth. After building a dictionary for each descriptor method, an image is globally described by building a frequency vector, where each bin corresponds to a visual word. For each feature, its visual word representation is computed by finding the nearest centroid in the dictionary, and its count is incremented in the respective bin in the frequency vector.

The database used for the object retrieval task is the result of the union of four datasets: *Simulated*, *Kinect 1*, *Kinect 2*, and *DeSurT*. After constructing the global descriptor vector for all images, the retrieval experiment consists in choosing the undeformed image as the query, while all images from all datasets compose a single large database of global descriptors. As a metric of comparison, we use the average accuracy of the retrieval over the top K nearest neighbors (Acc @ K), using a k-NN search in the frequency vector space. The accuracy is determined by the number of correct object classes it retrieves over the top search results.

Figure 4.18 shows the accuracy achieved for all methods, and Figure 4.19 displays qualitative examples of the retrieval application. GeoPatch achieved the best performance among all competitors for $K \leq 10$ and competitive results with DELF for $K > 10$, a descriptor specially designed for retrieval tasks and trained on the large Google-Landmarks dataset. GeoBit, on the other hand, performs worse than GeoPatch. This result can be explained by the fact that GeoBit computes 16 descriptors on different orientations and uses the smallest distance between the rotated versions of two keypoints to handle rotation invariance. This strategy works well when comparing pairs of images since there is a smaller probability that a rotated patch that does not correspond to the same physical keypoint will minimize the distance among all possible rotations. However, when a large number of comparisons are considered, such as in the retrieval application, such an event can happen more frequently, resulting in more ambiguity, ultimately decreasing the accuracy.

These results also show that, even GeoPatch being designed and trained to extract distinctive features (not discriminative features) and using a less elaborate architecture, it was capable of presenting competitive results compared to DELF, which uses its own detected keypoints optimized for image retrieval. These results concur with other independent works [149, 22], where the authors observed a clear trade-off between extracting invariant versus discriminative features. While an invariant descriptor can provide better matches, their discriminative power, which is better for classification tasks, is diminished. [125] also pointed out that the descriptor's entropy is related to its distinctiveness, and the matching quality is benefited by descriptors having high variance. GeoPatch was explicitly trained to be invariant and to provide distinctive features, thus it was able to deliver competitive results.

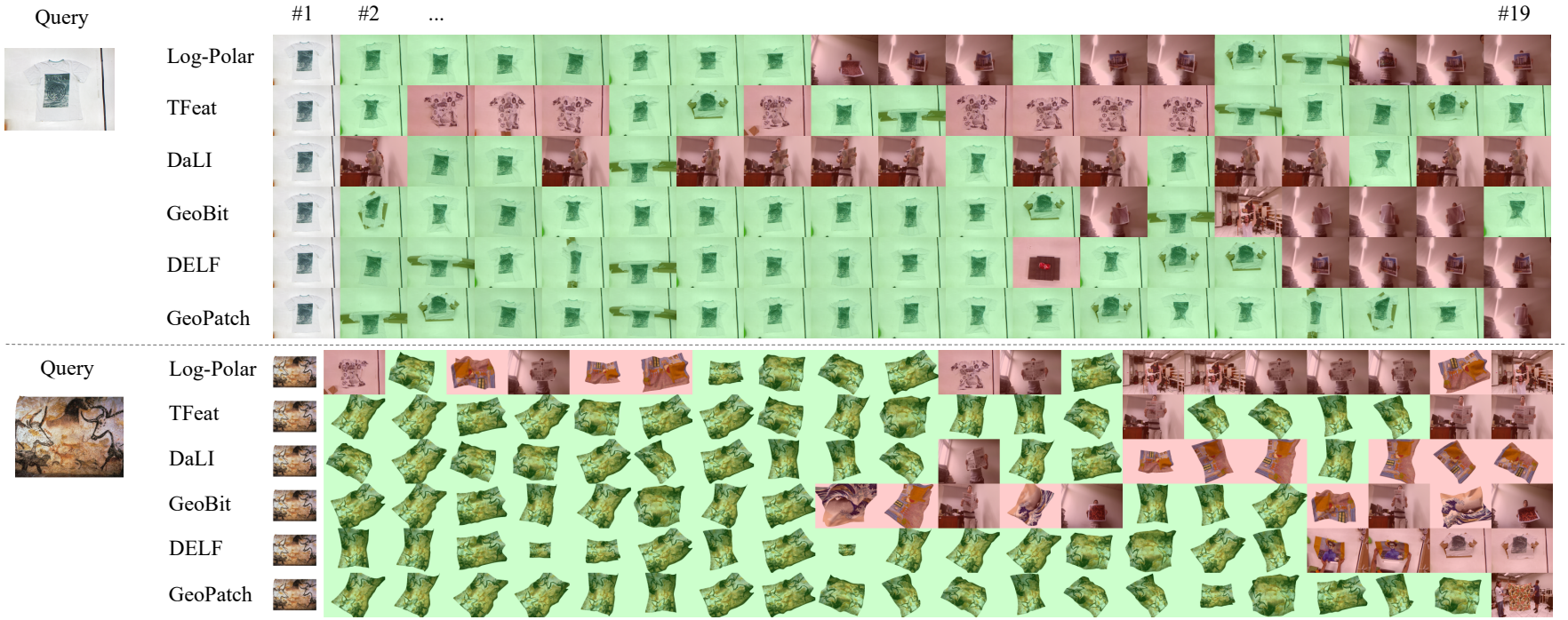


Figure 4.19: **Qualitative results from the object retrieval application.** Given a query image, we ranked the most similar images from the database according to the retrieval framework using each descriptor (incorrectly retrieved objects are colored in red, and correctly ones in green).

4.9.2 Deformable Surface Tracking

In this section, we present the performance of our descriptors when applied to track a region-of-interest of different textured meshes, subjected to large rotations, scale changes, variations on illumination, and strong nonrigid deformations. In this task, given a template patch in the reference image, the goal is to estimate the warp to follow the template on subsequent image frames.

From the previous experiments and results on recognition rate, AUC and matching scores metrics, we also selected the three best competitors in the experiments in addition to our descriptors: DaLI, Log-Polar, and TFeat. Following the image retrieval task, SIFT detector was used to select $2,048$ keypoints in the images. For each descriptor, we computed the matching distance matrix of all visible keypoints, within the template region-of-interest, and performed the correspondences between keypoints using the smallest distances. The template tracking, *i.e.*, the registration between the template (source image) and each current frame (target image), was performed using a Deformable-Affine TPS warp model [32]. To reduce the effects of outliers in the performance of all descriptors, we adopted a RANSAC outlier rejection strategy, following the method presented in [145], to filter out outliers matches. The coordinates of keypoints sets were normalized accordingly to the preprocessing discussed in [145]. Then, the parameters used for detecting outliers for all descriptors were a re-projection hyperplane error threshold $\theta = 0.1$ and a max number of iterations in RANSAC of $1,500$. These parameters were chosen to allow the best operational conditions to all descriptors with a four times margin of iterations when finding a set of good keypoint matches with a probability of 95%, even for descriptors giving up to 80% of wrong matches. Note that although the RANSAC filtering strategy reduces the effects of outliers in the tracking, the descriptors producing higher numbers of good matches result in more accurate template tracking. Furthermore, the higher the number of inlier matches generated by one's descriptor, the smallest the time requirement to find a suitable deformation model.

In order to measure the tracking accuracy quantitatively, we computed the state-of-the-art metric *Learned Perceptual Patch Similarity* distance (LPIPS) [172] to estimate the visual similarity between the tracked patches and the template. LPIPS distance closer to zero indicates higher patch similarity. For a more detailed performance assessment of the matches for each descriptor, we also present the average ratio of inlier matches (*i.e.*, the accuracy from RANSAC during the tracking) and the matching scores metric over the selected correct matches found by RANSAC using the ground-truth correspondences from the generated SIFT keypoints. Therefore, we present both the average accuracy from RANSAC (inlier rate) in the tracking, matching scores from the tracking (which indicates how many correspondences selected by RANSAC are indeed correct matches)

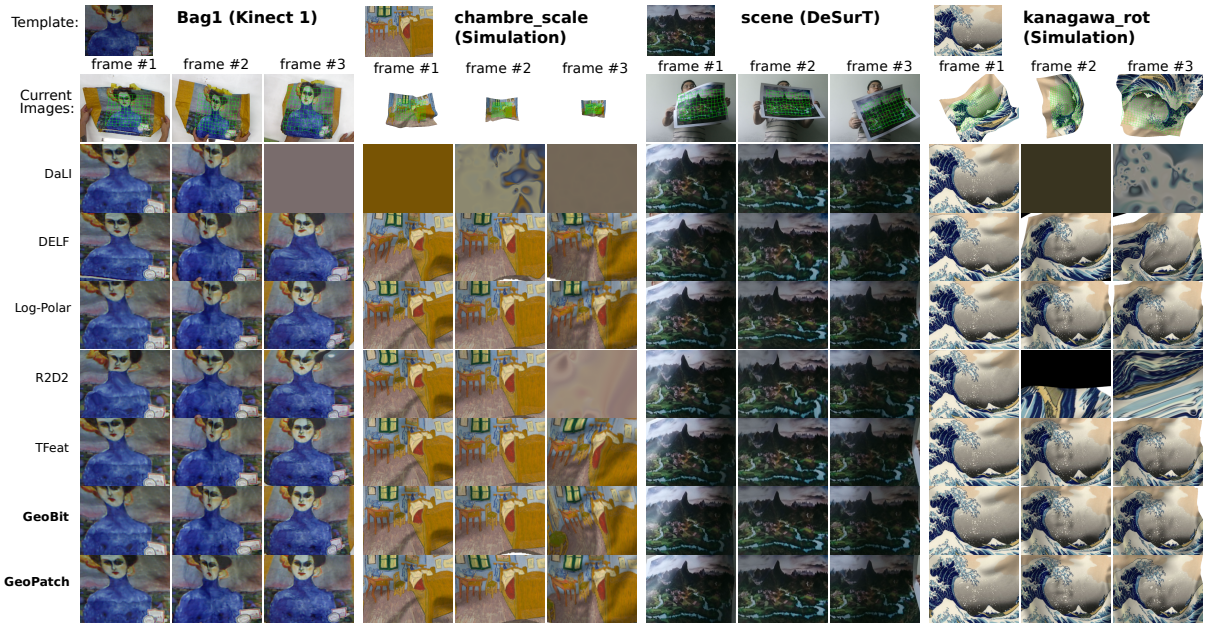


Figure 4.20: **Deformable tracking visual results**. These sequences illustrate the test set used with detected SIFT keypoints containing mild to strong surface deformations, in addition to illumination, orientation, and scale changes. The tracked template region is highlighted by the green grid in the first row of each sequence. Notice the proposed geodesic-aware descriptors performed well for the scenes with strong deformations and scale changes.

Table 4.5: **Evaluation of the tracking application**. Average values on all datasets. Best in bold, second-best in italic, and * indicates descriptors computed on their own detected keypoints.

	Inliers RANSAC	Matching Score	LPIPS
DaLI	0.32	0.31	0.45
DELF*	0.25	–	0.40
Log-Polar	0.36	0.36	0.33
R2D2*	*0.48*	–	0.36
TFeat	0.31	0.37	*0.29*
GeoBit	0.46	*0.42*	**0.27**
GeoPatch	**0.49**	**0.45**	*0.29*

and the LPIPS similarity distance. These quantitative results are shown in Figure 4.21 for the three most challenging sequences *DeSurT*, *Kinect 1*, and *Simulation*. As it can be noticed, GeoPatch and GeoBit presented the highest inlier rates and matching score (higher number of good correspondences) for all sequences, except for *cobble* and *cushion* sequences from DeSurT. Likewise, both geodesic-aware descriptors presented the highest perceptual similarities between the template and tracked regions overall. Table 4.5 also shows a summary of the results when comparing our descriptors against two other learning-based descriptors, namely R2D2 [113] and DELF [139]. Since these two descriptors are designed and trained to simultaneously detect and describe keypoints, we used the available

implementations independently to detect and describe keypoints per image along all our sequences. The results show that our descriptors achieved the best performance even when compared with descriptors using their own keypoints. R2D2 ranked second and third regarding "rate of inliers" and LPIPS in the tracking evaluation. GeoPatch presented higher inlier rates and matching scores than all descriptors, while both GeoBit and GeoPatch displayed the highest similarities overall.

Some visual qualitative results of the tracking are shown in Figure 4.20 with sequences from our proposed nonrigid dataset and from DeSurT. Although most descriptors were capable of handling viewpoint and illumination changes (as illustrated in Figure 4.20 for the sequence *scene* from DeSurT), only the proposed geodesic-aware descriptors were capable of handling frames with strong surface deformations and scale changes. These effects are illustrated for the results in *Blanket1*, *chambre_scale*, *lascaux_rot* and *kanagawa_rot* sequences. The geodesic-aware descriptors are also robust to strong illumination changes induced by these deformations (surface not respecting the Lambertian hypothesis) and by small specular reflections, as it can be noticed in sequences *scene* and *chambre_scale* results shown in Figure 4.20. Together these results suggest the tracking of the regions-of-interest using GeoPatch and GeoBit presents better stability and consistency for both qualitative and quantitative metrics.

Yet, we also found failure situations in challenging sequences, such as *Can1* (Kinect 1) and *cushion2* from DeSurT datasets, when tracking with detected SIFT keypoints for all descriptors. These failures were also observed when using detected keypoints from ORB in the tracking. Some examples are shown in Figure 4.22 for these sequences. They also displayed the worst perceptual similarity and lowest matching scores/RANSAC inlier rates in the metrics presented in Figure 4.21. The images from the can object are subjected to strong specular reflections and have noisy depth maps, while the *cushion2* contains a regular grid with repetitive texture patterns. Overall, all descriptors performed poorly for the cushion texture, although GeoBit succeeds in estimating the affine deformation for a few frames. The estimated TPS warps often did not even model the affine component of the deformation in this sequence with ambiguous keypoints. Concerning *Can1* results, we observed the descriptors using ORB detected keypoints were not capable of handling the local nonrigid deformations.

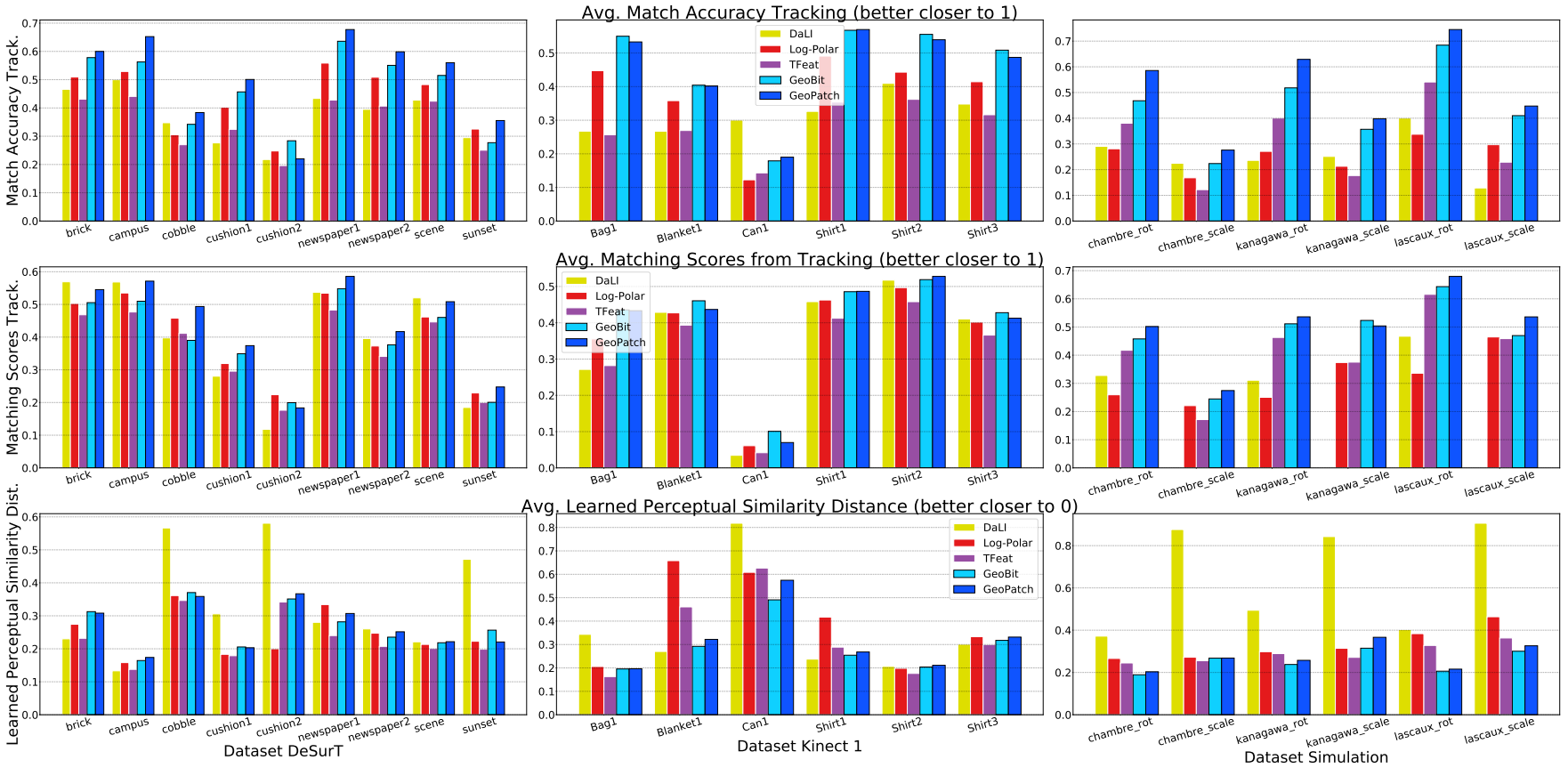


Figure 4.21: **Average matching accuracy (RANSAC inlier ratio), matching scores from the tracking and perceptual patch similarity metrics of tracked objects.** We provide the mean accuracy (RANSAC inlier rate), the matching score values from the correspondences found from RANSAC (using the ground truth correspondence set to check which correspondences are correct) and visual perceptual similarity (using LPIPS distance) in the three most challenging datasets. Notice the best performance in both quantitative metrics for the proposed geodesic-aware descriptors, notably in the sequences containing strong deformations *Kinect 1* and *Simulation*.

4.10 Discussion

In this chapter, we addressed the problem of extracting local features that are isometric invariant. We have proposed a methodology of constructing scale, non-rigid deformation, and rotation invariant descriptors based on intrinsic surface properties. We designed two descriptors, a binary hand-crafted and a learning-based approach. We also show an effective pooling strategy that endows our learned descriptor with rotation invariance without a prior orientation estimation step. Our experiments showed that our methods can achieve significant invariance to isometric deformation, rotation, and scale changes in image space.

Besides the experiments showing that our descriptors outperformed all other techniques in terms of standard metrics used to evaluate local descriptors, we also presented several experiments that demonstrate the benefits of using geometric-aware methods, including real application tasks in image retrieval and tracking. The applications were carefully designed to assess different aspects of the descriptors. Tracking a sequence of deforming objects, in essence, can demonstrate the effectiveness of the approaches in several other related tasks, such as non-rigid reconstruction and Structure-from-Motion. The image retrieval application, for its turn, reveals the distinctiveness of the descriptors in a larger scenario, where a bigger portion of the descriptor space is explored for tasks such as scalable image search and clustering.

The two proposed descriptors were designed to compliment each other's weaknesses. GeoBit can be computed and matched efficiently on CPU, which is an important feature when there is no availability of GPUs or scenarios limited computing resources. Moreover, GeoBit belongs to the family of binary descriptors, proven to be stable, robust and efficient in several critical vision tasks, in contrast to deep learning based descriptors, which are new approaches still being extensively studied. On the other hand, GeoPatch demonstrates generalization in our experiments, outperforming GeoBit in several real sequences and both applications, despite being only trained on simulated data, and can fully leverage modern GPU, being faster than GeoBit when the hardware is available.

We believe these results can promote the growth of approaches leveraging intensity and depth cues for descriptors on RGB-D images. Most descriptors are still limited to exploit only one information source, even when both intensity and color information are available, as for instance, when RGB-D image sequences are available, *e.g.*, Sun3D [162], NYU Depth V2 [128]. Since, to the best of our knowledge, GeoBit and GeoPatch are the first descriptors that fuse geometric and visual information to tackle non-rigid deformations, for the sake of a fair evaluation, we compared our descriptors against well-established methods that cover at least one of the aspects and/or kind of data covered/used by our descriptors. Therefore, we evaluated our descriptors, comparing their performance with

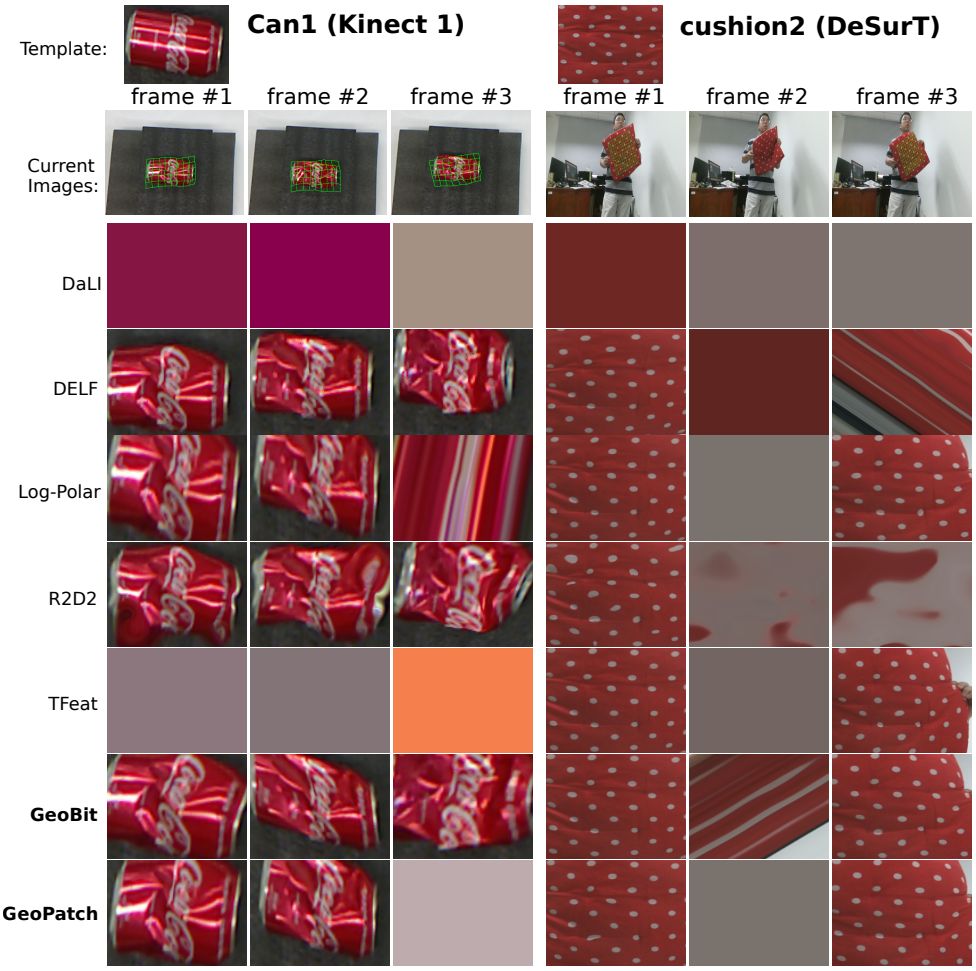


Figure 4.22: **Challenging and failure tracking cases**. The objects in *Can1* and *cushion2* sequences are affected by strong specular reflections, depth map noise, deformations as well as ambiguous texture patterns. All descriptors, using SIFT keypoints, did not provide enough correspondences to model the deformations.

descriptors for images only (ORB, DAISY, FREAK), RGB-D data (BRAND), a non-rigid descriptor for images only (DaLI), and recent learning-based descriptors (TFeat, Log-Polar, DELF, R2D2).

Our results take a step further towards combing photometric and geometric information to render higher performance on local patch description. These results extend the conclusion of [91, 66, 144, 89], where the combined use of intensity and shape information can provide invariance and distinctiveness, consequently improving the quality of keypoint matching. Thus, with the rapid progress being made in the production of multimodal sensors and the availability of inexpensive devices, it is of utmost importance to foster efficient methods that can extract and manipulate the invariance properties taking into account both sources of information.

4.10.1 Limitations

The key limiting factor of our methods when compared to RGB descriptors is the depth and camera calibration requirement, which is not available in many circumstances. Regardless of this limitation, rapid progress is being made in estimating metric depth from monocular images [58, 165], and our descriptor construction methods could benefit from these advances to operate on RGB images with methodological adaptations, which we leave for future work. However, since local feature extraction is often desired to run efficiently, using a large deep model to first extract metric depth and then use it for extracting low-level features is not an efficient choice in many contexts. In the next chapters, we demonstrate how the concepts of explicit deformation modelling can be applied to RGB-only images, particularly, by learning deformations with a deep network, in an efficient and more principled manner.

Another limiting factor is the quality of the depth map, which will directly impact on its performance since the geodesic estimation is derived from it. Finally, when extreme deformation arises on a surface, there is a physical limitation in both the depth and image sensor of a camera from sampling measurements from the surface. In this case, even if we could correctly compute the geodesics, there would be a loss of data due to the interpolation of sub-sampled pixels from the image. Nonetheless, as shown in the experiments, even with noisy measurements from a Kinect sensor, our methods are able to outperform state-of-the-art local descriptors on multiple benchmarks and applications. In addition, with the rapid advancements in time-of-flight depth sensing, commodity sensors can now achieve sub-millimetric accuracy, directly benefiting geodesic-aware descriptors.

Chapter 5

Data-Driven Deformation-Aware Descriptors

In the previous chapter, we introduced the concept of geodesic awareness in local feature extraction and demonstrated an efficient strategy to compute geodesic distances in a local keypoint reference frame using noisy depth data, enabling the construction of a mapping function that projects surface pixels onto rectified image patches. One of the main limitations of the geodesic-aware methods GeoBit and GeoPatch is the metric depth requirement, which significantly limits their applicability in real-world problems. Since depth data is not available in many contexts and use cases, this motivated us to conceptualize and implement strategies that enable local descriptors to correctly and reliably extract invariant features from corresponding points of deforming surfaces from single images. In our nonrigid-world with flexible humans, animals, tissues, and other materials with different deformation properties, the ability to create discriminative and deformation-invariant descriptors using consumer-grade RGB cameras plays a central role in achieving reliable matching in real-world scenarios.

Holding our position that reasoning about the geometric structure of the surfaces should be modelled as well, as stated in our main hypothesis, we first propose to use spatial transformers in order to build a deformation-aware network architecture with a differentiable deformation module, allowing the network to learn to reason about image deformations. In our context, reasoning implies that the network will adjust local geometric transformations based on higher-level image cues, such as shadows, local shape, and texture patterns. Subsequently, we also explore recent findings that suggests an interdependence between keypoints and descriptors [139, 138, 113], and propose a tighly coupled training scheme that allows the keypoints and descriptors to be optimized jointly, increasing matching scores and enhancing performance on real-world applications.

5.1 Extracting Deformation-Aware Local Features by Learning to Deform

In this section, we conceptualize a local descriptor that explicitly handle scene deformations. Since our goal is primarily focused on using solely RGB images, we opted to follow the current paradigm of data-driven neural network solutions, because the problem of estimating geometric properties from single images is a known ill-posed problem in projective geometry and require higher-level scene understanding, a task that CNNs demonstrated exceptional capabilities in recent advancements on object detection, semantic segmentation and monocular depth estimation tasks. However, since tradicional CNN architectures operate on a regular grid pattern, they are not deformation invariant by construction. Our proposed solution is to endow modern CNNs with a special module, called non-rigid warper, based on recent advancements in STNs [52, 53], allowing differentiable geometric transformations to be learned by the network.

By integrating the non-rigid warper into a CNN, we developed DEAL – DEformation-Aware Local descriptor. DEAL jointly learns to undeform and extract discriminative and invariant features from local regions. The proposed architecture is able to handle deformations and perspective distortions of sampled patches in the vicinity of the keypoint. It also attenuates noise in keypoint attributes which improves the local description performance and robustness to image transformations.

We designed our network as a hybrid approach, in the sense that it first extracts dense features and the final output is a local descriptor for each keypoint, unlike both trends of describing local patches like HardNet [87] or extracting dense features jointly with keypoints such as SuperPoint [31] and R2D2 [113]. Our approach combines the best of these two trends: i) the robustness to occlusion, strong illumination, and perspective changes of local patch-based descriptors that work directly with low-level image information; and ii) the capability of dense methods in encoding local semantics from the image to disambiguate hard pairs. Additionally, our model decouples the problem of learning distinctive descriptors robust to deformations in two complementary tasks, *i.e.*, first estimating a warp with a spatial attention mechanism and then describing the sampled local patches using the estimated non-rigid warp. Figure 5.1 outlines a schematic representation of the model.

For learning the contextual deformation information, we provide several views of deformed objects by applying isometric non-rigid transformations to the objects in a simulated environment as guidance to extract highly discriminative local deformation-aware features. Our approach learns deformation information at training time to produce a more informed descriptor extraction at test time. Unlike non-rigid aware deformations descriptors like GeoBit and GeoPatch, our approach does not require depth data to produce

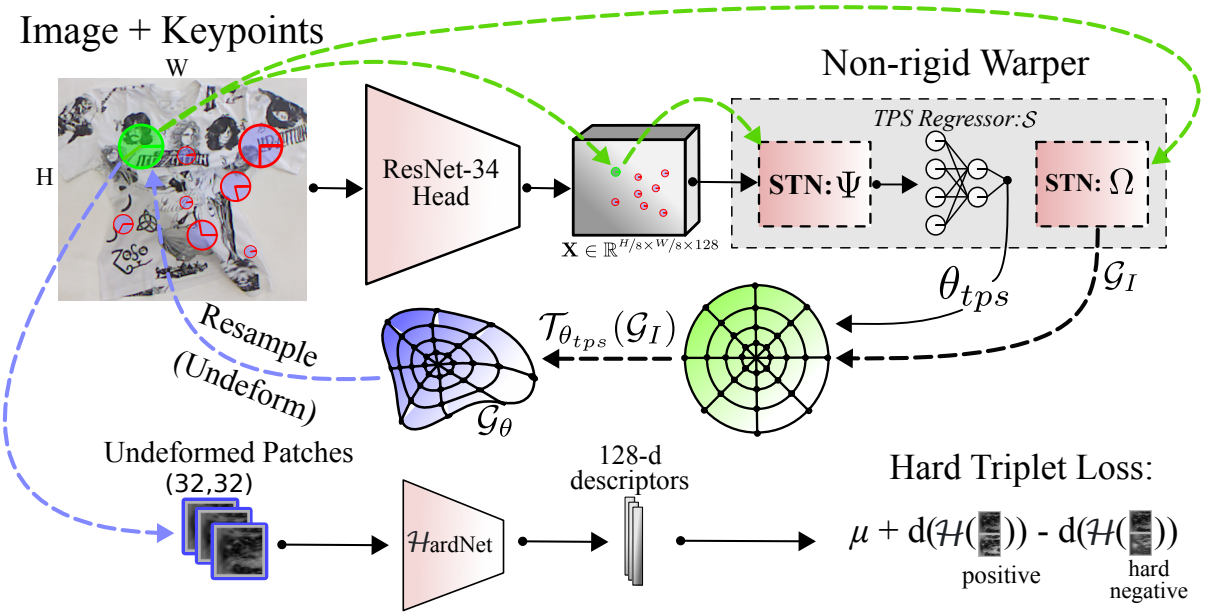


Figure 5.1: **Proposed formulation for computing descriptors of deforming objects.** The non-rigid warper component undeforms local image patches by applying a learned deformation for each keypoint. Higher level image information such as appearance and shape are encoded globally by the ResNet feature tensor $\mathbf{X}$. Then, the two carefully designed spatial transformers and the TPS regressor network decode the information locally, by estimating the warp parameters θ_{tps} used to rectify patches from the original image. The green arrows imply keypoint attribute information. The rectified patches are fed to HardNet that extracts discriminant descriptors. The full network model is trained end-to-end by optimizing the hard triplet loss. $d(\cdot)$ denotes the Euclidean distance.

descriptors robust to non-rigid deformations.

5.1.1 Deformation-aware network architecture

The proposed DEformation-Aware Local descriptor (DEAL) has been designed to jointly learn to undeform and extract discriminative and invariant features from local regions. The proposed architecture is able to handle deformations and perspective distortions of sampled patches in the vicinity of the keypoint. It also attenuates noise in keypoint attributes which improves the local description performance and robustness to image transformations. We designed our network as a hybrid approach, in the sense that it first extracts dense features and the final output is a local descriptor for each keypoint, unlike both trends of describing local patches like HardNet [87] or extracting dense features jointly with keypoints such as SuperPoint [31] and R2D2 [113]. Our approach combines the best of these two trends: i) the robustness to occlusion, strong illumination, and

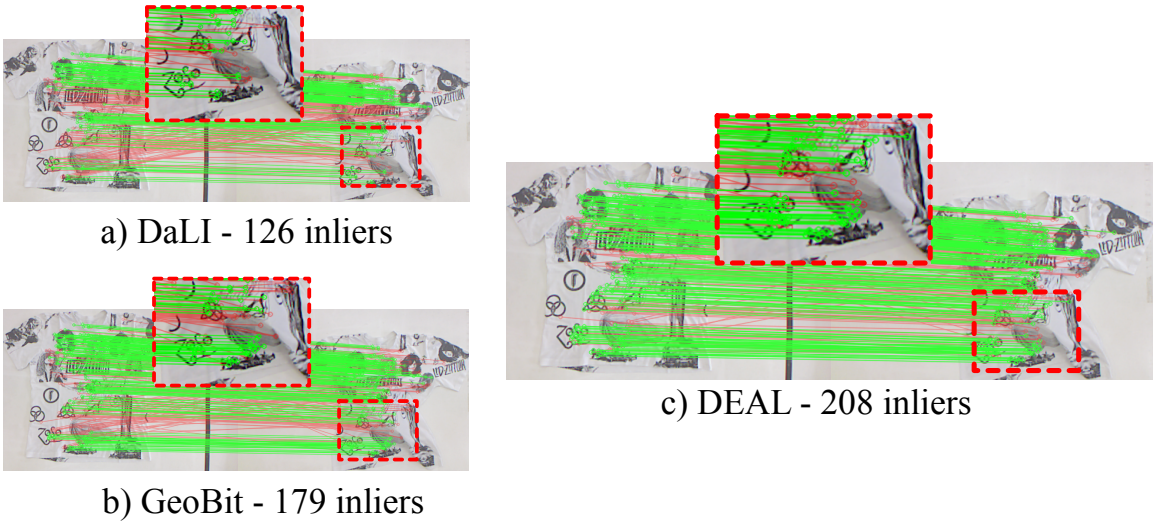


Figure 5.2: **Matching result of deformed shirt.** Correct matches are drawn as green, and wrong ones as red lines. Our descriptor better handles the strong deformation (highlighted in the zoomed boxes), among the patch-based deformation-aware competitors while using the same set of detected keypoints.

perspective changes of local patch-based descriptors that work directly with low-level image information; and ii) the capability of dense methods in encoding local semantics from the image to disambiguate hard pairs. Additionally, our model decouples the problem of learning distinctive descriptors robust to deformations in two complementary tasks, *i.e.*, first estimating a warp with a spatial attention mechanism and then describing the sampled local patches using the estimated non-rigid warp. Figure 5.1 outlines a schematic representation of the model, and Figure 5.2 shows a qualitative example of improvement in keypoint matching using the proposed approach.

5.1.1.1 Mid-level feature extraction

For the extraction of mid-level image features that encode local deformation information, we employ the ResNet-34 [47] and truncate it at the sixth convolutional block. The motivations to process features from ResNet instead of directly sampling local patches over the image are twofold: First, the mid-level feature maps provided by ResNet features allow the network to learn to encode local deformations from contextual cues such as local texture statistics, illumination and viewpoint changes; Secondly, the receptive field of our local feature descriptor is not tied to a fixed support region size and sampling pattern like traditional patch-based local descriptors.

Strong perspective and deformation changes drastically influence the optimal support region sizes for the same keypoint on different images; thus, we present an

architecture that is aware of these local deformations by design, since the dense feature map allows such information to flow to the non-rigid warper module. The ResNet-34 head outputs a feature map $\mathbf{X} \in \mathbb{R}^{H/8 \times W/8 \times 128}$ by performing convolutions until the spatial resolution achieves 1/8 of the original input resolution. The feature map $\mathbf{X}$ is used as input for the the non-rigid warper component, which computes the TPS parameters with the spatial transformer to model a non-rigid warp.

5.1.1.2 Non-rigid warper

The Non-Rigid Warper (NRW) is composed of two spatial transformers Ψ and Ω, in addition to the TPS parameter estimator $\mathcal{S}$. After obtaining the mid-level feature tensor of the entire image $\mathbf{X}$, for each keypoint, Ψ samples a local $5 \times 5 \times 128$ tensor centered at the coordinate of the keypoint, which are flattened and forwarded to $\mathcal{S}$, to estimate the local deformation parameters of the keypoint. Afterward, Ω initializes the polar transformation $\mathcal{G}_I$ with the keypoint attributes, and finally, patches are re-sampled from the original image and propagated to the descriptor extractor.

Feature sampler. We designed a Spatial Transformer Network Ψ to sample a local patch of mid-level features $\mathbf{Y}$ from the ResNet feature map $\mathbf{X}$ at the position of each keypoint in an end-to-end differentiable manner. For each keypoint, given a 2D affine transformation function $\mathcal{T}_{\mathbf{A}}$, where $\mathbf{A}$ is an affine matrix obtained from the downscaled keypoint position (to match the spatially downscaled spatial feature map $\mathbf{X}$ by a factor of 8), and the identity mesh grid $\mathcal{M}_I$, we generate transformed mesh grid coordinates as: $\mathcal{M}_{\mathbf{A}} = \mathcal{T}_{\mathbf{A}}(\mathcal{M}_I)$, as shown in Figure 5.1. Once the meshgrid is transformed, for each pixel's spatial coordinate $\mathbf{p}$, $\mathbf{p} \in \mathbb{R}^2$ in $\mathcal{M}_{\mathbf{A}}^{(i,j)}, \forall i \in [1, H], \forall j \in [1, W]$, we bilinearly interpolate values in $\mathbf{X}$ for each channel c to obtain the warped features $\mathbf{Y}$ *per keypoint*:

$$\mathbf{Y}_{\mathbf{p}}^{c} = \sum_{m=0}^{1} \sum_{n=0}^{1} \mathbf{X}_{\lfloor \mathbf{p}+(m,n) \rfloor}^{c} \mid 1 - m - \mathbf{p}'_{(1)} \mid \times \mid 1 - n - \mathbf{p}'_{(2)} \mid, \tag{5.1}$$

where $\mathbf{p}'$ is the decimal part of the pixel coordinate $\mathbf{p}$ [52] and $\mid x \mid$ is the absolute value of x.

The mesh grid $\mathcal{M}_{\mathbf{A}} \in \mathbb{R}^{5 \times 5 \times 2}$ encodes a local patch of 2D spatial coordinates centered at the keypoint. $\mathcal{M}_{\mathbf{A}}$ is used to interpolate the feature map $\mathbf{X}$ to sample the local mid-level features that encode local keypoint deformations. The local feature maps $\mathbf{Y}$ are flattened and forwarded to the localization network that regresses the parameters of the local deformations.

Localization network. We model the non-rigid object deformations with TPS [32]. A detailed explanation of TPS can be found in Section 2.3.3. The TPS formulation allows us to learn a set of parameters in a differentiable manner, meaning that we can train the network end-to-end. Recalling the TPS equation, the network estimates an affine matrix $\mathbf{A}$ and weight coefficients $\mathbf{w}_k$ given anchor points $\mathbf{c}_k$ established by the keypoint attributes, inducing a spatial geometric transformation $\mathbf{q} \rightarrow \mathbf{q}'$ of point $\mathbf{q}$ in image space according to Equation 2.10.

To estimate the coefficient weights of the TPS, we employ a regressor network $\mathcal{S}$, implemented as a Multi-Layer Perceptron (MLP) that takes as input the interpolated feature map $\mathbf{Y}$ of each keypoint to compute the TPS parameters θ_{tps}, that is composed by the affine parameters $\mathbf{A}$ and the weight coefficients $\mathbf{w}_k$. We use 64 control points in our implementation, which provides a good trade-off between non-rigid modeling capacity and computational cost according to our experiments. The estimated θ_{tps}, in conjunction with the identity polar grid $\mathcal{G}_I$ containing all control points $\mathbf{c}_k$ (initialized using the Polar Transformer) are used to sample a rectified image patch, as depicted in Figure 5.1.

Spatial transformer network sampler. Polar representations of a patch around keypoints increase the robustness of the descriptor extractor to image transformations, while also achieving rotation equivariance [36]. We thus adopt this representation in the sampled rectified patches. For that, a fixed regular polar transformation $\mathcal{G}_I$ is computed in the second Spatial Transformer Network Ω, parameterized by the keypoints' attributes. The attributes are obtained from a keypoint detector such as SIFT, and encode the keypoint's location (its coordinates in the image), its canonical orientation (used to achieve image in-plane rotation invariance), and its size (an estimate of the support region around the keypoint). We use those attributes to generate an identity polar transformation map $\mathcal{G}_I \in \mathbb{R}^{32\times32\times2}$ for each keypoint, encoding the point's initial position, rotation and size. The identity grid $\mathcal{G}_I$ is transformed to the warped grid $\mathcal{G}_\theta$ using the estimated θ_{tps}, which samples the final rectified patches by the second STN Ω. Finally, the patches are forwarded to the feature extraction network that computes a distinctive and compact descriptor for the rectified polar patch. We employed HardNet [87] as the backbone network in our implementation.

5.1.2 Local descriptor extraction and loss function

Our architecture can be trained end-to-end since all employed operations are differentiable, including the non-rigid warper component. HardNet $\mathcal{H}(.)$ takes as input

32×32 grayscale patches rectified by the non-rigid warper, and outputs a 128-dimensional L2 normalized feature vector for each keypoint. Then, the hardest in-batch triplet loss [87] is computed on a mini-batch of pairs of feature vectors, enforcing distinctiveness of the extracted descriptors. Let $\mathcal{F}_A \in \mathbb{R}^{N \times D}$ and $\mathcal{F}_B \in \mathbb{R}^{N \times D}$ be a matrix of N vertically stacked D-dimensional feature vectors of corresponding patches extracted by HardNet. Assuming that the descriptors are L2 normalized, we can calculate the Euclidean distances matrix $\mathbf{D}_{N \times N} = 2(1 - \sqrt{\mathcal{F}_A \mathcal{F}_B^T})$. The hardest negative example h_i for each row $\mathbf{D}'_i$, $\mathbf{D}' = \mathbf{D} + \alpha \mathbf{I}_{N \times N}$, where α is a constant ≥ 2 needed to supress the corresponding pairs from the diagonal of the distance matrix, is computed as $\delta_h^{(i)} = \min(\mathbf{D}'_i)$. The hardest in-batch margin ranking loss is then calculated as:

$$\mathcal{L}_H \left(\delta_+^{(.)}, \delta_h^{(.)} \right) = \frac{1}{N} \sum_{i=1}^{N} \max(0, \mu + \delta_+^{(i)} - \delta_h^{(i)}), \tag{5.2}$$

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

5.1.3 Implementation details

Training dataset. We employ a simulation physics engine to generate plausible non-rigid deformations (isometric transformations) of surfaces, developed in the previous chapters, and described in Section 4.4.2. The dataset is composed of 20,000 (960×720 resolution) image pairs with ground-truth correspondences. In order to generate realistic deformed images *in the simulation engine*, we performed texture mapping of the surfaces with real images extracted from large-scale Structure-from-Motion datasets [160]. We also added illumination variation such as intensity, global position, number of light sources, directional lighting, and color changes to enforce realistic non-linear illumination conditions in the simulation. The correspondences between frames are generated by first independently detecting up to 2,048 SIFT keypoints for each frame and then corresponding them using the simulation data under a threshold of 3 pixels. Each image pair contains, on average, approximately $1K$ ground-truth correspondences, summing up to a total of about $20M$ keypoint correspondences. Independently detecting the keypoints in the frames is a key step to improve generalization, since it considers repeatability properties of keypoint detectors.

Network and training setup. We implement [1] our network using PyTorch [102] and optimize it via Adam with initial learning rate of $5e^{-5}$, scaling it by 0.9 every $3,800$ steps. The network is trained end-to-end, setting the TPS regressor's weights of the last layer to zero, resulting in an identity warp at the beginning of training. We used a batch size of 8 image pairs containing up to 128 keypoint correspondences for each pair in our setup. The keypoint correspondences are randomly sampled from a uniform distribution with fixed seed during training, and we train the network for 10 epochs. The model is trained using a siamese scheme, where descriptors are extracted for the first and second set of corresponding keypoints (positive examples) using two networks with shared weights. The negative examples are calculated using a hard mining strategy in the batch as described by Mishchuk *et al.* [87] and used in the triplet loss. We also apply Average Pooling in the angle-axis of the polar patches at the end of HardNet feature maps to achieve rotation invariance. Our network implementation has $3.7M$ trainable parameters and takes about 5.5 hours to train on a GeForce GTX 1080 Ti GPU.

5.1.4 Ablation studies and processing time

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

Contribution of the TPS warper. We consider four different setups: (i) the use of the TPS on a standard regular grid patch, (ii) training the proposed architecture on a new dataset containing only rigid planar objects, (iii) using the second STN network Ω with fixed parameters at the identity (Fixed STN polar sampler – akin to Log-Polar); and (iv) with the NRW component for modeling the deformations. In this analysis, we use the scale simulation sequences, that were purposefully chosen because they exhibit mostly strong deformations across multiple image pairs. The Mean Matching Accuracy (MMA) achieved by each configuration can be seen in Table 5.1 (best in bold). All variations were trained until convergence with the same training procedure and HardNet initialization.

The ablation tests show that the polar sampler can improve matching accuracy compared to the Cartesian counterpart, and the non-rigid dataset also helps to improve accuracy.

[1] The training data and source code are available at www.verlab.dcc.ufmg.br/descriptors/neurips2021.

Most importantly, the NRW module alone can improve the accuracy of the descriptors by 4 p.p. when keypoints are affected by deformations, an important gain for tasks that require high matching accuracies such as deformable surface registration and non-rigid reconstruction.

Table 5.1: **Ablation of DEAL architecture.** Ablation study of the network design using the non-rigid datasets (Section 4.4).

Method	MMA ↑
Cartesian NRW	0.68
Planar Data NRW	0.75
Fixed Polar STN	0.75
Polar NRW	**0.79**

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

Sensitivity analysis. To evaluate the sensitivity and the influence of hyperparameters in our network, we performed the following experiments: (i) margin ranking parameter sensitivity analysis in the triplet loss; (ii) use of anchor swap in the triplet loss [151]; (iii) larger support region for the non-rigid warper (NRW) component; and (iv) use of dropout in the fully connected (FC) layers. Table 5.2 shows the MMA average achieved by testing different hyperparameters on the *Bag* sequence (Figure 4.13), which we used as a validation set and removed it from the benchmark experiments.

Table 5.2: **Effect of hyperparameters in DEAL.** The experiment is performed in the proposed non-rigid benchmark (Section 4.4).

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

The training steps and initialization are kept the same for all tested variations, and only one tested hyperparameter is changed while fixing all others for the sake of reproducibility and consistency. The baseline model uses the margin $\mu = 0.5$, no anchor swap, STN output of 3×3 and Dropout with probability $p = 0.1$ in the FC layers. We can observe that using the margin 0.75 and STN output of 5×5 increases the performance individually. We tested both changes simultaneously but it resulted in worse results than the individual changes. Thus, we update the final model (used in all experiments) to use STN output of 5×5 and keep other parameters from baseline unchanged, since they decrease the performance. We did not include larger STN outputs since we noticed marginal performance gains, while the model increases its computational requirements. The performance assessment in this test is done with the MMA @ 3 pixels metric as Revaud *et al.* [113].

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

5.2 Enhancing Deformable Local Features via Joint Keypoint Learning

Recently, several works [31, 113, 138] have consistently demonstrated that jointly learning keypoint detection and description yields significant improvements under challenging conditions, such as day-to-night matching [124] and long-term image matching across temporal scene changes and different seasons [67], indicating an entanglement of the detection and description tasks, since the keypoint detection can impact the performance of the descriptor. The descriptor for its turn can be used to determine reliable points optimized for specific goals. In contrast, handcrafted keypoints, typically used with patch-based learned local feature descriptors, often fail in challenging scenarios. This failure renders even the most robust patch-based descriptors ineffective due to the near-zero repeatability of the keypoints.

Keypoint learning remains an open research challenge. While a good keypoint detector should be repeatable across many image transformations [113], explicit supervision labels are lacking, aside from those generated by existing handcrafted keypoint detectors [118, 44, 78]. For this reason, keypoint learning is typically approached in a self-supervised manner using handcrafted loss functions [31, 113], which may introduce biases into the learning process, leading to sub-optimal performance across different dataset distributions.

In the context of non-rigid image matching, as previously discussed, all the deformation-aware methods to date neglect the keypoint detection phase, limiting their applicability in challenging deformations. The main shortcoming of our previously proposed local feature extractor DEAL is its dependency on existing keypoint detectors, which compromises the descriptor performance due to the lack of detection equivariance from most existing detectors in the presence of deformation changes. In contrast, we propose to learn detection and description in the same framework, achieving significant performance gains. We revisit joint keypoint detection learning and propose a new framework for

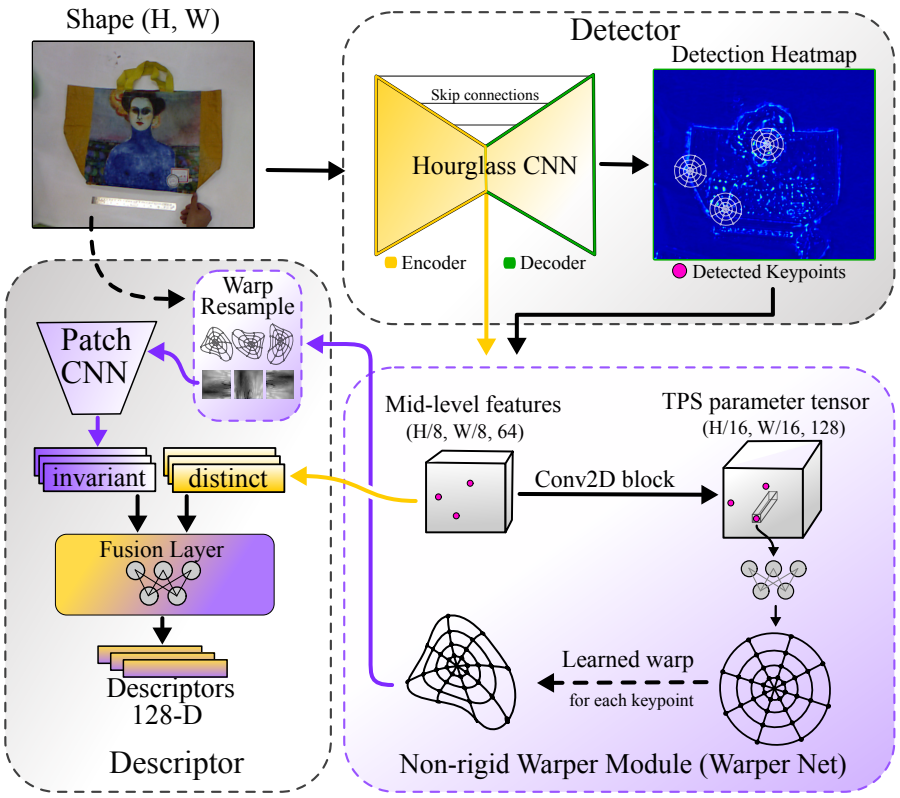


Figure 5.3: **DALF architecture**. Our architecture jointly optimizes non-rigid keypoint detection and description, and explicitly models local deformations for descriptor extraction during training. An hourglass CNN computes a dense heat map providing specialized keypoints that are used by the Warper Net to extract deformation-aware matches. A feature fusion layer balances the trade-off between invariance and distinctiveness in the final descriptors.

joint learning of keypoint detections and descriptors named DALF – Deformation-Aware Local Feature. DALF jointly learns to detect and describe points that are robust to non-rigid deformations, in addition to perspective and illumination changes, powered by the non-rigid warper module proposed in DEAL [108].

In DALF, both detector and descriptor are trained with a cooperative scheme aiming at the invariance of feature representations. Specifically, the keypoint detector is trained using policy gradient, seeking to increase the probability of detections that are both repeatable and reliable; Concurrently, the descriptor extractor learns to undeform and extract discriminative and invariant features from local regions. The model is only trained on synthetic warps, *i.e.*, it does not require expensive human annotation nor pseudo-ground-truth that may contain errors and bias, such as the output of an SfM pipeline that is used in several works [147, 122, 35, 113, 80]. Figure 5.3 outlines the proposed method.

5.2.1 Keypoint detection in deforming images

Since sampling keypoints in the image is inherently a discrete process, during training, we leverage the framework of reinforcement learning [147] to keep the computational graph differentiable. More specifically, we employ policy gradient [136] to optimize the expected reward for a distribution probability over the computed heat map. Finally, the estimated TPS parameters are used to warp a local patch according to the learned parameters. The warped patch is consumed by a CNN designed to extract a compact feature vector. The architecture is optimized by employing the hard triplet loss [87] for the descriptor branch, which enforces a distinctive representation according to local texture context, and the policy gradient method is used to optimize the detection branch.

The keypoint detection architecture uses a backbone hourglass CNN network $\mathbf{f}(\cdot)$, similar to a U-net [117]. This network enables computing a keypoint heat map in the original image resolution efficiently, while also producing mid-level feature representations that are useful to describe the keypoints. We employ three downsampling blocks for the encoder, and three upsampling blocks for the decoder, with skip connections, each having two convolutional layers composed of a 2D convolution followed by ReLU and batch normalization. Let $\mathcal{I} \in \mathbb{R}^{H \times W \times C}$ be the input image of size $H \times W$ and C channels, $\mathbf{f}(\mathcal{I})$ outputs two feature maps: mid-level representations $\mathbf{X} \in \mathbb{R}^{H/8 \times W/8 \times D}$ and detection heatmap $\mathbf{M} \in \mathbb{R}^{H \times W}$, where D is the local feature dimensionality.

An effective detector must output heatmaps $\mathbf{M} \in \mathbb{R}^{H \times W}$ with high responses in regions that can be matched well in non-rigid scenes containing view and illumination changes. Thus, during the training of the detection branch, we optimize $\mathbf{M}$ using a strategy similar to DISK [147], but applying only the probabilistic framework to learn the detection heatmap. A key difference compared to DISK is that we enforce the reliability of the detected keypoints by penalizing wrong matches even if the keypoints are repeatable. A probabilistic approach has several advantages as dealing with the inhering discreteness of sparse keypoint detection, and a simpler and more intuitive loss can be used for better convergence and regularization of the detection heatmap, in contrast to works that require elaborated handcrafted losses [113, 35, 80].

We seek to obtain high responses in confident regions not only for detection but also matching. To solve this problem with policy gradient, we divide the heatmap into a 2D grid of cells (Detection Heatmap in Figure 5.4). Analogously to the traditional use of the reinforcement learning technique in many artificial intelligence problems, our *agent* network considers a set of actions it can make to select keypoints. Each cell $\mathbf{c}_i$ has $m \times n$ pixels, where the network can learn the probability of detecting a keypoint within each cell. Given an image pair (A, B) of the same scene under different photometric and geometric transformations, and the ground-truth flow-field relating the two images, for each cell

$\mathbf{c}_i \in \mathbf{M}$, we consider a probability distribution over the cell $\mathbf{c} \in \mathbb{R}^{m \times n}$. Each logit value within the cell has a probability of being a keypoint. The probability mass function $\mathbf{p}_{\mathbf{c}_i}$ over the cell $\mathbf{c}_i$ is computed by applying the Softmax function.

Therefore, to train the detection branch, we employ the Reinforce algorithm [136]. During the forward pass of the network, we randomly sample an individual keypoint within each cell $\mathbf{c}_i$ according to the probability mass function $\mathbf{p}_{\mathbf{c}_i}$ alongside the keypoint's spatial coordinates, its probability p_s^i and its logit l_s^i. Note that each cell can have exactly one keypoint; however, in practice, it is common that low texture and ambiguous regions result in low-quality keypoints that cannot be reliably matched or detected in other images. For that reason, we accept a keypoint proposal from a cell with probability $\sigma(l_s^i)$, where σ is the sigmoid activation. This way, the network can learn to filter out unreliable keypoint proposals during training. The final probabilities of detection for the image I is given by the set $P_I = \{\sigma(l_s^i) \cdot p_s^i\}, \forall \mathbf{c}_i \in \mathbf{M}$, such that $\sigma(l_s^i) > 0.5$ (we only sampled keypoints that has positive values in the heatmap). Since we want the keypoints to be repeatable, we reward points that can be detected in both images A and B. Thus, given the pixel coordinate $\mathbf{p}_A^j \in \mathbb{R}^2$ of detected point j on image A, we define the reward function $\mathcal{R}(.)$ as follows:

$$\mathcal{R}(\mathbf{p}_A^j) = \begin{cases} 1 & \text{if } \exists \mathbf{p}_B^{(.)} \text{ s.t. } \|T(\mathbf{p}_A^j) - \mathbf{p}_B^{(.)}\| < \tau, \\ 0 & \text{otherwise,} \end{cases} \tag{5.3}$$

where $T(.)$ transforms pixels coordinates of image A to image B according to the ground-truth flow-field, and τ is a pixel threshold to determine if the detected keypoint in A has

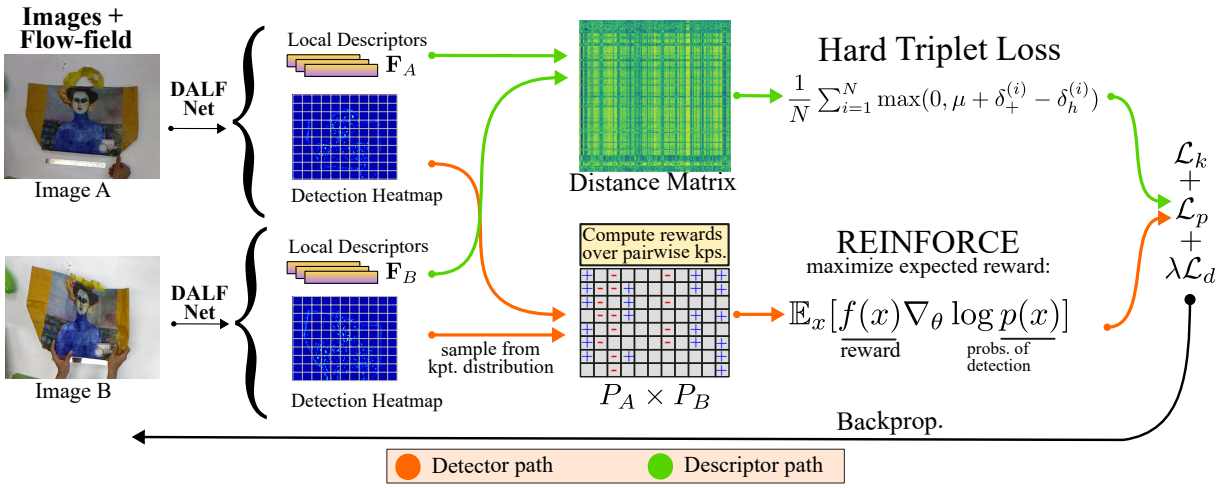


Figure 5.4: **Training strategy to learn to detect and describe keypoints aware of deformations.** DALF network is used to produce a detection heatmap and a set of local features for each image. In the detector path, the heatmaps are optimized via the REINFORCE algorithm considering keypoint repeatability under deformations. In the descriptor path, feature space is learned via the hard triplet loss. A siamese setup using image pairs is employed to optimize the network. Notice that we penalize keypoints that cannot be described accurately during the second training stage; thus, the keypoints and combined descriptors are optimized jointly to be robust to deformations.

a correspondence in image B. While very close keypoints can occur in neighboring cells, non-maximal suppression is used during inference to eliminate nearly repeated keypoints.

Once we have the set of probabilities P_A and P_B, we obtain the gradient of the parameter vector θ with respect to the expected rewards over all pairwise keypoints $\mathcal{K} = P_A \times P_B$, where $\times$ denotes the cartesian product (see Figure 5.4). The gradient ascent is used to maximize the expected sum of rewards:

$$\nabla_\theta \mathbb{E}_\mathcal{K}[\mathcal{R}(.)] = \sum_{(x,y)\in\mathcal{K}} \nabla_\theta(\log p(x;\theta) + \log p(y;\theta))\mathcal{R}(.), \tag{5.4}$$

where $p(.;\theta)$ denotes the probability of taking that action according to the network parametrized by θ. The variables x and y are the probabilities of detection from a pairwise combination of keypoints.

During the invariant feature learning stage, after 70% of the training progress, we zero out the reward of the keypoints if their descriptors are unreliable. Details about the penalization term for the keypoints are described in Section 5.2.5.

5.2.2 Keypoint descriptor

We observed that mid-level features extracted from the hourglass encoder do not explicitly model invariance to any kind of deformations, but tend to be highly distinctive on small to moderate photometric and geometric changes, such as varying illumination, and planar warps. Therefore, it is advantageous to supervise the mid-level features since we obtain informative descriptors with no additional cost at inference. For that, during the first training stage, we bilinearly interpolate the feature maps $\mathbf{X}$ at the detected keypoint positions to obtain a feature vector $\mathbf{f}_d \in \mathbb{R}^D$ for each keypoint coordinate. Let $\mathcal{F}_A \in \mathbb{R}^{N\times D}$ and $\mathcal{F}_B \in \mathbb{R}^{N\times D}$ be matrices of N L2-normalized feature vectors of corresponding descriptors $\mathbf{f}_d$ extracted by the hourglass decoder at keypoint positions, from images A and B, and $\mathbf{D}_{N\times N} = \sqrt{2 - 2\mathcal{F}_A\mathcal{F}_B^T}$ the distance matrix. To optimize the descriptors' embedding space, we employ the hard mining strategy [87] in the matrix $\mathbf{D}$ and minimize the hardest in-batch margin ranking loss defined in Equation 5.2 similar to our previous method DEAL.

5.2.3 Non-rigid warper module

CNNs' translation equivariance property makes local descriptors invariant to image translation, and multi-scale strategies increase the robustness of description extraction to in-plane scale changes [113, 147, 35]. However, when non-rigid deformations arise, the local texture can significantly change in appearance, introducing matching ambiguities. We demonstrated in DEAL [108] that TPS, coupled with STNs, can be used to model deformations for the task of local feature description. Here, we adopt the non-rigid warper module from DEAL to learn local invariance to non-rigid transformations affecting the patches with modifications for providing improved efficiency when estimating the parameters of the TPS transformations. Specifically, DEAL uses a STN to crop the feature map for sparse keypoints, which causes additional overhead for a large number of keypoint detections, while in DALF, the TPS parameter tensor is estimated densely for the entire image all at once, improving efficiency.

Spatial transformer network. We use the mid-level features from the backbone network to learn the parameters of the TPS with little additional overhead. The TPS parameter tensor $\mathbf{M}_\theta \in \mathbb{R}^{h/16 \times w/16 \times 2d}$ contains an intermediate representation useful to estimate a local non-rigid transformation for a keypoint. To obtain the parameter vector used in the TPS equation (defined in Equation 2.10), first, we bilinearly interpolate a feature vector from $\mathbf{M}_\theta$ at the spatial position of the keypoint, obtaining an intermediate parameter vector $\in \mathbb{R}^{2d}$. Then, an MLP is used to estimate the parameter vector μ_θ that is used in the TPS transformation. The parameter vector μ_θ encodes the affine matrix $\mathbf{A} \in \mathbb{R}^{2\times3}$, and the non-rigid components $\mathbf{w}_k \in \mathbb{R}^2$ separately representing offsets from the affine component. Given a homogeneous 2D point $\mathbf{q} \in \mathbb{R}^3$, weight coefficients and control points $\mathbf{c}_k$, $\in \mathbb{R}^2$, we use the parameters contained in μ_θ to apply the TPS transformation to a fixed polar grid centered at the keypoint, where n_c is the number of control points, $\mathbf{q}$ is a normalized spatial image coordinate from the fixed polar grid around the keypoint, and $\mathbf{p}$ is its transformed coordinate. Figure 5.3 (Warper Net) shows the patch warping and sampling step. Since we are using the TPS Radial Basis Function, $\rho = r^2 \log r$ is used. After the polar grid is transformed, a differentiable bilinear sampler [52] is used to obtain the transformed image patch that is used by a CNN similar to L2-Net architecture [140] to compute the invariant feature vector, supervised by the margin ranking loss (Equation 5.2). In our implementation, a major difference from the original L2-Net is that in the last convolutional block, we add an average pooling in the axis respective to the angular axis in the polar patch, attaining full rotation invariance.

5.2.4 Feature fusion layer

Distinctiveness and invariance are two desired attributes of a local feature descriptor. While invariance is vital for tasks that handle large appearance changes, such as rotation and scale, it usually implies distinctiveness loss [149]. By considering two complementary features, the distinctive ones coming from the backbone network with a larger receptive field but more sensitive to strong geometric transformations, and invariant features coming from the warper module that are robust to deformation and rotation by design, we propose to incorporate both information by a feature fusion step.

The fusion is performed by an attention-based MLP that predicts weight coefficients. The two descriptor vectors are first concatenated and forwarded to the Fusion Layer as depicted in Figure 5.3. Then, the concatenated descriptors are weighted by the predicted attention weights and L2-normalized to produce the final descriptor. During training, we optimize each descriptor loss individually and the loss of the fused descriptors simultaneously to enforce the network to learn how to fuse the feature vectors to achieve a better feature representation. In the experiments, we demonstrate that combining the features allows the final descriptor to handle strong image transformations while maintaining its distinctiveness.

5.2.5 Training strategy and model optimization

Stage-wise training. During experiments, we observed that training the network end-to-end in a single phase causes the model to focus on the invariant features and ignores the distinctive features coming from the backbone, even when re-weighting the loss terms. To solve the issue, we perform a two-stage training. During the first training stage, we only train the backbone network. The backbone features have a larger receptive field and higher-level semantics compared to the Warper Net features but with less invariance to rotation and low-level deformations. In the second training phase, the Decoder, Warper Net and Fusion Layer are optimized, where the final feature representation is optimized considering both representations through the fusion step. Moreover, the decoder of the network is further refined and encouraged to detect keypoints that are optimal for the fused descriptors.

Final loss. For the detection branch, we define the keypoint loss as $\mathcal{L}_k = -\mathbb{E}_{\mathcal{K}}[\mathcal{R}(.)]$ and add a regularization term for all the detected keypoints during training $\mathcal{L}_p = -\sum_x \log p(x)\cdot$

c, where c is a small negative constant to discourage the network from detecting low-quality points. We employ the margin ranking loss described in Section 5.2.2 for all the descriptor vectors computed by our network. The final loss is then computed as $\mathcal{L} = \mathcal{L}_k + \mathcal{L}_p + \lambda\mathcal{L}_d$, where λ is a weight term to balance the magnitude between the triplet loss and the policy-gradient losses.

5.2.6 Implementation details

We employed a synthetic data generation pipeline to create plausible non-rigid deformations of surfaces to supervise the training. In contrast to GeoPatch and DEAL that use a physics-based simulation in 3D, we only perform augmentation in image-space, significantly improving efficiency and simplicity of the data generation pipeline. Photometric and geometric transformations are applied to natural images obtained from large-scale Structure-from-Motion datasets [160]. We use only the raw images and do not use any other information. During training, we add random photometric changes, random perspective projection, and random TPS warps to obtain dense flows between image pairs depicting the same surface. Training starts with easier samples and progressively becomes more difficult, achieving the hardest difficulty at 60% of training iterations.

In experiments, we use the following hyperparameter values: $\mu = 0.5$ in the triplet loss as employed by DEAL [108]; pixel threshold $\tau = 1.5$; $\lambda = 0.005$ to balance the loss terms and keypoint penalization of $c = -7e^{-5}$ (values chosen by running the experiments on a smaller dataset); cell size $m = n = 8$ pixels as used in DISK [147] implementation; and the number of control points $n_c = 64$ as used in DEAL. As detailed in Section 5.2.5, we perform a two-stage training. Gradient accumulation was used for four forward passes before updating the weights to reduce gradient variance. We trained the network for 80,000 iterations in the first stage and 100,000 iterations in the second stage. Our network has about $1M$ trainable parameters and it takes about 48 hours to be fully trained on a single GeForce GTX Titan X GPU. We first trained the network encoder, then we froze the encoder and optimized the decoder and non-rigid warper module simultaneously. During inference, we use a non-maximal suppression of size 3×3 pixels in order to extract the keypoint coordinates from $\mathbf{M}$. Our network is implemented on PyTorch, has about $1M$ trainable parameters, and takes 48 hours to train on a GeForce GTX Titan X GPU. The source code for both training and inference is made publicly available [2].

[2]Training and inference code available at www.verlab.dcc.ufmg.br/descriptors/dalf_cvpr23/.

5.2.7 Ablation studies and processing time

Our ablation study comprises five different configurations of our method: (i) using the U-net backbone only without the non-rigid warper module, which is similar to DISK excepting the descriptor loss term; (ii) computing the descriptors using the non-rigid warper module only; (iii) fusing the invariant and distinct features from the non-rigid warper and backbone respectively; (iv) perform a stage-wise training where the backbone is optimized first and the non-rigid warper second, and finally (v) we perform stage-wise training with an additional attention layer to fuse the invariant and distinctive descriptors instead of simple concatenation.

Table 5.3: **Ablation study for DALF.** Performance of our method when considering different network components and training strategies.

Distinct	Invariant	2-Stage	Attn.	↑ MS / MMA
✓		-	-	0.48 / 0.72
	✓	-	-	0.51 / 0.78
✓	✓			0.53 / 0.79
✓	✓	✓		0.53 / 0.78
✓	✓	✓	✓	0.53 / 0.80

From Table 5.3, we can observe that the non-rigid warper contributes significantly to achieving more accurate matches when compared to using a convolutional backbone alone. Furthermore, by fusing the features, it is possible to obtain an improved descriptor that is both invariant and distinct with complementary properties.

The two-stage training provides similar matching scores and slightly reduced mean accuracy compared to end-to-end training. The invariant part tends to dominate the distinctive part during training, rendering the distinct part less useful in practice, which is not desired for applications needing more distinct features, such as image retrieval, and datasets without significant deformations. Finally, we test if an attention-based fusion layer can deliver better results than concatenating the descriptors in the fusion step. According to the results, it is possible to slightly increase the accuracy even further with a negligible cost in computation. Thus, we choose the model with the stage-wise training as the final architecture.

Table 5.4: **Stage-wise training impact on DALF.** Matching score @ 3 pixels for each configuration C following the order of Table 5.3, *e.g.*, C1 corresponds to distinct-only, C2 to invariant-only, etc. Best result in red, second best in green, third best in blue.

Dataset	C1	C2	C3	C4	C5
Kinect1	0.58	0.52	0.53	0.55	0.54
Kinect2	0.54	0.56	0.60	0.63	0.62
DeSurT	0.48	0.43	0.46	0.50	0.49
Simulation	0.27	0.52	0.50	0.34	0.42

Although the two-stage training is not mandatory in our learning pipeline, it offers a

better trade-off between invariance and distinctiveness, as shown in the top 3 performances on every dataset according to Table 5.4 (which presents the scores per dataset from Tab. 2 of the main paper), thus we opt for C5 as the final design choice. Note that the fusion of the invariant and distinct features (C3–5), one of our novel contributions, achieves much better rankings on average across all datasets.

Time efficiency. DALF is one of the most time efficient methods among the joint detection and description architectures. While our method runs at 9 FPS, DISK runs at 5 FPS and R2D2 at 2 FPS in an NVIDIA GeForce RTX 3060 GPU to extract 2,048 keypoints from 1024×768 images.

5.3 Quantitative Analyses

In this section, we evaluate our proposed RGB-only deformation-aware local feature methods alongside a wide range of descriptors from the literature on various publicly available datasets featuring deformable objects under diverse viewing conditions, including changes in illumination, viewpoint, and deformation. For this purpose, we have adopted our proposed dataset [107] described in Section 4.4 as well as the DeSurT [157] dataset. These datasets consist of color and depth images of 23 deforming real-world objects, where keypoints are independently detected, and ground-truth correspondences are established according to the image matching benchmark [54] protocol, which was adopted in several Image Matching Challenges[3].

5.3.1 Baselines and evaluation metrics

We compare our method with several patch-based descriptors [120, 1, 142, 91, 151, 36, 141], using the same set of SIFT [78] keypoints following the protocol of the image matching benchmark [54]. We also perform tests with a detector suitable for non-rigid correspondence [83] coupled with the deformation-aware descriptor DEAL [108], namely, Non-rigid Keypoint Detector (NKD) in Table 5.5. Finally, we also include in the comparison the state-of-the-art detect-and-describe methods [35, 99, 166, 31, 113, 80, 147, 106]. For each evaluated method, we detect the top 2,048 keypoints and match the descriptors using

[3] https://www.cs.ubc.ca/research/image-matching-challenge/

nearest neighbor search. In addition to the standard comparison, we include as the gold standard for sparse keypoint matching, the SuperPoint [31] with the SuperGlue [122] learned matcher, which holds the state-of-the-art for stereo and multi-view camera registration [54] of rigid scenes. As shown in Table 5.5, the methods are divided into three categories: (i) methods that only require RGB input (*RGB*), preferred over those needing additional information such as depth; (ii) Detect & Describe (*D&D*) methods that provide both detection and description jointly within a single pipeline; and (iii) Deformation-Aware (*D-A*) methods, which take into account deformation when computing the descriptors. Notice that a method may fulfill multiple categories simultaneously.

We used the Matching Scores (MS) [85] to evaluate the matching performance of both the detected keypoints and descriptors. Given a ground-truth transformation and a threshold in pixels, we compute the set of correct correspondences $\mathcal{S}_{gt}$ and obtain the score for an image pair (i, j) as $MS = |\mathcal{S}_{gt}|/\min(|keypoints_i|, |keypoints_j|)$. In addition, the mean matching accuracy (MMA) is also reported, which focuses on the accuracy of the descriptors to match the keypoints that were successfully detected on both images under the threshold denoted as the set $\mathcal{K}_{gt}$, and is computed as $MMA = |\mathcal{S}_{gt}|/|\mathcal{K}_{gt}|$.

5.3.2 Quantitative benchmarking on real images

Table 4.1 shows the MS and MMA scores achieved by all compared methods. Our proposed patch-based method DEAL displays the second-best results in MMA overall thanks to its deformation-aware module but has poor performance on the MS score, due to its reliance on SIFT keypoints, which are not designed for non-rigid transformations and may fail under moderate and strong non-rigid deformations, corroborating with our hypothesis that modelling keypoint invariance can significantly increase performance, as observed in DALF high scores. Nevertheless, among the patch-based methods, DEAL stands out using SIFT, obtaining the best metrics in both MS and MMA scores; This shows that the deformation-aware module is effective to cope with the deformations compared to other patch-based descriptors. To further validate the claim that keypoints should also be carefully modelled, we use the NKD [83] method proposed in a concurrent work from our team that focuses only on keypoint detection. We can observe that it drastically improve DEAL performance. However, NKD still produces inferior results compared to DALF, since it cooperatively learns keypoints suitable to the descriptor and vice-versa, while in NKD [83], the detection and description stages are still decoupled.

DALF outperformed all descriptors on average in both MS and MMA metrics, including the methods that use additional depth information to extract deformation-

Table 5.5: **Quantitative comparison with state-of-the-art local features**. The performance is assessed using top 2,048 keypoints [54]. The *RGB* methods only require color images. *D&D* methods perform joint detection and description. The deformation-aware methods are shown as *D-A*. The mark * indicates the use of a learned matcher. Best scores in bold and second-best underlined. The mean was calculated with full-precision values by averaging the scores of all 833 image pairs before rounding. The methods' names in bold font highlight the works proposed in this dissertation.

RGB	D&D	D-A	Method	**Datasets:** 833 **pairs total** – MS / MMA @ 3 pixels ↑				**Mean**
				Kinect 1 [107]	*Kinect 2* [107]	*DeSurT* [157]	*Sim.* [107]	
			BRAND [91]	0.17 / 0.34	0.22 / 0.49	0.14 / 0.33	0.04 / 0.09	0.16 / 0.34
✓			ORB [120]	0.19 / 0.38	0.25 / 0.55	0.18 / 0.40	0.14 / 0.30	0.20 / 0.43
✓			DAISY [142]	0.23 / 0.47	0.29 / 0.62	0.16 / 0.37	0.19 / 0.39	0.22 / 0.48
✓			FREAK [1]	0.24 / 0.49	0.33 / 0.72	0.16 / 0.38	0.15 / 0.31	0.23 / 0.51
✓			TFeat [151]	0.25 / 0.50	0.28 / 0.61	0.21 / 0.48	0.29 / 0.63	0.26 / 0.56
✓			Log-Polar [36]	0.28 / 0.58	0.30 / 0.65	0.23 / 0.54	0.22 / 0.49	0.26 / 0.57
✓			SOSNet [141]	0.17 / 0.34	0.25 / 0.55	0.17 / 0.38	0.26 / 0.57	0.22 / 0.47
✓	✓		LF-Net [99]	0.44 / 0.40	0.51 / 0.43	0.28 / 0.77	0.21 / 0.74	0.36 / 0.59
✓	✓		LIFT [166]	0.09 / 0.57	0.16 / 0.65	0.08 / 0.52	0.13 / 0.73	0.12 / 0.62
✓	✓		D2-Net [35]	0.20 / 0.50	0.23 / 0.82	0.14 / 0.47	0.11 / 0.30	0.17 / 0.57
✓	✓		SuperPoint [31]	0.45 / 0.74	0.54 / 0.85	0.39 / 0.68	0.18 / 0.34	0.41 / 0.69
✓	✓		R2D2 [113]	0.17 / 0.36	0.25 / 0.59	0.14 / 0.32	0.06 / 0.16	0.17 / 0.39
✓	✓		ASLFeat [80]	0.31 / 0.58	0.39 / 0.69	0.28 / 0.53	0.19 / 0.35	0.31 / 0.56
✓	✓		DISK [147]	0.53 / 0.76	0.52 / 0.81	0.44 / 0.61	0.26 / 0.34	0.45 / 0.66
✓	✓		SuperGlue* [122]	0.40 / 0.66	**0.62** / **0.99**	0.39 / 0.68	0.23 / 0.43	0.44 / 0.74
		✓	**GeoBit** [89]	0.31 / 0.65	0.35 / 0.77	0.20 / 0.47	0.32 / 0.71	0.30 / 0.66
		✓	**GeoPatch** [107]	0.32 / 0.66	0.35 / 0.80	0.26 / 0.60	0.39 / **0.86**	0.33 / 0.73
✓		✓	DaLI [129]	0.25 / 0.51	0.35 / 0.76	0.21 / 0.48	0.10 / 0.22	0.25 / 0.54
✓		✓	**DEAL**	0.33 / 0.68	0.38 / 0.85	0.27 / 0.63	0.36 / 0.80	0.34 / 0.75
✓		✓	NKD [83, 108]	0.44 / 0.74	0.49 / 0.82	0.33 / 0.64	0.31 / 0.74	0.40 / 0.74
✓	✓	✓	**DALF**	**0.54** / **0.82**	**0.62** / 0.90	**0.49** / **0.73**	**0.42** / 0.69	**0.53** / **0.80**

invariant features, improving the state-of-the-art in 8% p.p. in matching scores. Moreover, our method shows promising generalization properties to real deformations. DISK achieved the second-best results in MS, but the MMA indicates that its descriptors are more sensitive to non-rigid deformations.

We experimented with SuperGlue, a recent learned matcher, using both the indoor and outdoor pretrained weights, and the outdoor weights performed better than the indoor weights in all datasets. We thus report all SuperGlue results using the outdoor (best) weights. The results indicate that using SuperGlue with the pretrained weights offers marginal performance gains when matching deformable surfaces with SuperPoint, and a fine-tuning or even full re-training is required for improving results. We emphasize that our method can be easily coupled and trained with a learned matcher such as SuperGlue for matching deformable surfaces.

All the other methods achieve significantly worse scores due to their inability to cope with stronger deformations alongside illumination and affine transformations.

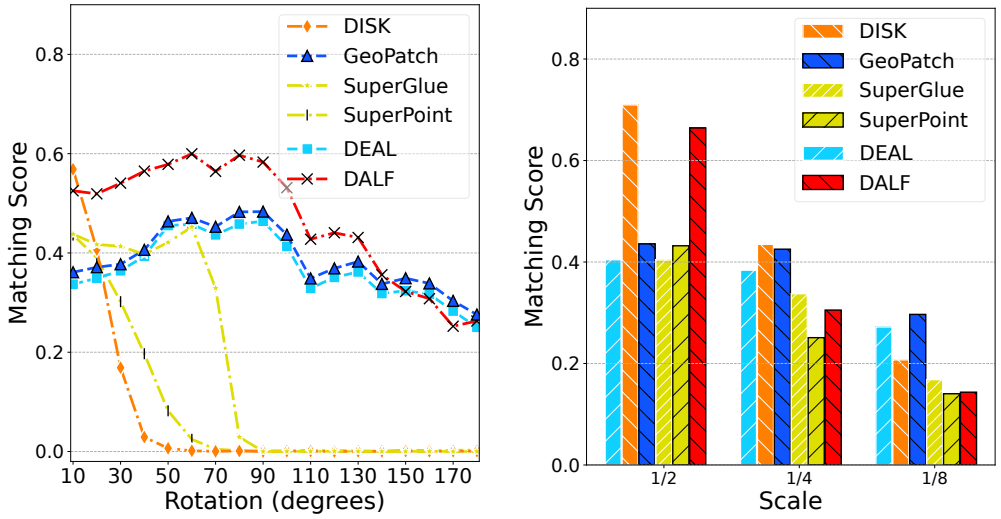


Figure 5.5: **Invariance to rotation & scale.** We evaluate the matching performance of the methods under rotation and scale changes between image pairs from the *Simulation* dataset. The objects are simultaneously deforming, rotating and scaling in image space.

5.3.3 Rotation and scale robustness

Aside from deformations, in-plane rotations and scale changes are two important geometric transformations that affects the descriptors' performance. Thus, we conduct a study using the Simulation sequences from [107] containing challenging rotation and scale changes in a controlled setting. Figure 5.5 clearly indicates that DALF holds the best performance in the presence of image in-plane rotations in addition to deformation changes compared to the five best competitors (Table 5.5). DALF also displays considerable robustness to scale changes, outperforming SuperPoint and providing a similar level of robustness of SuperGlue. GeoPatch (our proposed method in Section 4.3) display the smallest drop in performance, showing notable invariance to scale changes due to its geodesic sampling scheme that uses metric-depth to compute the sampling pixels, but requires the additional depth information. In contrast, our method DEAL (Section 5.1) which only uses RGB information, is on par with GeoPatch in the rotation sequences, and performs similarly in scale changes.

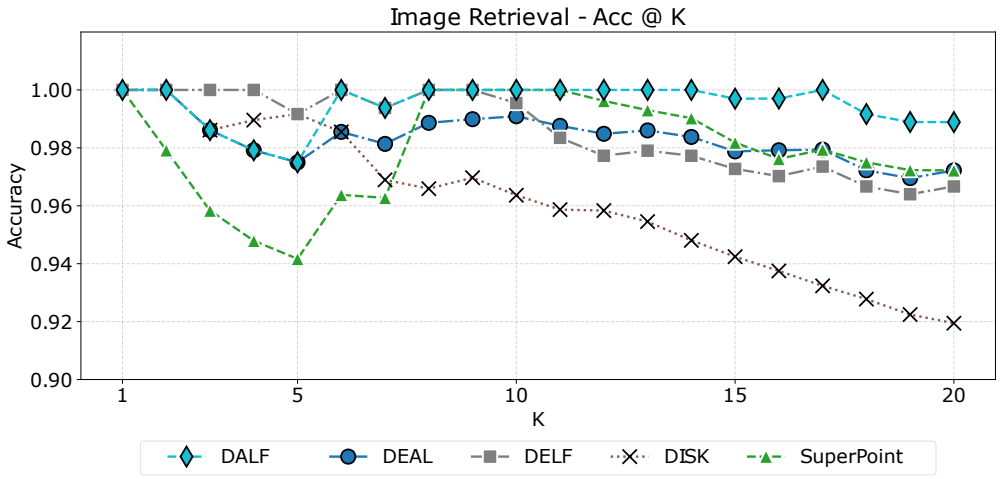


Figure 5.6: **Accuracy@K metric for the nonrigid object retrieval task.** The normalized area-under-the-curve for each method is the following DISK: 96.12%, SuperPoint: 97.92%, DEAL (ours): 98.34%. DELF: 98, 57%, and DALF (ours): 99, 49%.

5.4 Evaluation in real-world tasks

We apply state-of-the-art local features to two relevant real-world tasks: instance-level image retrieval of deformable objects and deformable registration of 3D surfaces. These complementary tasks were selected because their performance depends on the quality of image representations, particularly in terms of the descriptors' distinctiveness and invariance. We present both quantitative and qualitative metrics for these tasks, demonstrating the potential of our proposed approaches beyond low-level matching metrics, extending to practical applications.

5.4.1 Deformable object retrieval

The non-rigid datasets [93, 107] used for the retrieval task contains various sequences of different objects being deformed over time. We selected one frame for each sequence to serve as a query image. The other frames of all sequences compose the search database, from where the application must retrieve the results.

In the retrieval task, we assume a database that contains images from various deformed objects. Each object appears multiple times with different deformations. The top K images from the database corresponding to an image query are retrieved. To evaluate the methods, we use the retrieval accuracy for different K values. K-Nearest Neighbors is

used in conjunction with Bag-of-Visual-Words [26] approach over the descriptors as the retrieval engine. We compare our method against the state-of-the-art description methods that demonstrated the top performances in Table 5.5, in addition to DELF [139], a state-of-the-art descriptor designed and trained specifically for image retrieval. We calculated the normalized area under the curve of each method for $K = \{1, ..., 20\}$. DALF achieved the most accurate retrieval capabilities at 99.49%, while DELF, DEAL, SuperPoint, and DISK achieved 98.57%, 98.34%, 97.92%, and 96.12%, respectively. Figure 5.6 shows the qualitative results for the most important competing methods.

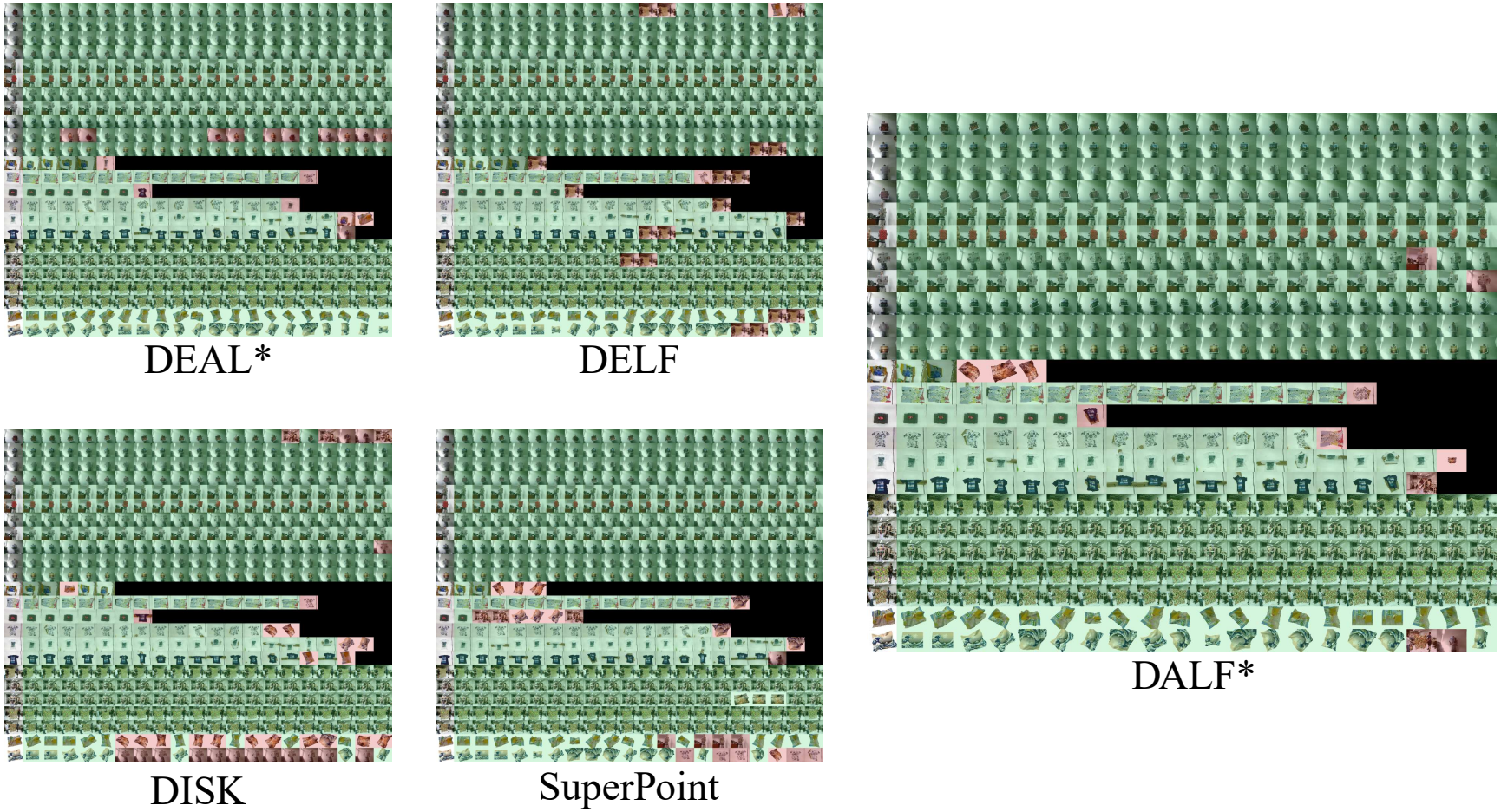


Figure 5.7: **Qualitative results of the non-rigid object retrieval task.** Our method has the best result in retrieving images of real and simulated deformed objects. The first column of each image shows the object queries, and the rows show the results from different queries. Green images correspond to the same object as the query, and red images do not correspond. Some objects are smaller and difficult to deform, so they may have less them 20 occurrences in the dataset. In that case, we lower the value of K to the exact number of occurrences of the object. For this reason, we can see some empty squares in the qualitative results. The black squares indicate no correspondent objects available. Our proposed methods are marked with *.

For each method, we detect and describe a maximum of 1,024 keypoints inside a mask delimiting the object pixels. Next, we sample an equal amount of descriptors for each image to collect about 10,000 descriptors. Then we use the sampled descriptors to compute 300 centroids using the K-Means algorithm. The centroids are then used to calculate one global representation for each image using the Bag-of-Visual-Words approach to aggregate all the described keypoints. Given a query, we use the global descriptor to retrieve the closest K images using K-Nearest Neighbors. We evaluate each method with the mean retrieval accuracy for each value of K from 1 to 20.

We compare our method against the best-performing description methods, in addition to DELF [139], a state-of-the-art descriptor designed and trained specifically for image retrieval. DALF achieved the best performance in the retrieval task, as shown in Figure 5.6. Note that at $K = 10$, all methods achieve similar scores because they can correctly retrieve the easy images. However, note that the task becomes hard when $K > 10$, where all methods but DALF degrade as they cannot reliably retrieve the images of the heavily deformed objects, while DALF exhibits superior performance. The full retrieval result for each query seen in Figure 5.7. The code for the retrieval task will be publicly available; its objective is to be an easy-to-run benchmark for detectors and descriptors.

5.4.2 Non-rigid 3D surface registration

In this section, we describe in detail the implementation of the surface registration application using the ARAP [133] optimization, and also show both quantitative and qualitative registration results. Non-rigid 3D surface registration aims to accurately align two RGB-D frames of the same surface, viewed from different viewpoints at the same time that the object is affected by non-rigid deformations. Figure 5.8 shows an overview of the registration pipeline. Surface alignment is a crucial step used by non-rigid reconstruction frameworks [51, 7] that allow complete 3D reconstruction of deforming objects. Improvements in registration accuracy can significantly increase the quality of the reconstruction, enabling the use of such systems in critical, challenging applications, such as the live reconstruction of human organs [81].

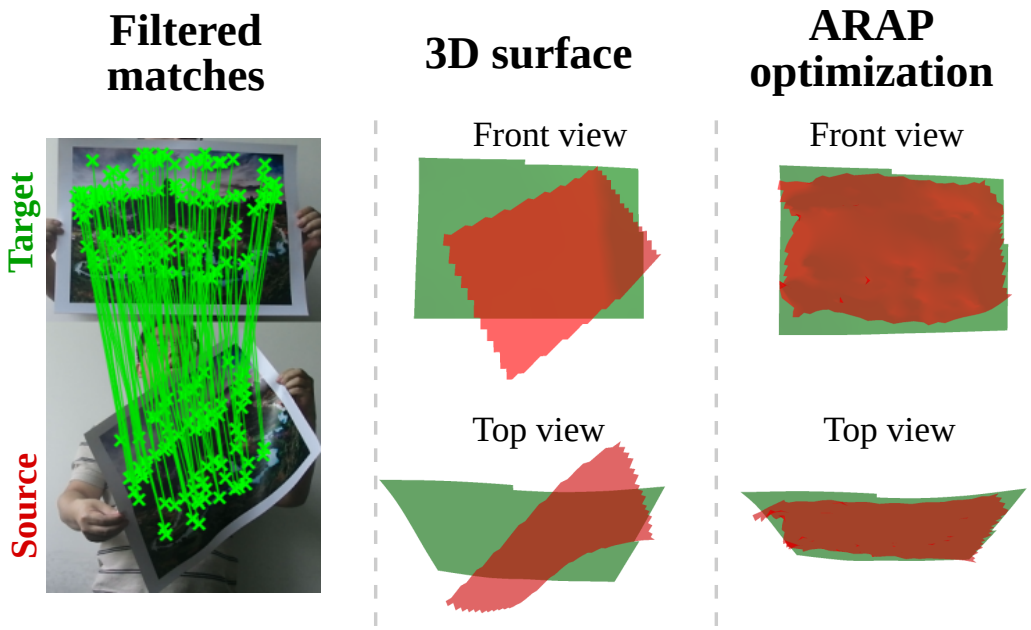


Figure 5.8: **Non-rigid 3D surface registration overview.** We use the filtered correspondences (left) to align two meshes of the same surface obtained from their respective RGB-D frames (middle) to the same reference pose and deformation (right), using as-rigid-as-possible (ARAP) refinement.

5.4.2.1 Implementation details

Our application considers the most difficult scenario: wide-baseline registration, where the object can be in an arbitrary viewpoint and deformed shape. Thus, it is challenging to filter outlier matches, in contrast with rigid registration, where it is possible to fit a homography or fundamental matrix using a minimal correspondence sample and perform RANSAC to remove the outlier correspondences with high confidence.

Our solution to this problem was to tune the AdaLAM [21] filtering method to perform outlier detection in the presence of image deformations. AdaLAM checks the affine consistency of local point clusters and filters the correspondences that are inconsistent with their neighboring matches. As we have observed empirically, the assumption of localized affine consistency is a reasonable approximation for non-rigid correspondences. We adjusted the sensitivity of the local affine RANSAC of AdaLAM to tolerate more deviation from the base affine transformation, which usually happens in the presence of significant deformations.

AdaLAM tends to provide erroneous consistent affine matches when the scene has repetitive patterns, which is inevitable in practice. Those inconsistent matches introduce large errors in the ARAP optimization, and the method fails to return a meaningful result. Thus, to improve the robustness of the registration, for all methods, we use the best 200 matches according to the Lowe's ratio test [78], which drastically reduces artifacts caused

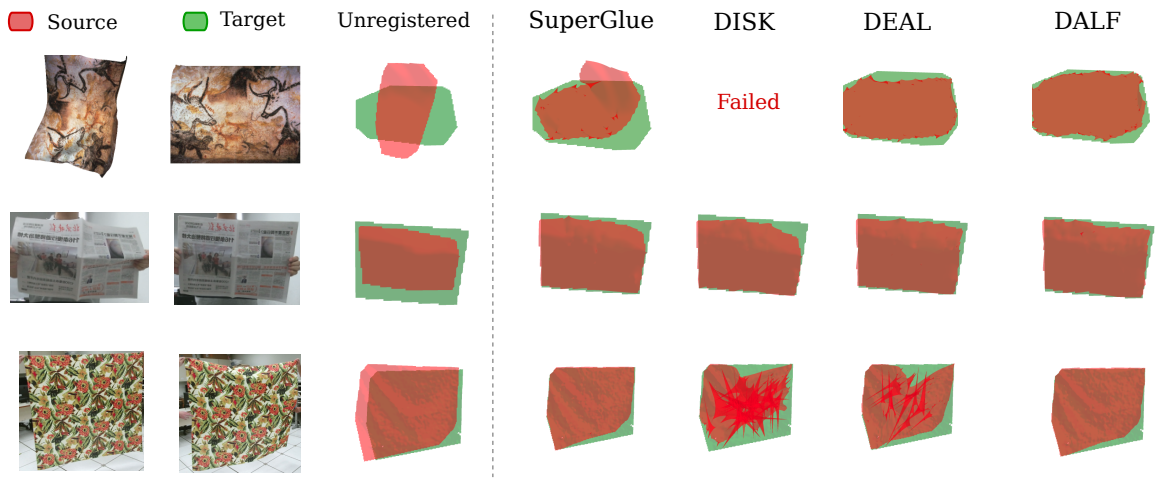


Figure 5.9: **Non-rigid registration under challenging scenarios**. Our method can achieve accurate non-rigid registration under large rotations, illumination changes caused by deformations, and highly repetitive patterns. In contrast, all other techniques produce low-quality results in at least one of the challenging scenarios. The sharp line artifacts in two registrations from DEAL and DISK indicates that the method produced inconsistent matches even after the filtering step, and the ARAP optimization failed due to local minima. Please check the supplementary video to visualize the registration results in 3D with the depicted image pairs and additional samples.

by repetitive patterns, and also accelerates the convergence of the ARAP optimization. The non-rigid registration application source-code will be released alongside the reference implementation of our proposed method.

5.4.2.2 Qualitative results

Figure 5.9 shows reconstruction results of challenging samples from the non-rigid datasets, including large rotations, sharp deformations and repetitive patterns. It is worth mentioning that SuperGlue [123], the best competing method, requires inputs in the form of image pairs, and employs global self and cross attention [152] across local features when matching them , *i.e.*, the matching problem is conditioned to the input image pair, which significantly improves its robustness, especially in ambiguous regions. In contrast, our method independently detects the local features without global awareness of the image, and a simple nearest neighbor search is used to perform matching. Still, our method achieves high-quality registration, as can be observed in all three challenging sequences. Our strategy can empower SuperGlue with deformation awareness by simply using our descriptors during training. In turn, SuperGlue's global self and cross-attention mechanisms could help our approach become much more robust to matching in challenging

Table 5.6: **3D surface registration metrics.** We select the top methods to conduct a detailed evaluation for the task of 3D surface registration. The 2D and 3D accuracy is computed under varying thresholds in centimeters for the 3D residuals and in pixels for the 2D residuals. Best in bold.

Method	2D Accuracy ↑			3D Accuracy ↑		
	@2px	@3px	@5px	@0.5cm	@1.0cm	@1.5cm
DISK	21.3	30.5	41.2	36.3	51.7	58.8
SuperPoint	23.2	34.4	47.4	42.6	60.5	68.4
SuperGlue	34.9	51.0	**68.1**	42.8	64.4	73.6
GeoPatch	28.9	41.5	55.9	41.0	62.3	71.1
DEAL	29.4	42.0	56.2	42.9	64.3	72.8
DALF	**36.6**	**51.6**	67.3	**46.2**	**66.9**	**74.8**

scenarios. Integrating DALF into a SuperGlue-like matcher is left for future work.

5.4.2.3 Quantitative results

To estimate the registration quality after the filtering and registration stage, the 2D error is computed using the ground-truth TPS transformation provided with the adopted datasets, given in pixels. We also estimate the residual 3D error in centimeters, assuming that the two corresponding surfaces must be perfectly adjusted in the 3D space, as their meshes are known beforehand. For each vertex of the source mesh, we query the nearest vertex in the target registered mesh and use the residual distance in centimeters as the error. Table 5.6 shows the performance of the top methods under different thresholds for the 2D and the 3D errors considering all the datasets used in Table 5.5, where our approach stands out, improving over 3 p.p. in 3D registration accuracy compared to the best current method (SuperGlue) in the tightest threshold of 0.5cm.

5.5 Discussion

In this chapter, we described how to enhance modern deep local feature extraction with explicit deformation awareness. First, we propose a learnable module that can be seamlessly integrated into modern deep architectures, such as CNNs, and does not rely on depth or other geometric information, but solely on a single RGB frame. Our

approaches explicitly model deformations via a TPS transformation for each keypoint in the image, enabling the network to reason about the local geometric structure of the surface based on learned priors. Second, our keypoint learning framework is trained end-to-end using a self-supervised guidance strategy that relies on synthetic warps. In addition to modeling deformations, we show that jointly learning keypoints adapted to deformations cooperatively with the descriptors significantly improves the robustness of local feature matching, both in terms of matching metrics and real-world applications.

From extensive experiments and two applications using real deformable objects, we draw the following key conclusions: (i) standard approaches for image matching that do not explicitly handle deformations deliver subpar results compared to deformation-aware features; (ii) optimizing the keypoint detection stage together with deformation-aware descriptors brings significant performance gains compared to existing deformation-aware methods that rely on affine keypoint detectors; and (iii) the balance between distinctiveness and invariance can be achieved effectively via a feature fusion component to increase the network expressiveness to deformations while keeping distinctiveness.

Compared to the *geodesic-aware* descriptors introduced in Chapter 4 of this dissertation, our new RGB-only methods provide a key advantage: depth is no longer a requirement to achieve improved invariance to non-rigid deformations, opening up the possibility of a much wider range of applications. It is worth mentioning that the two approaches are complementary. For instance, the geodesic-aware approaches are, by construction, inherently invariant to isometric deformations, and have a strong inductive bias in the patch sampling process that allows them to generalize to unseen domains. In contrast, the deformation-aware RGB methods are more versatile, but relies on learned priors of the trained dataset. We hypothesize that the worse performance of GeoPatch (geodesic-aware) compared to DEAL (deformation-aware) is due to the fact that the Kinect sensor noise hinders the full potential of GeoPatch. This is corroborated by our evaluation in the Simulation dataset (Table 5.5 – Sim.), where GeoPatch performs better than DEAL, because of the noise-free depth values.

We hope that our findings can bring more attention of the vision community to the under-explored problem of non-rigid correspondence, where several challenges in applications for non-rigid registration and 3D reconstruction of deforming surfaces remain unsolved.

5.5.1 Limitations

One of the primary limitations of both solutions presented in Chapter 5 is the simulation-to-real gap, a well-kown issue in deep learning approaches trained on simulated data [165]. Since we train both descriptor and non-rigid-warper module in synthetic image warps using a self-supervised learning strategy, it is expected that the synthetic deformations do not fully encompass the intricacies of real deformations. Nevertheless, we show in the experiments that the self-supervised learning strategy we adopted is effective enough to allow generalization to real deformations to the point we are able to surpass the competing approaches, including the geodesic-aware methods. An interesting research direction to improve the simulation-to-real gap is to use more realistic simulations, or weakly-supervised training schemes [116, 146], and also novel distillation strategies [165] to further increase the performance of our approaches in real deformations.

Another important limitation of our approaches its performance is compromised when dealing with surfaces having discontinuities, reflections, and poor texture. These conditions also hamper the competitors. However, since we depend on an additional learnable TPS module for deformation-awareness, errors may be accumulated not only on description but also in the deformation rectification step, which can result in worse performance. The non-rigid warper module may also introduce noise if the scene is not deforming at all, due to the distinctiveness-invariance trade-off principle [103]. This issue is partially addressed by the fusion module in DALF, that combines both distinct features from the main backbone with the deformation-aware features using a standard MLP.

Chapter 6

Accelerated Features

In the previous chapters, we have addressed the problem of invariance and distinctiveness of local features under challenging image transformations, including variations in illumination, viewpoint, and non-rigid deformations. We introduced geodesic-awareness and deformation-awareness in feature description, though computational efficiency was not the primary focus. Nevertheless, our methods maintained equivalent or superior computational performance compared to existing local feature extraction techniques. However, for tasks requiring real-time feature detection and extraction, such as in robot perception, autonomous driving, and embedded devices with limited compute (lightweight drones, Internet of Things, mobile robots, augmented reality glasses), existing deep learning solutions are often impractical due to their high computational demands, typically requiring expensive and power-hungry GPUs to run at near real-time speeds, which is highly undesired because image matching is a low-level task that is used by a myriad of more advanced and expensive solutions, such as VSLAM [18] and SfM [160, 126, 121].

Although lightweight handcrafted strategies such as ORB [120], FAST [118], and BRIEF [15] are capable of running efficiently on embedded systems, their robustness to adverse conditions, such as matching day-to-night images and handling significant appearance changes in the scene, is rather limited [121]. Furthermore, optimizing these solutions for specific problems is non-trivial, whereas developing a data-driven method that can adapt to the problem's specificities and fully exploit the scene properties is of utmost importance.

Another motivation for developing efficient deep learning solutions is the recent trend in the hardware industry towards highly efficient system-on-chips (SoCs) for deep learning deployment. Low-cost, energy-efficient SoCs, such as the Intel Myriad-X processor and the Nvidia Jetson family, are capable of performing terabyte operations with power consumption as low as 15W. Even when comparing highly efficient descriptors like ORB, modern architectures such as CNNs deployed on Neural SoCs may prove more efficient than running optimized algorithms on general-purpose processors like ARM or Intel.

We noticed that the current trend in the literature focuses on accuracy but often neglects compute efficiency, especially when deploying these solutions in the real-world. In this chapter, we introduce a lightweight and accurate architecture for resource-efficient

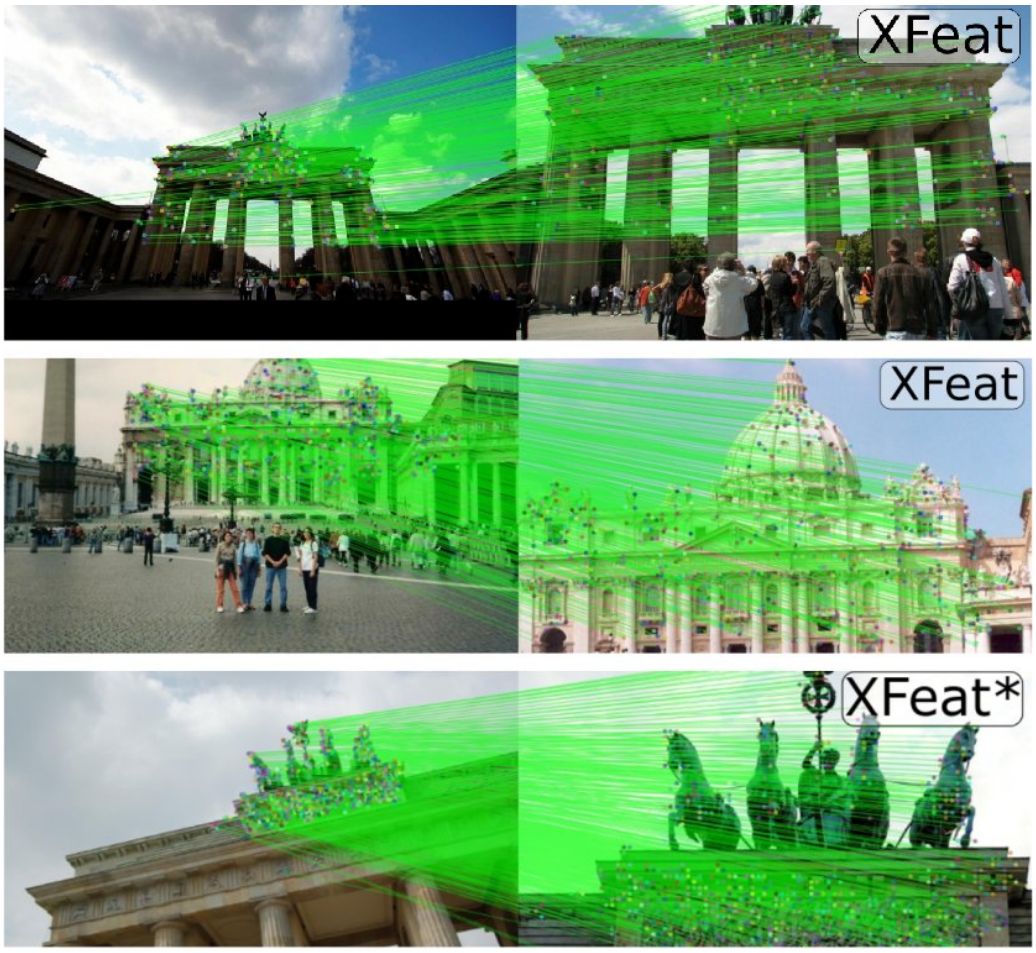


Figure 6.1: **Sparse (XFeat) and semi-dense (XFeat*) matching.** XFeat stands out with its dual ability to perform both sparse and semi-dense matching, providing fast feature extraction for a wide range of applications from visual localization with sparse matches to pose estimation and 3D reconstruction where denser correspondences deliver additional constraints and a more complete reconstruction.

visual correspondence. Our method, dubbed XFeat (Accelerated Features), revisits fundamental design choices in convolutional neural networks for detecting, extracting, and matching local features, satisfying a critical need for fast and robust algorithms suitable to resource-limited devices. In particular, accurate image matching requires sufficiently large image resolutions – for this reason, we keep the resolution as large as possible while limiting the number of channels in the network. Besides, our model is designed to offer the choice of matching at the sparse or semi-dense levels, each of which may be more suitable for different downstream applications, such as visual navigation and augmented reality. Our model is the first to offer semi-dense matching efficiently, leveraging a novel match refinement module that relies on coarse local descriptors. XFeat is versatile and hardware-independent, surpassing current deep learning-based local features in speed (up to 5x faster) with comparable accuracy, proven in pose estimation and visual localization. We showcase it running in real-time on an inexpensive laptop CPU without any specialized optimizations such as model quantization or compression.

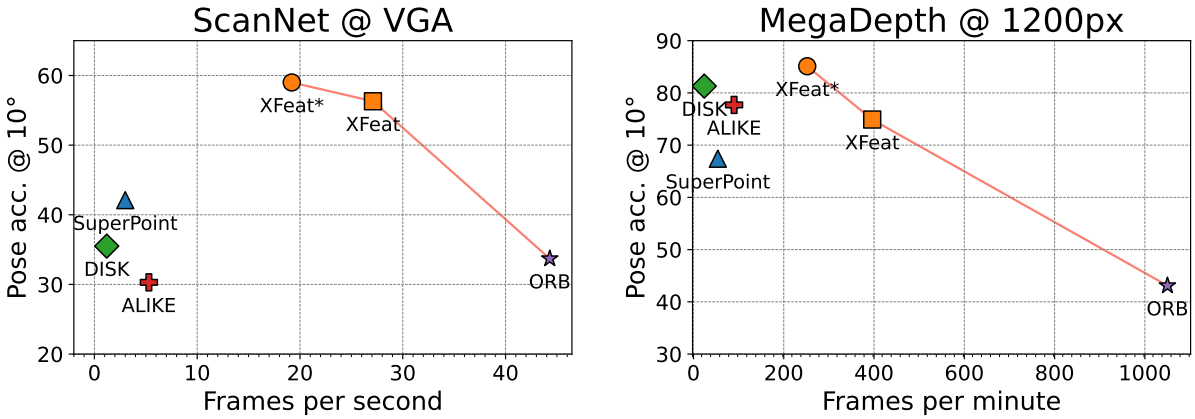


Figure 6.2: **In XFeat, accuracy meets efficiency.** XFeat delivers great trade-off between speed and relative pose estimation accuracy on both the Megadepth and ScanNet datasets, as evidenced by the Pareto-frontier curve in orange. Its lightweight architecture enables real-time feature extraction on GPU-free settings and resource-constrained devices without hardware-specific optimizations. Inference speed on a budget-friendly laptop (`Intel(R) i5-1135G7 @ 2.40GHz CPU`). * denotes semi-dense extraction.

XFeat is suitable to perform both sparse feature matching based on keypoints and dense matching of the coarse feature map, as shown in Figure 6.1. This versatility brings the best of both worlds: keypoint-based methods are more suitable to efficient visual localization based on Structure-from-Motion (SfM) maps [121], while dense feature matching can be more effective for relative camera pose estimation in poorly textured scenes [134, 23].

Compared with current methods available for image correspondence, our method significantly improves the trade-off ratio between matching accuracy and computational efficiency, outperforming all lightweight deep learning local feature alternatives by up to 5× in speed while being comparable to much larger models as SuperPoint [31] and DISK [147] in accuracy, as demonstrated in Figure 6.2. To mitigate computational costs while maintaining competitive accuracy, our proposed strategy for accelerated feature extraction brings three main contributions:

- A novel lightweight CNN architecture that can be deployed on resource-constrained platforms and downstream tasks that require high throughput or computational efficiency, without the requirement of time-consuming hardware-specific optimizations. Our method can readily replace existing lightweight handcrafted solutions [120], expensive deep models [147, 31] and lightweight deep models [176] in several downstream tasks such as visual localization and camera pose estimation;

- We design a minimalist, learnable keypoint detection branch that is fast and suitable for small extractor backbones, showing its effectiveness in visual localization, camera pose estimation, and homography registration;

- Lastly, a novel match refinement module for obtaining pixel-level offsets from coarse semi-dense matches is proposed. Our new strategy does not require high resolution features besides the local descriptors themselves as opposed to existing techniques [134, 23], greatly reducing compute and achieving high accuracy and matching density, shown in Figure 6.1, and with little additional computations with respect to traditional nearest neighbor search due to the small descriptor size.

6.1 Accelerated Local Feature Extraction

Local feature extraction accuracy heavily depends on input image resolution. For instance, in camera pose, visual localization, and SfM tasks, the correspondences should be fine-grained enough to allow pixel-level matches. However, feeding high-resolution images into network backbones increases computational requirements to undesired levels even for simple, small network backbones such as SuperPoint VGG-like architecture [131, 31]. In this section, we describe how to reduce significantly the computational cost using strategies to minimize the computational budget while mitigating robustness loss due to a considerably smaller CNN backbone.

6.1.1 Featherweight Network Backbone

Let $\mathcal{I} \in \mathbb{R}^{H \times W \times 1}$ be a gray-scale image, where H is the height and W the width in pixels. To decrease a CNN processing cost, a common approach is to start with shallow convolutions and then incrementally halve spatial dimensions (H_i, W_i) while doubling the channel count C_i in the i-th convolutional block [131]. Assuming a convolutional layer with unit stride, padding, no bias term and square kernel size $k \times k$, the cost of convolution in terms of floating point operations (F_{ops}) for the i-th layer can be expressed as:

$$F_{ops} = H_i \cdot W_i \cdot C_i \cdot C_{i+1} \cdot k^2. \tag{6.1}$$

Naively pruning channels C across the entire network compromises its capability of handling challenges like varying illumination and viewpoint as demonstrated in the ablation experiments (Section 6.2.7).

Efficient networks [50, 173] use depthwise separable convolutions to cut down F_{ops} by up to 9 times (with 3×3 kernel size) with fewer parameters than standard

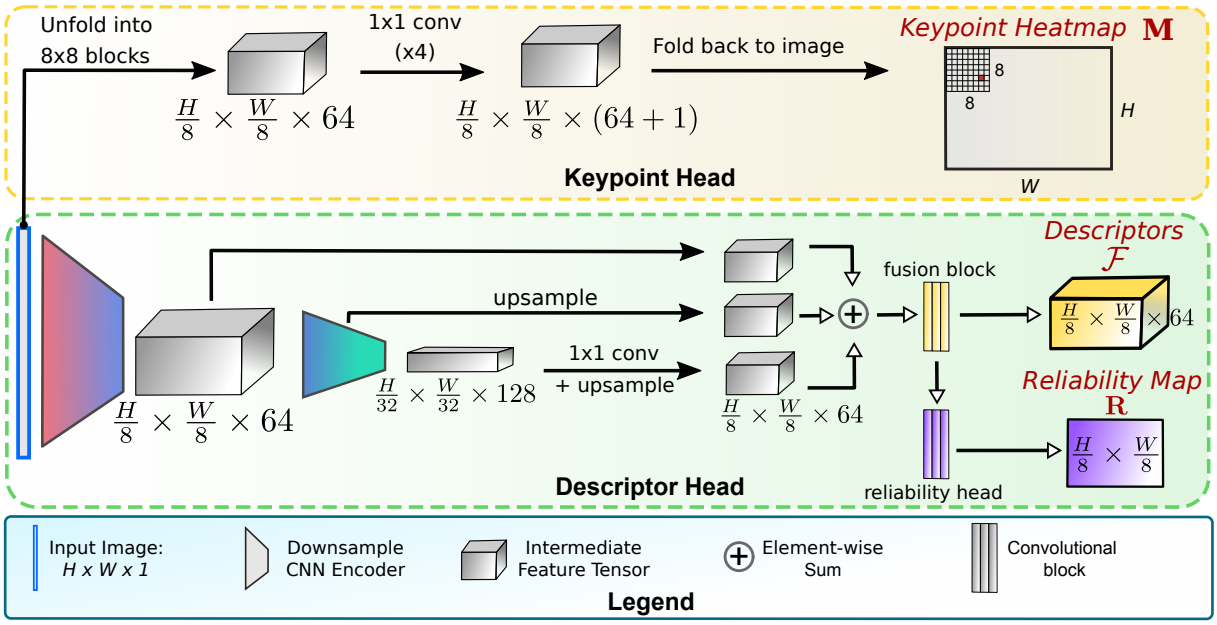


Figure 6.3: **Accelerated feature extraction network architecture.** XFeat extracts a keypoint heatmap **M**, a compact 64-D dense descriptor map $\mathcal{F}$, and a reliability heatmap **R**. It achieves unparalleled speed via early downsampling and shallow convolutions, followed by deeper convolutions in later encoders for robustness. Contrary to typical methods, it separates keypoint detection into a distinct branch, using 1×1 convolutions on an 8×8 tensor-block-transformed image for fast processing.

convolutions. However, in local feature extraction, where shallower networks handle larger image resolutions [31, 113, 40, 176, 80], this approach is less effective compared to their original use in low-resolution input scenarios like classification and object detection [131, 50, 47]. This leads to limited representational capacity and minor speed gains in shallow networks for local feature extraction.

In Equation 6.1, the $H_i * W_i$ terms emerge as the primary computational bottleneck impacting F_{ops} in CNN. SuperPoint [31] and ALIKE [176] reduce channel depth and layer count uniformly to alleviate the problem. We delve into the core of the issue, formulating a strategy to minimize early-layer depth and reconfigure channel distribution, significantly improving the accuracy-compute trade-off. Our proposed strategy involves reducing the channel count in initial convolution layers as much as possible due to the high spatial resolution. To counterbalance the parameter reduction, rather than adhering to the traditional VGG-like approach [131] of doubling channels, we propose tripling the channel count as the spatial resolution decreases, until a sufficient number of channels is reached (usually 128 for local feature backbones [147, 80, 113]). This strategy, marked by a triple rate increase in convolutional depth as spatial resolution halves, effectively redistributes the network's convolutional depth. It ensures minimal depth in early layers while compensating for the reduced parameter count accross the backbone. This approach not only significantly reduces the computational load in the early stages, particularly for high-resolution images, but also optimizes the network's overall capacity through more

effective management of convolutional depth. We found a good trade-off between spatial accuracy and speedup gains by starting with $C = 4$ channels and concluding at $C = 128$ in the final encoder block, achieving a spatial resolution of $H/32 \times W/32$.

To maintain the backbone's structural simplicity, we employ a primary unit termed the basic layer. This unit is structured with a 2D convolution with square kernel (1 or 3), complemented by ReLU activation and Batch Normalization. A stride of 2 in the convolution is applied for halving the spatial resolution as needed.

The network's architecture is modular, comprising several basic layers as a basic block, as depicted in Figure 6.4. Each block consists of two or three basic layers. The backbone of our network comprises six of these basic blocks, designed to halve the spatial resolution in each step while progressively augmenting the depth using the approach detailed in Section 6.1.1 The first basic layer on each block performs the spatial downsampling. Two additional basic blocks, in the end, are employed to perform the fusion of multi-resolution features and reliability map prediction, respectively. Preliminary experiments revealed that adding a single skip connection to the model as shown in Figure 6.4 slightly increased performance, which has led to its incorporation in the final backbone design.

6.1.2 Local Feature Extraction

In this section, we describe how our streamlined backbone is used to extract local features and perform dense matches. The architecture is formed by two branches: the descriptor head, and the keypoint head, shown in Figure 6.3. The two branches work independently to extract the keypoint heatmap $\mathbf{M}$, the local descriptors $\mathcal{F}$, and reliability map $\mathbf{R}$. While detection and description appear decoupled, they are linked through the reliability map $\mathbf{R}$, which is used to compute the final keypoint scores.

Descriptor head. The descriptor head extracts a dense feature map $\mathcal{F} \in \mathbb{R}^{H/8 \times W/8 \times 64}$, obtained by merging multi-scale features from the encoder. By using a feature pyramid strategy [74], we inexpensively increase the receptive field of the network by applying successive convolution blocks until 1/32 of original resolution is achieved, a strategy that has demonstrated success in local feature extraction to increase robustness to viewpoint changes [80, 176, 147] and a key ingredient for small network backbones to work well in practice. We merge the intermediate representation at three different scale levels: $\{1/8, 1/16, 1/32\}$ by bilinearly upsampling and projecting all intermediate representations to $H/8 \times W/8 \times 64$ followed by element-wise summation. Finally, a convolutional fusion block composed of three basic layers is used to combine the representations into the final feature representation $\mathcal{F}$. An additional convolutional block is used to regress a reliability map $\mathbf{R} \in \mathbb{R}^{H/8 \times W/8}$,

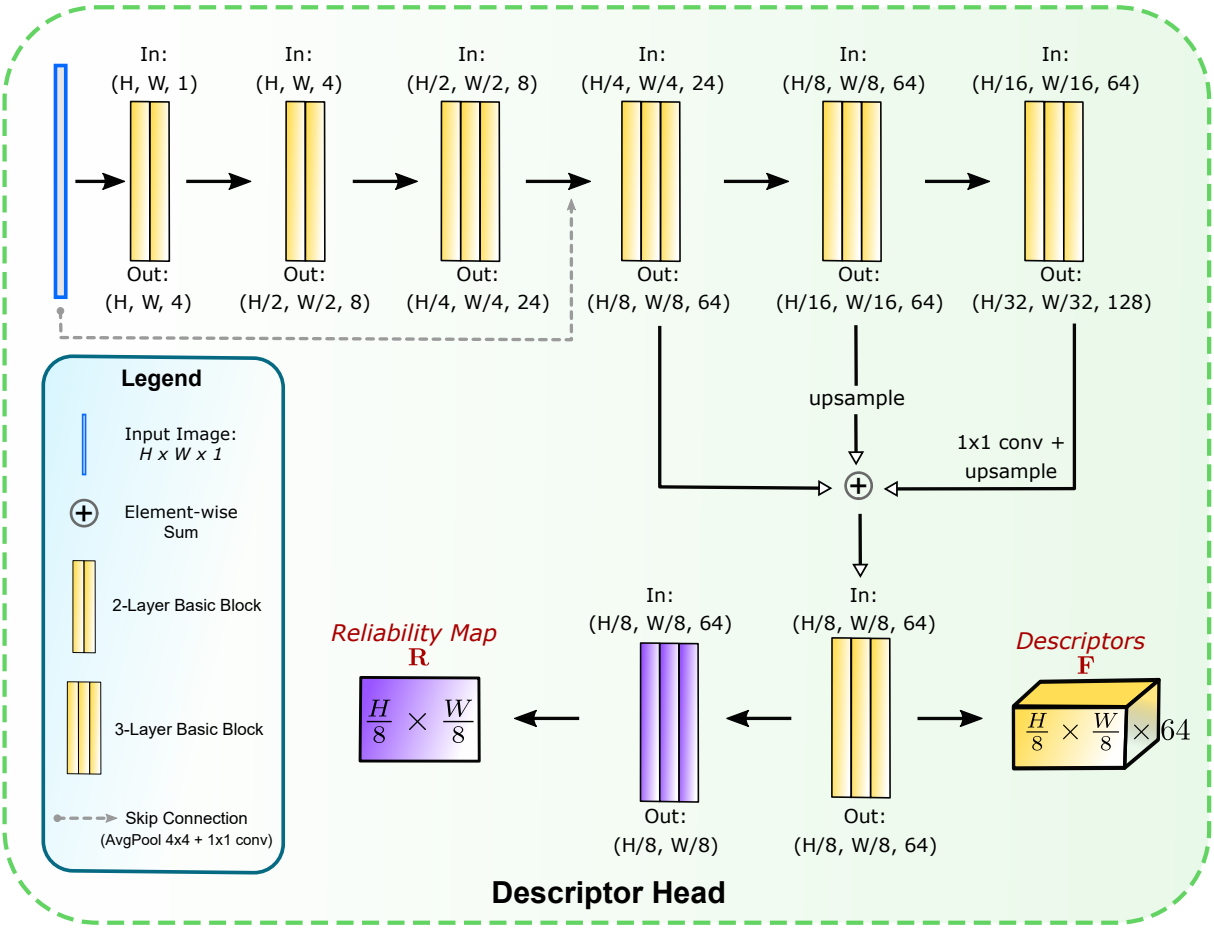


Figure 6.4: **Detailed descriptor backbone.** Our backbone is comprised of 23 convolutional layers, following the downsampling strategy described in Sec. 3.1 of the main paper. Our network is deeper compared to ALIKE [176] and SuperPoint [31] backbones in terms of layers, but due to the efficient downsampling strategy adopted, our network's inference is much faster.

which models the unconditional probability $\mathbf{R}_{i,j}$ that a given local feature $\mathcal{F}_{i,j}$ can be matched confidently. An overview of the feature extraction process is shown in Figure 6.3, and the detailed descriptor backbone is depicted in Figure 6.4.

Keypoint head. In general, backbones for local feature extracion rely on UNets [147], VGG [31], and ResNets [80]. The strategy used in SuperPoint [31] offers the fastest approach to extract pixel-level keypoints. It uses features in the final encoder with 1/8 of the original image resolution, and extracts pixel-level keypoints by classifying the coordinate of the keypoint in a flattened 8×8 grid from the feature embeddings. We adopt a strategy similar to SuperPoint, but with a major difference. We introduce a novel approach that employs a dedicated parallel branch for keypoint detection focused on low-level image structures. As shown in the ablation experiments (Section 6.2.7), by jointly training a descriptor and a keypoint regressor within a single neural network backbone significantly degrades the performance of semi-dense matching for compact CNN architectures.

Our key insight lies in the efficient utilization of the low-level features through a minimalist convolutional branch. To maintain spatial resolution without sacrificing speed, we represent the input image as a 2D grid comprised of 8×8 pixels on each grid cell, and we reshape each cell into 64-dimensional features. This representation preserves spatial granularity within individual cells, while exploiting rapid 1×1 convolutions for regressing keypoint coordinates. After four convolutional layers, we obtain a keypoint embedding $\mathbf{M} \in \mathbb{R}^{H/8 \times W/8 \times (64+1)}$ encoding the logits of keypoint distribution inside a cell $\mathbf{k}_{i,j} \in \mathbf{M}$, and classify the keypoint as one of the 64 possible positions inside $\mathbf{k}_{i,j} \in \mathbb{R}^{65}$ plus a dustbin to consider the case where no keypoint is found [31]. During inference, the dustbin is discarded and the heatmap is re-interpreted as an 8×8 cell. Figure 6.3 depicts the entire process of the Keypoint Head.

Dense matching. Recent research [134, 23] has demonstrated the benefits of dense image region matching, improving coverage and robustness. Our work proposes a lightweight module for dense feature matching, differing from other detector-free methods in two ways. Firstly, we can control memory and compute footprint by selecting top-K image regions according to their reliability score $\mathbf{R}_{i,j}$ and caching them for future matching. Secondly, we propose a simple and lightweight MLP to perform coarse-to-fine matching without high-resolution feature maps [134, 40], enabling us to perform semi-dense matching in resource-constrained settings.

Given the dense local feature map $\mathcal{F}$, which is at 1/8 of input spatial resolution, or a subset $\mathcal{F}_s \in \mathcal{F}$, we propose a simple refinement strategy to recover pixel-level offsets. Let $\mathbf{f}_a \in \mathcal{F}_1$ and $\mathbf{f}_b \in \mathcal{F}_2$ be two matching features obtained by traditional nearest neighbor matching from an image pair $(\mathcal{I}_1, \mathcal{I}_2)$. We predict offsets $\mathbf{o} = \text{MLP}(\text{concat}(\mathbf{f}_a, \mathbf{f}_b))$, classifying the offset (x, y) that leads to the correct pixel-level match at original image resolution:

$$(x, y) = \underset{\substack{i \in \{1,\dots,8\} \\ j \in \{1,\dots,8\}}}{\arg\max}\, \mathbf{o}(i, j), \tag{6.2}$$

where $\mathbf{o} \in \mathbb{R}^{8 \times 8}$ has the logits of a probability distribution over the possible offsets.

The match refinement module is trained in an end-to-end manner alongside the backbone network, ensuring that the intermediate feature representation retains fine-grained spatial details within a compact embedding space. The offset prediction is conditioned on the coarsely matched feature pair $(\mathbf{f}_a, \mathbf{f}_b)$, reducing the search space. Figure 6.5 illustrates the lightweight match refinement module.

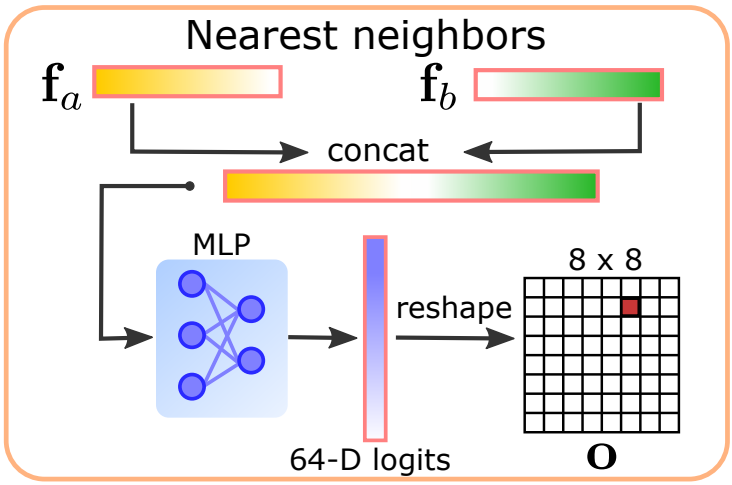


Figure 6.5: **Match refinement module for dense matching setting in XFeat.** This module learns to predict pixel-level offsets by only considering as input pairs of nearest neighbors from the original coarse-level features at 1/8 of original spatial resolution, significantly saving memory and compute.

6.1.3 Network Training

We train XFeat in a supervised manner with pixel-level ground truth correspondences. We assume image pairs $(\mathcal{I}_1, \mathcal{I}_2)$ with N matching pixels $M_{\mathcal{I}_1 \leftrightarrow \mathcal{I}_2} \in \mathbb{R}^{N \times 4}$, where the first two columns of $M_{\mathcal{I}_1 \leftrightarrow \mathcal{I}_2}$ encode the (x, y) coordinates of the points in $\mathcal{I}_1$, and the last two columns for $\mathcal{I}_2$.

Learning local descriptors. To supervise the local feature embeddings $\mathcal{F}$, we employ the negative log-likelihood (NLL) loss, as preliminary experiments showed that NLL converged faster and achieved higher accuracy compared to the triplet loss. Descriptor sets $\mathcal{F}_1$ and $\mathcal{F}_2$ are sampled from the dense maps $\mathcal{F}_{(\cdot,\cdot)}$, and each is represented in $\mathbb{R}^{N \times 64}$, comprising N 64-dimensional descriptors. The i-th rows $\mathcal{F}_1(i, \cdot)$ and $\mathcal{F}_2(i, \cdot)$ correspond to two descriptors of the same point from $\mathcal{I}_1$ and $\mathcal{I}_2$ respectively. Then, the similarity matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$ is obtained by: $\mathbf{S} = \mathcal{F}_1 \mathcal{F}_2^{\mathsf{T}}$. Given the symmetry of matching, we take both matching directions [134], resulting in the dual-softmax loss $\mathcal{L}_{ds}$, where the similarity measure of corresponding features lie in the main diagonal $\mathbf{S}_{ii}$ of $\mathbf{S}$ and softmax_r is performed row-wise:

$$\begin{aligned} \mathcal{L}_{ds} = &- \sum_i \log(\text{softmax}_r(\mathbf{S})_{ii}) \\ &- \sum_i \log\big(\text{softmax}_r(\mathbf{S}^{\mathsf{T}})_{ii}\big). \end{aligned} \tag{6.3}$$

Learning reliability. We supervise the reliability map during training by interpreting the dual-softmax probability as a confidence measure, denoted as $\bar{\mathbf{R}} \in \mathbb{R}^N$. $\bar{\mathbf{R}}_1$ and $\bar{\mathbf{R}}_2$ are

obtained by matching $\mathcal{F}_1$ and $\mathcal{F}_2$ with the dual-softmax strategy: $\bar{\mathbf{R}}_1 = \max_r(\text{softmax}_r(\mathbf{S}))$, and $\bar{\mathbf{R}}_2 = \max_r(\text{softmax}_r(\mathbf{S}^\mathsf{T}))$, similarly to Equation 6.3. As the training progresses, intuitively, distinct features will have high confidence matching probability. Thus, we supervise the reliability map directly with the L1 loss given the dual softmax scores $\bar{\mathbf{R}}_1$ and $\bar{\mathbf{R}}_2$:

$$\mathcal{L}_{rel} = |\sigma(\mathbf{R}_1) - \bar{\mathbf{R}}_1 \odot \bar{\mathbf{R}}_2| + |\sigma(\mathbf{R}_2) - \bar{\mathbf{R}}_1 \odot \bar{\mathbf{R}}_2|, \tag{6.4}$$

where σ is the sigmoid activation function and $\odot$ the Hadamard product. Note that for the reliability loss $\mathcal{L}_{rel}$, we only backpropagate the gradients through $\mathbf{R}$.

Learning pixel offsets. The match refinement module is supervised with pixel-level offsets obtained from the ground-truth correspondences $M_{\mathcal{I}_1 \leftrightarrow \mathcal{I}_2}$ at the original input image resolution. We also employ the NLL loss over the logits $\mathbf{o}$ described in Equation 6.1.2. During training, corresponding descriptors $\mathcal{F}_1(i, \cdot)$ and $\mathcal{F}_2(i, \cdot)$, together with their ground-truth offset $(\bar{x}, \bar{y})$ are obtained using $M_{\mathcal{I}_1 \leftrightarrow \mathcal{I}_2}(i, \cdot)$, and the fine matching loss $\mathcal{L}_{fine}$ becomes:

$$\mathcal{L}_{fine} = -\sum_i \log(\text{softmax}\,(\mathbf{o}_i))_{\bar{y}_i, \bar{x}_i}. \tag{6.5}$$

Learning keypoints. Our keypoint detection branch is minimalist by design. Whilst it is possible to supervise the keypoint head with existing keypoint losses [176, 113, 147], we chose to employ knowledge distillation from a larger teacher network to facilitate its learning. We opted for ALIKE [176] keypoints obtained from its tiny backbone to supervise our model. This choice is strategic, as the smaller backbone tends to concentrate on lower-level image features like corners, lines, and blobs, aligning well with our designed detector branch, given its limited receptive field size of 8×8 pixels. Given the keypoint raw logit map $\mathbf{M} \in \mathbb{R}^{H/8 \times W/8 \times (64+1)}$, we map keypoint coordinates from the teacher network (t_x, t_y) inside each cell $\mathbf{k}_{i,j} \in \mathbb{R}^{65}$ to linear index $t_{idx} = (t_x + t_y * 8), \quad t_{idx} \in \{0, 1, ..., 63\}$. To supervise the dustbin, when no keypoint is detected inside a cell $\mathbf{k}_{i,j}$, we set $t_{idx} = 64$. During training, we set an upper limit of samples for the no keypoint case to avoid class imbalance. Finally, the NLL loss is employed to compute the keypoint loss $\mathcal{L}_{kp}$:

$$\mathcal{L}_{kp} = -\sum_k \log(\text{softmax}\,(\mathbf{k}_{i,j}))_{t_{idx}}. \tag{6.6}$$

The final loss $\mathcal{L}$ is then a linear combination of all losses:

$$\mathcal{L} = \alpha\mathcal{L}_{ds} + \beta\mathcal{L}_{rel} + \gamma\mathcal{L}_{fine} + \delta\mathcal{L}_{kp}, \tag{6.7}$$

where $\{\alpha, \beta, \gamma, \delta\}$ are hyperparameters to adjust the magnitude of the different losses.

6.2 Quantitative Evaluation

In this section, we evaluate XFeat on three relevant tasks: (i) relative camera pose estimation, (ii) visual localization, and (iii) homography estimation. We compare our proposed solution with recent state-of-the-art works and also established descriptors. We also present ablations to justify our design decisions, and a comprehensive runtime analysis in a GPU-free setting.

6.2.1 Training & inference

XFeat was implemented on PyTorch [102] and trained on a blend of Megadepth [72] and synthetically warped COCO [75] images, using a 6:4 ratio, with images resized to $(W = 800, H = 600)$. Hybrid training was found to enhance generalization in our experiments (Section 6.2.7), aligning with recent findings [76]. The training involved batches of 10 image pairs using the Adam optimizer [61], leading to convergence within 36 hours on an NVIDIA RTX 4090 GPU.

Our ablations show that hybrid training significantly improves generalization for small CNN, as observed in high-capacity models [76]. The network was trained on batches of 10 image pairs using the Adam optimizer [61] with an initial learning rate of 3×10^{-4}, applying an exponential decay of 0.5 at every 30,000 gradient updates. Convergence is attained after 160,000 iterations, within 36 hours on a single NVIDIA RTX 4090 GPU, consuming 6.5 GB of VRAM in total, considering both training and synthetic warps done on the fly on GPU. Disk I/O is the predominant speed bottleneck due to the overhead of loading images and depth maps from the Megadepth dataset in their original resolution, which can be easily solved with a more careful data preparation scheme. The low memory usage of our method enables training on entry-level hardware, facilitating the fine-tuning or full training of our network for specific tasks and scene types.

XFeat inference. We considered two settings: Sparse (XFeat) and semi-dense matching (XFeat*), both utilizing the same pretrained backbone. In XFeat, we extracted up to 4,096 keypoints from the keypoint heatmap $\mathbf{M}$, using their scores derived from the keypoint and reliability confidences: $score = \mathbf{M}_{i,j} \cdot \mathbf{R}_{i,j}$. Local features were then bicubically interpolated from $\mathcal{F}$ at these keypoint locations and matched with Mutual Nearest Neighbor (MNN) search. For XFeat*, we enhanced features by processing images at 2 different scales (0.65 and 1.3, resizing the image internally after receiving the input), retaining the top 10,000

features according to their reliability. We used MNN search and offset refinement to match the features, retaining only those with offset prediction confidence above 0.2.

6.2.2 Baselines

Among the selected baselines, DISK [147] sets a high benchmark in accuracy at the cost of increased computational demand. This is followed by SiLK [40], SuperPoint [31], ZippyPoint [31], and ALIKE [176]. For SiLK and ALIKE, we opted for their smallest available backbones – *ALIKE-Tiny* and *VGGnp-μ* – aligning with our focus on models emphasizing compute efficiency. Finally, ORB [176] represents the upper limit in terms of speed, albeit with a large gap in robustness compared to the other solutions. Although there are additional compatible methods in the literature such as ALIKED [175], we focused on keeping the most representative method for each feature category (fast or accurate approaches). This diverse set of baselines enables a comprehensive assessment of our descriptor across the spectrum of computational expense and accuracy, ensuring a thorough validation against the current state-of-the-art. We use the top $4,096$ detected keypoints for all baselines, except for those marked with *, where the top $10,000$ keypoints are used. For matching, MNN search is employed. ZippyPoint model was used in its form as provided by the authors without hardware-specific compilation, due to the lack of clear instructions.

6.2.3 Relative pose estimation

Setup. Megadepth [72] and ScanNet [27] test sets are used as in previous works [134, 76], providing camera poses on scenes that do not overlap with our training set. The scenes contain significant viewpoint and illumination changes simultaneously and present repetitive structures, posing a significant challenge. LO-RANSAC [68] is used to estimate the essential matrix. We search for the optimal threshold for each method, and resize the images such that the maximum dimension becomes 1,200 pixels for Megadepth and use the default (VGA) resolution for ScanNet.

Metrics. We use the area under the curve (AUC) at thresholds of $\{5°, 10°, 20°\}$ [134, 76]. Additionally, we report the Acc@10°, which is the proportion of poses where the maximum

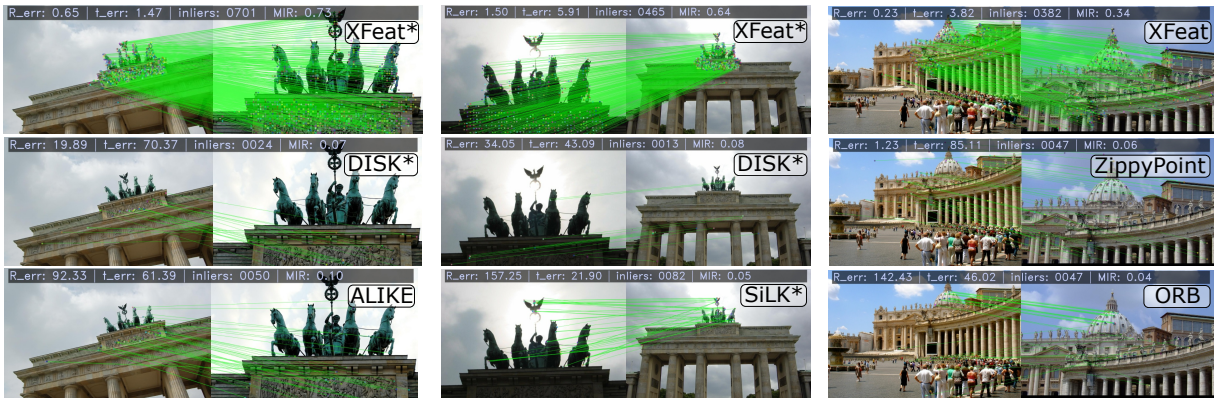


Figure 6.6: **Qualitative results on Megadepth-1500.** XFeat* and XFeat demonstrate exceptional robustness against variations in viewpoint and illumination. This is especially evident in challenging scenarios where heavy methods like DISK* breaks and XFeat* provide accurate relative pose 16× times faster in semi-dense settings with a comparable number of local features.

Table 6.1: **Megadepth-1500 relative camera pose estimation.** Our method achieves superior performance compared to other lightweight methods, while also outperforming SuperPoint at 9× speedup, and with comparable results to DISK at 16× speedup. * denotes 10k keypoints. FPS is the average of 30 frames ± standard deviation computed in VGA resolution. Best in bold, second best underlined, separated by method class (standard/fast). $^{+}$ indicates code used as provided by authors without hardware optimization. The methods proposed in this dissertation are highlighted in **bold** for reference.

	Method	AUC@5°	@10°	@20°	Acc@10°	MIR	inlier	dim	FPS
Standard	SiLK [40]	14.7	21.5	29.3	31.9	0.17	235	**32-f**	2.8 ± 0.08
	SiLK* [40]	16.2	23.2	31.8	34.7	0.14	478	**32-f**	2.9 ± 0.12
	SuperPoint [31]	37.3	50.1	61.5	67.4	0.35	495	256-f	**3.0 ± 0.07**
	DEAL [108]	39.5	53.0	64.9	71.7	0.36	389	128-f	0.13 ± 0.00
	DALF [105]	39.9	53.2	64.2	70.7	0.40	439	128-f	0.2 ± 0.00
	DISK [147]	53.8	65.9	75.0	**81.3**	**0.72**	1231	128-f	1.2 ± 0.01
	DISK* [147]	**55.2**	**66.8**	**75.3**	**81.3**	0.71	**1997**	128-f	1.2 ± 0.01
Fast	ORB [120]	17.9	27.6	39.0	43.1	0.25	288	**256-b**	**44.3 ± 1.18**
	ZippyPoint [56]	23.6	34.9	46.3	51.8	0.23	192	**256-b**	$^{+}$1.8 ± 0.06
	ALIKE [176]	49.4	61.8	71.4	77.7	0.47	333	64-f	5.3 ± 0.33
	XFeat	42.6	56.4	67.7	74.9	0.55	892	64-f	27.1 ± 0.33
	XFeat*	**50.2**	**65.4**	**77.1**	**85.1**	**0.74**	**1885**	64-f	19.2 ± 1.12

angular error is below 10 degrees, the mean inlier ratio (MIR), which is the ratio of matching points that comply with the estimated model after RANSAC, and the number of inlier points (#inlier). Finally, we measure the frames per second (FPS) of each method on a budget-friendly laptop **without** GPU and an Intel(R) i5-1135G7 @ 2.40GHz CPU. We also indicate whether the descriptor is floating-point (denoted by **f**) or binary-based (denoted by **b**) and report the descriptor dimensionality (dim).

Table 6.2: **ScanNet-1500 relative pose estimation.** XFeat and XFeat* exhibit better generalization to indoor scenes than current methods trained on outdoor datasets and handcrafted methods.

AUC	Super-Point	DISK (4k/10k)	ORB	ALIKE	XFeat/ XFeat*
@5°	12.5	9.6 / 11.3	9.0	8.0	16.7 / **18.4**
@10°	24.4	19.3 / 22.3	18.5	16.4	32.6 / **34.7**
@20°	36.7	30.4 / 33.9	29.9	25.9	47.8 / **50.3**

6.2.3.1 Results analysis

Table 6.1 shows the metrics on the relative camera pose estimation task on Megadepth-1500. Our method is much faster (5×) than the fastest available learning-based solution (ALIKE) and achieves competitive results in the sparse setting on several metrics. Moreover, it can deliver state-of-the-art results for the dense matching configuration using 10,000 descriptors on AUC@20°, Acc@10° and MIR in a fair comparison with DISK*, a much heavier model, considering the same number of descriptors. Figure 6.6 shows examples where XFeat stands out over existing solutions. Our method also allows more efficient matching with low-dimensional descriptors (64-f) compared to DISK and SuperPoint. Detailed timing analysis is provided in Section 6.2.8. It is worth mentioning that we obtain state-of-the-art results in more loose thresholds due to the requirement of interpolating the descriptors and predicting offsets at coarser resolution. Table 6.2 shows AUC values for the most competitive methods in ScanNet-1500 indoor imagery. Notice that none of the methods were retrained. DISK and ALIKE show signs of bias towards landmark datasets, while our approach demonstrates superior generalization.

XFeat and XFeat* surpass both fast and standard local feature extractors in pose accuracy while being significantly faster for indoor relative pose estimation. DISK and ALIKE, which were trained in the same Megadepth scenes as XFeat, display signs of overfitting in landmark imagery: they perform exceptionally well in strict thresholds (AUC@5°) on Megadepth-1500 test set, but their relative performance are similar or worse in tasks such as homography estimation and visual localization compared to XFeat and SuperPoint, as one can observe in Tables 6.3 and 6.4.

We conjecture that XFeat produces less biased local descriptors due to our hybrid training with synthetic warps on COCO. SuperPoint also demonstrate increased generalization accross different downstream tasks and datasets due to its inherent self-supervised training strategy on synthetic warps. Hybrid training can encourage local feature representations to focus less on distinctive textures often present in landmark outdoor imagery that could bias the CNN training. In addition, the large receptive field of our network, as

well as its increased layer depth compared to the other approaches, helps XFeat in indoor imagery (which often lacks distinctiveness at the local level), resulting in more consistent matches compared to DISK and ALIKE in ScanNet-1500, even though XFeat and the competitors were not trained on ScanNet data.

6.2.3.2 Results with deformation-aware local features

We evaluate the performance of our deformation-aware local feature extraction methods, DEAL and DALF, in matching rigid scenes for relative pose estimation. As shown in Table 6.1, both methods achieve comparable scores, performing better than SuperPoint, a widely adopted learning-based baseline designed for rigid scenes. These results suggest that DEAL and DALF also provide invariance for rigid scenes under challenging image transformations, attributed to their explicit deformation modeling that also accounts for rigid distortions. However, they may be computationally expensive compared to more efficient methods like XFeat, which achieves superior results with significantly less compute. Training DEAL and DALF on outdoor rigid datasets could enhance their performance on MegaDepth-1500 but might compromise their effectiveness in matching non-rigid scenes. This experiment could providing further insights into the invariance-distinctiveness trade-off in local feature extraction and is left for future work.

6.2.4 Homography estimation

Setup. We used the widely adopted HPatches [3] dataset containing sequences of images from planar scenes with moderate to strong viewpoint and illumination changes. Similarly to relative pose estimation, we used MAGSAC++ [4] to robustly estimate the homography transformation given the correspondences of each method.

Metrics. We followed ALIKE [176] protocol and estimated Mean Homography Accuracy (MHA). We used predefined thresholds of $\{3, 5, 7\}$ pixels. The accuracy was computed considering the average corner error in pixels by warping the four reference image corners to target images using the ground-truth homography and estimated one.

Table 6.3: **Homography estimation on HPatches.** All methods perform well due to RANSAC except ORB and SiLK which break on several illumination sequences. XFeat provides high quality homography estimation with a fraction of compute. Best in bold, second best underlined, separated by standard and fast methods. The methods proposed in this dissertation are highlighted in **bold** for reference.

Method	Illumination MHA			Viewpoint MHA		
	@3	@5	@7	@3	@5	@7
SiLK	78.5	82.3	83.8	48.6	59.6	62.5
SuperPoint	**94.6**	98.5	98.8	71.1	79.6	83.9
DEAL	88.8	94.2	95.0	**75.4**	**85.4**	**89.3**
DALF	92.7	96.5	97.3	58.6	81.4	86.8
DISK	**94.6**	**98.8**	**99.6**	66.4	77.5	81.8
ORB	74.6	84.6	85.4	63.2	71.4	78.6
ZippyPoint	94.2	96.9	98.5	66.1	76.8	80.7
ALIKE	94.6	**98.5**	**99.6**	68.2	77.5	81.4
XFeat	**95.0**	98.1	98.8	**68.6**	**81.1**	**86.1**

Results. Table 6.3 shows that our method is on par with the most accurate descriptors, reinforcing the robustness of our proposed keypoint and descriptor heads. In contrast, the performance of other lightweight solutions as ORB and SiLK are heavily compromised on the illumination and viewpoint splits, due to their limited capacity in handling agressive viewpoint and illumination changes present in the hardest image pairs. Our method also stands out for less strict thresholds, as discussed in Section 6.2.3.1.

Comparison to deformation-aware methods. In Table 6.3, we include our deformation-aware descriptors, DEAL and DALF, to evaluate whether explicit deformation modeling enhances invariance to rigid geometric transformations. In the viewpoint sequences, both DEAL and DALF demonstrate improved accuracy, achieving the highest homography accuracy at 5 pixels and beyond. DEAL exhibits superior invariance to viewpoint changes, leveraging SIFT's affine-invariance design, while DALF's keypoints, optimized for non-rigid scenes, may not perform optimally under rigid transformations. Nevertheless, compared to DEAL and DALF, XFeat is simpler to implement and significantly faster, offering a better compute-accuracy trade-off in several tasks requiring real-time inference capability and embedded deployment.

Table 6.4: **Visual localization on Aachen day-night.** XFeat enables fast and accurate localization, especially on the more challenging case of matching day-to-night images, being on-par with the state-of-the-art on thresholds above $0.5m, 5°$. Best in bold, second best underlined, separated by standard and fast methods.

Method	Day			Night		
	$0.25m$ $2°$	$0.5m$ $5°$	$5m$ $10°$	$0.25m$ $2°$	$0.5m$ $5°$	$5m$ $10°$
SuperPoint	**87.4**	<u>93.2</u>	<u>97.0</u>	<u>77.6</u>	<u>85.7</u>	<u>95.9</u>
DISK	<u>86.9</u>	**95.1**	**97.8**	**83.7**	**89.8**	**99.0**
ORB	66.9	76.1	81.7	10.2	12.2	19.4
ZippyPoint	80.7	88.6	93.7	61.2	70.4	79.6
ALIKE	**85.7**	**92.4**	**96.7**	**81.6**	88.8	**99.0**
XFeat	<u>84.7</u>	<u>91.5</u>	<u>96.5</u>	<u>77.6</u>	**89.8**	<u>98.0</u>

6.2.5 Visual localization

Setup. The hierarchical localization pipeline HLoc [121] is used to localize images of day and night scenes from the Aachen dataset [124]. Given the provided keypoint correspondences, HLoc triangulates an SfM map using the available ground-truth camera poses. A separate set of query images are then localized within the 3D map using the keypoint matches. For a fair comparison, we resize the images such that maximum dimension is held at $1,024$ pixels, and extract the top $4,096$ keypoints for all approaches.

Metrics. We use the standard metric provided by HLoc, which is the accuracy of correctly estimated camera poses within thresholds of position errors $\{0.25m, 0.5m, 5m\}$ and rotation errors $\{2°, 5°, 10°\}$ respectively.

Results. Table 6.4 presents the results of the visual localization experiment. Our method demonstrates similar performance to leading approaches as SuperPoint and DISK, while achieving a significant speed advantage, being at least 9 times faster and with a more compact descriptor. These findings challenge the prevailing trend in the literature to employ large and more intricate models for downstream tasks. Contrarily, they underscore the efficacy of simpler models that not only match accuracy but also offer the benefits of efficient operation on resource-constrained systems.

Table 6.5: **Comparison with state-of-the-art deformation-aware local features**. The *RGB* methods only require color images. *D&D* methods perform joint detection and description. The deformation-aware methods explicitly tackling deformations are shown as *D-A*. Best scores in bold and second-best underlined. The methods' names in bold font highlight the works proposed in this dissertation.

RGB	D&D	D-A	Method	**Datasets:** 833 **pairs total** – MS / MMA @ 3 pixels ↑				**Mean**
				Kinect 1 [107]	*Kinect 2* [107]	*DeSurT* [157]	*Sim.* [107]	
✓	✓		SuperPoint [31]	0.45 / 0.74	0.54 / 0.85	0.39 / 0.68	0.18 / 0.34	0.41 / 0.69
✓	✓		R2D2 [113]	0.17 / 0.36	0.25 / 0.59	0.14 / 0.32	0.06 / 0.16	0.17 / 0.39
✓	✓		ASLFeat [80]	0.31 / 0.58	0.39 / 0.69	0.28 / 0.53	0.19 / 0.35	0.31 / 0.56
✓	✓		DISK [147]	0.53 / 0.76	0.52 / 0.81	0.44 / 0.61	0.26 / 0.34	0.45 / 0.66
✓	✓		**XFeat**	0.45 / 0.74	0.23 / **0.90**	0.37 / 0.66	0.19 / 0.34	0.30 / 0.70
		✓	**GeoBit** [89]	0.31 / 0.65	0.35 / 0.77	0.20 / 0.47	0.32 / 0.71	0.30 / 0.66
		✓	**GeoPatch** [107]	0.32 / 0.66	0.35 / 0.80	0.26 / 0.60	0.39 / **0.86**	0.33 / 0.73
✓		✓	DaLI [129]	0.25 / 0.51	0.35 / 0.76	0.21 / 0.48	0.10 / 0.22	0.25 / 0.54
✓		✓	**DEAL** [108]	0.33 / 0.68	0.38 / 0.85	0.27 / 0.63	0.36 / 0.80	0.34 / 0.75
✓		✓	NKD [83, 108]	0.44 / 0.74	0.49 / 0.82	0.33 / 0.64	0.31 / 0.74	0.40 / 0.74
✓	✓	✓	**DALF** [105]	**0.54** / **0.82**	**0.62** / **0.90**	**0.49** / **0.73**	**0.42** / 0.69	**0.53** / **0.80**

6.2.6 Non-rigid image matching

To evaluate XFeat's robustness to transformations beyond rigid scenes, we conducted experiments on the non-rigid image matching benchmark [107], comparing its performance and that of other detect-and-describe methods against state-of-the-art deformation-aware approaches.

Setup. The performance of all methods is assessed using top 2,048 keypoints [54] on the non-rigid benchmark [107]. The overall mean was calculated with full-precision values by averaging the scores of all 833 image pairs before rounding.

Metrics. We adopt the metrics from DEAL [108], using Matching Scores (MS) [85], $MS = |\mathcal{S}_{gt}| / \min(|keypoints_i|, |keypoints_j|)$, which evaluates correct correspondences $\mathcal{S}_{gt}$ relative to the smaller number of detected keypoints, and mean matching accuracy (MMA), $MMA = |\mathcal{S}_{gt}|/|\mathcal{K}_{gt}|$, which measures accuracy over successfully detected keypoints $\mathcal{K}_{gt}$ under a pixel threshold.

Results. As shown in Table 6.5, XFeat achieves slightly lower MS score than DISK on average, but ranks first in MMA score considering only descriptors for rigid scenes, demonstrating some degree of robustness to deformations. The lower matching scores and higher mean matching accuracy suggest that the detected keypoints are more sensitive to deformations than the descriptors. This indicates that XFeat's minimalist keypoint

detector, with knowledge distilled from a standard keypoint detector, lacks robustness to adverse deformations. Hence, the MS drop is expected given its simple small CNN of limited receptive field and teacher network optimized for rigid scenes, exhibiting a clear trade-off between invariance and speed in local feature detection. These findings point to promising research directions, such as improving the efficiency of deformation-aware methods with XFeat's efficient image encoder backbone, and performing keypoint distillation from deformation-aware detectors like DALF [105] and NKD [83].

6.2.7 Ablation

We ablate our architecture in several configurations, which are listed in Table 6.6. We evaluate whether training with additional synthetic warps (i) may help the model to become more robust to challenging image pairs. Training on both real and synthetic warps is beneficial, especially for the dense matching setting. Second, we evaluate if we can further reduce channel count in the network (ii). We halve the channels of the last three convolutional blocks to 32 instead of 64, but performance significantly degrades for both sparse and dense settings.

We also demonstrate the rationale behind devising a parallel branch for keypoint detection. Without the proposed keypoint head (iii), an additional convolutional block is used on top of the output descriptor embeddings akin to SuperPoint. As shown in Table 6.6, XFeat* experiences degradation in performance when trained under this specific setup, since the limited network size constrains the capacity of intermediate embeddings, rendering them less effective for semi-dense matching in non-repeatable regions, adversely affecting the match refinement task. Thus, we opted to design a parallel branch which offers great trade-off between sparse and dense matching as shown in Table 6.6 – Default and (iii). Lastly, we evaluate the benefits of our proposed match refinement module. For XFeat*, the match refinement step is critical for enhancing accuracy. In our benchmarks, this module incurs only an additional 11% inference cost compared to MNN matching for an average of 10,000 descriptors, as discussed in Section 6.2.8.

Table 6.6: **Ablation on Megadepth-1500.** We ablate the architecture and training strategies for relative camera pose estimation.

Strategy	**AUC@5°**	
	XFeat	XFeat*
Default	42.6	50.2
(i) No synthetic data	41.5	33.9
(ii) Smaller model	37.4	40.7
(iii) Joint keypoint extraction	42.9	39.7
(iv) No match refinement	-	38.6

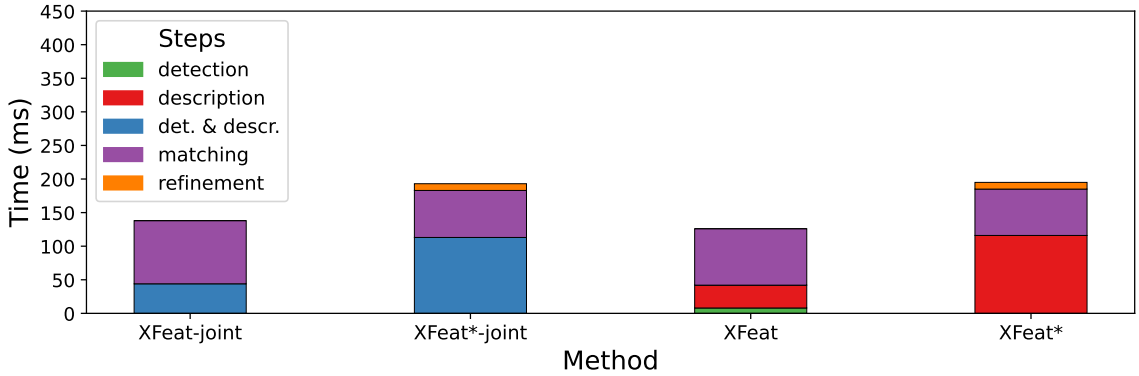


Figure 6.7: **XFeat detailed timing analysis on i7-6700K CPU.** Required time by each step of our ablated methods.

6.2.8 Detailed timing analysis

This section reports a detailed timing analysis of our proposed solutions in sparse and semi-dense matching settings. Regarding XFeat*'s match refinement step, we show in Figure 6.7 that the match refinement cost is negligible. More notably, even with the refinement step included, XFeat* achieves a similar matching time compared to XFeat with the same number of keypoints because refinement is performed after the nearest neighbor search. Additionally, we present the extraction running times for the most efficient methods available on an **Orange Pi Zero 3** equipped with a Cortex-A53 ARM processor. This device stands out as one of the *smallest and most affordable consumer-grade embedded computers* ($28). Considering its limited processing power, we adjusted the input resolution to 480×360 for all methods and used their standard PyTorch implementation without any deployment optimization. Our findings show that XFeat operates at an average of 1.8 FPS, SuperPoint at 0.16 FPS, and ALIKE at 0.58 FPS, respectively. This experiment shows that XFeat is the only learned method capable of running over one FPS on a highly constrained embedded device that is not optimized for neural network inference.

6.2.9 Comparison with learned matchers

Since XFeat* uses paired inputs when performing the refinement step, we provide additional comparisons of XFeat* (semi-dense matching) with popular learned matchers such as LoFTR [134] and LightGlue [76], and coarse-to-fine strategies as Patch2Pix [177], to elucidate the key differences. The results for these new approaches are shown in Table 6.7. Although XFeat* needs paired inputs for refinement, it fundamentally differs in

Table 6.7: **Matchers comparison on Megadepth-1500.** Inference speed in pairs per second (PPS) @ 1,200 px. (i7-6700K CPU).

Method	Type	AUC@5°	@10°	@20°	Acc@10°	MIR	#inliers	PPS
LoFTR	learned matcher	68.3	80.0	88.0	93.9	0.93	3009	0.06
LightGlue	learned matcher	61.4	75.0	84.8	91.8	0.92	475	0.31
Patch2Pix	coarse-fine	47.8	61.0	71.0	77.8	0.59	536	0.05
XFeat*	coarse-fine	**50.2**	**65.4**	**77.1**	**85.1**	**0.74**	**1885**	**1.33**

its methodology from learned matchers, being only comparable to Patch2Pix, as we rely on traditional nearest neighbor search for matching, followed by a lightweight refinement of matches, incurring a negligible computational load (see Figure 6.7). The requirement for paired inputs does not change the usual pipeline for SfM and visual localization tasks because XFeat*'s features can be stored for each image independently, as usually done for sparse settings. For instance, high-resolution feature maps are not required, unlike LoFTR, to produce refined matches.

Our techniques are, in fact, complementary to learned matchers; for example, LightGlue can be trained using both XFeat and XFeat* features. Learned matchers are more data hungry and much more expensive to train, *e.g.*, LoFTR uses 64 GPUs for 24 hours to be trained. XFeat*, for its turn, can be trained on a single 8 GB GPU. Furthermore, XFeat* offers up to 22× speedup over existing semi-dense solutions as shown in Table 6.7 and surpasses coarse-to-fine approaches such as Patch2Pix in accuracy, while being faster and delivering many more matches than sparse learned matchers as LightGlue. Naturally, XFeat, as a local descriptor, offers limited robustness to aggressive viewpoint changes and highly ambiguous image pairs compared to transformer-based feature matchers. Coupling a lightweight transformer such as LightGlue or LoFTR's linear transformer with XFeat's local features can open new directions in scalable, high-performance image matching tasks, facilitating advancements in both efficiency and accuracy that are pivotal for pushing the boundaries in visual navigation, augmented reality, and real-time VSLAM.

6.3 Qualitative Results

Figure 6.8 and Figure 6.9 show qualitative results of our two proposed approaches compared to the competing methods. For more challenging cases such as strong viewpoint and illumination changes, XFeat and XFeat* exhibit exceptional robustness even compared to DISK [147] – the largest CNN architecture regarding floating point operations. We hypothesize that this robustness is attributed to our network's large receptive field and depth

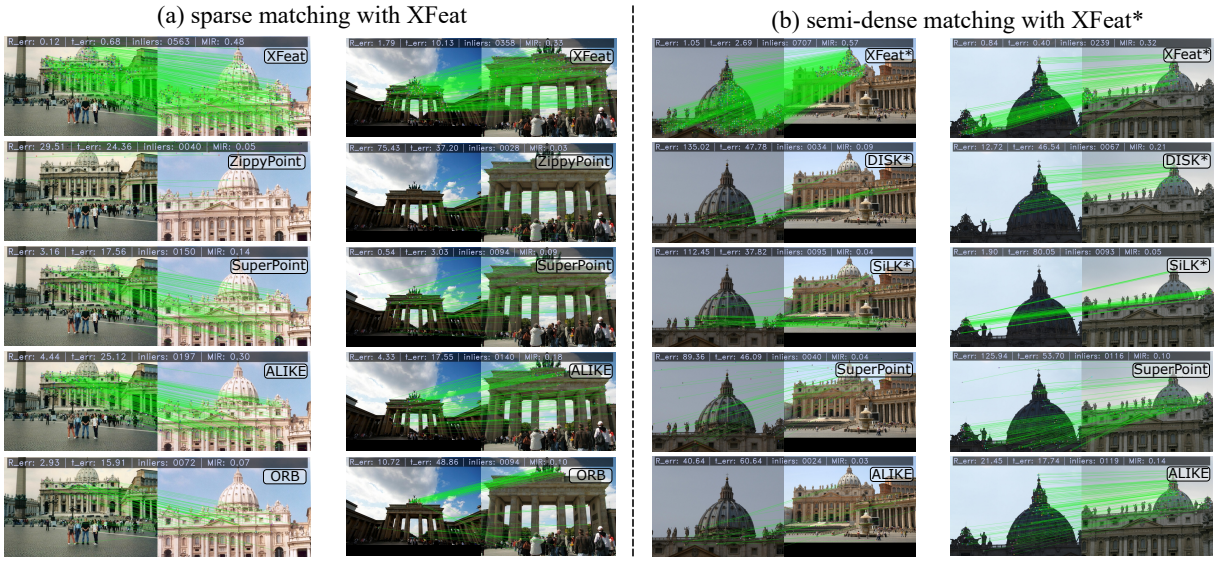


Figure 6.8: **Additional qualitative results on Megadepth-1500 [72, 134] landmark dataset.** XFeat and XFeat* are robust in demanding scenarios with significant viewpoint and illumination variations, outperforming even the more computationally intensive DISK model in semi-dense matching with 10,000 local features at a striking 16× speedup. In a sparse setting with 4,096 keypoints, our method, which is many times faster than ALIKE (5×) and SuperPoint (9×), demonstrates more robustness to wide baseline transformations due to the effective re-formulation of XFeat's backbone CNN.

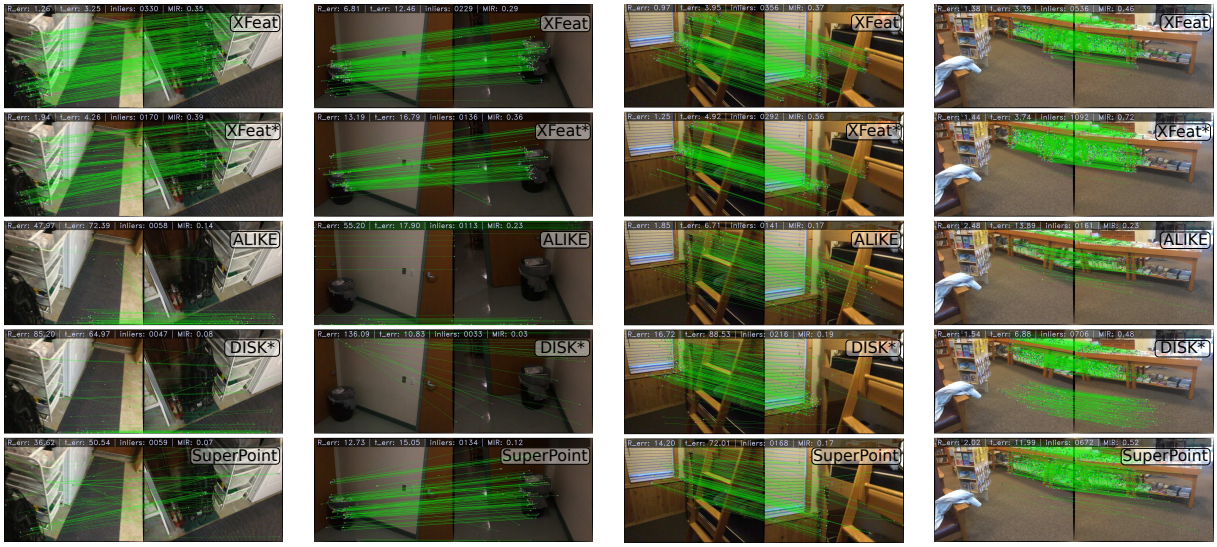


Figure 6.9: **Qualitative results on ScanNet-1500 [27, 134] indoor dataset.** Our proposed approaches consistently outperform state-of-the-art methods such as DISK and ALIKE in indoor imagery, both in terms of camera pose and inlier ratio. Notice that SuperPoint also often outperforms DISK and ALIKE. Section 6.2.3.1 provides a detailed discussion on the reasons behind our method's superiority.

compared to shallower models such as SuperPoint, ALIKE, and SiLK [40], demonstrating the effectiveness of our featherweight backbone in the compute-accuracy trade-off.

6.4 Discussion

In this chapter, we introduced XFeat, a lightweight CNN architecture designed for accelerated feature extraction, applicable to both sparse and semi-dense image matching. This versatility enables its use in tasks such as SfM and VSLAM, as well as semi-dense matching for tasks that benefit from denser point correspondences, such as relative camera pose estimation. Through experiments across three different tasks and comprehensive ablation studies, we demonstrated that existing methods present a significant gap in the compute-accuracy trade-off, with XFeat offering a substantially better balance in comparison.

On the one hand, fast approaches as ORB [120] offer cheap feature extraction but lack robustness in challenging scenarios such as day-to-night image matching and significant viewpoint changes. On the other hand, large CNN backbones, such as in DISK [147] architecture, provide robustness to such transformations but require powerful hardware for efficient inference. XFeat, for its turn, strikes a balance between robustness and efficiency (Figure 6.2), as demonstrated across three different tasks and ablation studies. Additionally, XFeat surpasses efficient methods like ORB in its adaptability, as it can be easily trained or fine-tuned for domain-specific tasks to achieve higher accuracy. With the advent of learned matchers [122, 76], it is also straightforward to couple XFeat features with existing and future learned matchers. We have trained a smaller version of the LightGlue [76] matcher, which is available in our repository. The full implementation, including training, evaluation, and inference code is publicly available [1]

The current scenario for high-performant, low-cost hardware tailored for deep learning architectures also favors XFeat in running efficiently in low-cost devices. It's simple architecture composed of standard convolution blocks allows its deployment using modern software stack as Open Neural Network Exchange (ONNX), effortlessly. There is several potential applications for XFeat in augmented reality and mobile robotics, where efficient and general data-driven solutions remain crucial for real-world deployment, particularly in mobile applications.

[1] https://github.com/verlab/accelerated_features.

6.4.1 Limitations

Since XFeat was designed for a balanced inference speed and accuracy, when it comes to robustness to extreme viewpoint changes, such as large scale changes and perspective distortions in the image pairs, the accuracy of our method is expected to drop, because of the limited representational capacity of the small feature extractor backbone. However, in many practical scenarios such as VSLAM, online frame registration, and augmented reality, where high throughput is required, the frames are usually redundant, having enough overlap. In such scenarios, XFeat provides accurate and fast pointwise correspondences, as demonstrated in pose estimation, homography registration and visual localization. In cases where matching more challenging image pairs are needed, it is possible to train a lightweight learned matcher [76] for improved robustness while keeping efficiency in extraction. Devising a lightweight learned matcher tailored for embedded platforms and real-time inference on low-budget hardware is a potential next step that could achieve state-of-the-art compute-accuracy trade-off.

XFeat also inherits the disadvantages of standard CNNs, lacking invariance to large image in-plane rotations. Rotation equivariance CNNs [159] can be seamlessly integrated into XFeat backbone but they may introduce unwanted speed overhead due to the equivariant kernel computation and inefficiencies in current implementations. A simple solution often adopted is test-time rotation augmentation. Since XFeat is very fast, it is cheap to re-compute the descriptors in several rotated versions of the image (usually in 90° intervals) and match rotated versions of the reference image to the target image. Nevertheless, recent works as Steerers [6] demonstrate that it is possible to transform the local features by a linear projection to achieve rotation equivariance without having to re-compute the local features, offering rotation invariance at a small additional cost in matching.

Chapter 7

Conclusion

In this chapter, we discuss the broader implications and overall contributions of this dissertation. While the previous chapters provided technical details about the methods and their limitations, here we aim to highlight the importance of local feature representations, our key findings during the dissertation development, and consider their significance within the wider context of the field of Computer Vision.

Throughout this dissertation, we approached the problem of local image feature representations robust to viewing changes, and notably to non-rigid object deformations, considering two key aspects: (i) invariance and (ii) efficiency. With the advent of deep-learning, it may appear that local features are being replaced by large end-to-end CNN backbones [77] and Vision Transformers (ViT) [33]. Increasingly large foundation models as DINO series [100] and multi-modal models as Florence [161] can deliver incredible zero-shot performance on several vision tasks. However, we argue that large models have two critical weaknesses, specifically for the image matching problem: compute and data. Achieving accurate image matching requires high spatial resolution, making large models impractical even with powerful hardware. Moreover, training and fine-tuning such models still demand substantial amounts of data, limiting their applicability to domain-specific tasks often encountered in industrial settings, medical science, space exploration, non-rigid correspondence, and other fields with limited data availability. This further restricts their usability in many real-world applications.

Starting from the core argument of the dissertation statement, in *Chapter* 4, we introduced *geodesic-aware* image sampling as an inductive bias mechanism that enables local feature representations to remain invariant to scene deformations. This framework is applied in both handcrafted and learned local feature extraction, highlighting its potential usage in different feature extraction methods. The proposed framework is general and can be extended beyond local feature extraction to modern architectures for tasks such as object detection, segmentation, classification, and many others. The main limitation of this approach is that metric depth must be available. Given that accurate depth data is becoming increasingly common and cheap with the introduction of new depth sensing technologies, this approach can reduce data requirements and model complexity while maintaining robustness to deformations, suggesting promising directions for future

research.

Chapter 5 tackled the significant depth requirement limitation of geodesic-aware approaches, introducing the *deformation-aware module* to endow local representations with an explicit learned non-rigid transformation from single color images, trained end-to-end to increase matching performance. Specifically, we replaced the geodesic mapping function with spatial transformations learned from data priors. We showed that although deformation-aware description improves performance, learning keypoints optimized for deformation is also equally important for true invariance to scene deformations. We show that integrating those two enhancements improves the performance of local representations on several tasks, while keeping the same input modality (RGB) and network size comparable to standard state-of-the-art local feature extraction methods. Furthermore, we show that synthetic labels can be used to train the networks and still obtain improvements in real-world data, highlighting the potential of synthetic datasets for future improvements in non-rigid image matching, where very limited data is available to train and evaluate methods.

Finally, in *Chapter* 6, we addressed a central issue in modern deep local feature architectures that is often overlooked: high inference costs. Contrary to the trend of designing large and complex neural networks for the essential task of pointwise matching, we proposed a streamlined, data-driven architecture that balances the compute-accuracy trade-off in a Pareto-optimal sense. The simple building blocks of our CNN network facilitate the deployment of the architecture across a wide range of hardware optimized for neural inference, opening new research directions for efficient and accurate image matching in real-world tasks.

7.1 Perspectives and Future Work

In this section, we present a general perspective on local feature detection and extraction and outline future research directions inspired by the achievements of this dissertation. One important aspect of the proposed *geodesic-aware* local features is that geodesics estimation rely on the quality of geometric details for accurate computation. Recently, significant progress has been made in monocular depth estimation using deep learning models [165], as well as in novel technologies such as neural radiance fields [86] and 3D Gaussian splatting [59] for novel view synthesis, alongside the recovery of accurate scene geometry from these representations. These advancements provide a rich source of 3D scene geometry that can be explored for learning geodesic-aware features. A promising research direction involves improving local feature representations without requiring depth

at inference by learning geodesic-aware representations with 3D geometry exclusively during training. This approach can leverage learned geodesic-aware priors during inference, enabling deformation-aware descriptor extraction without access to depth, thereby enhancing applicability. This strategy is fully compatible with the learned *deformation-aware* local features proposed in this dissertation, allowing the explicit incorporation of geodesic priors as additional inductive bias in the deformation-aware module.

A limitation we encountered in addressing non-rigid transformations is the scarcity of both synthetic and real training data containing non-rigid deformations. To partially address this challenge, we proposed a real benchmark and a simulation framework for training and evaluating image matching approaches. However, significant gaps remain in deformation realism and scene diversity for both synthetic and real data. We believe that realistic and diverse synthetic datasets, leveraging the latest generation of graphical engines and large-scale 3D datasets [30], can help bridge this gap for training and evaluating non-rigid correspondence methods. For real datasets, developing new weakly supervised training schemes for correspondence [116, 146], which rely only on image-level labels, represents a promising research direction. Such approaches could significantly reduce the reliance on costly annotation processes for real non-rigid scenes.

Another potential research topic is the study of the distinctiveness versus invariance properties of local features [103], aiming to better understand their interplay under various image transformations, including non-rigid deformations. For instance, the feature fusion module in DALF's architecture could be extended to incorporate dynamic feature selection mechanisms, enabling adaptive handling of non-rigid or rigid transformations selectively. This could be achieved, for example, by leveraging attention mechanisms [152] commonly employed in transformer architectures.

Learned matchers [122, 76] represent a promising complement to the methods presented in this dissertation. However, a significant research challenge in incorporating a learned matcher for non-rigid matching tasks lies in the scarcity of diverse and abundant supervision sources from real deformations. Current learned matchers for rigid scenes have demonstrated a tendency to overfit to training scenes, often necessitating pre-training on diverse datasets of synthetic homographies [76], particularly due to their reliance on data-hungry transformer-based architectures.

Another line of work, comprising end-to-end feed-forward reconstruction methods such as DUST3R [156], VGGT [154], and others, offers promising directions for obtaining reconstruction directly from deep network outputs without relying on classical geometry-based camera calibration pipelines. While these methods demonstrate strong performance in domains like indoor reconstruction of well-structured scenes, they require substantial amounts of data and computation, and often struggle to generalize to unseen training scenes.

Regarding further optimizations for inference time in local feature detection and

extraction, although XFeat achieved significant speed improvements, it is evident that XFeat could benefit from quantization methods [56] to enable highly efficient inference on embedded hardware. This represents an interesting research direction to further optimize the compute-accuracy trade-off. During our qualitative evaluations, we observed that XFeat keypoints tend to be noisier compared to more computationally intensive methods such as DISK and ALIKE [147, 176], likely due to XFeat's simple keypoint extraction head. Recent advancements in learning to refine keypoint positions [60] suggest promising avenues for improving keypoint accuracy efficiently. Additionally, there is substantial room for progress in the development of efficient learned matchers, as current methods remain computationally expensive for real-time deployment on embedded systems.

Future advancements in invariance, efficiency, and the development of novel architectures for feature extraction and learned matching remain critical research areas with the potential to drive progress across all domains relying on the foundational task of image matching.

References

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

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

[3] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, pages 5173–5182, 2017.

[4] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. Magsac++, a fast, reliable and accurate robust estimator. In *CVPR*, pages 1304–1312, 2020.

[5] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *CVIU*, 110:346–359, June 2008.

[6] Georg Bökman, Johan Edstedt, Michael Felsberg, and Fredrik Kahl. Steerers: A framework for rotation equivariant keypoint descriptors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4885–4895, 2024.

[7] Aljaz Bozic, Michael Zollhofer, Christian Theobalt, and Matthias Nießner. Deepdeform: Learning non-rigid rgb-d reconstruction with semi-supervised data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7002–7012, 2020.

[8] Gary Bradski. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000.

[9] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008.

[10] Michael M Bronstein and Iasonas Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1704–1711. IEEE, 2010.

[11] Matthew Brown, Gang Hua, and Simon Winder. Discriminative learning of local image descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):43–57, 2010.

[12] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74:59–73, 2007.

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

[14] Andrei Bursuc, Giorgos Tolias, and Hervé Jégou. Kernel local descriptors with implicit rotation matching. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 595–598, 2015.

[15] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. Brief: Computing a local binary descriptor very fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1281–1298, 2012.

[16] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 778–792. Springer, 2010.

[17] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary Robust Independent Elementary Features. In *ECCV*, September 2010.

[18] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[19] Rodrigo L Carceroni and Kiriakos N Kutulakos. Multi-view scene capture by surfel sampling: From video streams to non-rigid 3d motion, shape and reflectance. *International Journal of Computer Vision*, 49:175–214, 2002.

[20] T. Cavallari, S. Golodetz, N. A. Lord, J. Valentin, V. A. Prisacariu, L. Stefano, and P. S. Torr. Real-time rgb-d camera pose estimation in novel scenes using a relocalisation cascade. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2465–2477, oct 2020.

[21] Luca Cavalli, Viktor Larsson, Martin Ralf Oswald, Torsten Sattler, and Marc Pollefeys. Adalam: Revisiting handcrafted outlier detection. *arXiv preprint arXiv:2006.04250*, 2020.

[22] Ken Chatfield, Victor S Lempitsky, Andrea Vedaldi, and Andrew Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, volume 2, page 8, 2011.

[23] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David Mckinnon, Yanghai Tsin, and Long Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In *ECCV*, pages 20–36. Springer, 2022.

[24] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

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

[26] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Katharina Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *European Conference on Computer Vision*, 2002.

[27] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017.

[28] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[29] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[30] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv preprint arXiv:2307.05663*, 2023.

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

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

[33] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[34] Ivan Dryanovski, Roberto G Valenti, and Jizhong Xiao. Fast visual odometry and mapping from rgb-d data. In *2013 IEEE international conference on robotics and automation*, pages 2305–2310. IEEE, 2013.

[35] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features. *arXiv preprint arXiv:1905.03561*, 2019.

[36] Patrick Ebel, Anastasiia Mishchuk, Kwang Moo Yi, Pascal Fua, and Eduard Trulls. Beyond cartesian representations for local descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 253–262, 2019.

[37] Johan Edstedt, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Dedode: Detect, don't describe—describe, don't detect for local feature matching. In *2024 International Conference on 3D Vision (3DV)*, pages 148–157. IEEE, 2024.

[38] Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[39] William T Freeman, Edward H Adelson, et al. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9):891–906, 1991.

[40] Pierre Gleize, Weiyao Wang, and Matt Feiszli. Silk: Simple learned keypoints. In *ICCV*, pages 22499–22508, October 2023.

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

[42] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[43] Maciej Halber, Yifei Shi, Kai Xu, and Thomas Funkhouser. Rescan: Inductive instance segmentation for indoor rgbd scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

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

[45] Christopher Harris and Mike Stephens. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference*, pages 23.1–23.6, 1988.

[46] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[48] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[49] Judy Hoffman, Saurabh Gupta, and Trevor Darrell. Learning with side information through modality hallucination. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[50] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*, 2017.

[51] Matthias Innmann, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. In *European conference on computer vision*, pages 362–379. Springer, 2016.

[52] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *arXiv preprint arXiv:1506.02025*, 2015.

[53] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, 2015.

[54] 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. *International Journal of Computer Vision*, pages 1–31, 2020.

[55] Andrew E Johnson and Martial Hebert. Efficient multiple model recognition in cluttered 3-d scenes. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*, pages 671–677. IEEE, 1998.

[56] Menelaos Kanakis, Simon Maurer, Matteo Spallanzani, Ajad Chhatkuli, and Luc Van Gool. Zippypoint: Fast interest point detection, description, and matching through mixed precision discretization. In *CVPRW*, pages 6113–6122, 2023.

[57] Asako Kanezaki, Zoltan-Csaba Marton, Dejan Pangercic, Tatsuya Harada, Yasuo Kuniyoshi, and Michael Beetz. Voxelized Shape and Color Histograms for RGB-D. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop on Active Semantic Perception and Object Search in the Real World*, San Francisco, CA, USA, September, 25–30 2011.

[58] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[59] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[60] Shinjeong Kim, Marc Pollefeys, and Daniel Barath. Learning to make keypoints sub-pixel accurate. In *The European Conference on Computer Vision (ECCV)*, 2024.

[61] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[62] Jan J Koenderink and Andrea J van Doorn. Representation of local geometry in the visual system. *Biological cybernetics*, 55(6):367–375, 1987.

[63] Iasonas Kokkinos, Michael M. Bronstein, Roee Litman, and Alex M. Bronstein. Intrinsic shape context descriptors for deformable shapes. In *CVPR*, pages 159–166, June 2012.

[64] Axel Barroso Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Key. net: Keypoint detection by handcrafted and learned cnn filters. *arXiv preprint arXiv:1904.00889*, 2019.

[65] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view RGB-D object dataset. In *Proc. ICRA*, May 2011.

[66] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Sparse distance learning for object recognition combining RGB and depth information. In *Proc. ICRA*, 2011.

[67] Mans Larsson, Erik Stenborg, Lars Hammarstrand, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. A cross-season correspondence dataset for robust semantic

segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9532–9542, 2019.

[68] Viktor Larsson and contributors. PoseLib - Minimal Solvers for Camera Pose Estimation, 2020.

[69] Vincent Lepetit and Pascal Fua. Keypoint recognition using randomized trees. *IEEE transactions on pattern analysis and machine intelligence*, 28(9):1465–1479, 2006.

[70] Stefan Leutenegger, Margarita Chli, and Roland Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. In *ICCV*, 2011.

[71] Stefan Leutenegger, Margarita Chli, and Roland Siegwart. BRISK: Binary robust invariant scalable keypoints. In *ICCV*, 2011.

[72] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050, 2018.

[73] Di Lin and Hui Huang. Zig-zag network for semantic segmentation of rgb-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 2019*, 42(10):2642–2655, 2020.

[74] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017.

[75] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

[76] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17627–17638, 2023.

[77] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[78] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, pages 91–110, 2004.

[79] Zixin Luo, Tianwei Shen, Lei Zhou, Siyu Zhu, Runze Zhang, Yao Yao, Tian Fang, and Long Quan. Geodesc: Learning local descriptors by integrating geometry constraints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 168–183, 2018.

[80] 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*, pages 6589–6598, 2020.

[81] Lena Maier-Hein, Peter Mountney, Adrien Bartoli, Haytham Elhawary, D Elson, Anja Groch, Andreas Kolb, Marcos Rodrigues, J Sorger, Stefanie Speidel, et al. Optical techniques for 3d surface reconstruction in computer-assisted laparoscopic surgery. *Medical image analysis*, 17(8):974–996, 2013.

[82] Renato Martins, Eduardo Fernandez-Moral, and Patrick Rives. Adaptive direct rgb-d registration and mapping for large motions. In *Asian Conference on Computer Vision (ACCV)*, 2016.

[83] 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 *SIBGRAPI*, volume 1, pages 61–66. IEEE, 2022.

[84] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60:63–86, 2004.

[85] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *International journal of computer vision*, 65(1-2):43–72, 2005.

[86] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[87] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems*, pages 4826–4837, 2017.

[88] Yoshikatsu Nakajima, Byeongkeun Kang, Hideo Saito, and Kris Kitani. Incremental class discovery for semantic segmentation with rgbd sensing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

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

[90] Erickson R Nascimento, G Oliveira, M Campos, and A Vieira. Improving object detection and recognition for semantic mapping with an extended intensity and

shape based descriptor. In *IEEE IROS Workshop on Active Semantic Perception*, 2011.

[91] Erickson R. Nascimento, Gabriel L. Oliveira, Mario F. M. Campos, Antônio W. Vieira, and William Robson Schwartz. BRAND: A Robust Appearance and Depth Descriptor for RGB-D Images. In *Proc. IROS*, 2012.

[92] Erickson R. Nascimento, Gabriel L. Oliveira, Antônio W. Vieira, and Mario F. M. Campos. On the development of a robust, fast and lightweight keypoint descriptor. *Neurocomputing*, 120(0):141–155, 2013.

[93] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10004–10012, 2019.

[94] Erickson R. Nascimento, William Robson Schwartz, and Mario F. M. Campos. EDVD - enhanced descriptor for visual and depth data. In *IAPR International Conference on Pattern Recognition (ICPR)*, 2012.

[95] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 343–352, 2015.

[96] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2161–2168. Ieee, 2006.

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

[98] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51 – 59, 1996.

[99] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: learning local features from images. In *Advances in neural information processing systems*, pages 6234–6244, 2018.

[100] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra,

Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[101] Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for k-medoids clustering. *Expert systems with applications*, 36(2):3336–3341, 2009.

[102] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NeurIPS Workshops*, 2017.

[103] Rémi Pautrat, Viktor Larsson, Martin R Oswald, and Marc Pollefeys. Online invariance selection for local feature descriptors. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 707–724. Springer, 2020.

[104] Nathan Piasco, Désiré Sidibé, Valérie Gouet-Brunet, and Cédric Demonceaux. Learning scene geometry for visual localization in challenging conditions. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[105] Guilherme Potje, Felipe Cadar, André Araujo, Renato Martins, and Erickson R Nascimento. Enhancing deformable local features by jointly learning to detect and describe keypoints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1306–1315, 2023.

[106] Guilherme Potje, Felipe Cadar, André Araujo, Renato Martins, and Erickson R Nascimento. Xfeat: Accelerated features for lightweight image matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2682–2691, 2024.

[107] 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.

[108] Guilherme Potje, Renato Martins, Felipe Chamone, and Erickson Nascimento. Extracting deformation-aware local features by learning to deform. *Advances in Neural Information Processing Systems*, 34:10759–10771, 2021.

[109] Guilherme Potje, Gabriel Resende, Mario Campos, and Erickson R Nascimento. Towards an efficient 3d model estimation methodology for aerial and ground images. *Machine Vision and Applications*, 28:937–952, 2017.

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

[111] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[112] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020.

[113] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2D2: Reliable and repeatable detector and descriptor. In *Advances in Neural Information Processing Systems*, volume 32, pages 12405–12415, 2019.

[114] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[115] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. End-to-end weakly-supervised semantic alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[116] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Ncnet: Neighbourhood consensus networks for estimating image correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2):1020–1034, 2020.

[117] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[118] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):105–119, 2008.

[119] Peter J Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984.

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

[121] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, pages 12716–12725, 2019.

[122] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, pages 4938–4947, 2020.

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

[124] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8601–8610, 2018.

[125] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *International Journal of computer vision*, 37(2):151–172, 2000.

[126] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[127] Gil Shamai and Ron Kimmel. Geodesic distance descriptors. In *CVPR*, pages 3624–3632, July 2017.

[128] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012.

[129] Edgar Simo-Serra, Carme Torras, and Francesc Moreno-Noguer. DaLI: deformation and light invariant descriptor. *International Journal of Computer Vision*, 115(2), 2015.

[130] 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*, 2015.

[131] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[132] Noah Snavely, Steven Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. acm trans graph 25(3):835-846. *ACM Trans. Graph.*, 25:835–846, 07 2006.

[133] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116. Citeseer, 2007.

[134] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, pages 8922–8931, 2021.

[135] Vitaly Surazhsky, Tatiana Surazhsky, Danil Kirsanov, Steven J Gortler, and Hugues Hoppe. Fast exact and approximate geodesics on meshes. In *ACM Trans. Graph. (TOG)*, volume 24, pages 553–560. Acm, 2005.

[136] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[137] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.

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

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

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

[141] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11016–11025, 2019.

[142] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830, 2010.

[143] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique Signatures of Histograms for Local Surface Description. In *ECCV*, 2010.

[144] Federico Tombari, Samuele Salti, and Luigi Di Stefano. A combined texture-shape descriptor for enhanced 3D feature matching. In *ICIP*, 2011.

[145] 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.

[146] Prune Truong, Martin Danelljan, Fisher Yu, and Luc Van Gool. Warp consistency for unsupervised learning of dense correspondences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10346–10356, 2021.

[147] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *Advances in Neural Information Processing Systems*, 33:14254–14265, 2020.

[148] Luc Van Gool, Theo Moons, and Dorin Ungureanu. Affine/photometric invariants for planar intensity patterns. In *European Conference on Computer Vision*, pages 642–651. Springer, 1996.

[149] Manik Varma and Debajyoti Ray. Learning the discriminative power-invariance trade-off. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[150] Levi O. Vasconcelos, Erickson R. Nascimento, and Mario F. M. Campos. KVD: Scale invariant keypoints by combining visual and depth data. *Pattern Recognition Letters*, 86:83 – 89, 2017.

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

[152] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[153] Andrea Vedaldi and Brian Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1469–1472, 2010.

[154] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. *arXiv preprint arXiv:2503.11651*, 2025.

[155] Lizhen Wang, Xiaochen Zhao, Tao Yu, Songtao Wang, and Yebin Liu. Normalgan: Learning detailed 3d human from a single rgb-d image. In *Computer Vision – ECCV 2020*, pages 430–446. Springer International Publishing, 2020.

[156] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

[157] Tao Wang, Haibin Ling, Congyan Lang, Songhe Feng, and Xiaohui Hou. Deformable surface tracking by graph matching. In *IEEE International Conference on Computer Vision*, 2019.

[158] Yiqun Wang, Jianwei Guo, Dong-Ming Yan, Kai Wang, and Xiaopeng Zhang. A robust local spectral descriptor for matching non-rigid shapes with incompatible shape structures. In *CVPR*, 2019.

[159] Maurice Weiler and Gabriele Cesa. General e(2)-equivariant steerable cnns. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

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

[161] Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. *arXiv preprint arXiv:2311.06242*, 2023.

[162] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE international conference on computer vision*, pages 1625–1632, 2013.

[163] L. Xu, Z. Su, L. Han, T. Yu, Y. Liu, and L. Fang. Unstructuredfusion: Realtime 4d geometry and texture reconstruction using commercial rgbd cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2508–2522, 2020.

[164] Jun Yang, Yu-Gang Jiang, Alexander G Hauptmann, and Chong-Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 197–206, 2007.

[165] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024.

[166] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned invariant feature transform. In *ECCV*, 2016.

[167] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *European Conference on Computer Vision*, pages 467–483. Springer, 2016.

[168] Qinghua Yu, Jie Liang, Junhao Xiao, Huimin Lu, and Zhiqiang Zheng. A novel perspective invariant feature transform for rgb-d images. *Computer Vision and Image Understanding*, 167:109 – 120, 2018.

[169] Andrei Zaharescu, Edmond Boyer, Kiran Varanasi, and Radu P. Horaud. Surface Feature Detection and Description with Applications to Mesh Matching. In *CVPR*, Miami Beach, Florida, June 2009.

[170] Julio Zaragoza, Tat-Jun Chin, Michael S Brown, and David Suter. As-projective-as-possible image stitching with moving dlt. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2339–2346, 2013.

[171] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: Learning local geometric descriptors from rgb-d reconstructions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 199–208, 2017.

[172] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[173] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, pages 6848–6856, 2018.

[174] Qiang Zhao, Wei Feng, Liang Wan, and Jiawan Zhang. Sphorb: A fast and robust binary feature on the sphere. *International journal of computer vision*, 113(2):143–159, 2015.

[175] Xiaoming Zhao, Xingming Wu, Weihai Chen, Peter CY Chen, Qingsong Xu, and Zhengguo Li. Aliked: A lighter keypoint and descriptor extraction network via deformable transformation. *IEEE Transactions on Instrumentation and Measurement*, 72:1–16, 2023.

[176] Xiaoming Zhao, Xingming Wu, Jinyu Miao, Weihai Chen, Peter CY Chen, and Zhengguo Li. Alike: Accurate and lightweight keypoint detection and descriptor extraction. *IEEE TMM*, 2022.

[177] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. Patch2pix: Epipolar-guided pixel-level correspondences. In *CVPR*, pages 4669–4678, 2021.