UNIVERSIDADE FEDERAL DE GOIÁS
INSITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

NIGEL JOSEPH BANDEIRA DIAS

Spatio-Temporal Affine Regression for Feature Tracking

Goiânia
2026

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO (TECA) PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do material bibliográfico

[ ] Dissertação [X] Tese [ ] Outro*:_____

*No caso de mestrado/doutorado profissional, indique o formato do Trabalho de Conclusão de Curso, permitido no documento de área, correspondente ao programa de pós-graduação, orientado pela legislação vigente da CAPES.

Exemplos: Estudo de caso ou Revisão sistemática ou outros formatos.

2. Nome completo do autor

Nigel Joseph Bandeira Dias

3. Título do trabalho

Spatio-Temporal Affine Regression for Feature Tracking

4. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante:
a) consulta ao(à) autor(a) e ao(à) orientador(a);
b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação.
O documento não será disponibilizado durante o período de embargo.
Casos de embargo:
- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

Obs. Este termo deverá ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Nigel Joseph Bandeira Dias**, **Discente**, em 06/05/2026, às 11:34, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Ronaldo Martins Da Costa**, **Professor do Magistério Superior**, em 08/05/2026, às 07:52, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **6176809** e o código CRC **69A7C0DF**.

Referência: Processo nº 23070.000774/2026-50 SEI nº 6176809

NIGEL JOSEPH BANDEIRA DIAS

Spatio-Temporal Affine Regression for Feature Tracking

Tese apresentada ao Programa de Pós–Graduação do Insituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Área de concentração: Ciência da Computação.

Linha de pesquisa: Sistemas Inteligentes e Aplicações.

Orientador: Prof. Dr. Ronaldo Martins da Costa

Co-Orientador: Prof. Dr. Gustavo Teodoro Laureano

Goiânia
2026

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Dias, Nigel Joseph Bandeira
Spatio-Temporal Affine Regression for Feature Tracking [manuscrito] = Regressão Afim Espaço-Temporal para Rastreamento de Características / Nigel Joseph Bandeira Dias. - 2026.
75 f.: il. 2026

Orientador: Prof. Dr. Ronaldo Martins da Costa; co-orientador: Dr. Gustavo Teodoro Laureano
Tese (Doutorado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação, Goiânia, 2026.
Ilustrações.
Bibliografia.
Inclui: tabelas, lista de figuras, lista de tabelas.

1. Feature. 2. Tracking. 3. Spatio-temporal. 4. Deep Neural Networks. 5. Deep Learning.

I. Costa, Ronaldo Martins da, orient. II. Laureano, Gustavo Teodoro, co-orient. III. Título.

CDU 004

NIGEL JOSEPH BANDEIRA DIAS

Spatio-Temporal Affine Regression for Feature Tracking

Tese defendida no Programa de Pós–Graduação do Insituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Doutor em Ciência da Computação, aprovada em 16 de Fevereiro de 2026, pela Banca Examinadora constituída pelos professores:

Prof. Dr. Ronaldo Martins da Costa
Insituto de Informática – UFG
Presidente da Banca

Prof. Dr. Gustavo Teodoro Laureano
Insituto de Informática – UFG

Prof. Dr. Fernando Santos Osório
Instituto de Ciências Matemáticas e de Computação – USP São Carlos

---

Prof. Dr. Aldo André Díaz Salazar
Instituto de Informática – UFG

Prof. Fabrízzio Alphonsus Alves de Melo Nunes Soares
Instituto de Informática – UFG

Prof. Anderson da Silva Soares
Instituto de Informática – UFG

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

ATA DE DEFESA DE TESE

Ata nº **51** da sessão de Defesa de Tese de **Nigel Joseph Bandeira Dias**, que confere o título de Doutor em Ciência da Computação, na área de concentração em Ciência da Computação.

Aos dez dias do mês de março de dois mil e vinte e seis, a partir das catorze horas, na sala 250 do INF, realizou-se a sessão pública de Defesa de Tese intitulada **"Spatio-Temporal Affine Regression for Feature Tracking"**. Os trabalhos foram instalados pelo Orientador, Professor Doutor Ronaldo Martins da Costa (INF/UFG) com a participação dos demais membros da Banca Examinadora: Professor Doutor Gustavo Teodoro Laureano (INF/UFG), coorientador; Professor Doutor Fernando Santos Osório (USP São Carlos), membro titular externo; Professor Doutor Fabrízzio Alphonsus Alves de Melo Nunes Soares (INF/UFG), membro titular interno; Professor Doutor Anderson da Silva Soares (INF/UFG), membro titular interno; e Professor Doutor Aldo André Diaz Salazar (INF/UFG), membro titular externo. A participação do professor Fernando Santos Osório ocorreu por meio de videoconferência. Durante a arguição os membros da banca não fizeram sugestão de alteração do título do trabalho. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Tese, tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pelo Professor Doutor Ronaldo Martins da Costa, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos dez dias do mês de março de dois mil e vinte e seis.
TÍTULO SUGERIDO PELA BANCA

Documento assinado eletronicamente por **Fabrizzio Alphonsus Alves De Melo Nunes Soares**, **Professor do Magistério Superior**, em 10/03/2026, às 17:53, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Gustavo Teodoro Laureano**, **Professor do Magistério Superior**, em 10/03/2026, às 17:53, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Ronaldo Martins Da Costa**, **Professor do Magistério Superior**, em 10/03/2026, às 17:53, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Anderson Da Silva Soares**, **Professor do Magistério Superior**, em 10/03/2026, às 17:54, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Fernando Santos Osório**, **Usuário Externo**, em 10/03/2026, às 17:54, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Aldo Andre Diaz Salazar**, **Professor do Magistério Superior**, em 10/03/2026, às 17:54, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Nigel Joseph Bandeira Dias**, **Discente**, em 11/03/2026, às 10:20, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **6024931** e o código CRC **7A0B7974**.

Referência: Processo nº 23070.000774/2026-50 SEI nº 6024931

Abstract

Dias, Nigel. **Spatio-Temporal Affine Regression for Feature Tracking**. Goiânia, 2026. 71p. PhD. Thesis. Programa de Pós-Graduação em Ciência da Computação, Insituto de Informática, Universidade Federal de Goiás.

Feature association is a fundamental prerequisite for visual localization pipelines. Typically, these methods rely on feature matching to estimate relative motion based on projective geometric constraints. Despite significant advances in feature association, most existing techniques rely on pairwise matching paradigms and often neglect the rich temporal context inherent in image sequences. In this thesis, we revisit the canonical Kanade-Lucas-Tomasi (KLT) feature tracker. We reformulate this classic algorithm by integrating deep neural network mechanisms for spatiotemporal and geometric learning. The proposed methodology uses a convolutional neural network trained to regress affine transformation parameters across consecutive frame patches. This capability allows for precise tracking of interest points considering the consistency of the projective geometry. To perform the training of the proposed network, we introduce a versatile protocol to synthesize feature tracking annotations from already available datasets. This methodology leverages state-of-the-art feature extraction and matching along with a model selection criterion based on epipolar geometry. Experimental evaluations on the TUM RGB-D benchmark demonstrate the consistent superiority of the proposed method in estimating relative camera motion compared to KLT and the Pips++ method. Although our method exhibits a lower inlier ratio, the resulting correspondence subset possesses significantly higher geometric fidelity. These results establish the proposed method as a robust solution, suitable for deployment in embedded systems with limited resources.

Keywords

Feature, Tracking, Spatio-temporal, Deep Neural Networks.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Local feature tracking in images, often referred to as point tracking, is a fundamental task in several computer vision applications. Systems that require 3D scene information, such as the underlying geometric structure or camera trajectory, typically rely on point-tracking methods to estimate these parameters. By identifying the position of the same point across multiple viewpoints, the system can constrain the possible solutions to a rigid geometric transformation model that accurately describes the observed data [Hartley e Zisserman 2004].

Notable applications of point tracking include visual self-localization for mobile robots [Sellak, Alj e Salih-Alj 2024, Al-Tawil et al. 2024], eye in hand fixed base robotic systems [Klingensmith, Sirinivasa e Kaess 2016], and the generation of orthomosaics from aerial imagery [Gómez-Reyes et al. 2022]. Furthermore, this technique is crucial for the digital reconstruction of tissues using medical imaging [Schmidt et al. 2024], digital video stabilization [Sarıgül 2023], 3D environment reconstruction, augmented reality position tracking [Baker et al. 2024], and space exploration [Andolfo, Petricca e Genova 2022]. These diverse applications share a core processing pipeline grounded in the fundamentals of Structure from Motion (SfM), Visual Odometry (VO), and Simultaneous Localization and Mapping (SLAM). Solving the central challenges in these domains requires precise knowledge of the 3D arrangement of points scattered throughout the observed scene.

In fields like Structure from Motion (SfM), achieving a high-quality 3D reconstruction relies heavily on datasets that accurately describe the details and shapes of the target objects. A technically sound dataset must be dense, featuring numerous 3D points with significant spatial diversity and high positional accuracy. Similarly, visual self-localization and environment mapping in mobile robotics require this same foundational information but can generally tolerate sparser point distributions. Furthermore, vision-based localization techniques are fundamentally grounded in the geometric modeling of the world, where algorithms attempt to infer these models directly from observed data [Hartley e Zisserman 2004]. Consequently, the overall performance of these systems depends critically on the feature association step. This crucial step enables the estima-

tion of relative motion between consecutive frames through the principles of projective geometry [Fraundorfer e Scaramuzza 2012]

Local features, typically extracted from high-contrast regions such as mountain peaks, building corners, and doorways, are often referred to as keypoints or interest points. They are preferred in a wide variety of applications because they can be matched even in the presence of occlusion, scale, and orientation changes. These points are often characterized by the pixel patches surrounding their locations. These regions are then converted into a more compact, stable vector called a descriptor, which can be matched against other descriptors [Szeliski 2022].

Despite significant advances in feature association, particularly with deep neural networks, most existing methods focus on pairwise feature-matching approaches. In general, these methods rely on sparse interest points, matched using high-dimensional representations that encode their local visual appearance. Such representations often come at the cost of high computational requirements and increased implementation complexity [Sarlin et al. 2020, Potje et al. 2024]. Likewise, as they rely on pairwise matching, they typically overlook the rich temporal context in the image sequence.

The tracking-based approach serves as an alternative to the feature matching paradigm. While feature matching requires exhaustive extraction and computationally intensive association for every frame pair, the tracking method relies on detecting features in an initial frame and subsequently propagating them across the temporal sequence. This strategy significantly reduces computational overhead by eliminating redundant extraction and matching steps. Therefore, it is highly advantageous for real-time applications where processing efficiency is crucial.

Recent advancements in point tracking have been largely driven by deep learning methodologies. However, existing frameworks [Harley, Fang e Fragkiadaki 2022, Doersch et al. 2023] primarily target long-term temporal associations. While this focus enables the recovery of occluded points, it often results in high computational latency. Furthermore, these approaches typically lack strict geometric consistency. This omission limits their applicability to geometry-dependent tasks such as Structure-from-Motion (SfM).

When consecutive frames exhibit large visual overlap and limited appearance change, a window can be tracked by optimizing a matching criterion over a constrained range of transformations [Shi e Tomasi 1994]. This strategy is particularly suitable for applications like Visual Odometry (VO) and Visual SLAM (V-SLAM). In these contexts, the motion and appearance deformation between adjacent frames are typically small.

In this work, we revisit the well-known Kanade-Lucas-Tomasi (KLT) [Lucas e Kanade 1981, Shi e Tomasi 1994] feature tracker and reformulate this classical approach using deep neural network concepts for spatio-temporal and geometric

learning. The proposed model is a fully convolutional neural network that learns to predict the affine transformation parameters between patches from consecutive frames, enabling precise tracking of interest points. The model comprises three primary components: a patch-based Spatio-temporal Feature Extractor [Carreira e Zisserman 2017], a Cost Volume Block [Teed e Deng 2020], and a Regression Block [Jaderberg et al. 2015] designed to estimate the affine parameters for each local region.

We hypothesize that an end-to-end fully convolutional neural network reformulation of the classical gradient-based KLT feature tracker will provide more precise and robust affine transformation estimates between consecutive frames than traditional mathematical methods. Traditional KLT relies on raw pixel intensities and local spatial gradients that often fail under drastic lighting changes or motion blur. In contrast, a neural network can learn deep high-level spatial and temporal features that are highly robust to illumination changes and noise, providing a much stronger foundation for tracking. Additionally, rather than solving a complex system of linear equations, a specialized regression network can directly map the relationships in the cost volume to the exact geometric affine parameters for any local patch.

This work aims to contribute to the development of an alternative approach to traditional point of interest tracking methods, capable of increasing the geometric consistency of associations between points in consecutive frames. To achieve this, the present work introduces a novel pipeline for generating geometrically consistent tracking datasets alongside a deep neural network architecture. This network aims to estimate local point displacements by learning the underlying epipolar geometric relationships between successive images.

Experimental evaluations on the TUM RGB-D benchmark [Sturm et al. 2012] demonstrate the consistent superiority of the proposed method in estimating relative camera motion. Our approach outperforms both the KLT baseline [Bouguet 1999] and Pips++ [Zheng et al. 2023] across the majority of the Freiburg 3 sequences. Although our method consistently exhibits the lowest inlier ratio, the resulting subset of correspondences possesses higher geometric fidelity.

We also attribute this performance enhancement to the rigorous curation of the training dataset. We propose a flexible protocol that generates feature-tracking annotations from any real-world dataset for VO and V-SLAM tasks. This methodology leverages state-of-the-art feature extraction and matching techniques, in conjunction with a model-selection criterion based on epipolar geometry. Consequently, the process yields a set of tracking states where points remain geometrically consistent with the camera trajectory.

In summary, the primary contributions of this research are as follows:

- We introduce a flexible framework for generating feature-tracking labels from any real-world dataset for VO and V-SLAM tasks.

- We propose a computationally efficient Spatio-temporal Affine Regression model designed for robust feature tracking.

A preliminary version of this research was published in [Dias, Laureano e Costa 2026], while an updated version has been submitted to the 23rd Conference on Robots and Vision (CRV 2026), which is sponsored by the Canadian Image Processing and Pattern Recognition Society (CIPPRS).

The remainder of this thesis is organized as follows. Chapter 2 reviews foundational and contemporary literature on feature association and long-term point tracking relevant to this study. Chapter 3 covers the fundamental principles of feature tracking and epipolar geometry. Chapter 4 details both the dataset generation protocol and the proposed tracking architecture. Subsequently, Chapter 5 presents the experimental setup, quantitative evaluations, and a discussion of the results. Finally, Chapter 6 provides concluding remarks and summarizes the research findings.

CHAPTER 2

Related works

Feature-based correspondence techniques have been studied since the foundational research in stereo matching and have subsequently become central to image stitching, vision-based localization, and mapping [Szeliski 2022]. Broadly, two primary strategies exist for establishing these correspondences: detecting features in one image and tracking them via local search, or detecting features independently in each image and matching them based on a local appearance descriptor.

A more comprehensive description of alternative techniques can be found in in [Balntas et al. 2017, Jin et al. 2020]. However, in this chapter, we present fundamental and contemporary works on feature association and long-term point tracking that are most relevant to the method proposed in this thesis.

For applications such as video sequences or rectified stereo pairs, local motion around feature points may be predominantly translational. Consequently, simple error metrics, such as the Sum of Squared Differences (SSD) or Normalized Cross-Correlation (NCC), can be used to directly compare intensities in small patches around each feature [Lucas e Kanade 1981]. However, many scenarios involve significant changes in orientation, scale, or even affine deformations. Therefore, to match features across these challenging conditions, descriptors must be invariant to such transformations [Szeliski 2022].

The field of feature descriptors remains a highly active area of research. Among classical approaches, GLOH [Mikolajczyk e Schmid 2005] has demonstrated superior performance, followed closely by SIFT [Lowe 2004]. However, while these descriptors are highly discriminative, they are relatively computationally intensive to extract and match, which can be a significant drawback for real-time applications. To achieve a better trade-off between accuracy and efficiency, several works [Calonder et al. 2010, Rublee et al. 2011, Leutenegger, Chli e Siegwart 2011] have proposed binary descriptors that leverage fast Hamming distance operators available in modern computer architectures. More recently, the focus has shifted toward learning-based descriptors [Jin et al. 2020]. Some of these methods [Yi et al. 2016, Balntas et al. 2017, Barroso-Laguna et al. 2019] operate on lo-

cal patches, much like the classical SIFT approach. In contrast, architectures such as [DeTone, Malisiewicz e Rabinovich 2018, Dusmanu et al. 2019, Zhao et al. 2022] process the entire image to compute dense descriptor vectors.

Assuming descriptors are designed so that Euclidean distances in feature space effectively rank potential matches, the simplest matching strategy is to set a distance threshold and retrieve all candidates from other images within this range. The problem is that a fixed threshold is difficult to set, therefore, a better strategy is to match the nearest neighbor in feature space. In this case, a commonly used heuristic is the *Nearest Neighbor Distance Ratio* (NNDR) [Mikolajczyk e Schmid 2005], which compares the nearest neighbor distance to that of the second-nearest neighbor. Furthermore, efficient search methods are required to avoid comparing all features against all others. A widely used approach utilizes indexing structures, such as multi-dimensional search trees, to rapidly search for features near a given location [Muja e Lowe 2014]. Due to unmatchable keypoints and imperfect descriptors, some correspondences may be incorrect. These are typically rejected using heuristics, such as Lowe's ratio test [Lowe 2004], followed by robust geometric verification methods, such as RANSAC [Fischler e Bolles 1981].

More recently, several works have proposed deep networks trained to jointly match local features and reject outliers given an input image pair. SuperGlue [Sarlin et al. 2020] was a pioneering work in learning-based local feature matching. This approach takes two sets of interest points and their descriptors as input and predicts matches using a Graph Neural Network (GNN), combining the expressive representations of Transformers [Vaswani et al. 2023] with optimal transport [Peyré e Cuturi 2020] to solve a partial assignment problem. Conversely, methods like LoFTR [Sun et al. 2021] propose dense approaches that match points distributed on dense grids rather than sparse locations, boosting performance in textureless regions.

Despite their remarkable robustness and accuracy in wide-baseline scenarios, these transformer-based architectures are computationally expensive to train, and their complexity grows quadratically with the number of keypoints. Consequently, recent research has focused on efficiency. In [Lindenberger, Sarlin e Pollefeys 2023], the authors revisit SuperGlue's design to improve both memory and computational efficiency. They propose an adaptive depth and width mechanism that reduces the layer count based on image difficulty and prunes confidently rejected points early, thereby saving inference time. In contrast to the prevalent transformer trend, [Potje et al. 2024] propose a lightweight Convolutional Neural Network (CNN) architecture for accelerated feature extraction, applicable to both sparse and semi-dense matching.

As an alternative to extracting and matching descriptors across all images, one can identify salient features in the first image and search for their corresponding locations in subsequent frames. This *detect-then-track* approach is widely used in video

applications, where motion and appearance deformations between adjacent frames are expected to be small.

Feature tracking is classically formulated as a localized image registration problem [Lucas e Kanade 1981], relying on the brightness constancy constraint to minimize intensity differences between frames. The standard Kanade-Lucas-Tomasi (KLT) tracker [Shi e Tomasi 1994] extends this by selecting features with high spatial gradients and employing a dual-model strategy: pure translation is used for robust frame-to-frame tracking, while an affine motion model is applied to monitor feature quality and detect drift. Since their original work, Shi and Tomasi's approach has generated several follow-on papers. Tommasini et al. [Tommasini et al. 1998] extended the tracker by introducing an automatic scheme for rejecting spurious features. Bouguet [Bouguet 1999] proposed a pyramidal version for hierarchical motion estimation to handle large displacements. Furthermore, Collins [Collins e Liu 2003] proposed an improved mechanism for feature selection and for handling larger appearance changes. Despite the rise of deep learning-based approaches, the pyramidal version of the KLT tracker remains a standard approach for feature association, particularly for computationally constrained platforms [Qin, Li e Shen 2018, Zheng et al. 2024].

While feature-based tracking remains widely used in real-time applications such as SLAM, autonomous navigation, and augmented reality, significant research effort has shifted toward dense motion estimation methods, such as optical flow [Teed e Deng 2020]. Whereas feature tracking maintains a sparse set of salient points over long durations, optical flow typically estimates a dense motion field between consecutive frames. A third research direction explores semi-dense strategies that combine these paradigms to produce motion estimates that are both spatially dense and temporally long-range [Sand e Teller 2006]

A more recent paradigm formulates pixel tracking as a long-range motion estimation problem, where arbitrary pixels are associated with trajectories across multiple future frames [Harley, Fang e Fragkiadaki 2022, Doersch et al. 2023, Karaev et al. 2023]. These methods typically employ Transformer-based architectures, leveraging modern techniques such as dense cost maps, iterative optimization, and learned appearance updates. However, similar to transformer-based feature matching, these models are often difficult to train, and their computational complexity tends to grow with the number of tracked points.

In summary, the evolution of feature tracking algorithms reveals a clear divide between historical efficiency and modern robustness. As shown in 2.1, classical methods advanced from basic intensity comparisons to efficient descriptors, with tools like the KLT tracker remaining essential for real-time systems due to their low computational overhead. Nevertheless, these traditional techniques often fail under extreme deformations and

complex environmental changes. On the other hand, the rise of deep learning paradigms, detailed in 2.2, has introduced transformer-based matchers and long-range motion estimators that achieve unprecedented accuracy. However, their significant computational demands restrict their use in highly dynamic or resource-constrained environments. This dichotomy highlights a critical gap, emphasizing the need to revisit traditional tracking methods using modern deep learning techniques. Future research should aim to merge the efficiency of classical architectures with the robust data-driven representations of neural networks. By designing lightweight machine learning trackers, the field can attain the high accuracy of modern artificial intelligence while meeting the strict performance demands of SLAM, autonomous navigation, and mobile robotics.

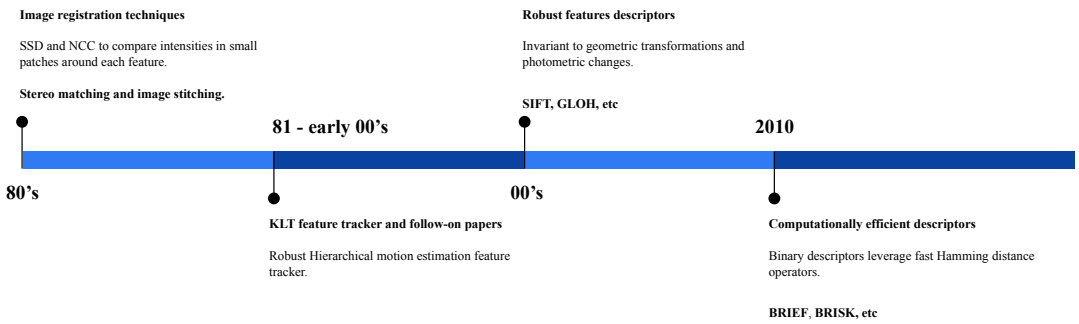


Figure 2.1: *Timeline detailing the advancement of feature matching and tracking algorithms. Early decades focused on image registration using intensity comparisons (SSD, NCC) and hierarchical motion estimation (KLT tracker). The 2000s marked a shift toward robust descriptors invariant to geometric and photometric changes, such as SIFT. By the 2010s, the focus transitioned to computational efficiency, leveraging fast Hamming distance operators with binary descriptors like BRIEF and BRISK.*

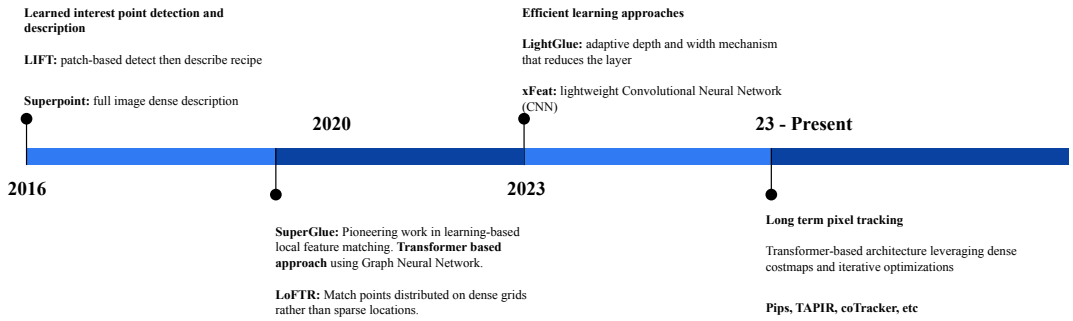


Figure 2.2: *Timeline highlighting the advancement of learning-based approaches in feature matching and tracking. The progression begins in 2016 with learned interest point detection and description. By 2020, the focus shifted to transformer-based feature matching, pioneered by architectures like SuperGlue and LoFTR. From 2023 onward, the field emphasizes computational efficiency (e.g., LightGlue, xFeat) alongside sophisticated, long-term pixel tracking techniques using dense costmaps and iterative optimizations (e.g., TAPIR, coTracker).*

CHAPTER 3

Background

This chapter establishes the fundamental theoretical concepts required to understand visual motion estimation and robust point tracking. The discussion begins by exploring feature-based motion estimation, detailing how distinct local patches are identified and subsequently tracked across consecutive image frames using both translational and advanced parametric models. However, because raw feature correspondences are inherently susceptible to noise and tracking errors, the second half of this chapter introduces the crucial concept of geometric verification. By leveraging the principles of epipolar geometry and the essential matrix, strict algebraic constraints are established between multiple views. Finally, robust estimation algorithms such as RANSAC are detailed as the standard method for isolating true inlier correspondences, thereby ensuring the generation of highly accurate and geometrically consistent motion trajectories.

3.1 Feature-Based Motion Estimation

Feature-based motion estimation is a fundamental process that involves identifying and tracking distinct local regions across consecutive image frames. The theoretical foundation of this process rests upon three essential pillars. The robust identification of points and patches is vital to ensure precise spatial localization and establish stable areas to track. Following this localization, translational alignment provides the mathematical basis for calculating foundational pixel displacements. To achieve true robustness, advanced parametric models are required to handle the geometric deformations caused by complex camera dynamics, such as rotation and scaling. United, these foundational elements create a framework capable of tracking accurate feature trajectories across highly dynamic environments.

3.1.1 Points and Patches

Localized features, such as mountain peaks, building corners, and doorways, are often referred to as keypoints or interest points. They are preferred in a wide variety of

applications because they can be matched more precisely. As observed in Figure 3.1, textureless regions are difficult to localize, whereas patches with significant contrast variations are easily distinguishable.

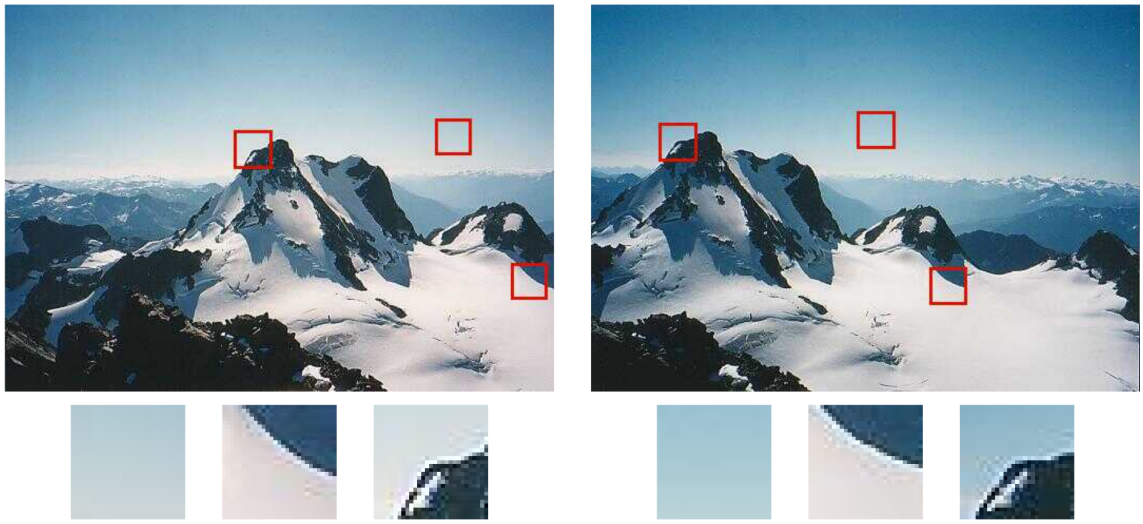


Figure 3.1: *The figure below displays image pairs alongside their extracted patches. It is evident that certain patches allow for more precise localization or matching than others [Szeliski 2022].*

Mathematically, these intuitions can be formalized by comparing an image patch against itself with respect to a small variation in position $\Delta\mathbf{u}$, a formulation known as the *auto-correlation function*:

$$E_{\mathrm{AC}}(\Delta\mathbf{u}) = \sum_i w(\mathbf{x}_i)\,[I_0(\mathbf{x}_i+\Delta\mathbf{u}) - I_0(\mathbf{x}_i)]^2 \tag{3-1}$$

where I_0 is the image, $\mathbf{u} = (u, v)$ is the displacement vector, w is a spatial window function, and the summation is performed over all pixels i in the patch. Figure 3.2 illustrates that the auto-correlation surface for a corner patch exhibits a distinct global minimum, indicating robust localization. In contrast, the edge surface reveals ambiguity along the edge direction, while the flat region lacks any stable minimum.

[Lucas e Kanade 1981] showed that using the Taylor Series expansion of the image function the auto-correlation surface can be approximated as

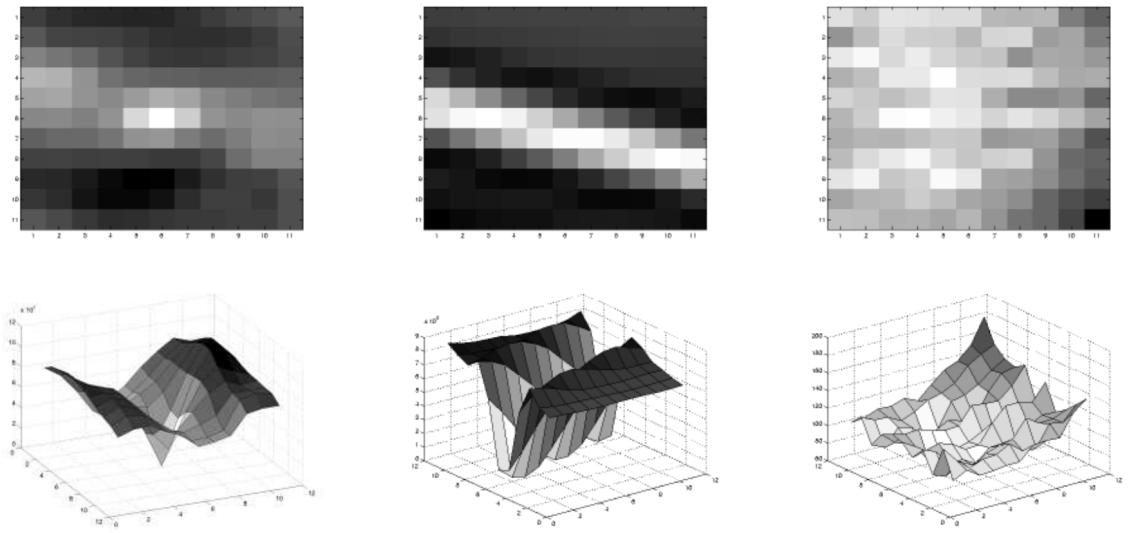


Figure 3.2: *The figure below displays the auto-correlation surfaces for a corner patch, an edge patch, and a flat region, respectively [Szeliski 2022].*

$$
\begin{aligned}
E_{\mathrm{AC}}(\Delta\mathbf{u}) &= \sum_i w(\mathbf{x}_i)\left[I_0(\mathbf{x}_i+\Delta\mathbf{u}) - I_0(\mathbf{x}_i)\right]^2 \\
&\approx \sum_i w(\mathbf{x}_i)\left[I_0(\mathbf{x}_i) + \nabla I_0(\mathbf{x}_i)\cdot\Delta\mathbf{u} - I_0(\mathbf{x}_i)\right]^2 \\
&= \sum_i w(\mathbf{x}_i)\left[\nabla I_0(\mathbf{x}_i)\cdot\Delta\mathbf{u}\right]^2 \\
&= \Delta\mathbf{u}^T\mathbf{A}\Delta\mathbf{u},
\end{aligned}
$$

where

$$
\nabla I_0(\mathbf{x}_i) = \left(\frac{\partial I_0}{\partial x}, \frac{\partial I_0}{\partial y}\right)(\mathbf{x}_i) \tag{3-2}
$$

denotes the image gradient at x_i, which can be approximated using discrete kernels such as the Sobel operator, or computed by convolving the image with the horizontal and vertical derivatives of a Gaussian.

The auto-correlation matrix $\mathbf{A}$ can be written as:

$$
\mathbf{A} = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}, \tag{3-3}
$$

where the weighted summation is replaced by a discrete convolution with the weighting kernel w. This matrix summarizes the predominant directions of the gradient in a local neighborhood, and the eigenvalues indicate the magnitude of the principal intensity changes in two orthogonal directions. In flat regions, both eigenvalues are small. Con-

versely, in an edge window, one large eigenvalue corresponds to the dominant gradient direction, while the other remains small. In a corner region, however, both eigenvalues are significant, indicating intensity variations in orthogonal directions (see Figure 3.3).

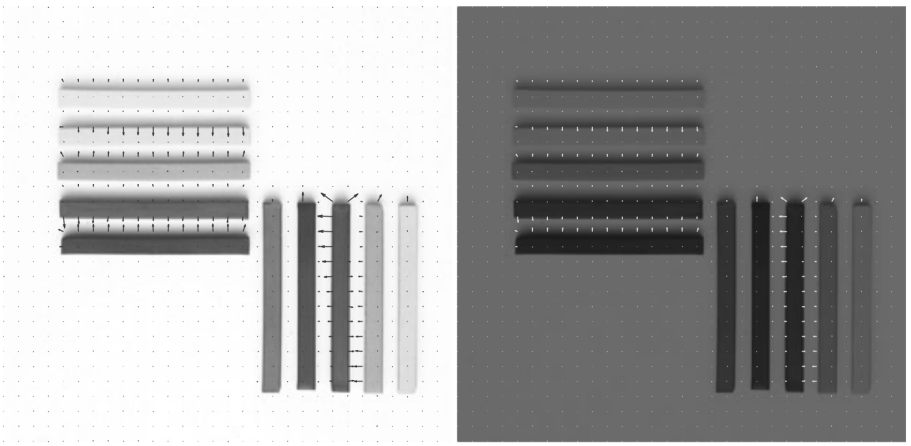


Figure 3.3: *The figure below displays the image gradient on flat regions, edges, and corners. Note how the gradient direction changes in the corner regions [Forsyth e Ponce 2012].*

[Harris e Stephens 1988] proposed using the local maxima of a rotationally invariant scalar measure, derived from the autocorrelation matrix, to locate keypoints. Their method quantifies a pixel's 'cornerness' score based on the determinant and trace of this matrix:

$$C = \det(\mathbf{A}) - k(tr(\mathbf{A})^2, \tag{3-4}$$

where k is an empirical constant. The local maxima of this function correspond to regions where both eigenvalues are large. Crucially, this detector is invariant to translation and rotation.

The matrix $\mathcal{A}$ is a reliable indicator of which patches can be tracked effectively. Indeed, regions containing high gradients in orthogonal directions provide stable locations for finding correspondences. While Harris combined the eigenvalues into a single 'cornerness' score to avoid explicit eigenvalue decomposition, Shi and Tomasi [Shi e Tomasi 1994] argued that a feature is good to track if and only if the smaller eigenvalue is sufficiently large:

$$C = min(\lambda_1, \lambda_2). \tag{3-5}$$

3.1.2 Translational alignment

The principle described in Equation 3-1 can also be applied to align images or patches. Given a template image $I_0(\mathbf{x})$ sampled at discrete pixel locations $\mathbf{x}_i = (x_i, y_i)$, the goal is to locate this template within a target image $I_1(\mathbf{x})$. A standard least-squares solution involves finding the displacement that minimizes the *Sum of Squared Differences* (SSD) function:

$$E_{\text{SSD}}(\mathbf{u}) = \sum_i [I_1(\mathbf{x}_i + \mathbf{u}) - I_0(\mathbf{x}_i)]^2 = \sum_i e_i^2, \tag{3-6}$$

where $\mathbf{u} = (u, v)$ is the displacement vector and $e_i = I_1(\mathbf{x}_i + \mathbf{u}) - I_0(\mathbf{x}_i)$ represents the residual error. The proposed solution relies on the *brightness constancy constraint*, which posits that the perceived brightness of an object remains constant under varying conditions. In other words, it assumes that corresponding pixel values remain invariant across consecutive images.

The most straightforward strategy to minimize the cost function in Equation 3-6 is to perform an exhaustive search over a range of shifts, however, this approach is computationally expensive. To accelerate this process, *hierarchical motion estimation* is often employed. An image pyramid (Figure 3.4) is constructed, allowing the search to be performed at coarser levels over a smaller discrete domain. At the coarsest level, we search for the optimal displacement $\mathbf{u}^{(l)}$ that minimizes the difference between images $I_0^{(l)}$ and $I_1^{(l)}$. This search is conducted over a displacement range $\mathbf{u}^{(l)} \in 2^{-l}[-S, S]^2$, where S represents the desired search range at the original resolution. Once a suitable motion vector has been estimated, it is used to predict a likely displacement:

$$\hat{\mathbf{u}}^{(l-1)} \leftarrow 2\mathbf{u}^{(l)} \tag{3-7}$$

for the next finer level. The search over displacements is then repeated at the finer level over a much narrower range of displacements.

The strategy described above limits alignment to integer-pixel resolution, however, many applications require significantly higher accuracy to achieve robust results. To obtain *sub-pixel* estimates, a common approach is to refine the alignment by employing an iterative optimization on the cost function (Equation 3-1), utilizing a Taylor series expansion to linearize the image intensity

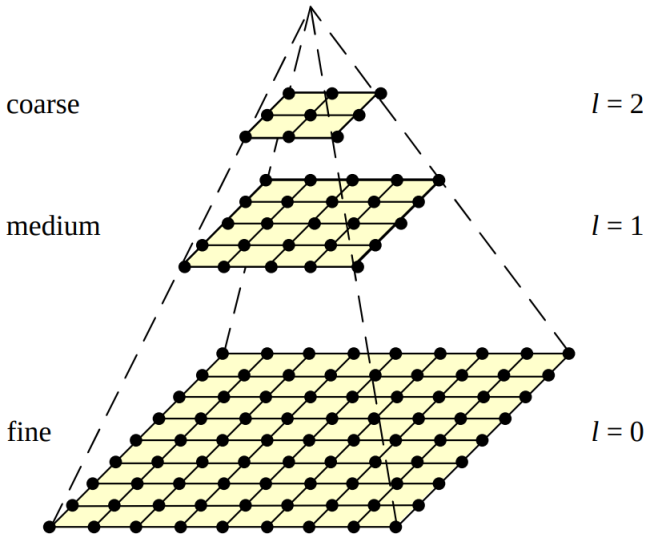


Figure 3.4: *In a standard image pyramid, every successive layer is scaled down, reducing both the width and height by 50%. Consequently, each new level contains only one-fourth of the total pixel count found in the layer immediately above it [Szeliski 2022].*

$$E_{\text{SSD}}(\mathbf{u}+\Delta\mathbf{u}) = \sum_i [I_1(\mathbf{x}_i+\mathbf{u}+\Delta\mathbf{u}) - I_0(\mathbf{x}_i)]^2 \tag{3-8}$$

$$\approx \sum_i [I_1(\mathbf{x}_i+\mathbf{u}) + \mathbf{J}_1(\mathbf{x}_i+\mathbf{u})\Delta\mathbf{u} - I_0(\mathbf{x}_i)]^2 \tag{3-9}$$

$$= \sum_i [\mathbf{J}_1(\mathbf{x}_i+\mathbf{u})\Delta\mathbf{u} + e_i]^2, \tag{3-10}$$

where

$$\mathbf{J}_1(\mathbf{x}_i+\mathbf{u}) = \nabla I_1(\mathbf{x}_i+\mathbf{u}) = \left(\frac{\partial I_1}{\partial x}, \frac{\partial I_1}{\partial y}\right)(\mathbf{x}_i+\mathbf{u}) \tag{3-11}$$

is the *image gradient* or *Jacobian* at $(\mathbf{x}_i+\mathbf{u})$ and

$$e_i = I_1(\mathbf{x}_i+\mathbf{u}) - I_0(\mathbf{x}_i), \tag{3-12}$$

is the current intensity error. This least squares problem can be minimized by solving the associated *normal equations*

$$\mathbf{A}\Delta\mathbf{u} = \mathbf{b} \tag{3-13}$$

where

$$\mathbf{A} = \sum_i \mathbf{J}_1^T(\mathbf{x}_i+\mathbf{u})\mathbf{J}_1(\mathbf{x}_i+\mathbf{u}) \tag{3-14}$$

and

$$\mathbf{b} = -\sum_i e_i \mathbf{J}_1^T(\mathbf{x}_i + \mathbf{u}) \tag{3-15}$$

are called the *Hessian* and *gradient-weighted residual vector*, respectively. These matrices are also often written as

$$\mathbf{A} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = -\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}. \tag{3-16}$$

3.1.3 Parametric motion

As the camera moves, image intensity patterns undergo complex transformations. Nevertheless, images captured at consecutive time instants are typically highly correlated, as they depict the same scene from slightly varying viewpoints. Equation 3-6 relies on the assumption that pixels move rigidly between frames under a pure translation model, preserving their shape, size, and orientation. However, when the camera undergoes complex motion (e.g., rotation or scaling), the translation model fails to account for geometric deformations, leading to high residual errors. To mitigate this, a more effective strategy is to enrich the motion description with more sophisticated motion models.

Transformation	Matrix	# DoF	Preserves
translation	$[I \mid t]_{2\times3}$	2	orientation
rigid (Euclidean)	$[R \mid t]_{2\times3}$	3	lengths
similarity	$[sR \mid t]_{2\times3}$	4	angles
affine	$[A]_{2\times3}$	6	parallelism
projective	$[\tilde{H}]_{3\times3}$	8	straight lines

Table 3.1: *This table presents a hierarchy of 2D coordinate transformations where invariant properties are cumulative. Each transformation maintains its specific attributes as well as those defined in the categories below it [Szeliski 2022].*

To accommodate parametric motion, instead of relying on a single constant translation vector $\mathbf{u}$, we employ a spatially varying *motion field* (or *correspondence map*) $\mathbf{x}'(\mathbf{x};\mathbf{p})$. This field is parameterized by a low-dimensional vector $\mathbf{p}$, where $\mathbf{x}'$ represents any of the motion models listed in Table 3.1. Consequently, the parametric incremental motion update rule becomes:

$$E_{\text{LK-PM}}(\mathbf{p}+\Delta\mathbf{p}) = \sum_i [I_1(\mathbf{x}'(\mathbf{x}_i;\mathbf{p}+\Delta\mathbf{p})) - I_0(\mathbf{x}_i)]^2 \tag{3-17}$$

$$\approx \sum_i [I_1(\mathbf{x}'_i) + \mathbf{J}_1(\mathbf{x}'_i)\Delta\mathbf{p} - I_0(\mathbf{x}_i)]^2 \tag{3-18}$$

$$= \sum_i [\mathbf{J}_1(\mathbf{x}'_i)\Delta\mathbf{p} + e_i]^2, \tag{3-19}$$

where the Jacobian is now

$$\mathbf{J}_1(\mathbf{x}'_i) = \frac{\partial I_1}{\partial \mathbf{p}} = \nabla I_1(\mathbf{x}'_i)\frac{\partial \mathbf{x}'}{\partial \mathbf{p}}(\mathbf{x}_i), \tag{3-20}$$

the *Hessian* and *gradient-weighted* residual vector become

$$\mathbf{A} = \sum_i \mathbf{J}_{\mathbf{x}'}^{\top}(\mathbf{x}_i)[\nabla I_1^T(\mathbf{x}'_i)\nabla I_1(\mathbf{x}'_i)]\mathbf{J}_{\mathbf{x}'}(\mathbf{x}_i)$$

and

$$\mathbf{b} = -\sum_i \mathbf{J}_{\mathbf{x}'}^{\top}(\mathbf{x}_i)[e_i \nabla I_1^T(\mathbf{x}'_i)].$$

Computing the Hessian matrix and residual vectors for parametric motion is significantly more computationally demanding than in the pure translational case. With n parameters and N pixels, the accumulation step requires $O(n^2N)$ operations. To mitigate this computational cost, a common strategy is to partition the image into smaller sub-blocks (patches) P_j. This allows for the efficient accumulation of the simpler 2×2 spatial gradient terms (e.g., $\nabla I \nabla I^T$) at the pixel level before assembling the full parametric system:

$$\mathbf{A}_j = \sum_{i\in P_j} \nabla I_1^T(\mathbf{x}'_i)\nabla I_1(\mathbf{x}'_i)$$

$$\mathbf{b}_j = \sum_{i\in P_j} e_i \nabla I_1^T(\mathbf{x}'_i).$$

The full Hessian and residual can then be approximated as

$$\mathbf{A} \approx \sum_j \mathbf{J}_{\mathbf{x}'}^T(\hat{\mathbf{x}}_j)\left[\sum_{i\in P_j} \nabla I_1^T(\mathbf{x}'_i)\nabla I_1(\mathbf{x}'_i)\right]\mathbf{J}_{\mathbf{x}'}(\hat{\mathbf{x}}_j) = \sum_j \mathbf{J}_{\mathbf{x}'}^T(\hat{\mathbf{x}}_j)\mathbf{A}_j\mathbf{J}_{\mathbf{x}'}(\hat{\mathbf{x}}_j)$$

and

$$\mathbf{b} \approx -\sum_j \mathbf{J}_{\mathbf{x}'}^T(\hat{\mathbf{x}}_j) \left[\sum_{i \in P_j} e_i \nabla I_1^T(\mathbf{x}_i') \right] = -\sum_j \mathbf{J}_{\mathbf{x}'}^T(\hat{\mathbf{x}}_j) \mathbf{b}_j,$$

where $\mathbf{x}^j$ denotes the center of each patch P_j. This simplification is effectively equivalent to replacing the true, spatially varying motion Jacobian with a piecewise-constant approximation.

3.2 Geometric Verification

Feature correspondences are frequently contaminated by outliers, i.e., incorrect data associations that deviate from the true motion. Common sources of these errors include image noise, occlusions, motion blur, and significant changes in viewpoint or illumination, as the standard mathematical formulations described previously do not explicitly account for these artifacts. The solution to outlier removal consists of exploiting the geometric and algebraic constraints that hold between two or more views of the same scene.

3.2.1 Epipolar Geometry

Epipolar geometry describes the intrinsic projective relationship between two views. Crucially, it is independent of the scene structure, relying solely on the cameras' internal parameters and their relative pose. While traditionally motivated by the need to constrain the correspondence search in stereo matching, this geometry is also extensively used to evaluate geometric consistency between consecutive frames in a temporal sequence.

Figure 3.5 illustrates the fundamental geometric entities of *epipolar geometry*. The *epipoles*, denoted as $\mathbf{e}$ and $\mathbf{e}'$, are the points where the baseline joining the two camera centers intersects the respective image planes. Equivalently, each epipole represents the projection of the opposing camera center onto the current view. The *epipolar plane* is defined by the two camera centers and the 3D scene point $\mathbf{X}$. Finally, the *epipolar lines* $\mathbf{l}$ and $\mathbf{l}'$ are formed by the intersection of this epipolar plane with the image planes. In projective terms, the epipolar line $\mathbf{l}'$ corresponds to the image of the ray back-projected from point $\mathbf{x}$ in the first view.

As illustrated in Figure 3.5, consider the case where the 3D points $\mathbf{X}$ lie on a specific plane π. The set of image points $\mathbf{x}_i$ in the first view and their corresponding points $\mathbf{x}_i'$ in the second view are projectively equivalent, as both sets are projections of the planar points $\mathbf{X}_i$. Consequently, there exists a 2D Homography $\mathbf{H}_\pi$ mapping each $\mathbf{x}_i$ to $\mathbf{x}_i'$. Geometrically, the *epipolar line* $\mathbf{l}'$ is defined as the line passing through the

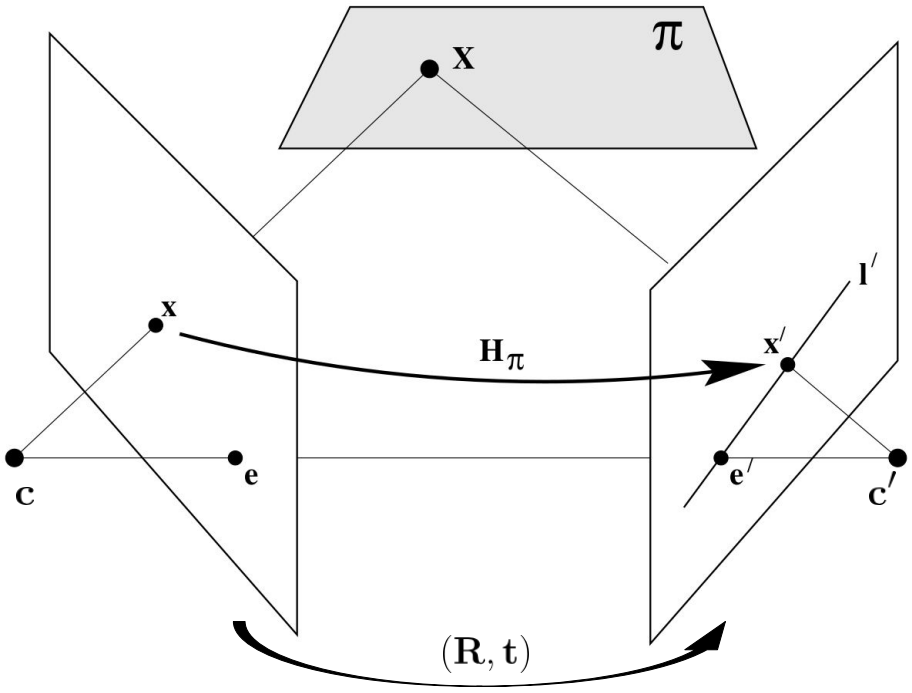


Figure 3.5: ***Point correspondence geometry****: The points* $\boldsymbol{x}$ *and* $\boldsymbol{x}'$ *are both images of the 3D point* $\boldsymbol{X}$*. The image point* $\boldsymbol{x}$ *backprojects to a ray in 3D space defined by the first camera centre and this ray is imaged as a line* $\boldsymbol{l}'$ *in the second image. So the image of* $\boldsymbol{X}$ *in the second view must lie on* $\boldsymbol{l}'$ *[Hartley e Zisserman 2004].*

epipole $\mathbf{e}'$ and the point $\mathbf{x}'$. This can be expressed algebraically using the cross product: $\mathbf{l}' = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_\times \mathbf{x}'$. By substituting the homography relation $\mathbf{x}' = \mathbf{H}_\pi \mathbf{x}$, we obtain:

$$\mathbf{l}' = [\mathbf{e}']_\times \mathbf{H}_\pi \mathbf{x} = \mathbf{F}\mathbf{x} \tag{3-21}$$

where $\mathbf{F} = [\mathbf{e}']_\times \mathbf{H}_\pi$, called the fundamental matrix, is the algebraic representation of the *epipolar geometry*.

The epipolar geometry imposes a strong constraint: for every point $\mathbf{x}$ in the first image, there exists a corresponding epipolar line $\mathbf{l}'$ in the second image. Consequently, any candidate match $\mathbf{x}'$ in the second image must lie on this line $\mathbf{l}'$. This point-to-line projective mapping is encapsulated by the fundamental matrix $\mathbf{F}$, providing a rigorous mechanism to validate the geometric consistency of feature correspondences. The algebraic constraint is expressed as:

$$\mathbf{x}_i'^{\top} \mathbf{F} \mathbf{x}_i = 0 \tag{3-22}$$

which holds for all true corresponding point pairs $\mathbf{x}_i \leftrightarrow \mathbf{x}_i'$.

The set of corresponding point pairs $\mathbf{x} \leftrightarrow \mathbf{x}'$ is geometrically degenerate with respect to $\mathbf{F}$ if it fails to uniquely define the epipolar geometry, or equivalently, if there exist

two or more linearly independent rank-2 matrices that satisfy Equation 3-22. A critical degeneracy case occurs when all points lie on a plane. In this instance, the two views are related by a 2D projective transformation (homography) such that $\mathbf{x}' = \mathbf{H}\mathbf{x}$. Substituting this relation into the epipolar constraint (Equation 3-22) reveals that the fundamental matrix must satisfy $\mathbf{x}'^{\top}(\mathbf{F}\mathbf{H}^{-1})\mathbf{x}' = 0$. This condition holds for all points whenever $\mathbf{F}\mathbf{H}^{-1}$ is skew-symmetric. Consequently, the correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ determine a three-parameter family of possible fundamental matrices $\mathbf{F}$ [Hartley e Zisserman 2004].

Another degeneracy case arises when the camera undergoes pure rotation without translation. In this instance, the epipolar geometry is undefined because the two camera centres are coincident. Analogous to the planar case, this leads to a two-parameter family of possible solutions for $\mathbf{F}$ [Hartley e Zisserman 2004].

3.2.2 The Essential Matrix

The essential matrix represents the specialization of the fundamental matrix to normalized image coordinates. This formulation assumes that the calibration matrix $\mathbf{K}$ (containing the camera's *intrinsic parameters*) is known. Consequently, the essential matrix possesses fewer degrees of freedom (five, compared to seven for the fundamental matrix) and satisfies stricter algebraic constraints.

Consider a camera matrix decomposed as $\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]$, where $\mathbf{R}$ and $\mathbf{t}$ represent the relative pose (rotation and translation) between the two cameras, as illustrated in Figure 3.5. Let $\mathbf{x} = \mathbf{P}\mathbf{X}$, if the calibration matrix $\mathbf{K}$ is known, we can apply its inverse to the image point to obtain $\hat{\mathbf{x}} = \mathbf{K}^{-1}\mathbf{x}$, which represents the point in *normalized coordinates*. Similarly, the matrix $\mathbf{K}^{-1}\mathbf{P} = [\mathbf{R} \mid \mathbf{t}]$ is termed the *normalized camera matrix*, as the effect of the intrinsic parameters has been removed. Consequently, the projection $\hat{\mathbf{x}} = [\mathbf{R} \mid \mathbf{t}]\mathbf{X}$ can be interpreted as the image of the point $\mathbf{X}$ captured by a virtual camera $[\mathbf{R} \mid \mathbf{t}]$ with the identity matrix $\mathbf{I}$ as its internal calibration.

Since the absolute global reference frame is arbitrary, we can, without loss of generality, align it with the first camera. This sets the first camera center at the origin ($\mathbf{c} = \mathbf{0}$) with a canonical orientation ($\mathbf{R} = \mathbf{I}$). Consequently, we consider the pair of normalized camera matrices $\mathbf{P} = [\mathbf{I} \mid \mathbf{0}]$ and $\mathbf{P}' = [\mathbf{R} \mid \mathbf{t}]$. The fundamental matrix associated with this pair of normalized cameras is defined as the *essential matrix*, which takes the form:

$$\mathbf{E} = [\mathbf{t}]_{\times}\mathbf{R} \tag{3-23}$$

The essential matrix possesses only five degrees of freedom. While the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$ each contribute three degrees of freedom, the system is subject to an overall scale ambiguity. Unlike the fundamental matrix, which allows only

for projective reconstruction, the essential matrix enables the retrieval of camera pose up to scale. This latter property holds because the essential matrix is a 3×3 matrix of rank 2 with two equal non-zero singular values.

The decomposition of the essential matrix $\mathbf{E}$ using Singular Value Decomposition (SVD) yields two possible rotation matrices and two possible translation vectors. Consequently, for a given essential matrix $\mathbf{E} = \mathbf{U}\text{diag}(1,1,0)\mathbf{V}^\top$ and a canonical first camera $\mathbf{P} = [\mathbf{I} \mid \mathbf{0}]$, there are four valid configurations for the second camera matrix $\mathbf{P}'$. These solutions arise from the combinations of the two possible rotations $\mathbf{R}$ and the sign ambiguity of the translation $\mathbf{t}$:

$$P' = [UWV^\top \mid +\mathbf{u}_3] \text{ or } [UWV^\top \mid -\mathbf{u}_3] \text{ or } [UW^\top V^\top \mid +\mathbf{u}_3] \text{ or } [UW^\top V^\top \mid -\mathbf{u}_3] \quad (3\text{-}24)$$

where W is an orthogonal matrix with form

$$\mathbf{W} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3\text{-}25)$$

Geometrically, these four solutions represent the four possible configurations of the camera pair that satisfy the *epipolar geometry*. However, only one of these configurations is physically valid. As illustrated in Figure 3.6, the correct solution is uniquely identified by the condition that the reconstructed point $\mathbf{X}$ must lie in front of both cameras.

3.2.3 RANSAC

As previously noted, feature correspondences are often contaminated by outliers, leading to incorrect data associations. While conventional robust estimation techniques can mitigate these errors, they typically fail when the outlier ratio is significant. A more effective strategy is to isolate a subset of *inlier* correspondences—points that are geometrically consistent with the dominant motion. To achieve this, the RANdom SAmple Consensus (RANSAC) algorithm is widely employed. The process begins by selecting a minimal random subset of k correspondences, which is then used to compute an initial estimate of the motion model.

The fundamental principle of RANSAC is to generate model hypotheses from randomly sampled sets of data points. The hypothesis that maximizes consensus with the remaining dataset is selected as the optimal solution. This process is illustrated in Figure 3.7 using a line fitting model. Since an exhaustive search of all possible subsets is computationally infeasible, a sufficient number of trials N must be performed to guarantee

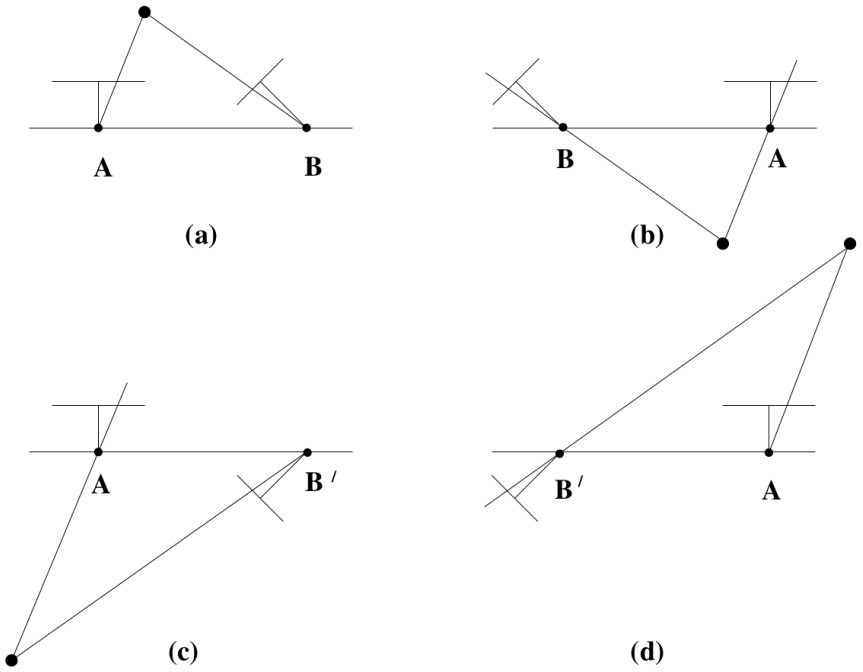


Figure 3.6: *The four possible solutions for calibrated reconstruction derived from the essential matrix* **E**. *A baseline reversal distinguishes the left and right columns, while the top and bottom rows differ by a* 180° *rotation of the second camera about the baseline. Note that only in (a) does the reconstructed point lie in front of both cameras [Hartley e Zisserman 2004].*

success. Specifically, the iteration count N is chosen to ensure, with probability p, that at least one random sample of size s is entirely free from outliers:

$$N = \frac{\log(1-p)}{\log(1-(1-\varepsilon)^s)} \tag{3-26}$$

where s is the minimal number of data points required to instantiate the model, ε represents the proportion of outliers in the dataset, and p denotes the desired probability of success.

In the context of two-view motion estimation, outlier removal relies on the geometric constraints imposed by the motion model, specifically the *epipolar geometry*. Theoretically, the optimal solution minimizes the geometric reprojection error over the full set of correspondences:

$$r_i^{geom} = \sum_i \left(d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 \right) \tag{3-27}$$

where $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ are the measured correspondences, and $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ represent the estimated 'true' correspondences that perfectly satisfy the epipolar constraint $\hat{\mathbf{x}}'^{\top}\mathbf{F}\hat{\mathbf{x}}_i = 0$.However, minimizing this non-linear function is computationally intensive. In practice, RANSAC

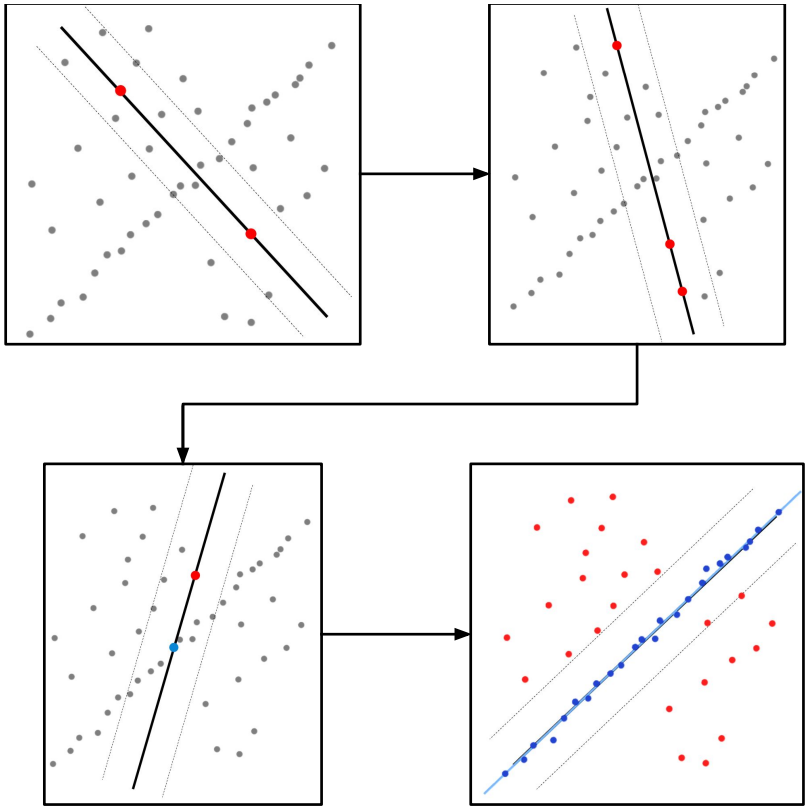


Figure 3.7: *Illustration of the RANSAC algorithm applied to line fitting. In each iteration, a minimal subset of two points is randomly selected to define a line hypothesis. The support for this hypothesis is quantified by counting the number of data points falling within a specified distance threshold. This process is repeated for multiple trials, and the model with the maximum support is identified as the robust fit. The resulting points within the threshold constitute the inliers (or consensus set).*

often evaluates inlier hypotheses using the Sampson distance, which provides a first-order approximation to the geometric error:

$$r_i^{sampson} = \sum_i \frac{(\mathbf{x}_i'^T \mathbf{F}\mathbf{x}_i)^2}{(\mathbf{F}\mathbf{x}_i)_1^2 + (\mathbf{F}\mathbf{x}_i)_2^2 + (\mathbf{F}^T\mathbf{x}_i')_1^2 + (\mathbf{F}^T\mathbf{x}_i')_2^2} \tag{3-28}$$

where $(\mathbf{F}\mathbf{x}_i)_j^2$ denotes the square of the j-th entry of the vector $\mathbf{F}\mathbf{x}_i$.

3.3 Final consideration

In summary, the mathematical formulations and geometric constraints detailed in this chapter form the core pipeline of classical point tracking architectures. The initial phase of feature-based motion estimation provides the necessary tools to extract stable image patches and calculate their continuous displacements under complex camera dynam-

ics. Subsequently, the application of geometric verification through epipolar geometry and robust estimators like RANSAC ensures that only physically valid correspondences are retained. Together, these theoretical pillars provide a reliable mechanism for rejecting outliers and estimating camera motion with high precision. A deep understanding of these fundamental principles is essential for developing novel tracking frameworks that seek to improve upon these classical paradigms by increasing geometric consistency and overall computational efficiency.

CHAPTER 4

Methodology

To address the critical gap between classical efficiency and modern robustness highlighted previously, this chapter details a comprehensive methodology for a novel deep learning tracking framework. The section begins by outlining a rigorous protocol for generating geometrically consistent feature tracking datasets from standard visual odometry benchmarks, ensuring the data adheres strictly to rigid body motion constraints. Following the data generation strategy, the core of the proposed architecture is presented. This includes a differentiable tracking model heavily inspired by the traditional Kanade Lucas Tomasi algorithm, yet entirely modernized to extract robust spatial and temporal features directly from video sequences. By leveraging an inflated 3D convolutional backbone alongside a local cost volume and a dedicated affine regression head, the proposed method accurately estimates complex geometric deformations at the local patch level to ensure highly precise point tracking across consecutive frames.

4.1 Feature Tracking Dataset generation

The available datasets for feature correspondence [Li e Snavely 2018] are primarily designed for feature matching approaches, which aim to establish correspondences between images with large deformations. Such datasets are not suitable for the proposed method. Similarly, existing datasets for point tracking in videos [Doersch et al. 2023, Greff et al. 2022] were not created for feature correspondence evaluation. These datasets generally aim to track arbitrary points in videos, with ground-truth data often generated by following objects that move independently of the camera. Consequently, the resulting data violate the rigid-body motion assumption, which is fundamental to Visual Odometry (VO) and Visual SLAM (V-SLAM) tasks.

Therefore, we design a protocol to build a versatile, generalizable feature-tracking dataset from any publicly available dataset for VO and V-SLAM tasks. Figure 4.1 illustrates the workflow employed for dataset generation. To avoid temporal leakage, we adopt a video-level data splitting scheme: 70% of sequences are allocated for training,

15% for validation, and 15% for testing. The original video sequences are segmented into fixed-length clips (i.e., sets of consecutive frames)

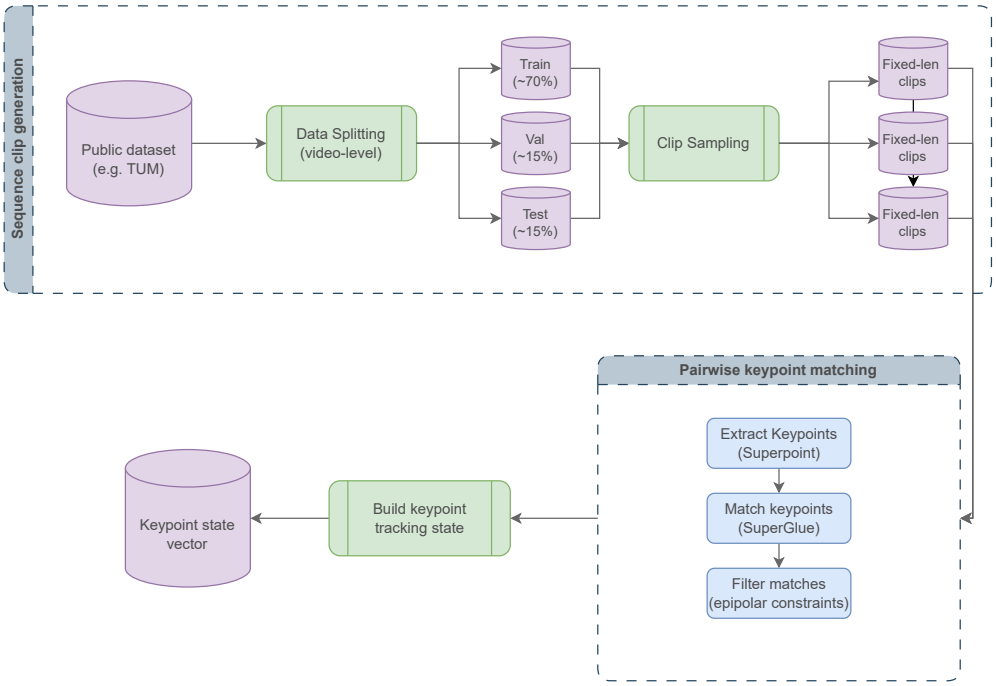


Figure 4.1: *Overview of the Proposed Dataset generation procedure.*

Two parameters govern the clip generation process: *clip length* and *step between clips*. The *clip length* parameter specifies the number of consecutive frames per clip, thereby establishing the temporal duration of the tracking sequence. The *step between clips* parameter determines the stride (or interval) between successive clips, which regulates the degree of overlap between them.

Following clip generation, a keypoint state vector is constructed by extracting and matching features across consecutive frames. The SuperPoint [DeTone, Malisiewicz e Rabinovich 2018] extractor was selected for its high repeatability and uniform spatial distribution, ensuring comprehensive coverage of the image domain. For inter-frame correspondence, SuperGlue [Sarlin et al. 2020] was employed due to its inherent compatibility with SuperPoint and robustness against occlusions. Both architectures utilize default hyperparameters and pre-trained weights. To emulate a continuous tracking process, matching in the current frame is conditioned on the successful association of keypoints in the preceding frame pair.

Successful associations must satisfy the geometric constraints outlined in Section 3.2. To address degenerate configurations strictly, a model selection strategy is employed. For a given match set $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, we estimate both the Homography and Fundamental matrix using the OpenCV [Bradski 2000] RANSAC implementation. The *Geometric Information Criterion* (GRIC) [Lasenby et al. 1998] is subsequently applied to select the model that best fits the data. GRIC scores are calculated as:

$$\text{GRIC} = \sum \rho(e_i^2) + \lambda_1 dn + \lambda_2 k, \tag{4-1}$$

where e_i is the residual error, $\rho(\cdot)$ is a robust loss function, k represents the number of parameters, d indicates the structural dimension, n is the point count, and λ_1, λ_2 serve as weighting factors. The component $\lambda_1 dn$ penalizes the data representation cost, whereas $\lambda_2 k$ penalizes model complexity. Table 4.1 summarizes these properties.

model	c	k	d	constraint	parameters
fundamental matrix	7	7	3	$x'^{\top} F x = 0$	$F = \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix}$
homography	4	8	2	$x' = Hx$	$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$

Table 4.1: ***Model descriptions****: c is the minimum number of correspondences needed in a sample to estimate the constraint. k is the number of parameters in the model; d is the dimension of the constraint.*

To quantify residual errors, the best-fit model from the RANSAC framework is evaluated against all available correspondences. The Sampson Distance (see Section 3.2.3) is used to assess residuals for the Fundamental matrix. For the Homography matrix, the Symmetric Transfer Error is applied:

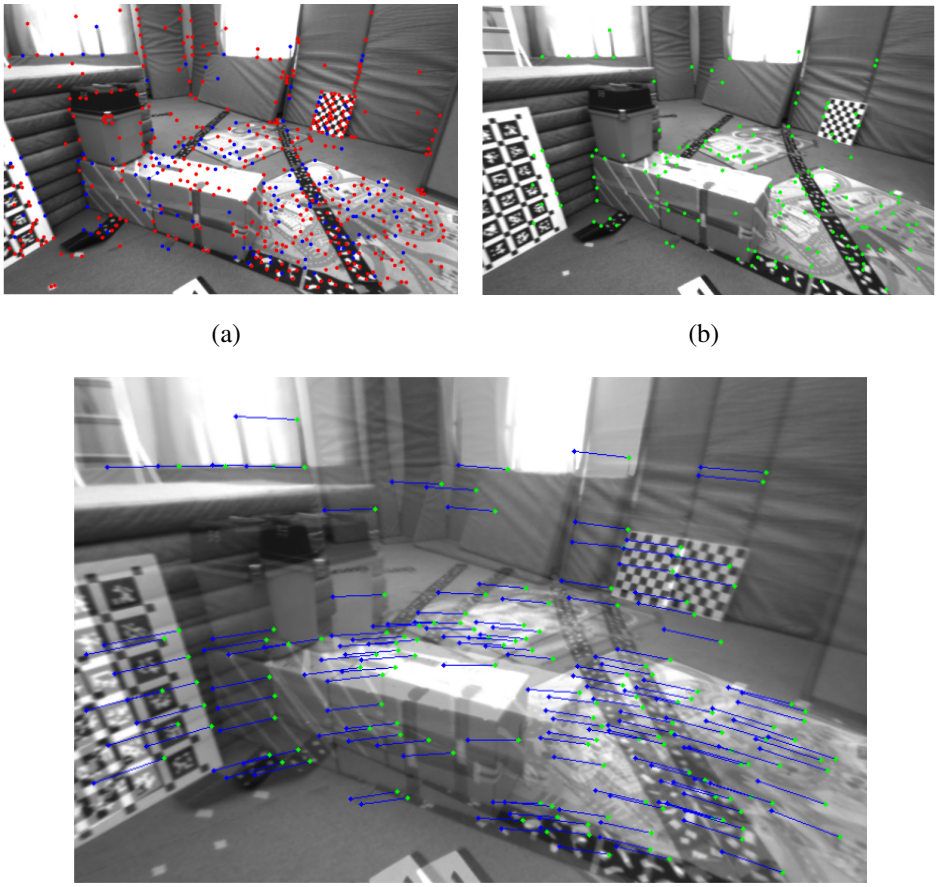
$$E_{\text{sym}} = \sum_i \left(\left\| \mathbf{x}'_i - \mathbf{H}\mathbf{x}_i \right\|^2 + \left\| \mathbf{x}_i - \mathbf{H}^{-1}\mathbf{x}'_i \right\|^2 \right) \tag{4-2}$$

where $\|\cdot\|$ denotes the Euclidean distance, $\mathbf{H}\mathbf{x}_i$ represents the forward projection of point $\mathbf{x}_i$ from the source to the target image and $\mathbf{H}^{-1}\mathbf{x}'_i$ represents the backward projection of point $\mathbf{x}'_i$ from the target back to the source image. Figure 4.2 demonstrates the efficacy of the proposed model selection strategy using a sample from the Tsukuba Stereo Dataset [Martull, Peris e Fukui 2012].

(a) (b)

Figure 4.2: *Outlier rejection using GRIC. Comparison of feature matching results on the Tsukuba stereo pair. (a): Putative matches without model selection. (b): Inliers identified by the GRIC algorithm. The reduction in point density illustrates the removal of outliers that do not conform to the dominant geometric motion model.*

The set of inliers associated with the model yielding the highest GRIC score is designated as the optimal correspondence set for each consecutive frame pair. Subsequently, a keypoint state tensor $\mathbf{S} \in \mathbb{R}^{N \times T \times 2}$ is constructed, encoding the (x, y) coordinates of the N keypoints (initialized in the first frame) across the temporal sequence of T frames. Unmatched keypoints are assigned the value $(-1, -1)$, indicating that the keypoint was lost, which can be useful for models incorporating occlusion reasoning. Figure 4.3 depicts the tracking results obtained from a four-frame clip sourced from the EuRoC MAV Dataset [Burri et al. 2016].



(a) (b)

(c)

Figure 4.3: *Feature tracking results: (a) The reference frame displaying initial feature points. Blue points indicate reference features that were successfully tracked to the target frame. Red points represent reference features that were lost and not matched. (b) The target frame showing only the green points, which are the successfully matched locations of the blue reference points from (a). (c) A visualization of the feature correspondences. The blue reference points are connected to their corresponding green target points by lines, illustrating the tracked motion vectors between the frames.*

4.2 Spatio-Temporal Affine Regression for Feature Tracking

We propose a novel, differentiable tracking model inspired by the classic Kanade-Lucas-Tomasi (KLT) [Lucas e Kanade 1981, Shi e Tomasi 1994] algorithm. This model functions as a fully convolutional neural network that learns to predict the affine transformation parameters between patches from consecutive frames, enabling the pre-

cise tracking of interest points. The architecture consists of a patch-based Spatio-temporal Feature Extractor, a Cost Volume Block, and a Regression Block that predicts the affine parameters for each patch. The proposed method is illustrated in Figure 4.4.

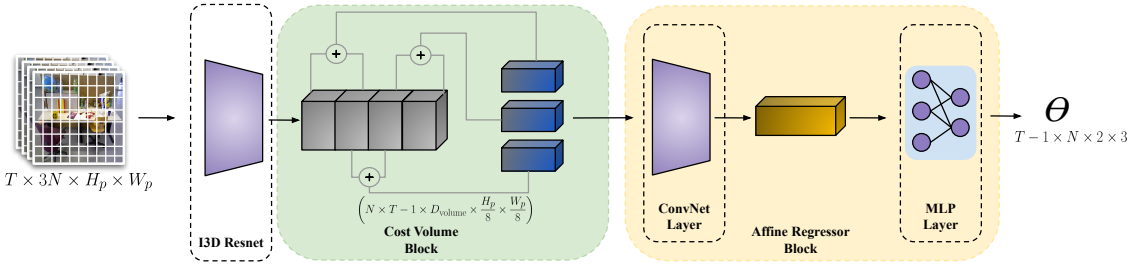


Figure 4.4: *Overview of the Proposed Tracking Architecture.*

4.2.1 Patch-Based Spatio-Temporal encoder

The model's backbone uses an Inflated 3D ConvNet (I3D) [Carreira e Zisserman 2017] architecture for feature extraction. This architecture builds upon successful 2D ConvNets, such as ResNet, by inflating their filters and pooling kernels into 3D. This inflation process allows the model to learn seamless spatio-temporal features directly from video while retaining the proven designs of ImageNet [Deng et al. 2009] architectures. Consequently, a key advantage is the ability to use pre-trained 2D weights as a valuable initialization, which significantly accelerates the training process.

The I3D ConvNet architecture is designed to extract semantic-level spatio-temporal features, making it well-suited for tasks such as action recognition and event understanding [Wang, Jabri e Efros 2019]. However, given our focus on establishing correspondence at the patch level, we utilize early-to-mid feature maps rather than the deepest layers. These intermediate representations retain higher spatial resolution and preserve structural details, attributes that are essential for capturing local spatial and short-term temporal patterns such as edges, corners, textures, and localized motion dynamics.

The encoder takes as input a set of non-overlapping patches $\mathbf{X} \in \mathbb{R}^{N \times T \times C \times H_p \times W_p}$, where N is the number of patches, T is the temporal dimension, and (H_p, W_p) are the patch's spatial dimensions. It outputs a dense feature map $\mathbf{F} \in \mathbb{R}^{N \times T \times D \times H_f \times W_f}$, where D represents the feature dimension. Each patch is processed independently, enabling parallel computation. This localized analysis also ensures the extraction of coherent spatio-temporal features from regions likely to share a similar depth.

4.2.2 Cost Volume Block

As discussed in Section 3.1, the basis of feature tracking lies in identifying correspondences between regions in distinct images. Given a patch p_t in image I_t, the

goal is to locate the most similar patch p_{t+1} within image I_{t+1}. While one could compare all reference patches from the first image against all target patches in the second, or even apply an attention mechanism to score the relationship between patches based on appearance, this process is computationally expensive and requires significant memory. Therefore, leveraging the *brightness constancy constraint* described in Section 3.1.2, we implement a Local Correlation Layer (often referred to as a cost volume), which is a core building block in modern Optical Flow algorithms [Sun et al. 2018].

The purpose of the Local Correlation Layer is to measure the similarity between a pixel in image I_t and its neighbors in image I_{t+1}. By identifying the most similar neighbor, the model can estimate the magnitude of object motion.

Let $F_t, F_{t+1} \in \mathbb{R}^{C\times H\times W}$ denote the feature maps for the current and subsequent frames, respectively. To ensure the matching process remains robust to illumination or gain variations between frames, we apply L_2 normalization to the feature vectors

$$\hat{F}(\mathbf{x}) = \frac{F(\mathbf{x})}{\|F(\mathbf{x})\|_2}. \tag{4-3}$$

Note that normalized correlation represents the dot product of unit vectors, which is mathematically equivalent to cosine similarity.

Given a search radius r, we define a displacement offset $\mathbf{d} = (i, j)$, where $i, j \in \{-r, \ldots, r\}$. The correlation volume V at spatial location $\mathbf{x}$ for a specific neighbor offset $\mathbf{d}$ is the dot product of the feature vector at time t and the shifted feature vector at time $t+1$

$$V(\mathbf{x}, \mathbf{d}) = \langle \hat{F}_t(\mathbf{x}), \hat{F}_{t+1}(\mathbf{x}+\mathbf{d}) \rangle \tag{4-4}$$

Expanding the dot product as a summation over the channels C

$$V(u, v, i, j) = \sum_{c=1}^{C} \hat{F}_t(u, v)_c \cdot \hat{F}_{t+1}(u+i, v+j)_c \tag{4-5}$$

Motivated by recent studies [Oord, Li e Vinyals 2018, Hénaff et al. 2020, Jabri, Owens e Efros 2020], we project the feature vectors into a compact latent embedding space rather than processing raw pixels. The primary intuition is to learn representations that encode the underlying shared information across parts of the data with high dimensionality. Furthermore, this approach improves efficiency regarding memory usage and computational cost. The proposed correlation layer is illustrated in Figure 4.5.

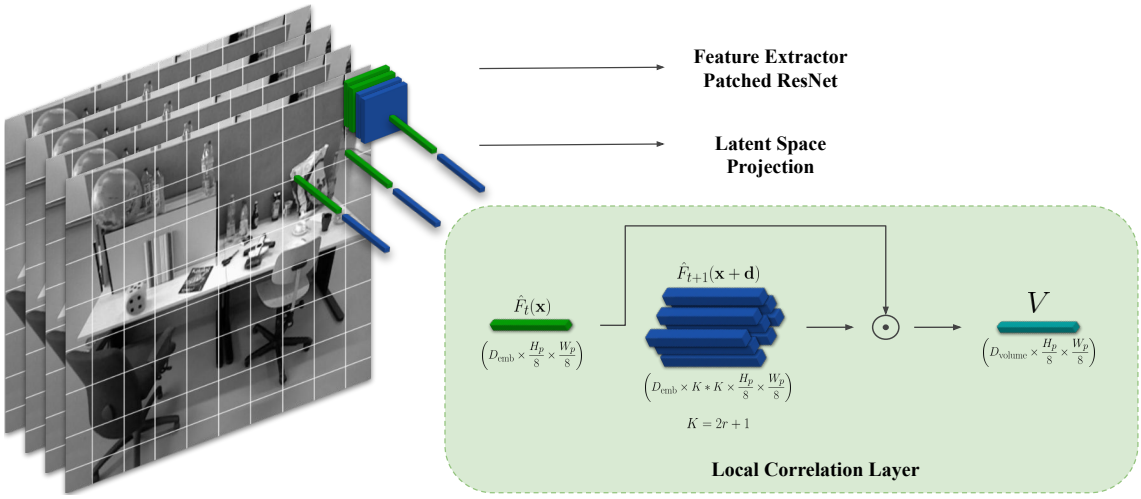


Figure 4.5: *Local Correlation Volume computation.*

4.2.3 Affine Regressor Block

As stated in Section 3.1.3, the camera can undergo complex motion, such as rotation and scaling. Consequently, a simple translation model may fail to account for the resulting geometric distortions. To address this, we implemented an Affine Regressor Block comprising a Convolutional Neural Network and a lightweight regression head designed to predict the parameters of the 3×2 affine transformation matrix for each patch across consecutive frames. This architecture is akin to the Localization Network introduced in the Spatial Transformer Network(STN) [Jaderberg et al. 2016].

The standard STN utilizes the parameters θ to spatially transform feature maps, aiming to improve the invariance of image classification layers. In contrast, within our model, θ constitutes the primary prediction. It quantifies the geometric transformation that accounts for the motion observed from Frame t to Frame $t+1$.

Let $F_t^i \in \mathbb{R}^{D_{\text{emb}} \times \frac{H_p}{8} \times \frac{W_p}{8}}$ denote the feature embedding of a patch at time t, and let $S_{t,t+1}^i \in \mathbb{R}^{(D_{\text{volume}} \times \frac{H_p}{8} \times \frac{W_p}{8})}$ represent its corresponding correlation score volume (similarity scores). For every patch, we concatenate the embedding and the scores along the channel dimension. This results in a new tensor that encodes visual appearance and motion correlation

$$X_{t,t+1}^i = \text{Concat}(F_t^i, S_{t,t+1}^i). \tag{4-6}$$

The Convolutional Neural Network processes all the concatenated tensors. It comprises two basic layers that utilize Batch Normalization and ReLU activation to ensure training stability and nonlinearity while preserving spatial resolution. This stage processes the spatial relationships within the correlation volume. Since the correlation peak might extend across adjacent pixels, convolution aggregates this local neighborhood information. The resulting tensor, representing spatially encoded motion cues, is flattened

and passed to the Multilayer Perceptron (MLP). The MLP layer consists of a dense layer with ReLU activation, followed by a final linear projection layer that outputs the 6 parameters of the affine matrix.

In summary, the proposed Affine Regressor Block is defined by the composition of a spatial aggregation function and a parametric projection function. The resulting vector $\theta \in \mathbb{R}^6$ represents the flattened Affine Transformation Matrix $\mathcal{T}$

$$\mathcal{T} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \\ \theta_4 & \theta_5 & \theta_6 \end{bmatrix} \tag{4-7}$$

- θ_1, θ_5: Scaling/Zoom parameters.
- θ_2, θ_4: Shear/Rotation parameters.
- θ_3, θ_6: Translation (x, y) parameters.

4.2.4 Loss Function

To train the affine regression head, we employ a supervised trajectory loss that minimizes the reprojection error within a normalized coordinate space. This strategy decouples geometric alignment from image resolution, ensuring consistent gradient magnitudes. The process involves spatial association, coordinate normalization, and homogeneous projection.

Given a set of ground-truth keypoint trajectories $\mathbf{P}_{gt} \in \mathbb{R}^{N \times T \times 2}$, we first perform a spatial association step in the reference frame $t = 0$. For each keypoint trajectory i, the starting coordinate $p_0^{(i)}$ serves to identify the corresponding grid cell index m and the local patch origin $\mathbf{o}_m$.

All subsequent ground-truth coordinates $p_t^{(i)}$ (for $t > 0$) are mapped into the local coordinate system of patch m and normalized to the interval $[-1, 1]$. Let $\mathbf{r} = [r_h, r_w]^T$ represent the half-width and half-height of the patch. The normalized coordinate $\bar{p}_t^{(i)}$ is computed as

$$\bar{p}_t^{(i)} = \frac{p_t^{(i)} - \mathbf{o}_m - \mathbf{r}}{\mathbf{r}} \tag{4-8}$$

To enable the application of the affine transformation, the reference normalized coordinate $\bar{p}_0^{(i)}$ is augmented into its homogeneous form. This conversion is essential to incorporate the translation component of the affine matrix via a standard matrix multiplication operation

$$\tilde{p}_0^{(i)} = \begin{bmatrix} \bar{x}_0^{(i)} \\ \bar{y}_0^{(i)} \\ 1 \end{bmatrix} \in \mathbb{R}^3 \tag{4-9}$$

The predicted location at time t is computed by applying the predicted affine transformation matrix $\theta_t^{(m)} \in \mathbb{R}^{2\times 3}$, which corresponds to patch m, to the homogeneous reference point. This operation determines the projected coordinates of point $p_0^{(i)}$ within the target frame based on the local motion estimated by the network

$$\hat{p}_t^{(i)} = \theta_t^{(m)} \cdot \tilde{p}_0^{(i)} \tag{4-10}$$

The objective function is defined as the summed Smooth-L_1 distance between the predicted projected coordinate $\hat{p}_t^{(i)}$ and the ground-truth normalized target $\bar{p}_t^{(i)}$. A validity mask $m_{valid}(i,t)$ is applied to exclude occluded or invalid points, which are denoted by -1 in the ground truth, from the loss calculation.

$$\mathcal{L} = \frac{1}{N_{valid}} \sum_{i=1}^{N} \sum_{t=1}^{T} m_{valid}(i,t) \cdot \left\| \hat{p}_t^{(i)} - \bar{p}_t^{(i)} \right\|_1 \tag{4-11}$$

where $N_{valid} = \sum_{i,t} m_{valid}(i,t)$ ensures the loss is normalized by the number of valid tracking instances.

4.3 Final consideration

In conclusion, the methodology presented in this chapter provides a highly cohesive pipeline for both generating reliable tracking data and performing robust motion estimation. The custom dataset generation protocol ensures that the training environment strictly respects epipolar geometry, effectively filtering out dynamic outliers through rigorous mathematical validation. Furthermore, the proposed Spatiotemporal Affine Regression architecture successfully elevates classical tracking paradigms into the realm of modern artificial intelligence. By projecting raw pixel patches into a compact latent space and computing local correlation volumes, the network learns highly resilient visual representations. Coupled with an affine regression block optimized via a normalized trajectory loss, the model is specifically designed to predict complex geometric transformations and maintain accurate point correspondences under challenging environmental conditions.

CHAPTER 5

Experimental Results and Discussions

This chapter presents a comprehensive empirical evaluation of the proposed tracking architecture. To rigorously assess the network's capability to generalize to unseen and highly dynamic environments, a zero-shot evaluation is conducted using the challenging EuRoC MAV dataset. This includes an ablation study comparing different convolutional backbones to weigh the trade-offs between computational efficiency and tracking accuracy. Furthermore, the geometric reliability of the proposed method is quantified through pose-derived metrics and epipolar error analysis, rather than relying solely on a simple end-point-error metric. By directly comparing these results against the classical KLT algorithm and modern deep learning approaches like Pips++, this chapter validates the robustness, efficiency, and geometric consistency of the proposed feature tracking framework.

5.1 Implementation Details

To achieve the objectives of this study, a custom dataset was developed based on the TUM RGB-D dataset [Sturm et al. 2012]. This dataset was selected because it offers a wide range of scenes and camera motions that are appropriate for this application. The dataset comprises 47 sequences, each containing images with a resolution of 640×480 recorded at 30 Hz in both office environments and an industrial hall. As detailed in Section 4.1, the total number of clips is determined by the clip length and the step size parameters. After experimenting with multiple configurations, it was found that a clip size of 8 and a step size of 1 frame yielded optimal results. Further information regarding the dataset split and clip statistics is presented in Table 5.1 and Table 5.3.

Table 5.1: *Statistics for the Training Split*

Sequence Name	Num. Clips	Mean Valid Patches
fr1/xyz	765	26.20
fr1/floor	1114	24.99
fr1/plant	907	14.95
fr1/teddy	848	16.25
fr1/desk	486	18.04
fr2/pioneer_slam2	1464	20.88
fr2/dishes	2640	17.89
fr2/pioneer_slam	1878	18.85
fr2/coke	2220	20.44
fr2/360_kidnap	1277	33.24
fr2/pioneer_360	741	18.34
fr2/desk_with_person	3624	28.60
fr2/flowerbouquet	2852	21.09
fr2/rpy	3214	32.45
fr2/metallic_sphere2	1619	22.76
fr2/360_hemisphere	2584	28.83
fr2/desk	2201	27.49
fr2/large_no_loop	617	26.48
fr3/walking_xyz	792	26.25
fr3/structure_notexture_far	7	6.00
fr3/long_office_household	2403	23.25
fr3/walking_rpy	601	21.49
fr3/sitting_static	669	45.85
fr3/nostructure_texture_near_withloop	1615	26.16
fr3/structure_notexture_near	1	4.00
fr3/walking_halfsphere	916	19.66
fr3/structure_texture_far	897	36.04
fr3/structure_texture_near	1023	23.16
fr3/cabinet	35	8.57
fr3/nostructure_texture_far	333	24.28
Total / Mean	**40343**	**22.75**

Table 5.2: *Statistics for the Validation Split*

Sequence Name	Num. Clips	Mean Valid Patches
fr1/desk2	385	15.74
fr1/room	1148	15.10
fr2/large_with_loop	1189	28.50
fr2/xyz	3608	32.29
fr3/teddy	1033	13.64
fr3/sitting_rpy	715	25.35
fr3/sitting_halfsphere	965	19.44
fr3/large_cabinet	685	13.58
Total / Mean	**9728**	**20.46**

Table 5.3: *Statistics for the Test Split*

Sequence Name	Num. Clips	Mean Valid Patches
fr1/360	372	47.05
fr1/rpy	346	52.15
fr2/flowerbouquet_brownbackground	1117	34.23
fr2/pioneer_slam3	1111	46.57
fr2/metallic_sphere	1110	36.65
fr3/walking_static	357	60.22
fr3/sitting_xyz	608	59.82
Total / Mean	**5021**	**48.10**

During training, each clip is treated as an independent sample. Uniform random sampling with replacement is applied across all clips, and the sample order is randomized at each epoch to improve generalization and reduce sequence-specific temporal bias. Samples within a single batch may originate from different sequences. This approach ensures that batch-level statistics reflect the full distribution of the dataset rather than biases specific to individual sequences or time frames.

A series of data augmentations is applied to improve robustness. These include random adjustments to brightness and contrast along with the addition of Gaussian noise, speckle noise, and motion blur. Subsequently, the input frames are resized to 256×256 pixels. From these resized frames, non-overlapping 32×32 patches are extracted, which results in a total of 64 patches per frame. To accommodate the backbone's downsampling factor, we use a correlation search radius of $r = 4$. This value effectively enforces a global correlation within the local window, allowing the network to search the entire spatial extent of the patch.

The model was trained for 120,000 steps using the Adam optimizer with a learning rate of 6×10^{-5} and a weight decay of 1×10^{-5}. A batch size of 12 was utilized for training on a single NVIDIA GeForce RTX 4090 GPU. The memory consumption of the model scales primarily with the clip length, T. To ensure optimal performance, the key hyperparameters were optimized via a Bayesian search algorithm [Dewancker, McCourt e Clark 2015] with the goal of maximizing validation accuracy.

The proposed method, detailed in Section 4, is compared to the Kanade-Lucas-Tomasi (KLT) Feature Tracker implementation from [Bouguet 1999]. This version is a pyramidal implementation of the classic KLT algorithm, designed to address the aperture problem described in Section 3. To ensure a direct and fair comparison, the KLT baseline is configured to use a window size equal to the patch size of the proposed method. Additionally, the study includes a comparison with Pips++ [Zheng et al. 2023], which is an algorithm with a transformer architecture that treats tracking as a long-term sequence problem. This approach addresses the Tracking Any Point (TAP) problem, similar to those found in other recent works [Doersch et al. 2023, Karaev et al. 2023, Harley, Fang e Fragkiadaki 2022].

The experiments were performed using an NVIDIA GeForce RTX 3060 GPU and an AMD Ryzen 7 3700X CPU.

5.2 Generalization Performance

To evaluate the model's performance on unseen data, we measure its match precision on the EuRoC MAV Dataset [Burri et al. 2016]. This dataset comprises two sequences recorded by a micro aerial vehicle (MAV). The first is the Industrial Machine Hall, which represents a challenging industrial environment for SLAM, and the second is the Vicon Room, designed to evaluate multi-view reconstruction performance.

For the evaluation, matches were generated on non-overlapping clips of 8 frames. Given the reference keypoint position, the predicted location at time t is computed by applying the predicted affine transformation, following the same spatial association described in the loss function. Finally, we calculate the Euclidean distance between the ground-truth position and the predicted position in the last frame ($t = 8$) of the clip.

To evaluate the trade-off between tracking accuracy and computational efficiency, we conducted an ablation study comparing the standard ResNet-18 backbone with the deeper ResNet-50 architecture. The models were trained exclusively on the TUM RGB-D dataset and evaluated zero-shot on the EuRoC MAV dataset. This setup introduces a significant domain shift, testing the model's ability to generalize from handheld office sequences to aggressive MAV flight in industrial environments.

The results in Table 5.4 demonstrate that ResNet50 consistently achieves lower tracking error across all evaluated sequences. For instance, in the Machine Hall 01 sequence, ResNet50 reduces the mean matching error from 3.37 to 3.14 pixels (an improvement of $\approx 6.7\%$). This suggests that in cross-domain scenarios, the deeper ResNet50 backbone extracts high-level semantic features that are more invariant to the lighting changes, motion blur, and texture variations inherent in the EuRoC dataset.

Table 5.4: *Resnet backbones evaluation comparison on EuRoC MAV Dataset.*

	Resnet18		Resnet50	
	Mean (px)	Std (px)	Mean (px)	Std (px)
MH_01	3.366130	2.357810	**3.141082**	**2.200913**
MH_02	3.765397	3.324947	**3.520760**	**3.181175**
V1_01	4.023218	1.859242	**3.749356**	**1.868905**
V2_01	5.222234	8.670275	**5.088775**	**8.796874**

On the other hand, while ResNet50 offers superior generalization accuracy, ResNet18 remains a viable candidate for strictly resource-constrained embedded systems, where a slight increase in error is permissible in exchange for real-time throughput. As shown in Figure 5.1, although ResNet50 is more accurate, the performance gap is relatively narrow. In constrained environments, sacrificing $\approx 10\%$ accuracy at a 4-pixel tolerance to gain $\approx 50\%$ in memory efficiency and faster inference speeds may be a strategic trade-off.

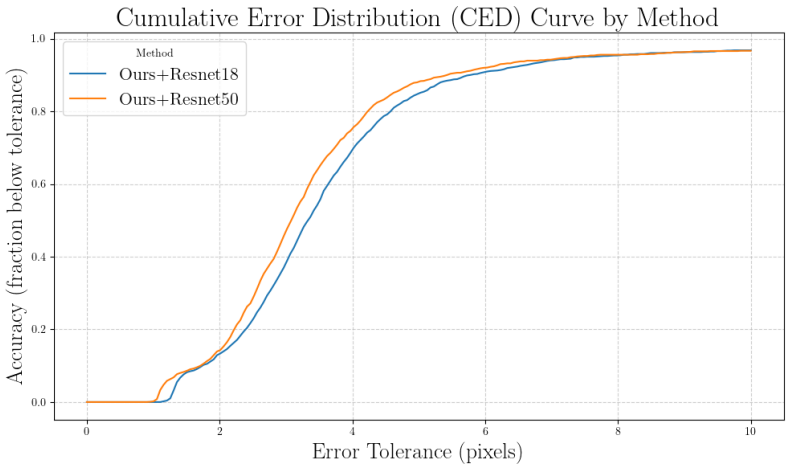


Figure 5.1: *Cumulative Error Distribution Curve by Method.*

Notably, in the Vicon Room 2 sequence, both backbones struggle equally. Analyzing the mean error per clip in Figure 5.2, we observe that certain clips exhibit substantial error spikes. Visual inspection (Figure 5.3) confirms that the tracker fails when motion becomes excessively aggressive. This indicates that the failures stem from the fixed search radius and patch size. Essentially, this is an architectural limitation: when

the feature displacement exceeds the search window, the method fails. While present in other sequences, this issue is most severe here. More qualitative results can be found in Figure 5.4.

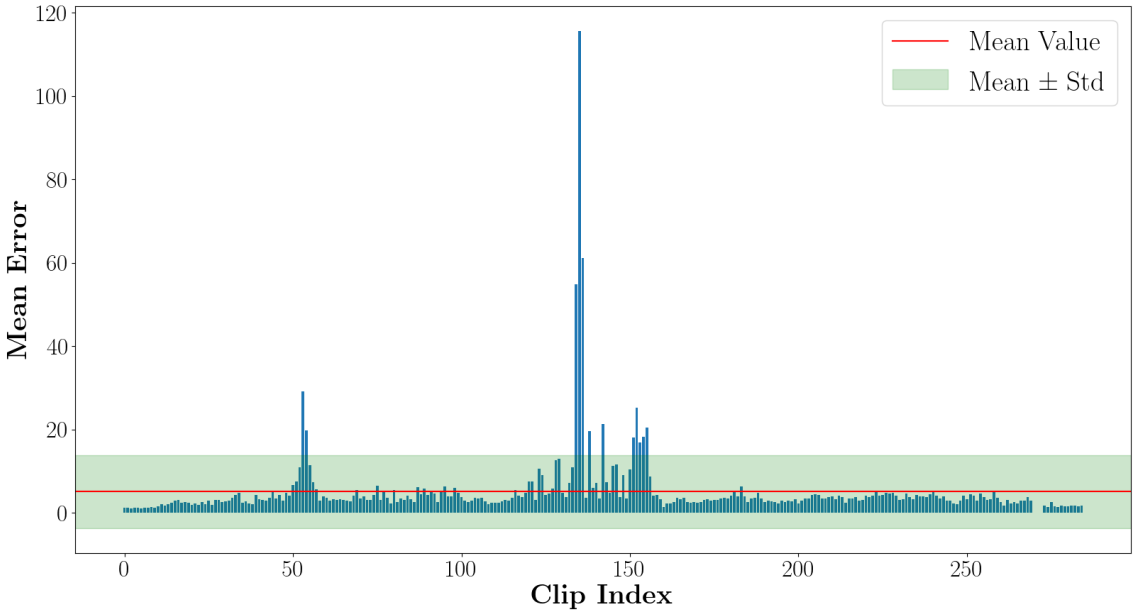
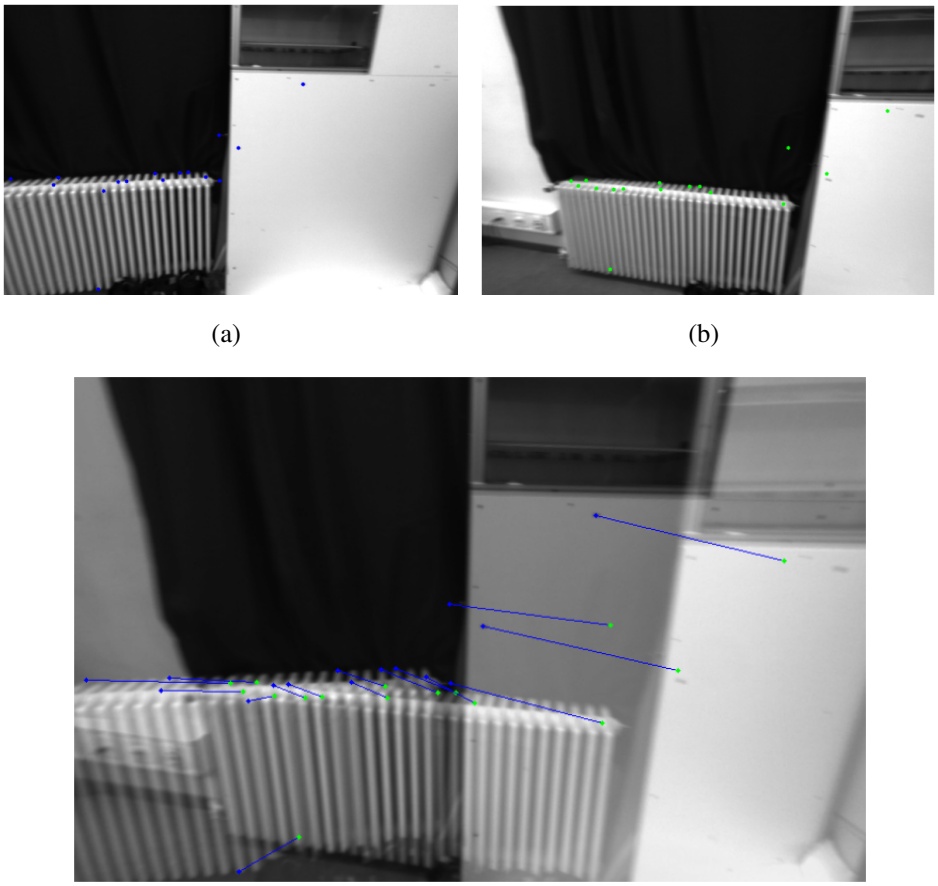


Figure 5.2: *Resnet50 backbone Vicon Room* 2 *mean error per clip.*



(a) (b)

(c)

Figure 5.3: *Tracking failure in aggressive motions.*

5.3 Methods comparison

To compare the methods, we defined a set of metrics to quantify the geometric accuracy of the feature trackers. These metrics aim to measure how well the tracked points obey geometric constraints imposed by camera motion, such as epipolar geometry [Hartley e Zisserman 2004], which is useful to evaluate their utility for downstream tasks such as VO, V-SLAM and 3D reconstruction. All experiments were conducted on the Freiburg 3 sequences, which was selected because its color and IR images are already undistorted, meaning the distortion parameters are all zero [Sturm et al. 2012]. The sequences are divided into clips of length 8. Within each clip, SuperPoint features [DeTone, Malisiewicz e Rabinovich 2018] are extracted from the first frame and tracked to the last, generating a set of correspondences. This resulting set of correspondences is then used for further evaluation. The implementation of all methods detailed below is based on the OpenCV library [Bradski 2000].

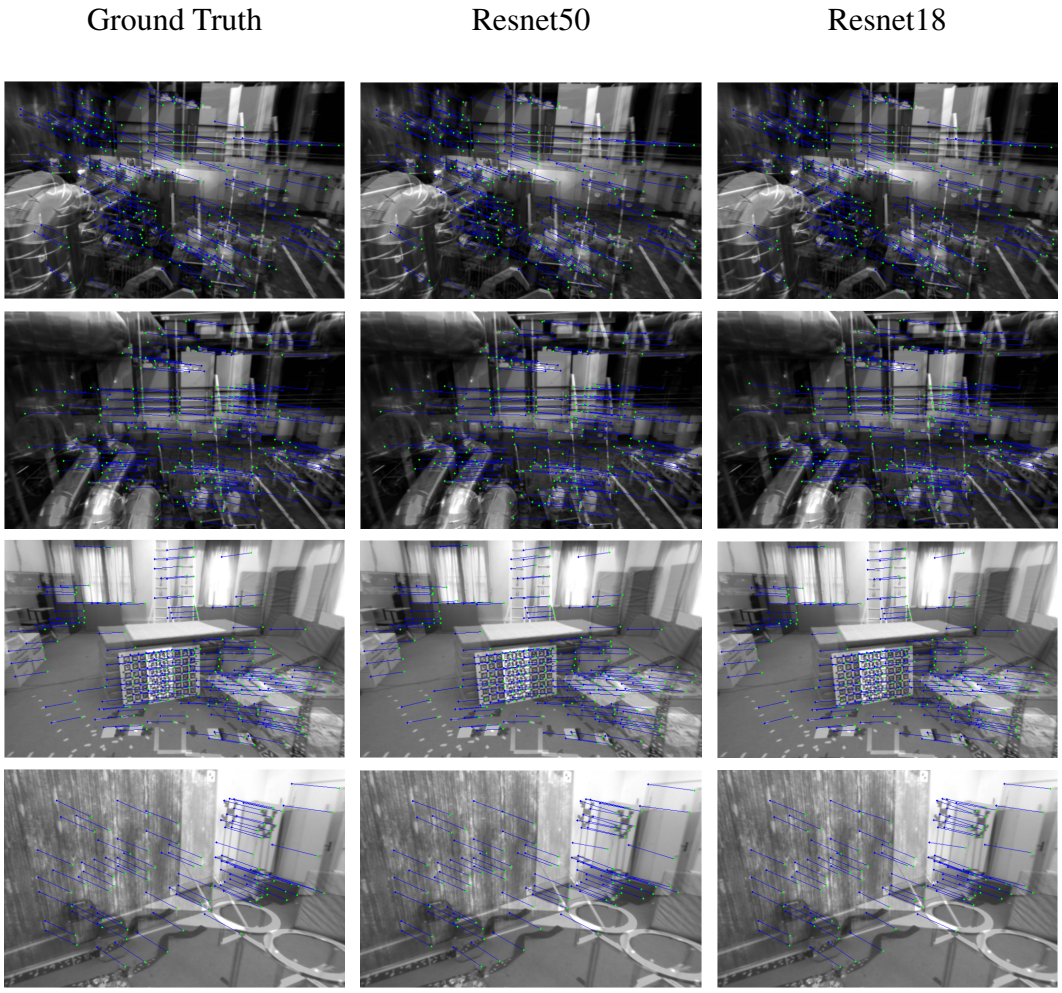


Figure 5.4: *Qualitative Results on the EuRoc Dataset.*

5.3.1 Pose-derived metrics

Pose-derived metrics are task-oriented, as they directly measure how the quality of feature tracking translates into pose estimation performance. The evaluation process is as follows: for each clip, the generated set of feature correspondences is used to compute the Essential matrix that best fits the data. This is done within a RANSAC framework to ensure robustness to outliers. The Essential matrix is then decomposed to yield an estimated relative pose $(\mathbf{R}_{est}, \mathbf{t}_{est})$, which is compared to the ground-truth relative pose $(\mathbf{R}_{gt}, \mathbf{t}_{gt})$ derived from the dataset's trajectory.

The rotation error, which quantifies the angular drift caused by correspondence inaccuracies, is computed through the residual rotation $\mathbf{R}_{rel} = \mathbf{R}_{gt}^T \cdot \mathbf{R}_{est}$:

$$\theta_{error} = \cos^{-1}\left(\frac{\text{trace}(\mathbf{R}_{rel}) - 1}{2}\right) \tag{5-1}$$

Table 5.5 presents a comparison of the rotation error in degrees for each sequence. As the results indicate, the proposed method marginally outperforms both base-

lines across the majority of sequences.

Table 5.5: *Comparison of Rotation Error (deg) on TUM RGB-D Sequences*

Sequence	KLT	Ours	Pips++
fr3/cabinet	5.7999	**5.7439**	5.8808
fr3/large_cabinet	5.2421	**5.2251**	5.3730
fr3/long_office_household	5.7369	**5.6531**	5.7218
fr3/nostructure_notexture_far	4.2936	4.0212	**3.0183**
fr3/nostructure_notexture_near_withloop	13.1939	9.5322	**9.4578**
fr3/nostructure_texture_far	**2.8265**	3.0251	2.9059
fr3/nostructure_texture_near_withloop	4.5257	**4.4754**	4.7819
fr3/sitting_halfsphere	10.4065	**10.1914**	10.3907
fr3/sitting_rpy	12.8966	12.8601	**12.8504**
fr3/sitting_static	2.1136	**1.7028**	1.7050
fr3/sitting_xyz	2.9413	**2.9017**	2.9396
fr3/structure_notexture_far	2.9370	2.6982	**2.6446**
fr3/structure_notexture_near	4.8400	4.9052	**4.4984**
fr3/structure_texture_far	2.7505	**2.7112**	2.7582
fr3/structure_texture_near	4.3487	4.3323	**4.3274**
fr3/teddy	11.3787	**10.9479**	11.4666
fr3/walking_halfsphere	10.2862	**10.0714**	10.2434
fr3/walking_rpy	11.7476	**11.6586**	11.8588
fr3/walking_static	**1.3268**	1.3347	1.3508
fr3/walking_xyz	**4.2470**	4.3029	4.2833

The second metric is the translation error, defined as the Euclidean distance between the estimated and ground-truth translation vectors. It is worth noting that the translation vector obtained from Essential matrix decomposition has an inherent scale ambiguity, it can only be recovered up to an unknown scale factor. Therefore, before computing the Euclidean distance, the estimated translation vector must be aligned with the ground-truth scale. This is achieved by first scaling the estimate:

$$\mathbf{t}_{est}^{escaled} = \mathbf{t}_{est} \cdot \left\| \mathbf{t}_{gt} \right\| \tag{5-2}$$

the final translation error is then calculated as:

$$\text{trans}_{error} = \left\| \mathbf{t}_{gt} - \mathbf{t}_{est}^{escaled} \right\| \tag{5-3}$$

Table 5.6 illustrates the comparison of the translation error for each sequence.

Consistent with the rotation error results, the proposed method marginally outperforms the alternative approaches.

Table 5.6: *Comparison of Translation Error (m) on TUM RGB-D Sequences*

Sequence	KLT	Ours	Pips++
fr3/cabinet	0.0874	**0.0821**	0.0920
fr3/large_cabinet	0.1643	**0.1572**	0.1669
fr3/long_office_household	0.1244	**0.1223**	0.1237
fr3/nostructure_notexture_far	0.0770	**0.0715**	0.0749
fr3/nostructure_notexture_near_withloop	0.1257	**0.1225**	0.1264
fr3/nostructure_texture_far	0.1275	0.1205	**0.1202**
fr3/nostructure_texture_near_withloop	0.1131	0.1139	**0.1072**
fr3/sitting_halfsphere	0.0885	**0.0835**	0.0884
fr3/sitting_rpy	0.0190	**0.0151**	0.0170
fr3/sitting_static	0.0064	**0.0060**	0.0062
fr3/sitting_xyz	0.0661	**0.0623**	0.0639
fr3/structure_notexture_far	0.0679	**0.0650**	0.0677
fr3/structure_notexture_near	0.0458	**0.0439**	0.0458
fr3/structure_texture_far	0.0946	0.0920	**0.0918**
fr3/structure_texture_near	0.0699	**0.0680**	0.0685
fr3/teddy	0.1245	**0.1154**	0.1263
fr3/walking_halfsphere	0.1044	**0.0990**	0.1031
fr3/walking_rpy	0.0404	0.0384	**0.0383**
fr3/walking_static	0.0051	**0.0050**	0.0053
fr3/walking_xyz	0.0984	**0.0897**	0.0931

5.3.2 Geometric accuracy metric

The geometric accuracy metric evaluates the robustness and reliability of a feature tracker by verifying whether its feature correspondences are geometrically consistent with camera motion. As outlined in Section 3.2, epipolar geometry defines the intrinsic projective relationship between two views. This relationship is independent of the scene structure and relies solely on the camera's internal parameters and relative pose.

Therefore, for each tracker, the resulting set of correspondences is utilized to estimate the Fundamental matrix within a RANSAC framework. As previously established, the estimated matrix is the one with the largest number of inliers among the correspondences. In other words, it is the model that best fits the data. Once the matrix is determined,

we evaluate it against the entire set of correspondences using the Sampson distance. Given the presence of outliers, the median error is employed rather than the mean.

Table 5.7 presents the comparison of the epipolar error in pixels, while Table 5.8 displays the corresponding number of inliers. The results indicate the superiority of the KLT method, which exhibits a lower median Sampson error and a higher percentage of inliers. However, an analysis of the distribution of median Sampson errors across all clips reveals a different perspective. As illustrated in Figure 5.5, the KLT method displays the highest variance. This suggests frequent and significant deviations from the median performance. In contrast, the proposed method demonstrates the most robust performance. It proves to be the most consistent approach and is the least prone to extreme error values among the evaluated methods.

Table 5.7: *Comparison of Epipolar / Sampson Error (px) on TUM RGB-D Sequences*

Sequence	**KLT**	**Ours**	**Pips++**
fr3/cabinet	0.2423	0.1247	**0.0860**
fr3/large_cabinet	**0.1795**	0.3196	0.2254
fr3/long_office_household	**0.1350**	0.3470	0.2195
fr3/nostructure_notexture_far	**0.0207**	0.0621	0.0393
fr3/nostructure_notexture_near_withloop	0.3743	0.0138	**0.0100**
fr3/nostructure_texture_far	**0.0995**	0.2085	0.1424
fr3/nostructure_texture_near_withloop	**0.0631**	0.2281	0.1411
fr3/sitting_halfsphere	**0.3495**	1.0188	0.3642
fr3/sitting_rpy	**0.2286**	0.9386	0.2629
fr3/sitting_static	**0.0353**	0.1370	0.1334
fr3/sitting_xyz	**0.0729**	0.2534	0.1964
fr3/structure_notexture_far	**0.0368**	0.0503	0.0398
fr3/structure_notexture_near	**0.0112**	0.0198	0.0147
fr3/structure_texture_far	**0.0375**	0.1658	0.1421
fr3/structure_texture_near	**0.1062**	0.2573	0.1782
fr3/teddy	2.0891	2.8538	**0.9544**
fr3/walking_halfsphere	3.7820	2.1833	**2.1480**
fr3/walking_rpy	3.0255	2.5206	**1.6855**
fr3/walking_static	**0.0696**	0.2343	0.2294
fr3/walking_xyz	**0.4685**	0.8816	0.5875

Table 5.8: *Comparison of Inlier Ratio on TUM RGB-D (fr3) Sequences*

Sequence	KLT	Ours	Pips++
fr3/cabinet	0.6804	0.6666	**0.7055**
fr3/large_cabinet	**0.7111**	0.6253	0.6593
fr3/long_office_household	**0.7465**	0.6044	0.6732
fr3/nostructure_notexture_far	**0.8022**	0.7111	0.7590
fr3/nostructure_notexture_near_withloop	0.7097	0.7079	**0.7166**
fr3/nostructure_texture_far	**0.8033**	0.6735	0.7364
fr3/nostructure_texture_near_withloop	**0.8858**	0.6821	0.7762
fr3/sitting_halfsphere	**0.6430**	0.5063	0.5903
fr3/sitting_rpy	**0.6903**	0.4975	0.6471
fr3/sitting_static	**0.9137**	0.7971	0.7935
fr3/sitting_xyz	**0.8398**	0.6644	0.7075
fr3/structure_notexture_far	**0.7855**	0.7396	0.7840
fr3/structure_notexture_near	0.6949	0.6767	**0.6959**
fr3/structure_texture_far	**0.9233**	0.7447	0.7728
fr3/structure_texture_near	**0.8003**	0.6557	0.7191
fr3/teddy	**0.5548**	0.4606	0.5400
fr3/walking_halfsphere	**0.5533**	0.4432	0.5111
fr3/walking_rpy	**0.5644**	0.4526	0.5324
fr3/walking_static	**0.8147**	0.6912	0.6995
fr3/walking_xyz	**0.6504**	0.5235	0.5551

5.4 Discussion

To validate the reliability of the proposed method for 3D vision tasks, we evaluated the quality of the correspondences generated by the trackers with respect to epipolar geometry constraints. Unlike simple endpoint error, the metrics derived from the Fundamental and Essential matrices provide a more rigorous assessment of whether the tracked correspondences are consistent with a valid rigid transformation. This geometric consistency is critical for Odometry and Visual SLAM tasks.

The quantitative evaluation performed on the TUM RGB-D benchmark reveals notable performance characteristics. These results highlight the advantages of the proposed method for rigid-body motion estimation. As evidenced in Tables 5.5 and 5.6, our method demonstrates a consistent advantage in the estimation of relative camera motion. It outperforms both the KLT baseline and the deep tracker Pips++ in the majority of se-

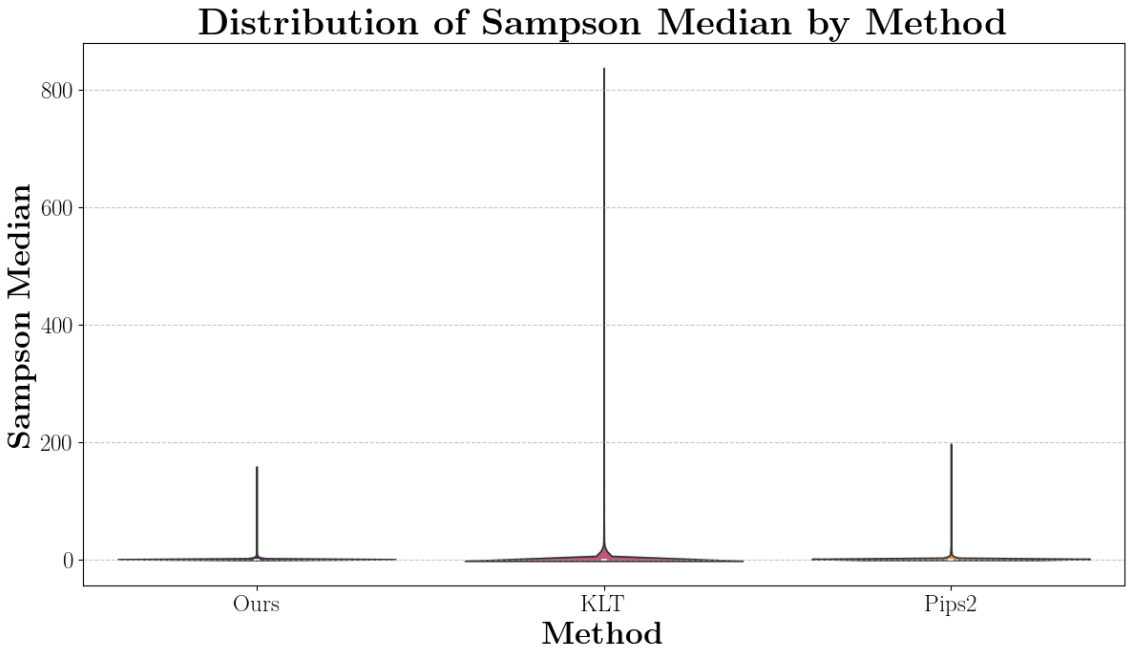


Figure 5.5: *Violin plot of the Sampson distance in pixels.*

quences. We attribute this performance advantage to the training dataset composition, where points were filtered based on epipolar geometry constraints.

The analysis of the Sampson error presented in Table 5.7 provides a nuanced picture. Although the KLT tracker frequently achieves lower median error values, the proposed method demonstrates superior robustness. As visualized in the violin plot distributions in Figure 5.5, our approach yields a distribution with the fewest extreme outliers.

Table 5.8 reveals a paradoxical finding. Although our method consistently exhibits the lowest inlier ratio, it yields significantly superior rotation and translation estimates. We posit that this discrepancy underscores a critical distinction between tracking quantity and geometric quality. The subset of correspondences generated by our network is smaller yet possesses higher geometric fidelity.

A fundamental difference between the approaches lies in the tracking formulation. KLT and Pips++ treat points independently. In contrast, our method adopts a patch-based affine paradigm. This formulation introduces a distinct trade-off. It enforces local geometric rigidity by constraining all pixels within a patch to move coherently. However, this relies on a strict assumption of local planarity. Consequently, the network struggles when a patch spans a depth discontinuity.

A further instance of performance divergence arises in sequences lacking strong visual features, specifically *fr3/nostructure_notexture_far* and *fr3/structure_notexture_near*. In these specific scenarios, Pips++ tends to yield superior results. Since our approach depends on local patch correlation, it degrades when the

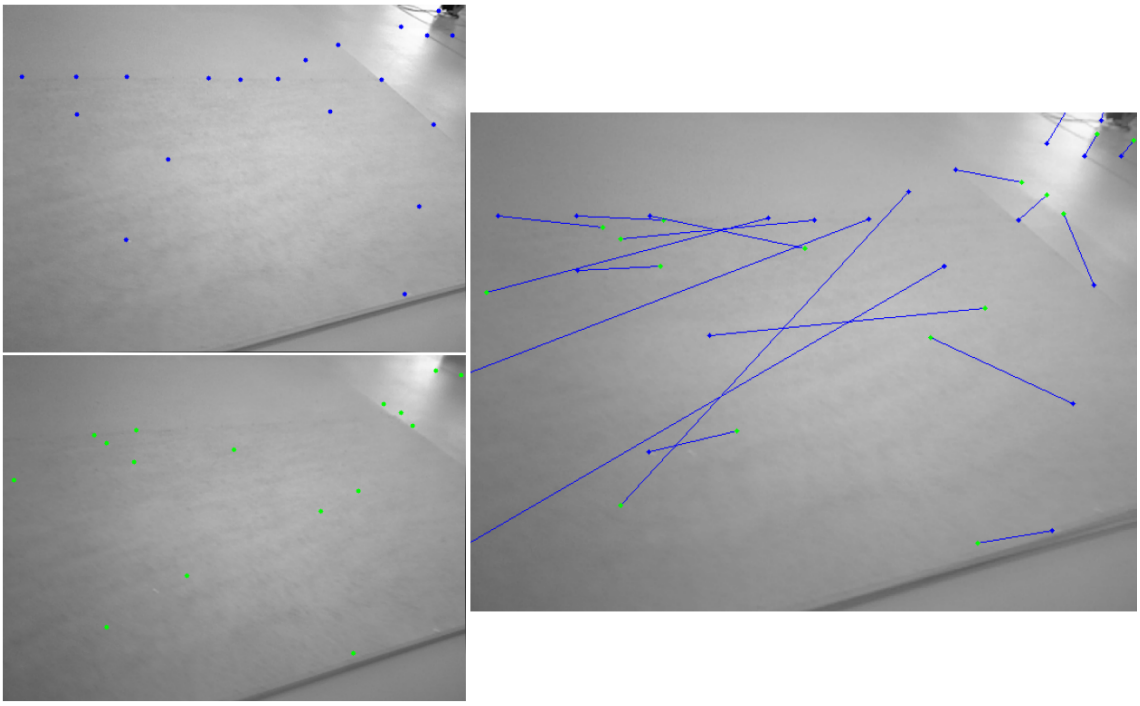


Figure 5.6: *Qualitative Evaluation of the KLT Algorithm in an Unstructured and Textureless Environment*

local window lacks discernible texture. In contrast, Pips++ employs a broader receptive field and global temporal context to maintain tracking in featureless regions. Nevertheless, our method remains significantly more robust than the KLT baseline under these adverse conditions. As illustrated in Figures 5.6 and 5.7, KLT frequently succumbs to catastrophic failure when intensity gradients are ambiguous. This indicates that learned representations are more robust than raw pixel intensities for sparse texture tracking.

This conclusion is further supported by the explicit visual comparisons provided in Figures 5.8 and 5.9. These figures broaden the evaluation to include highly structured indoor scenes alongside the challenging textureless environments. By overlaying the mean and standard deviation of the End Point Error directly onto the predicted frames, these qualitative results provide a mathematical validation of the tracking accuracy. They clearly illustrate how our proposed architecture significantly minimizes tracking drift compared to the classical KLT method and maintains highly competitive error metrics when evaluated against the Pips2 algorithm.

5.5 Final consideration

In summary, the extensive experimental evaluations validate the efficacy of the proposed tracking architecture for rigid body motion estimation. The quantitative results demonstrate that the network consistently outperforms both classical and contemporary

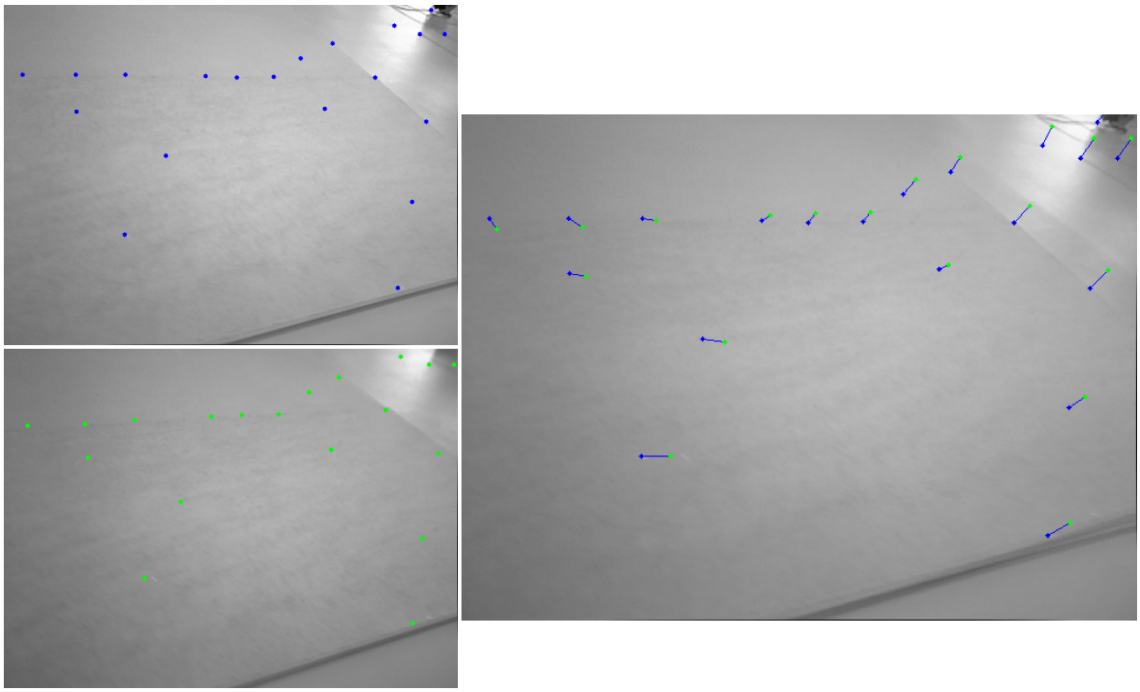


Figure 5.7: *Qualitative Evaluation of the proposed method in an Unstructured and Textureless Environment*

baselines in estimating relative camera pose, yielding highly accurate rotation and translation metrics. Although the patch based formulation results in a lower overall inlier ratio compared to traditional methods, the retained correspondences exhibit significantly higher geometric fidelity and robust resilience to extreme outliers. Furthermore, the qualitative and zero shot generalization tests confirm that the learned visual representations effectively handle challenging illumination changes and sparse textures where raw pixel methods typically fail. Ultimately, these experiments confirm that integrating classical tracking principles with modern convolutional networks provides a geometrically reliable and highly competitive solution for advanced visual odometry and spatial mapping applications.

Figure 5.8: *Comparison of feature tracking accuracy against the KLT baseline across four sequences from the TUM dataset. The mean and standard deviation of the End Point Error (EPE) are overlaid on the respective frames, illustrating the improved robustness and reduced drift of our method in challenging environments.*

Figure 5.9: *Comparison of feature tracking accuracy against the Pips2 baseline across four sequences from the TUM dataset. The mean and standard deviation of the End Point Error (EPE) are overlaid on the respective frames, illustrating the comparative performance and drift of our method relative to Pips2.*

CHAPTER 6

Final considerations

6.1 Conclusions

In this thesis, we propose a spatio-temporal affine regression model for robust feature tracking. Inspired by the Kanade-Lucas-Tomasi (KLT) paradigm, our method integrates a patch-based spatio-temporal feature extractor, a Cost Volume Block, and a Regression Block designed to regress the affine motion parameters, aligning patches across consecutive frames. Furthermore, we introduce a versatile framework for synthesizing feature-tracking annotations from arbitrary real-world datasets suitable for VO and V-SLAM applications. This data curation protocol yields a set of tracking states where keypoints exhibit strict geometric consistency with the camera trajectory.

To assess the reliability of the proposed method, we evaluated the quality of the correspondences with respect to epipolar geometry constraints. While endpoint error measures positional deviation, the evaluation criteria employed provide a more rigorous assessment of adherence to a valid rigid transformation. This geometric consistency is a prerequisite for robust Visual Odometry and Visual SLAM.

Quantitative evaluations on the TUM RGB-D benchmark demonstrate the consistent superiority of the proposed method in estimating relative camera motion. Our approach outperforms both the KLT baseline and the Pips++ deep tracker across the majority of the Freiburg 3 sequences. Additionally, analysis of the Sampson error indicates that although our network yields a smaller subset of correspondences, these matches possess higher geometric fidelity, suggesting a considerable improvement in tracking quality.

In conclusion, the experimental results position the proposed method as a robust intermediate solution. It effectively combines the local precision essential for vision-based localization with the semantic robustness required to handle real-world illumination and dynamic environments. Although deep global trackers such as Pips++ exhibit superior performance in texture-less regions, our local affine approach yields the highest metric fidelity in scenes where trackable geometric features are present. Furthermore, the proposed method demonstrates greater memory efficiency and lower latency than Pips++.

Consequently, it is particularly suitable for deployment on resource-constrained embedded systems.

6.2 Publications

- Dias NJB, Laureano GT, Da Costa RM. "Keyframe Selection for Visual Localization and Mapping Tasks: A Systematic Literature Review". Robotics. 2023; 12(3):88. https://doi.org/10.3390/robotics12030088
- N. J. Bandeira Dias, G. T. Laureano, and R. Martins Da Costa, “KLT Revisited: Spatio-Temporal Affine Regression for Feature Tracking”, RITA, vol. 33, no. 2, pp. 234–242, Mar. 2026. https://doi.org/10.22456/2175-2745.150903
- Dias NJB, Laureano GT, Da Costa RM. "Beyond Point Tracking: Learning Local Affine Deformations for Robust and Geometrically Consistent Feature Tracking". **Submitted** to the 23rd Conference on Robots and Vision. 2026. (**Under Review**)

6.3 Limitations and Future works

A significant variation in performance across the Freiburg 1 sequences becomes evident when analyzing the matching precision on the test set (Table 6.1). Corresponding velocity data from Table 6.2 indicates that the proposed method struggles specifically with aggressive camera dynamics. This issue arises from the architectural constraint of the search radius. Because the model relies on a fixed size correlation window, large displacements between frames may exceed the effective receptive field, causing the feature to fall outside the search range. Unlike coarse to fine strategies that manage large motions through downsampling, the current single scale design possesses an inherent limit on maximum detectable displacement, which ultimately leads to tracking loss during episodes of rapid motion.

Table 6.1: *Proposed method match precision on TUM testing set.*

Sequence	Mean	Std
fr1/360	**10.183481**	**11.496540**
fr1/rpy	**7.836737**	**5.121746**
fr2/flowerbouquet_brownback	2.567945	0.915945
fr2/metallic_sphere	2.259686	1.057704
fr2/pioneer_slam3	3.543999	8.542287
fr3/sitting_xyz	1.785724	0.673866
fr3/walking_static	1.545256	0.565464

To mitigate the search radius limitation, future iterations of this work could incorporate a pyramidal feature extraction strategy. By initially estimating motion at a

Table 6.2: *TUM dataset sequence velocities information.*

Sequence	Avg. translation velocity (m/s)	Avg. angular velocity (deg/s)
fr1/360	**0.210**	**41.600**
fr1/rpy	**0.062**	**50.147**
fr2/flowerbouquet_brownback	0.157	10.598
fr2/metallic_sphere	0.148	10.422
fr2/pioneer_slam3	0.164	12.339
fr3/sitting_xyz	0.132	3.562
fr3/walking_static	0.012	1.388

lower resolution, such as ResNet Layer 3 or 4, the model would be capable of recovering large displacements. These coarse estimates could then serve as an initialization for the fine-grained correlation at the original resolution. This coarse-to-fine approach would preserve the precision of the method. Simultaneously, it would significantly expand the basin of attraction, rendering the system more robust for high-speed robotics applications.

Bibliography

[Al-Tawil et al. 2024]AL-TAWIL, B. et al. A review of visual slam for robotics: evolution, properties, and future applications. *Frontiers in Robotics and AI*, Volume 11 - 2024, 2024. ISSN 2296-9144. Disponível em: <https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2024.1347985>.

[Andolfo, Petricca e Genova 2022]ANDOLFO, S.; PETRICCA, F.; GENOVA, A. Visual odometry analysis of the nasa mars 2020 perseverance rover's images. In: *2022 IEEE 9th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*. [S.l.: s.n.], 2022. p. 287–292.

[Baker et al. 2024]BAKER, L. et al. Localization and tracking of stationary users for augmented reality. *The Visual Computer*, v. 40, p. 227–244, 2024. Disponível em: <https://ieeexplore.ieee.org/abstract/document/10099250>.

[Balntas et al. 2017]BALNTAS, V. et al. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In: *CVPR*. [S.l.: s.n.], 2017.

[Barroso-Laguna et al. 2019]BARROSO-LAGUNA, A. et al. *Key.Net: Keypoint Detection by Handcrafted and Learned CNN Filters*. 2019. Disponível em: <https://arxiv.org/abs/1904.00889>.

[Bouguet 1999]BOUGUET, J.-Y. Pyramidal implementation of the lucas kanade feature tracker. In: . [s.n.], 1999. Disponível em: <https://api.semanticscholar.org/CorpusID:9350588>.

[Bradski 2000]BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[Burri et al. 2016]BURRI, M. et al. The euroc micro aerial vehicle datasets. *Int. J. Rob. Res.*, Sage Publications, Inc., USA, v. 35, n. 10, p. 1157–1163, set. 2016. ISSN 0278-3649. Disponível em: <https://doi.org/10.1177/0278364915620033>.

[Calonder et al. 2010]CALONDER, M. et al. Brief: Binary robust independent elementary features. In: DANIILIDIS, K.; MARAGOS, P.; PARAGIOS, N. (Ed.). *Computer Vision –*

ECCV 2010. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 778–792. ISBN 978-3-642-15561-1.

[Carreira e Zisserman 2017]CARREIRA, J.; ZISSERMAN, A. Quo vadis, action recognition? a new model and the kinetics dataset. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017. p. 4724–4733.

[Collins e Liu 2003]COLLINS; LIU. On-line selection of discriminative tracking features. In: *Proceedings Ninth IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2003. p. 346–352 vol.1.

[Deng et al. 2009]DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2009. p. 248–255.

[DeTone, Malisiewicz e Rabinovich 2018]DETONE, D.; MALISIEWICZ, T.; RABINOVICH, A. *SuperPoint: Self-Supervised Interest Point Detection and Description*. 2018. Disponível em: <https://arxiv.org/abs/1712.07629>.

[Dewancker, McCourt e Clark 2015]DEWANCKER, I.; MCCOURT, M.; CLARK, S. Bayesian optimization primer. *URL https://app. sigopt. com/static/pdf/SigOpt_ Bayesian_Optimization_Primer. pdf*, 2015.

[Dias, Laureano e Costa 2026]DIAS, N. J. B.; LAUREANO, G. T.; COSTA, R. M. D. Klt revisited: Spatio-temporal affine regression for feature tracking. *Revista de Informática Teórica e Aplicada*, v. 33, n. 2, p. 234–242, Mar. 2026. Disponível em: <https://seer.ufrgs.br/index.php/rita/article/view/150903>.

[Doersch et al. 2023]DOERSCH, C. et al. *TAP-Vid: A Benchmark for Tracking Any Point in a Video*. 2023. Disponível em: <https://arxiv.org/abs/2211.03726>.

[Doersch et al. 2023]DOERSCH, C. et al. TAPIR: Tracking any point with per-frame initialization and temporal refinement. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2023. p. 10061–10072.

[Dusmanu et al. 2019]DUSMANU, M. et al. *D2-Net: A Trainable CNN for Joint Detection and Description of Local Features*. 2019. Disponível em: <https://arxiv.org/abs/1905.03561>.

[Fischler e Bolles 1981]FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, Association for Computing Machinery, New York,

NY, USA, v. 24, n. 6, p. 381–395, jun. 1981. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/358669.358692>.

[Forsyth e Ponce 2012]FORSYTH, D. A.; PONCE, J. *Computer Vision: A Modern Approach*. 2nd. ed. [S.l.]: Pearson, 2012. ISBN 978-0136085928.

[Fraundorfer e Scaramuzza 2012]FRAUNDORFER, F.; SCARAMUZZA, D. Visual odometry : Part ii: Matching, robustness, optimization, and applications. *IEEE Robotics and Automation Magazine*, v. 19, n. 2, p. 78–90, 2012.

[Greff et al. 2022]GREFF, K. et al. Kubric: a scalable dataset generator. 2022.

[Gómez-Reyes et al. 2022]GóMEZ-REYES, J. K. et al. Image mosaicing applied on uavs survey. *Applied Sciences*, v. 12, n. 5, 2022. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/12/5/2729>.

[Harley, Fang e Fragkiadaki 2022]HARLEY, A. W.; FANG, Z.; FRAGKIADAKI, K. Particle video revisited: Tracking through occlusions using point trajectories. In: *ECCV*. [S.l.: s.n.], 2022.

[Harris e Stephens 1988]HARRIS, C.; STEPHENS, M. A combined corner and edge detector. In: *Proceedings of the 4th Alvey Vision Conference*. [S.l.: s.n.], 1988. p. 147–151.

[Hartley e Zisserman 2004]HARTLEY, R.; ZISSERMAN, A. *Multiple View Geometry in Computer Vision*. 2. ed. [S.l.]: Cambridge University Press, 2004.

[He et al. 2015]HE, K. et al. *Deep Residual Learning for Image Recognition*. 2015. Disponível em: <https://arxiv.org/abs/1512.03385>.

[Hénaff et al. 2020]HéNAFF, O. J. et al. Data-efficient image recognition with contrastive predictive coding. In: *Proceedings of the 37th International Conference on Machine Learning*. [S.l.]: JMLR.org, 2020. (ICML'20).

[Jabri, Owens e Efros 2020]JABRI, A. A.; OWENS, A.; EFROS, A. A. Space-time correspondence as a contrastive random walk. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2020. (NIPS '20). ISBN 9781713829546.

[Jaderberg et al. 2015]JADERBERG, M. et al. Spatial transformer networks. In: *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 2*. Cambridge, MA, USA: MIT Press, 2015. (NIPS'15), p. 2017–2025.

[Jaderberg et al. 2016]JADERBERG, M. et al. *Spatial Transformer Networks*. 2016. Disponível em: <https://arxiv.org/abs/1506.02025>.

[Jin et al. 2020]JIN, Y. et al. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision*, Springer Science and Business Media LLC, v. 129, n. 2, p. 517–547, out. 2020. ISSN 1573-1405. Disponível em: <http://dx.doi.org/10.1007/s11263-020-01385-0>.

[Karaev et al. 2023]KARAEV, N. et al. CoTracker: It is better to track together. In: . [S.l.: s.n.], 2023.

[Klingensmith, Sirinivasa e Kaess 2016]KLINGENSMITH, M.; SIRINIVASA, S. S.; KAESS, M. Articulated robot motion for simultaneous localization and mapping (arm-slam). *IEEE Robotics and Automation Letters*, v. 1, n. 2, p. 1156–1163, 2016.

[Lasenby et al. 1998]LASENBY, J. et al. Geometric motion segmentation and model selection. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, v. 356, n. 1740, p. 1321–1340, 05 1998. ISSN 1364-503X. Disponível em: <https://doi.org/10.1098/rsta.1998.0224>.

[Leutenegger, Chli e Siegwart 2011]LEUTENEGGER, S.; CHLI, M.; SIEGWART, R. Y. Brisk: Binary robust invariant scalable keypoints. *2011 International Conference on Computer Vision*, p. 2548–2555, 2011. Disponível em: <https://api.semanticscholar.org/CorpusID:1211102>.

[Li e Snavely 2018]LI, Z.; SNAVELY, N. Megadepth: Learning single-view depth prediction from internet photos. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 2041–2050.

[Lindenberger, Sarlin e Pollefeys 2023]LINDENBERGER, P.; SARLIN, P.-E.; POLLEFEYS, M. *LightGlue: Local Feature Matching at Light Speed*. 2023. Disponível em: <https://arxiv.org/abs/2306.13643>.

[Lowe 2004]LOWE, D. G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, Kluwer Academic Publishers, USA, v. 60, n. 2, p. 91–110, nov. 2004. ISSN 0920-5691. Disponível em: <https://doi.org/10.1023/B:VISI.0000029664.99615.94>.

[Lucas e Kanade 1981]LUCAS, B. D.; KANADE, T. An iterative image registration technique with an application to stereo vision. In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981. (IJCAI'81), p. 674–679.

[Martull, Peris e Fukui 2012]MARTULL, S.; PERIS, M.; FUKUI, K. Realistic cg stereo image dataset with ground truth disparity maps. In: *Proceedings of the International*

Workshop on Tracking Methods and Applications (TrakMark). [S.l.: s.n.], 2012. v. 111, n. 430, p. 117–118.

[Mikolajczyk e Schmid 2005]MIKOLAJCZYK, K.; SCHMID, C. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 27, n. 10, p. 1615–1630, 2005.

[Muja e Lowe 2014]MUJA, M.; LOWE, D. G. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 36, n. 11, p. 2227–2240, 2014.

[Oord, Li e Vinyals 2018]OORD, A. van den; LI, Y.; VINYALS, O. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018. Disponível em: <https://api.semanticscholar.org/CorpusID:49670925>.

[Peyré e Cuturi 2020]PEYRé, G.; CUTURI, M. *Computational Optimal Transport*. 2020. Disponível em: <https://arxiv.org/abs/1803.00567>.

[Potje et al. 2024]POTJE, G. et al. Xfeat: Accelerated features for lightweight image matching. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2024. p. 2682–2691.

[Potje et al. 2024]POTJE, G. et al. *XFeat: Accelerated Features for Lightweight Image Matching*. 2024. Disponível em: <https://arxiv.org/abs/2404.19174>.

[Qin, Li e Shen 2018]QIN, T.; LI, P.; SHEN, S. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, v. 34, n. 4, p. 1004–1020, 2018.

[Rublee et al. 2011]RUBLEE, E. et al. Orb: An efficient alternative to sift or surf. In: *Proceedings of the 2011 International Conference on Computer Vision*. USA: IEEE Computer Society, 2011. (ICCV '11), p. 2564–2571. ISBN 9781457711015. Disponível em: <https://doi.org/10.1109/ICCV.2011.6126544>.

[Sand e Teller 2006]SAND, P.; TELLER, S. Particle video: Long-range motion estimation using point trajectories. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. [S.l.: s.n.], 2006. v. 2, p. 2195–2202.

[Sarlin et al. 2020]SARLIN, P.-E. et al. Superglue: Learning feature matching with graph neural networks. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2020. p. 4937–4946.

[Sarıgül 2023]SARıGüL, M. A survey on digital video stabilization. *Multimedia Tools and Applications*, v. 82, p. 40181–40207, 2023. Disponível em: <https://link.springer.com/article/10.1007/s11042-023-14726-1>.

[Schmidt et al. 2024]SCHMIDT, A. et al. Tracking and mapping in medical computer vision: A review. *Medical Image Analysis*, v. 94, p. 103131, 2024. ISSN 1361-8415. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1361841524000562>.

[Sellak, Alj e Salih-Alj 2024]SELLAK, S.; ALJ, Y.; SALIH-ALJ, Y. Monocular visual odometry in mobile robot navigation. In: *2024 IEEE 12th International Symposium on Signal, Image, Video and Communications (ISIVC)*. [S.l.: s.n.], 2024. p. 1–6.

[Shi e Tomasi 1994]SHI, J.; TOMASI. Good features to track. In: *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 1994. p. 593–600.

[Sturm et al. 2012]STURM, J. et al. A benchmark for the evaluation of rgb-d slam systems. In: *Proc. of the International Conference on Intelligent Robot Systems (IROS)*. [S.l.: s.n.], 2012.

[Sun et al. 2018]SUN, D. et al. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume . In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, 2018. p. 8934–8943. Disponível em: <https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00931>.

[Sun et al. 2021]SUN, J. et al. *LoFTR: Detector-Free Local Feature Matching with Transformers*. 2021. Disponível em: <https://arxiv.org/abs/2104.00680>.

[Szeliski 2022]SZELISKI, R. *Computer Vision: Algorithms and Applications*. 2nd. ed. [S.l.]: Springer, 2022. ISBN 978-3-030-34371-2.

[Teed e Deng 2020]TEED, Z.; DENG, J. *RAFT: Recurrent All-Pairs Field Transforms for Optical Flow*. 2020. Disponível em: <https://arxiv.org/abs/2003.12039>.

[Tomasi 1991]TOMASI, C. Detection and tracking of point features. In: . [s.n.], 1991. Disponível em: <https://api.semanticscholar.org/CorpusID:238434334>.

[Tommasini et al. 1998]TOMMASINI, T. et al. Making good features track better. In: . [S.l.: s.n.], 1998. p. 178–183.

[Vaswani et al. 2023]VASWANI, A. et al. *Attention Is All You Need*. 2023. Disponível em: <https://arxiv.org/abs/1706.03762>.

[Wang, Jabri e Efros 2019]WANG, X.; JABRI, A.; EFROS, A. A. *Learning Correspondence from the Cycle-Consistency of Time*. 2019. Disponível em: <https://arxiv.org/abs/1903.07593>.

[Wang, Simoncelli e Bovik 2003]WANG, Z.; SIMONCELLI, E.; BOVIK, A. Multiscale structural similarity for image quality assessment. In: *The Thrity-Seventh Asilomar Conference on Signals, Systems and Computers, 2003*. [S.l.: s.n.], 2003. v. 2, p. 1398–1402 Vol.2.

[Yi et al. 2016]YI, K. M. et al. *LIFT: Learned Invariant Feature Transform*. 2016. Disponível em: <https://arxiv.org/abs/1603.09114>.

[Zhao et al. 2022]ZHAO, X. et al. *ALIKE: Accurate and Lightweight Keypoint Detection and Descriptor Extraction*. 2022. Disponível em: <https://arxiv.org/abs/2112.02906>.

[Zheng et al. 2024]ZHENG, F. et al. Lrpl-vio: A lightweight and robust visual–inertial odometry with point and line features. *Sensors*, v. 24, n. 4, 2024. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/24/4/1322>.

[Zheng et al. 2023]ZHENG, Y. et al. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In: *ICCV*. [S.l.: s.n.], 2023.