Vision-centric State Estimation and Mapping for Visually Challenging Scenarios

Jianeng Wang

St. Edmund Hall
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

2025

Abstract

Reliable 3D scene understanding is essential for enabling autonomous robot operation in complex environments. This thesis addresses the challenges of vision-based state estimation and mapping in challenging scenarios, where conventional methods often struggle due to motion blur, low light, or high dynamic motion. The overarching goal is to develop vision-centric systems that enhance state estimation and scene interpretation by leveraging both novel sensing technologies and robust multi-session mapping strategies.

The first contribution of this thesis presents a stereo event-based visual odometry (VO) system that fully exploits the asynchronous and high-temporal-resolution nature of event cameras. Unlike traditional frame-based VO systems that estimate robot states at a fixed rate in a discrete manner, the proposed system models camera motion as a continuous-time trajectory, enabling per-event state estimation. It combines asynchronous feature tracking with a physically-grounded motion prior to estimate a smooth trajectory that allows pose query at any time within the measurement window. Experimental results demonstrate that this system achieves competitive performance under high-speed motion and challenging lighting conditions, offering a promising alternative for continuous-time state estimation on asynchronous data streams.

The second contribution introduces *Exosense*, a scene understanding system tailored for self-balancing exoskeletons. Building upon a wide field-of-view multi-camera device, Exosense can generate rich, semantically annotated elevation maps that integrate geometry, terrain traversability, and room-level semantics. The system supports indoor navigation by providing reusable environment representations for localization and planning. Designed as a wearable sensing platform, Exosense emphasizes modularity and adaptability, with the potential for integration into a broader wearable sensor ecosystem.

Building upon Exosense, the third contribution is *LT-Exosense*, a change-aware, multi-session mapping system designed for long-term operation in dynamic environments. LT-Exosense incrementally merges scene representations built during repeated traversals of an environment, detects environmental changes, and updates a unified global map. This map representation enables adaptive path planning in response to the dynamic environment. The system supports persistent spatial memory and demonstrates compatibility with different sensor configurations, offering a flexible and scalable foundation for lifelong assistive mobility.

Together, these contributions cover different topics under vision-centric state estimation and mapping in challenging scenarios, including high-speed sensing, semantic scene interpretation, and long-term map maintenance. The thesis opens up new possibilities for robust autonomy on resource-constrained platforms, such as drones and self-balancing exoskeletons, where reliable environmental understanding is critical to safe and intelligent operation.

Vision-centric State Estimation and Mapping for Visually Challenging Scenarios

Jianeng Wang

St. Edmund Hall

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

2025

Declaration

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. I declare that this thesis is entirely my own work, and except where otherwise stated, describes my own research.

Jianeng Wang, St. Edmund Hall

Acknowledgements

It really took me a while to complete this journey and present this thesis to you. Over the past five years, so much has happened. The path was full of twists and turns, with moments of both joy and challenge. But now, at last, I could put a period here. Along the way, I was fortunate to meet mentors who guided me, colleagues who collaborated with me, friends who encouraged me, and family who never stopped believing in me. Because of all these people, I was able to make it and grow in ways I never expected. I would like to express my thankfulness here.

First and foremost, I would like to thank my supervisor, Prof. Maurice Fallon, for taking on my supervision when I was in a dilemma and for welcoming me into the Dynamic Robotics Systems (DRS) Group. I sincerely appreciate all the support and advice received over the years. This thesis would not have been possible without his unwavering help.

I would also like to express my profound gratitude to my supervisor, Prof. Jonathan Gammell, for introducing me to the wonderful world of robotics and for providing insightful guidance on how to approach research. I really appreciate his patience and detailed feedback on my first work, which has significantly influenced my research style.

I owe my special thanks to Dr. Matias Mattamala. I'm fortunate to collaborate with him, and I appreciate all the in-depth technical discussions. They helped me significantly improve my engineering skills.

I would like to acknowledge my colleagues at the Oxford Robotics Institute (ORI). Thank you to Christina Kassab and Lintong Zhang for your research collaboration; Nived Chebrolu for many insightful discussions; Rowan Boarder for socials and casual talks; Ethan Yifu Tao for advice on university matters; Tobit Fletcher for sharing technical expertise—and many more.

My thanks go as well to my collaborators at Wandercraft, Guillaume Burger, Fabio Elnecave, and Marine Petriaux for hosting my visit and supporting the exoskeleton experiments.

I would like to extend my heartfelt thanks to my friends back home: Chenyang, Hongyi, Jincheng, Shengyang, Xiaodong, Xinyu, Yiqiang, Yuhui, Yuyang, Zhaoyang. I appreciate all the joyful moments with you and your encouragement throughout my DPhil.

Lastly, I want to express my forever appreciation to my parents and grandparents for their unconditional support and love. I could never have made it this far without them. I hope this work makes them proud.

To all of you, thank you from the bottom of my heart.

Abstract

Reliable 3D scene understanding is essential for enabling autonomous robot operation in complex environments. This thesis addresses the challenges of vision-based state estimation and mapping in challenging scenarios, where conventional methods often struggle due to motion blur, low light, or high dynamic motion. The overarching goal is to develop vision-centric systems that enhance state estimation and scene interpretation by leveraging both novel sensing technologies and robust multi-session mapping strategies.

The first contribution of this thesis presents a stereo event-based visual odometry (VO) system that fully exploits the asynchronous and high-temporal-resolution nature of event cameras. Unlike traditional frame-based VO systems that estimate robot states at a fixed rate in a discrete manner, the proposed system models camera motion as a continuous-time trajectory, enabling per-event state estimation. It combines asynchronous feature tracking with a physically-grounded motion prior to estimate a smooth trajectory that allows pose query at any time within the measurement window. Experimental results demonstrate that this system achieves competitive performance under high-speed motion and challenging lighting conditions, offering a promising alternative for continuous-time state estimation on asynchronous data streams.

The second contribution introduces *Exosense*, a scene understanding system tailored for self-balancing exoskeletons. Building upon a wide field-of-view multi-camera device, Exosense can generate rich, semantically annotated elevation maps that integrate geometry, terrain traversability, and room-level semantics. The system supports indoor navigation by providing reusable environment representations for localization and planning. Designed as a wearable sensing platform, Exosense emphasizes modularity and adaptability, with the potential for integration into a broader wearable sensor ecosystem.

Building upon Exosense, the third contribution is *LT-Exosense*, a change-aware, multi-session mapping system designed for long-term operation in dynamic environments. LT-Exosense incrementally merges scene representations built during repeated traversals of an environment, detects environmental changes, and updates a unified global map. This map representation enables adaptive path planning in response to the dynamic environment. The system supports persistent spatial

memory and demonstrates compatibility with different sensor configurations, offering a flexible and scalable foundation for lifelong assistive mobility.

Together, these contributions cover different topics under vision-centric state estimation and mapping in challenging scenarios, including high-speed sensing, semantic scene interpretation, and long-term map maintenance. The thesis opens up new possibilities for robust autonomy on resource-constrained platforms, such as drones and self-balancing exoskeletons, where reliable environmental understanding is critical to safe and intelligent operation.

Contents

List of Abbreviations

AER	Address-Event Representation
BA	Bundle Adjustment
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DoF	Degrees of Freedom
EKF	Extended Kalman Filter
ESDF	Euclidean Signed Distance Field
FoVs	Fields of View
GPU	Graphics Processing Unit
IMU	Inertial Measurement Unit
iSAM	Incremental Smoothing and Mapping
MAP	Maximum A Posteriori
MLE	Maximum Likelihood Estimation
MSCKF	Multi-State Constraint Kalman Filter
PnP	Perspective-n-Points
RANSAC	Random Sample Consensus
SAE	Surface of Active Events
SDF	Signed Distance Field
SfM	Structure-from-Motion
SLAM	Simultaneous Localisation And Mapping
ToF	Time-of-Flight
TSDF	Truncated Signed Distance Field
VIO	Visual-Inertial Odometry
VO	Visual Odometry

1 Introduction

Contents

1.1 Motivation

3D scene understanding refers to the fundamental task of using sensor measurements to analyze and interpret the geometric and semantic contents of the environment around a robot or agent [14]. It plays a vital role in enabling a robot to perceive, understand and interact with its surroundings, thereby facilitating high-level operations such as safe navigation in arbitrary 3D environments. The task intersects multiple research areas in robotics and computer vision, including motion tracking [111], depth estimation [75], terrain representation [98][207] and scene graph construction [103][15].

To achieve reliable 3D scene understanding, state estimation and mapping

serves as the foundational techniques that enable a robot to determine its pose and reconstruct its environment. Their ability to accurately model the environment anchors the overall system performance. After two decades of development, the field is increasingly mature. Various sensor modalities have been incorporated into the reconstruction pipelines, such as cameras [275] and LiDARs [274], and solve the problem in an incremental manner via Simultaneous Localisation And Mapping (SLAM) [29] or in a batch manner by using Structure-from-Motion (SfM) [221]. The emphasis on the map representation has also shifted from metric accuracy [30] to a richer representation which incorporates semantics and achieves deeper scene understanding [190].

Vision sensors are widely used for state estimation and mapping due to their rich and dense information content [2]. However, the performance of conventional vision-based methods often degrades under visually challenging scenarios. For instance, the quality of the image content is highly dependent on exposure settings. High exposure times lead to motion blur, while low exposure times result in dark or noisy images, especially under poor lighting conditions. Additionally, vision systems may struggle in high dynamic motion environments, such as high-speed drone flight [71], or on platforms subject to significant acceleration and jerk, such as wearable devices or walking exoskeletons [200].

Although augmenting vision systems with active sensors (e.g., LiDAR) can alleviate individual limitations, this is not always feasible on resource-constrained platforms like drones or wearable robots. As a result, there is growing interest in improving the robustness of vision-centric state estimation and mapping through hardware advances—such as using fisheye cameras [276], multi-camera systems [275], or novel technologies like event cameras [71]. These approaches are promising progress which may help to achieve reliable 3D scene understanding under visually challenging conditions.

1.2 Objective

The main objective of this thesis is to develop vision-centric state estimation and mapping systems that can work robustly in visually challenging scenarios, and in doing so enhance 3D scene understanding. To achieve this, the thesis investigates two systems:

The first system explores the use of a novel vision sensing technology—event cameras. Specifically, it involves the development of an event-based Visual Odometry (VO) pipeline designed to fully exploit the asynchronous and high-temporal-resolution nature of event data. Event cameras are intended to enable more reliable motion estimation in conditions where conventional cameras typically fail, such as high-speed motion or low-light environments.

The second system leverages a variety of vision sensors to improve robustness and versatility. It utilizes a wide field-of-view multi-camera setup, supplemented with RGB-D sensors (i.e., a standard RGB camera paired with a depth sensor), to enable a wide range of functionalities within the 3D scene understanding pipeline. These include accurate sensor state estimation, dense environment reconstruction, semantic terrain labeling, traversability estimation, and multi-session mapping capabilities.

1.3 Contributions

The main contribution of this thesis lies in the design and development of vision-centric state estimation and mapping systems tailored for 3D scene understanding in visually challenging scenarios, as illustrated in Fig. 1.1.

The specific contributions are summarized as follows:

- **An event-based state estimation pipeline:** A complete visual odometry system was developed leveraging the unique properties of event cameras. By fully exploiting the asynchronous nature and high temporal resolution of event data, the proposed system achieves competitive performance in estimating high-dynamic drone motion compared to state-of-the-art methods.

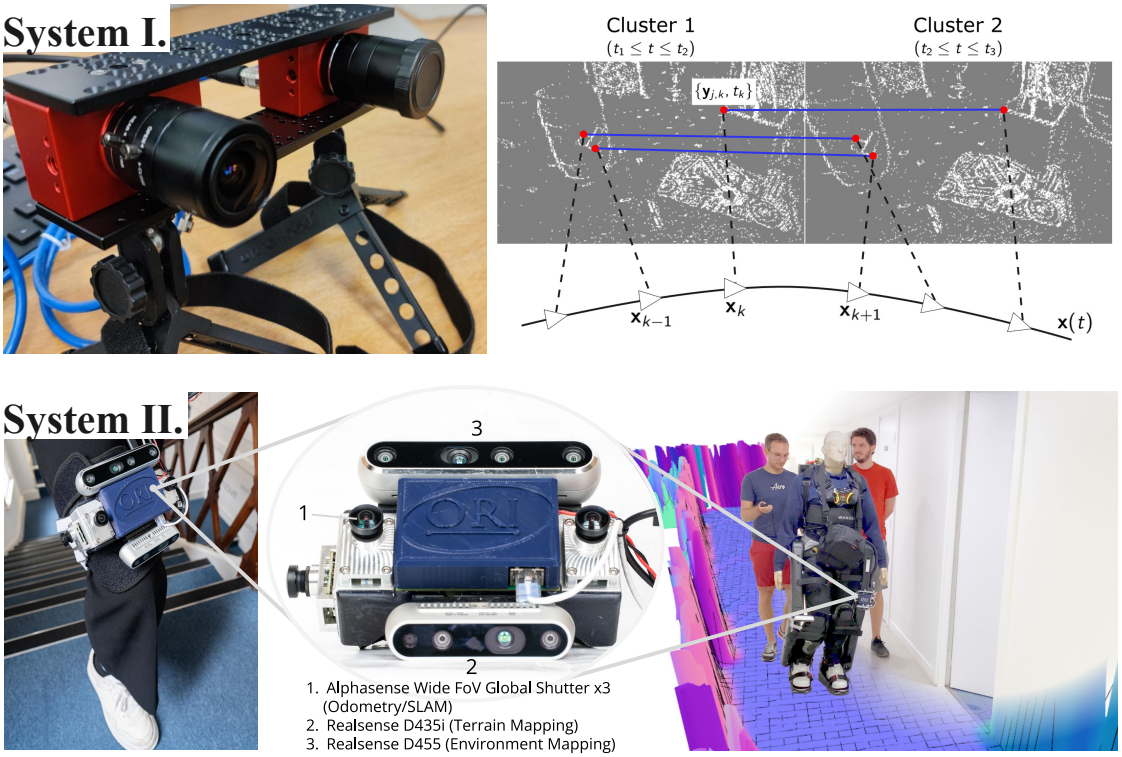


Figure 1.1: Vision-centric state estimation and mapping systems presented in this DPhil thesis. **I.** shows a stereo event camera setup (*Left*), which was used to develop the VO pipeline (*Right*). **II.** is a vision-centric scene understanding system, Exosense, which is equipped with various types of visual sensors (*Middle*) for use either leg-mounted (*Left*) or on an exoskeleton (*Right*).

- **A vision-centric 3D scene understanding system for an exoskeleton:** A robust system is presented based on a wide field-of-view multi-camera configuration. It is capable of generating globally consistent and semantically enhanced elevation maps, integrating both terrain semantics and traversability information. This system is designed for operation in indoor environments, particularly targeting walking exoskeleton use cases.

- **Multi-session mapping with a multi-camera setup:** Building upon this multi-camera system, the work was further extended to support multi-session mapping capabilities. This enhancement enables deployment across a wide range of visually challenging environments, improving robustness and generalizability regardless of the application scenario.

1.4 Publications

The list of publications related to this DPhil thesis is shown as follows:

1.4.1 First-author Publications

- **Wang, J.** and Gammell, J. D. (2022). A Event-Based Stereo Visual Odometry With Native Temporal Resolution via Continuous-Time Gaussian Process Regression. *IEEE Robotics and Automation Letters (RA-L)*. [253]

- **Wang, J.**, Mattamala, M., Kassab, C., Burger, G., Elnecave, F., Zhang, L., Petriaux, M., and Fallon, M. (2025). "Exosense: A Vision-Based Scene Understanding System for Exoskeletons". *IEEE Robotics and Automation Letters (RA-L)* [254]

- **Wang, J.**, Mattamala, M., Kassab, C., Chebrolu, N., and Fallon, M. (2025) "LT-Exosense: A Vision-centric Multi-session Mapping System for Lifelong Safe Navigation of Exoskeletons" *IEEE Robotics and Automation Letters (RA-L)* (To Be Submitted)

1.4.2 Co-authored Publication

- Wang, Z., Bian, W., Li, X., Tao, Y. **Wang, J.**, Fallon, M, and Prisacariu, V. (2025). "Seeing in the Dark: Benchmarking Egocentric 3D Vision with the Oxford Day-and-Night Dataset". *Conference on Neural Information Processing Systems (NeurIPS)* (Under Review) [259]

1.5 Thesis Outline

This thesis will be presented in an "integrated thesis" style, consisting of peer-reviewed publications, alongside in-depth discussion and additional details of each work.

The remainder of the thesis is structured as follows:

- **Chapter 2 – Background** introduces fundamental concepts, theory and requisite notation related to this thesis.
- **Chapter 3 – Literature Review** surveys the related works on state estimation, mapping and scene understanding.
- **Chapter 4 – Event-based Visual Odometry** presents a continuous-time event-based stereo visual odometry system. This chapter demonstrates the approach of leveraging the asynchronous nature of event data to achieve per event state estimation.
- **Chapter 5 – Vision-Based Scene Understanding For Exoskeletons** presents *Exosense*, a scene understanding system centered around a multi-camera setup designed primarily for an exoskeleton. The chapter describes the system's capabilities of estimating robot state, mapping its surroundings, analyzing the terrain and localizing within a prior map, which can reliably function while the exoskeleton is dynamically walking.
- **Chapter 6 – Multi-session Visual Mapping** builds on top of the core Exosense system to enable multi-session mapping capability, which is able to fuse spatial knowledge across multiple sessions, detect environmental changes, and update a lifelong global map that empower the long-term usage of exoskeletons.
- **Chapter 7 – Conclusion and Future Work** summarizes the key findings of this DPhil thesis, discusses the limitations of the proposed systems and suggests future directions of this line of research.

2
Background

Contents

This chapter presents the fundamental concepts and theoretical foundations that underpin the work in this thesis and are referenced throughout the subsequent chapters.

Sec. 2.1 introduces the basic notation and conventions adopted in this thesis.

Sec. 2.2 discusses the principles of 3D rigid-body transformations and the associated mathematical operations. Sec. 2.3 gives the background knowledge of the optimization problem in state estimation. Sec. 2.4 provides an overview of the 3D metric map representations explored during this work. Finally, Sec. 2.5 outlines the primary vision sensors utilized in the projects.

2.1 Notations

This thesis follows the conventions defined in Tab. 2.1 to distinguish different mathematical objects such as scalars, vectors and matrices.

Table 2.1: Notation used for mathematical objects.

Quantity	**Description**	**Example**
Scalars	Upper/Lowercase italic	The optimization cost: J
Vectors	Lowercase bold	The 3D point: $\mathbf{p}$
Matrices	Uppercase bold/ bold calligraphic	The rotation matrix: $\mathbf{R}$
Sets/Manifolds	Uppercase calligraphic	The landmark set: $\mathcal{P}$
Common Number Set	Uppercase blackboard bold	Real number set: $\mathbb{R}$
Frame	Lowercase text	The *map* frame: map
	(legacy expression)	The *map* frame: $\underrightarrow{\mathcal{F}}_{\text{map}}$

2.2 3D Rigid-body Transformation

This section introduces the fundamental concepts of 3D rigid-body transformations, their applications in robotics and computer vision, and a brief overview of relevant Lie theory concepts that are used throughout this thesis.

2.2.1 Frame

A *frame* refers to a coordinate system used to represent the pose (i.e., position and orientation) of an object or robot expressed with respect to (w.r.t.) another coordinate system. Typically, multiple frames are defined relative to a common *reference frame*. When the reference frame changes, the poses of all associated

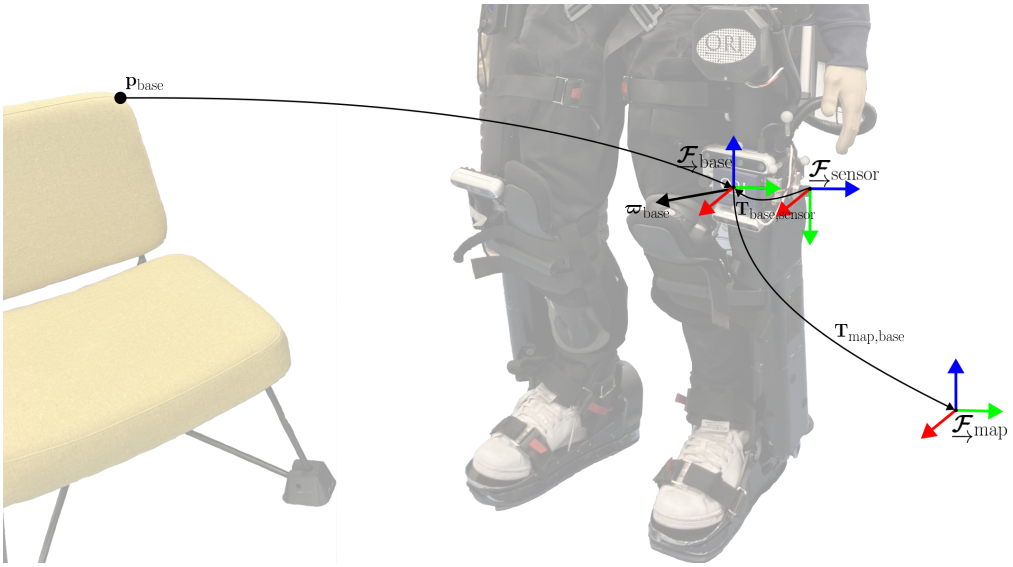


Figure 2.1: Example reference frames and frame-endowed notation used in this thesis. The axes colour follows the convention with the x-axis in **red**, y-axis in **green**, and z-axis in **blue**. The map frame $\underrightarrow{\mathcal{F}}_{\text{map}}$ is fixed. The base frame $\underrightarrow{\mathcal{F}}_{\text{base}}$ is associated to the device and moves with respect to the fixed frame, while the sensor frame $\underrightarrow{\mathcal{F}}_{\text{sensor}}$ is rigidly connected to $\underrightarrow{\mathcal{F}}_{\text{base}}$. The 6DOF velocity expressed in $\underrightarrow{\mathcal{F}}_{\text{base}}$ is denoted as $\boldsymbol{\varpi}_{\text{base}}$ and landmarks seen by it is written as $\mathbf{p}_{\text{base}}$.

frames would change accordingly to reflect their new relationships—this process is referred to as a *transformation*.

Fig. 2.1 illustrates example coordinate frames and related notations used in the remainder of the thesis. It is noted that all frames in this thesis follow the right-hand rule convention.

2.2.2 Pose and Transformation Matrix

For a rigid body, $\underrightarrow{\mathcal{F}}_{\text{body}}$, its pose describes its position and orientation w.r.t. a reference frame, $\underrightarrow{\mathcal{F}}_{\text{ref}}$. This pose is commonly represented using a homogeneous transformation matrix, which encodes both rotation and translation in a single matrix form. In 3D space, this matrix is expressed as:

$$\mathbf{T}_{\text{ref,body}} = \begin{bmatrix} \mathbf{R}_{\text{ref,body}} & \mathbf{t}_{\text{ref}}^{\text{body,ref}} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4\times4}, \tag{2.1}$$

where $\mathbf{R}_{\text{ref,body}} \in \mathbb{R}^{3\times3}$ is the rotation matrix, which belongs to the *Special Orthogonal Group*, $SO(3)$, and $\mathbf{t}_{\text{ref}}^{\text{body,ref}} \in \mathbb{R}^{3\times1}$ is the translation vector, and also represent the position of the $\underrightarrow{\mathcal{F}}_{\text{body}}$ in $\underrightarrow{\mathcal{F}}_{\text{ref}}$.

Although the transformation matrix is parameterized by 12 entries (9 from $\mathbf{R}_{\text{ref,body}}$ and 3 from $\mathbf{t}_{\text{ref}}^{\text{body,ref}}$), the underlying pose belongs to the *Special Euclidean Group* $SE(3)$ and can be fully characterized by a 6 Degrees of Freedom (DoF) vector—three for translation and three for rotation. The mathematical structure and parameterization of $SE(3)$, along with its associated Lie algebra, will be further discussed in Sec. 2.2.3.

Applications

A common use of transformation matrices is to express the spatial relationship between a point or pose from one coordinate frame to another.

- **Transforming a point:**

  Given a 3D point $\mathbf{p}_{\text{base}} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ in the base frame, $\underrightarrow{\mathcal{F}}_{\text{base}}$, the transformation from $\underrightarrow{\mathcal{F}}_{\text{base}}$ to map frame, $\underrightarrow{\mathcal{F}}_{\text{map}}$, can be applied to express this point in map frame via

  $$\mathbf{p}_{\text{map}} = \mathbf{T}_{\text{map,base}} \begin{bmatrix} \mathbf{p}_{\text{base}} \\ 1 \end{bmatrix}, \tag{2.2}$$

  where $\begin{bmatrix} \mathbf{p}_{\text{base}} \\ 1 \end{bmatrix}$ represents the point in its homogeneous coordinates.

- **Transforming a pose:**

  Given the robot pose as base frame expressed in the odometry frame, $\mathbf{T}_{\text{odom,base}}$, if the transformation from odom to map is available, the base can subsequently be expressed in the map frame via

  $$\mathbf{T}_{\text{map,base}} = \mathbf{T}_{\text{map,odom}} \mathbf{T}_{\text{odom,base}} \tag{2.3}$$

2.2.3 Lie Group and Lie Algebra Geometry

A *Lie group* is a mathematical structure that combines the properties of a group with those of a differentiable manifold [87].

As a group, a Lie group is a non-empty set equipped with a binary operation (e.g., matrix multiplication) that satisfies the group axioms: closure, associativity, the existence of an identity element and an inverse element. Taking a transformation

matrix as an example, it can perform matrix multiplication, which is a binary operation and associative. Amongst all transformation matrices, one particular matrix is the identity matrix, and the inverse can be computed for each matrix in the group.

As a differentiable manifold, each element of a Lie group can be locally parameterized by a continuous variable in a Euclidean space, $\mathbb{R}^n$. The space of these parameters forms the associated *Lie algebra*. A small variation in this parameter can also result in a small variation in the corresponding Lie group element. Hence, the mapping from the Lie algebra to the Lie group is differentiable. One can take derivatives of functions that involve Lie group elements by differentiating with respect to their Lie algebra representation. Take the rotation matrix as an example, it can be parameterized by a 3D vector using Euler angles or an axis-angle representation [11]. If the mapping from the 3D vector to a rotation matrix is treated as a function, it is differentiable and the derivative gives the Jacobian of the rotation matrix.

In this thesis, two special Lie groups are of primary interest:

- The *Special Orthogonal Group*, $SO(3)$, which represents 3D rotations.
- The *Special Euclidean Group*, $SE(3)$, which represents 3D poses (i.e., rotation and translation).

Lie Group

The *Special Orthogonal Group*, $SO(3)$, is defined as a set of valid 3D rotation matrices,

$$SO(3) = \left\{ \mathbf{R} \in \mathbb{R}^{3\times 3} \mid \mathbf{R}\mathbf{R}^T = \mathbf{1}, \det \mathbf{R} = 1 \right\}, \tag{2.4}$$

which suggests all valid rotation matrices are orthonormal.

The *Special Euclidean Group*, $SE(3)$, is the set of valid 3D rigid-body transformation matrices,

$$SE(3) = \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \mid \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3 \right\}, \tag{2.5}$$

which contains both rotation and translation components.

Lie Algebra

Every Lie group element has a corresponding Lie algebra element. Lie algebra represents the tangent space of its corresponding Lie group, making it essential in linearizing $SE(3)$ and $SO(3)$ estimation functions.

A Lie algebra consists of a vector space over some field and has a binary operation known as Lie bracket that satisfies closure, bilinearity, alternation, and Jacobi identity properties [11].

The Lie algebra of $SO(3)$ is defined as

$$\begin{aligned} \text{vectorspace:} \quad & \mathfrak{so}(3) = \left\{ \boldsymbol{\Phi} = \boldsymbol{\phi}^{\wedge} \in \mathbb{R}^{3\times 3} \mid \boldsymbol{\phi} \in \mathbb{R}^3 \right\}, \\ \text{field:} \quad & \mathbb{R}, \\ \text{Lie bracket:} \quad & \left[\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2\right] = \boldsymbol{\Phi}_1\boldsymbol{\Phi}_2 - \boldsymbol{\Phi}_2\boldsymbol{\Phi}_1, \end{aligned} \tag{2.6}$$

where $\boldsymbol{\phi}$ is the tangent vector which parameterizes the rotation and $(\cdot)^{\wedge}$ is the skew-symmetric operator for $\mathbb{R}^3$

$$\boldsymbol{\phi}^{\wedge} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}^{\wedge} = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 3}, \quad \boldsymbol{\phi} \in \mathbb{R}^3. \tag{2.7}$$

The Lie algebra of $SE(3)$ is defined as

$$\begin{aligned} \text{vectorspace:} \quad & \mathfrak{se}(3) = \left\{ \boldsymbol{\Xi} = \boldsymbol{\xi}^{\wedge} \in \mathbb{R}^{4\times 4} \mid \boldsymbol{\xi} \in \mathbb{R}^6 \right\}, \\ \text{field:} \quad & \mathbb{R}, \\ \text{Lie bracket:} \quad & \left[\boldsymbol{\Xi}_1, \boldsymbol{\Xi}_2\right] = \boldsymbol{\Xi}_1\boldsymbol{\Xi}_2 - \boldsymbol{\Xi}_2\boldsymbol{\Xi}_1, \end{aligned} \tag{2.8}$$

where $\boldsymbol{\xi}$ is the tangent vector that parameterizes the pose, with first three entries for translation and last three entries for rotation. $(\cdot)^{\wedge}$ is overloaded here for $\mathbb{R}^6$

$$\boldsymbol{\xi}^{\wedge} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}^{\wedge} & \boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4}, \quad \boldsymbol{\rho}, \boldsymbol{\phi} \in \mathbb{R}^3. \tag{2.9}$$

It is noted that both Eq. (2.7) and Eq. (2.9) have the inverse operation denoted as $(\cdot)^{\vee}$.

Conversion between Lie Group and Lie Algebra

In Lie theory, the exponential map is the operation to convert Lie algebra to Lie group, and its inverse converts Lie group to Lie algebra. Mathematically, they correspond to matrix exponentials and logarithms. For a square matrix $\mathbf{M} \in \mathbb{R}^{m\times m}$, the two operations are defined as

- **Matrix Exponential:**

$$\exp(\mathbf{M}) = \mathbf{1} + \mathbf{M} + \frac{1}{2!}\mathbf{M}^2 + \frac{1}{3!}\mathbf{M}^3 + \cdots = \sum_{n=0}^{\infty} \frac{1}{n!}\mathbf{M}^n, \tag{2.10}$$

  where the conversions from Lie algebra to Lie group are written as

$$\mathbf{R} = \exp(\boldsymbol{\phi}^\wedge), \quad \text{where } \mathbf{R} \in SO(3) \text{ and } \boldsymbol{\phi}^\wedge \in \mathfrak{so}(3), \tag{2.11}$$

  and

$$\mathbf{T} = \exp(\boldsymbol{\xi}^\wedge), \quad \text{where } \mathbf{T} \in SE(3) \text{ and } \boldsymbol{\xi}^\wedge \in \mathfrak{se}(3). \tag{2.12}$$

- **Matrix Logarithm:**

$$\ln(\mathbf{M}) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n}(\mathbf{M} - \mathbf{1})^n, \tag{2.13}$$

  where the conversions from Lie group to Lie algebra tangent vector are written as

$$\boldsymbol{\phi} = \ln(\mathbf{R})^\vee, \quad \text{where } \mathbf{R} \in SO(3) \text{ and } \boldsymbol{\phi} \in \mathbb{R}^3, \tag{2.14}$$

  and

$$\boldsymbol{\xi} = \ln(\mathbf{T})^\vee, \quad \text{where } \mathbf{T} \in SE(3) \text{ and } \boldsymbol{\xi} \in \mathbb{R}^6. \tag{2.15}$$

Adjoint

The *Adjoint* representation allows a Lie group to directly operate on its own Lie algebra tangent vector. The Adjoint of $SO(3)$ is itself, while for $SE(3)$, it is defined as

$$\boldsymbol{\mathcal{T}} = \mathrm{Ad}(\mathbf{T}) = \begin{bmatrix} \mathbf{R} & \mathbf{t}^\wedge\mathbf{R} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \in \mathbb{R}^{6\times 6}, \quad \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in SE(3). \tag{2.16}$$

Jacobian

For a vector function with multiple variables,

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(x_1, x_2, \cdots x_n) \\ f_2(x_1, x_2, \cdots x_n) \\ \vdots \\ f_m(x_1, x_2, \cdots x_n) \end{bmatrix}, \tag{2.17}$$

its *Jacobian* matrix is the matrix spanning all its first-order partial derivatives

$$\mathbf{J}(\mathbf{f}(\mathbf{x})) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}. \tag{2.18}$$

In the context of Lie groups, Jacobians describe how small perturbations in the Lie algebra affect the corresponding group elements after linearization. These Jacobians are crucial in optimization-based state estimation (e.g., bundle adjustment or pose graph optimization), where objectives are often defined in the group but updates are applied in the tangent space.

Depending on where the perturbation is applied to the original Lie group, the Jacobian matrix can be classified as a left or right Jacobian. This thesis considers the case of left-multiplying the perturbation to the original Lie group, hence all representations are in left Jacobian format.

For $SO(3)$, the Jacobian matrix is about the exponential map function in Eq. (2.11), where the tangent vector is now considered as the perturbation. The Jacobian matrix then takes the form of

$$\mathbf{J} = \sum_{n=0}^{\infty} \frac{1}{(n+1)!} (\boldsymbol{\phi}^\wedge)^n \in \mathbb{R}^{3\times 3} \tag{2.19}$$

Likewise, in $SE(3)$, the Jacobian matrix is about Eq. (2.12), and is written as

$$\boldsymbol{\mathcal{J}} = \sum_{n=0}^{\infty} \frac{1}{(n+1)!} (\boldsymbol{\xi}^\curlywedge)^n \in \mathbb{R}^{6\times 6}, \tag{2.20}$$

where $(\cdot)^\curlywedge$ converts $\mathbb{R}^6$ to $\mathbb{R}^{6\times 6}$

$$\boldsymbol{\xi}^\curlywedge = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix}^\curlywedge = \begin{bmatrix} \boldsymbol{\phi}^\wedge & \boldsymbol{\rho}^\wedge \\ \mathbf{0} & \boldsymbol{\phi}^\wedge \end{bmatrix} \in \mathbb{R}^{6\times 6}. \tag{2.21}$$

This section presents key geometric foundations of 3D rigid-body transformations, which are extensively employed in Chapter 4. To maintain conciseness, some of the detailed derivations are omitted. For a more thorough discussion, the reader is encouraged to consult [11, 124].

2.3 Optimization in State Estimation

State estimation problems are commonly formulated as nonlinear least-squares optimization tasks, in which a cost function defined over residual errors is minimized to obtain the optimal state estimates. This general formulation can be written as:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} J(\mathbf{e}(\mathbf{x}, \mathbf{y})), \tag{2.22}$$

where $J(\cdot)$ is the cost function and $\mathbf{e}(\mathbf{x}, \mathbf{y})$ is the error function, which computes the residual between the predicted state, $\mathbf{x}$, against the sensor-derived measurement, $\mathbf{y}$.

The remainder of this section approaches the problem from a probabilistic perspective, establishes its equivalence to a least-squares formulation under certain assumptions, and finally introduces factor graph-based optimization as a powerful and structured framework for solving state estimation problems.

2.3.1 Maximum a Posteriori Estimation

Maximum A Posteriori (MAP) estimation provides a probabilistic framework for state estimation. It aims to determine an optimal estimate given a set of observations. Specifically, it seeks the state that maximizes the posterior distribution conditioned on the observed measurements. By definition [43], MAP estimation is formulated as

$$\hat{\mathbf{x}}_{\text{MAP}} = \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}), \tag{2.23}$$

where $p(\mathbf{x}|\mathbf{y})$ is the posterior distribution of $\mathbf{x}$ given $\mathbf{y}$. This posterior distribution can be further characterized by Bayes rule [241],

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})}, \tag{2.24}$$

where $p(\mathbf{y}|\mathbf{x})$ is the likelihood function, representing the probability of observing $\mathbf{y}$ given a particular state $\mathbf{x}$, and $p(\mathbf{x})$ and $p(\mathbf{y})$ are known as prior distribution and marginal distribution, used to model the individual distribution without knowing others.

Since $p(\mathbf{y})$ does not depend on $\mathbf{x}$, it can be treated as a constant during optimization. Therefore, the MAP estimate simplifies to:

$$\hat{\mathbf{x}}_{\text{MAP}} = \arg\max_{\mathbf{x}} p(\mathbf{y}|\mathbf{x})p(\mathbf{x}). \tag{2.25}$$

2.3.2 Maximum Likelihood Estimation

When the prior distribution is uniform (i.e., constant across all possible states), the MAP estimation reduces to *Maximum Likelihood Estimation (MLE)*. The MLE aims to find the state that maximizes the likelihood of the observed measurements, and is given by:

$$\hat{\mathbf{x}}_{\text{MLE}} = \arg\max_{\mathbf{x}} p(\mathbf{y}|\mathbf{x}). \tag{2.26}$$

Unlike MAP estimation, which incorporates a prior distribution, MLE assumes no prior knowledge about the state and instead seeks the state that best explains the observed data. In essence, MLE directly maximizes the likelihood function, as shown in Eq. (2.26), by finding the state that best fits the incoming measurements.

2.3.3 Linking Maximum a Posteriori Estimation to Least Squares Estimation

The negative log likelihood function, $f = -\ln(\cdot)$, is monotonically decreasing. When its variable reaches the maximum, the function gives the minimum value. By taking the negative log likelihood of $p(\mathbf{x}|\mathbf{y})$, the MAP estimation becomes a problem of finding the optimal state that minimizes the negative log likelihood of the posterior distribution of the state conditioned on the measurement:

$$\hat{\mathbf{x}}_{\text{MAP}} = \arg\min_{\mathbf{x}} -\ln(p(\mathbf{x}|\mathbf{y})). \tag{2.27}$$

With more than one state to estimate, MAP estimation is the product of the individual posterior distribution of each state:

$$\hat{\mathbf{x}}_{\text{MAP}} = \arg\min_{\mathbf{x}} \prod_i -\ln(p(\mathbf{x}_i|\mathbf{y}_i)). \tag{2.28}$$

Assuming that the measurements and prior distributions of the states follow Gaussian distributions, the MAP estimation can be expressed as a least-squares optimization problem. This leads to the standard sum of least squares formulation [11]:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_i \|\mathbf{y}_i - \mathbf{h}(\mathbf{x}_i)\|^2_{\mathbf{\Sigma}_i} + \|\mathbf{x}_0 - \check{\mathbf{x}}_0\|^2_{\mathbf{\Sigma}_0}, \tag{2.29}$$

where $\mathbf{h}(\cdot)$ is the measurement model given the state. $\check{\mathbf{x}}_0$ is the prior measurement of the first state. $\mathbf{\Sigma}$ is the covariance matrix associated with the corresponding measurement of each state.

If the prior distribution is uniform, this implies that there is no prior knowledge about the state estimate before the measurement arrives. The MLE can also be written in least squares form by dropping the prior term, $\|\mathbf{x}_0 - \check{\mathbf{x}}_0\|^2_{\mathbf{\Sigma}_0}$, in Eq. (2.29).

In real-world state estimation problems, the error function in Eq. (2.29) can represent various types of measurement discrepancies, such as image feature reprojection error [126] or photometric loss [53], among others. These error functions are typically non-linear. As a result, the problem becomes a non-linear least squares estimation problem, which requires an iterative solution approach. Standard gradient descent-based methods, such as Gauss-Newton (GN) or Levenberg-Marquardt (LM) approaches, are commonly applied to solve for the optimal state estimate [76].

The Gauss-Newton method assumes that the squared error function is locally approximated by a quadratic surface. Around an operating point, the state vector is perturbed, and the cost function is linearized by computing its first-order Taylor expansion [11]. The optimal perturbation is then obtained by solving a linear system where the gradient is set to zero. By iteratively updating the state with this perturbation, the algorithm converges toward a local minimum of the cost function. However, Gauss-Newton may fail to converge if the initial estimate is far from the local minimum or if the system is poorly conditioned [48].

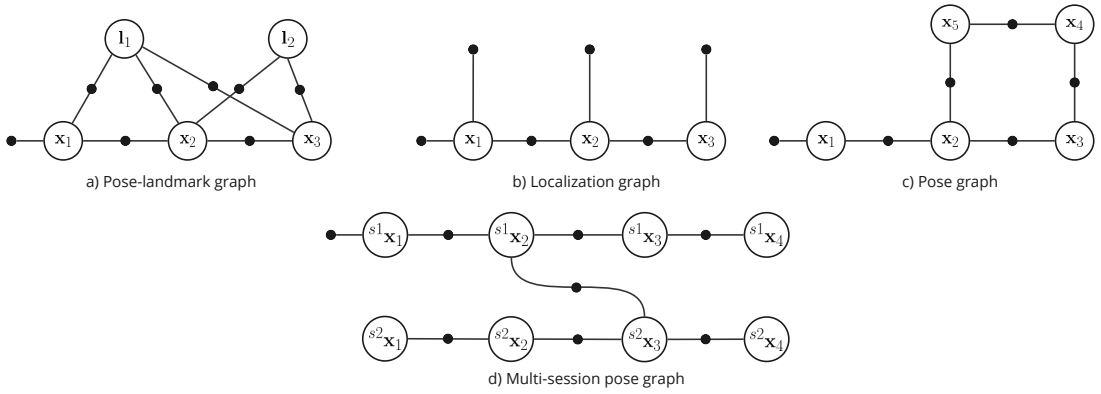


Figure 2.2: Factor graph examples used in robot state estimation. In the factor graph, factors (black dots) are always connected to the nodes (circles) via edges, which indicate that a particular factor depends on a particular set of variables.**a)** Pose-landmark graph, **b)** Localization graph, **c)** Pose graph, **d)** Multi-session pose graph.

To improve robustness in such scenarios, the Levenberg-Marquardt (LM) method introduces a damping factor into the optimization. This modification restricts updates to directions that reduce the cost function, thereby avoiding large, unstable steps that could lead to divergence [76]. As a result, LM achieves better convergence properties, particularly during early iterations when the linear approximation is less accurate.

2.3.4 Factor Graph based Optimization

A *factor graph* is a bipartite graph composed of factors and nodes [239]. In the context of state estimation, the nodes represent the unknown states to be estimated, while the factors represent functions that encode constraints or observations involving subsets of these states.

Referring to Eq. (2.29), each factor corresponds to an error function that enforces consistency between a predicted state and a measurement, assuming Gaussian-distributed priors and measurement noise. Factor graphs provide a flexible and modular framework to incorporate heterogeneous types of measurements and constraints, making them well-suited for modeling real-world estimation problems.

Fig. 2.2 illustrates several example factor graphs commonly used in state estimation tasks.

The optimization of a factor graph involves adjusting the values of each node to best satisfy the constraints imposed by the factors, effectively minimizing the joint error across the entire graph. Furthermore, real-time applications require performing optimization in an incremental manner as data comes in sequentially. Several efficient open-source libraries are available for performing such incremental optimization, including g2o [129] and GTSAM [42], where algorithms like HOG-Man [82] and iSAM [114] can be adopted.

Optimization plays a foundational role in state estimation, enabling accurate and consistent estimates of system states. It is a technique which is heavily used throughout this DPhil thesis. This section presents the fundamental knowledge for estimator design, which are the core techniques for the work shown in Chapter 4. Moreover, the factor graph-based optimization framework introduced here is adopted in the system designs presented in Chapter 5 and Chapter 6.

2.4 3D Metric Map Representation

A map integrates sensor data to model the environment in which a robot operates [29]. Maps can be broadly classified into two categories: metric and topological. Metric maps represent the geometric properties of the environment, while topological maps encode the connectivity or relational structure between different locations [240].

This thesis primarily focuses on 3D metric maps, which are further categorized into three main types based on their representation of the scene: point-based, surface-based, and volumetric representations, as illustrated in Fig. 2.3.

2.4.1 Point-based Representation

A point-based representation (Fig. 2.3**a**) models the 3D environment using a collection of discrete points, commonly referred to as a point cloud. These points can be directly obtained from range sensors such as LiDAR, or derived via the reprojection of depth images.

Point clouds are straightforward to acquire and can be easily converted to other types of map representations. However, they lack explicit information about surface

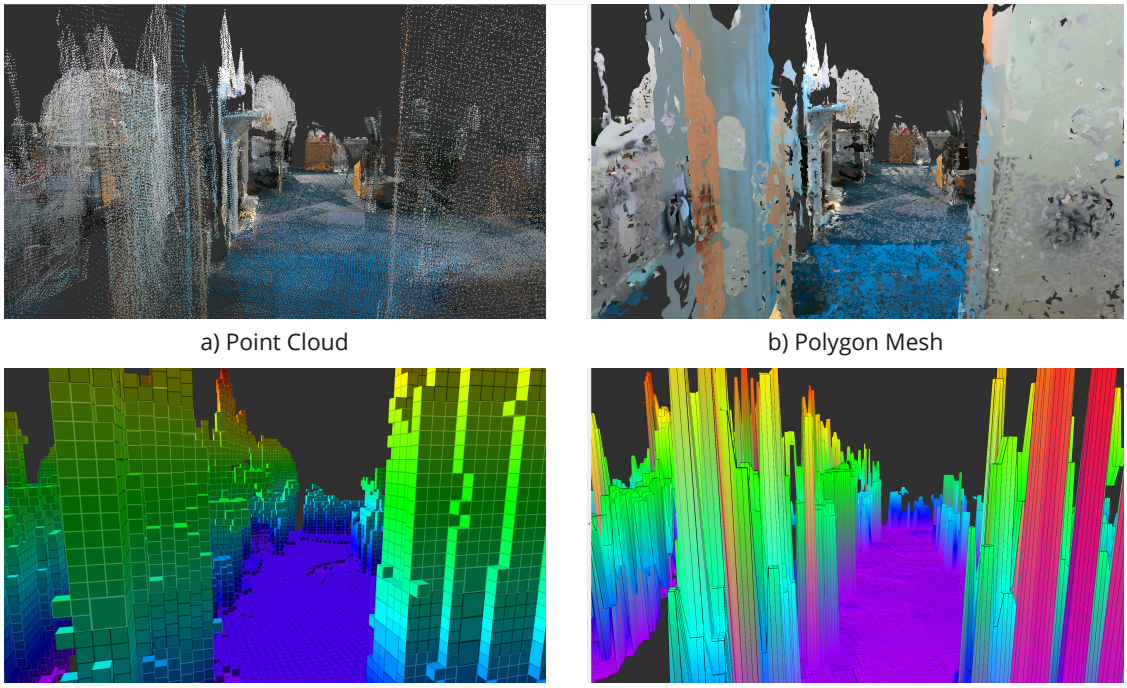


Figure 2.3: 3D metric map representations investigated in this work. The four subfigures show the same office scene but using different map representations. **a)** Point cloud (Sec. 2.4.1) represents the scene as a collection of 3D points, which are measurements directly coming from the depth sensor. **b)** Polygon mesh (Sec. 2.4.2) is created by connecting the points as a set of triangular faces. **c)** Octomap (Sec. 2.4.3) partitions the space using an octree structure to indicate the occupancy information. Here only the occupied blocks are visualized and colored by their heights. **d)** Elevation map (Sec. 2.4.3) is a 2.5D representation that represents the space as a 2D grid, where each cell contains the corresponding elevation information.

connectivity and occupancy, which limits their direct applicability in downstream robotic tasks such as navigation, path planning, or object manipulation. Additionally, point clouds often contain redundant measurements of the same location, making them inefficient and cumbersome when modeling large-scale environments.

2.4.2 Surface-based Representation

Surface-based representations define the frontier of known surfaces, making them effective for modeling the shape and contours of 3D objects. They are widely used in computer graphics and geometric modeling due to their descriptive power. However, they do not explicitly provide free space information, which is critical for robot path planning and navigation.

This section introduces two common types of surface representations: surfels and polygon meshes.

Surfel

Surfels is an abbreviation for surface elements, which essentially are point clouds with more attributes. Each surfel typically contains a 3D coordinate of the surface centroid and its normal. Additional attributes may include surface radius, color, and statistics of local point distribution. Surfels are often used for efficient, scalable 3D mapping and are well-suited for online applications due to their lightweight nature.

Polygon Mesh

A polygon mesh represents a 3D surface using a collection of vertices, edges and faces. Vertices are connected by edges, and a closed set of edges forms a face. Faces are commonly triangular or quadrilateral. Polygon meshes are highly expressive and widely used for detailed surface reconstruction. An example of a triangular mesh reconstruction is shown in Fig. 2.3**b**.

2.4.3 Volumetric Representation

Volumetric representations model the 3D structure of an environment by partitioning space into grids, either explicitly or implicitly. These grids capture volumetric occupancy or the estimated distance to a surface, providing a rich foundation for scene reconstruction and robotic interaction.

In explicit volumetric representations, the 3D environment is voxelized into identical 3D cubes, known as voxels. Each voxel contains its occupancy information, indicating whether it is occupied, free, or unknown. The partitioning scheme can be further improved using structures like octree [98], voxel hashing [177] or wavelet compression [207] for memory and computational efficiency. For relatively flat environments with few or no overhanging surfaces, a more lightweight representation such as an elevation map can be used. An elevation map partitions the world into a 2D grid, where each cell stores a height value, effectively forming a 2.5D

representation. Fig. 2.3**c** and **d** show the same environment modeled using a 3D occupancy map and an elevation map, respectively.

Alternatively, the world can be partitioned into a 3D voxel grid, where each cell is characterized by the Signed Distance Field (SDF). The SDF computes the perpendicular distance from a given point to the boundary of a set in a metric space, where the boundary can be the surface of a geometric shape. One commonly used variant is the Truncated Signed Distance Field (TSDF), which computes projective distances along the sensor ray within a small truncation region around observed surfaces (e.g., building TSDF in the camera frame and all SDF is computed along the camera ray). In the simplest form, given the camera view point $\mathbf{o}$ and a custom 3D point $\mathbf{p}$ along the ray from $\mathbf{o}$ to the nearest surface $\mathcal{S}$, the distances of the view point to the custom point and surface are denoted as $d_{\mathbf{o}\text{-to-}\mathbf{p}}$ and $d_{\mathbf{o}\text{-to-}\mathcal{S}}$, respectively. The SDF can then be formulated as

$$SDF = d_{\mathbf{o}\text{-to-}\mathcal{S}} - d_{\mathbf{o}\text{-to-}\mathbf{p}} \tag{2.30}$$

When a point on the ray is inside the surface, its SDF value is negative, when it is outside the boundary, the SDF value is positive. Alternatively, TSDF can also be converted to a Euclidean Signed Distance Field (ESDF) [185], where every cell contains the Euclidean signed distance to the nearest obstacle in a global map frame. Since SDF-based volumetric representation implicitly captures the surface information, it can retrieve the surface-based representation using zero crossings of the iso-surface [22].

3D metric maps provide a rich geometric representation of the robot's operating environment and serve as a foundation for high-level robotic tasks such as navigation, planning, and interaction. The aforementioned map representations have been investigated and tested throughout the works presented in this thesis. In particular, Chapter 5 incorporates both elevation maps and submapping structures [24], offering a step toward long-term, reliable operation of exoskeletons. Meanwhile, Chapter 6 adopts a point-based representation for multi-session mapping, yielding globally consistent maps that can be further converted into alternative forms to support navigation and scene understanding tasks.

Figure 2.4: Vision sensors used in the works presented in this thesis. **a)** shows a example of frame-based camera—Sevensense Core Research multi-camera visual-inertial sensor (Source: Sevensense [224]). **b)** is a Realsense D435i RGB-D camera (Source: Realsense [107]). **c)** is a Prophesee event camera without a lens attached (Source: Prophesee [196]).

2.5 Vision Sensors

Vision sensors capture and process visual information from the environment, enabling the extraction of useful features for a wide range of robotic applications. This section specifically focuses on its sub-category—sensors termed as cameras, which broadly speaking are devices that capture 2D intensity information of the scene in the form of pixel arrays. Fig. 2.4 illustrates the various cameras utilized in this research. Their roles, characteristics, and modeling approaches will be discussed in the remainder of this section.

2.5.1 Frame-based Camera

The frame-based camera is one of the most commonly used vision sensors that is low-cost and provides rich visual information. It is a passive sensor that captures the absolute intensity information of the scene and outputs images at a fixed frame rate [71].

Pinhole Camera Model and Front Projection Model

The pinhole camera is the simplest representation of the image capture process. It describes the geometric relationship between a 3D point in the camera coordinate frame and its projection onto a 2D image plane. An ideal pinhole camera consists of a small aperture through which light rays pass and form an inverted image on the opposite side of the aperture, assuming no lens distortion (Fig. 2.5**a**). To simplify

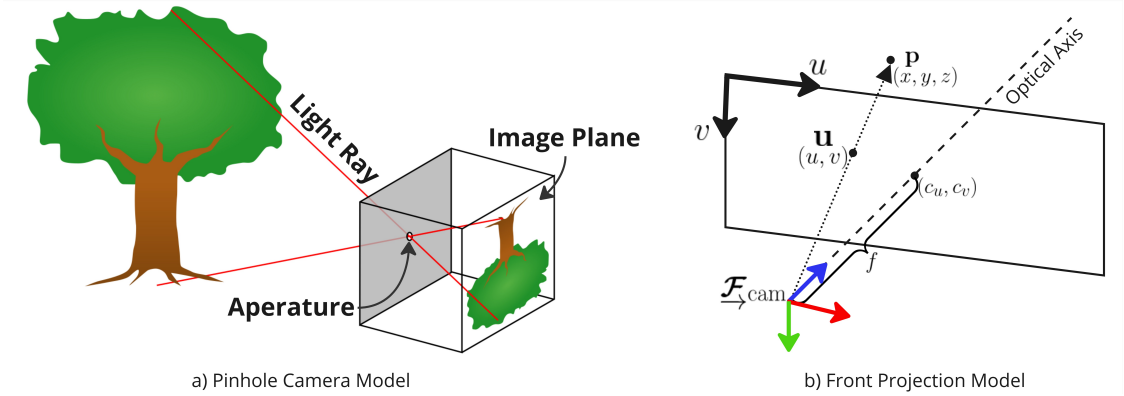


Figure 2.5: Illustration of pinhole camera model and front projection model. **a)** The real world object would be flipped on the image plane through the pinhole camera. The figure is adapted from Kirkpatrick and Wheeler [125]. **b)** The front projection camera model is equivalent to pinhole camera model but moving the image plane to the front of the camera. The 3D-to-2D projection formulation is shown in Eq. (2.31). It is noted that the color of the axes follows the same rule in Fig. 2.1.

visualization and computation, the front projection model is often adopted. In this model, the image plane is conceptually presented in front of the pinhole, resulting in a non-inverted image that is more intuitive to work with (Fig. 2.5**b**).

Formally, given a 3D point, $\mathbf{p}$, in camera frame, $\underrightarrow{\mathcal{F}}_{\text{cam}}$, the camera front projection model maps this point to 2D pixel coordinates in the image plane using:

$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} = \mathbf{P}\mathbf{K}\frac{1}{z}\mathbf{p} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \frac{1}{z} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \tag{2.31}$$

$\mathbf{u}$ is the projected pixel position of the 3D point in the image space. $\mathbf{P}$ is the projection matrix to convert $\mathbb{R}^3$ to $\mathbb{R}^2$. $\mathbf{K}$ is the intrinsic matrix that encodes the focal length f_u and f_v in horizontal and vertical direction and optical center position (c_u, c_v), where all units are in pixel. It is noted that if the pixel is square, f_u is equal to f_v, which gives the same focal length shown in Fig. 2.5**b**.

It is important to note that during projection, the depth information of the 3D point (i.e., the z coordinate) is lost. To recover the actual 3D point from a pixel coordinate, external range measurements are required. One way is to rely on a range measurement device such as LiDAR or a depth camera. Alternatively, depth can also be estimated using a stereo camera, which leverages the disparity between two synchronized camera views to triangulate 3D structure.

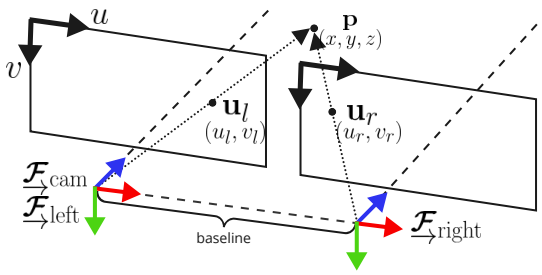


Figure 2.6: Illustration of (left) stereo camera model, where the overall setup is defined to be the left camera frame. The right camera frame is translated along the x-axis of the left camera frame by a distance known as the baseline.

Stereo Camera Model

The stereo camera model consists of two rigidly attached pinhole cameras with a known relative transformation between them. A common configuration is one camera translated along the x-axis of the other by a fixed distance known as the stereo baseline. In most cases, the coordinate frame of the stereo setup is defined to coincide with the left camera frame. This configuration is often referred to as the Left Stereo Camera Model [11], as illustrated in Fig. 2.6:

$$\mathbf{u}_{\text{stereo}} = \begin{bmatrix} u_l \\ v_l \\ d \end{bmatrix} = \mathbf{K}_{\text{stereo}} \frac{1}{z} \mathbf{p} = \begin{bmatrix} f_u & 0 & c_u & 0 \\ 0 & f_v & c_v & 0 \\ 0 & 0 & 0 & f_u b \end{bmatrix} \frac{1}{z} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \tag{2.32}$$

b is the stereo baseline distance and d is the disparity

$$d = u_l - u_r = \frac{1}{z} f_u b \tag{2.33}$$

Triangulation with Stereo Setup By capturing images from two slightly different viewpoints, the stereo system enables depth estimation through triangulation using the disparity in much the same way as our two eyes function (Eq. (2.33)):

$$z = \frac{f_u b}{d}. \tag{2.34}$$

This approach provides a passive and geometry-driven method for reconstructing 3D structure from 2D observations. A 3D point can be obtained by rearranging

the Front Projection Model in Eq. (2.31) and approximating the same focal length in horizontal and vertical directions

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{u-c_u}{d} b \\ \frac{v-c_v}{d} b \\ \frac{f_u b}{d} \end{bmatrix} \tag{2.35}$$

Lens Distortion

Most camera devices are equipped with lenses composed of convex and concave optical elements that bend incoming light rays to achieve various Fields of View (FoVs), enabling diverse photographic and sensing capabilities [238]. This however introduces lens distortion, which violates the assumptions of the ideal 3D-to-2D projection model described earlier. Depending on the lens type, distortion is commonly modeled using either the radial-tangential distortion model [28] or the equidistant projection model [268]. The radial-tangential model is generally applicable for modeling distortion in standard and wide-angle lenses, accounting for radial warping and slight decentering of the optical axis. Commercially available cameras like the Intel Realsense series [107] can be modeled with radial-tangential model to undistort the image. In contrast, the equidistant model is specifically designed to handle the extreme distortion introduced by fisheye lenses, which offer ultra-wide FoVs. In the work discussed in Chapter 5, the system hardware uses Alphasense cameras [224] with a FoV of 126°×92.4° in width and height, providing wide-angle view to facilitate visual state estimation in high acceleration and jerky motion scenarios. This type of camera uses an equidistant model to represent its lens distortion. Cameras employing either model can be calibrated using open-source tools such as Kalibr [69] or OpenCV [26], which estimate the intrinsic parameters and distortion coefficients required for various robotics tasks.

2.5.2 RGB-D Camera

An RGB-D camera integrates a color (RGB) camera and a depth sensor to provide dense depth measurements for each pixel in the color image. This enables the generation of colored point clouds. Early RGB-D cameras typically employed

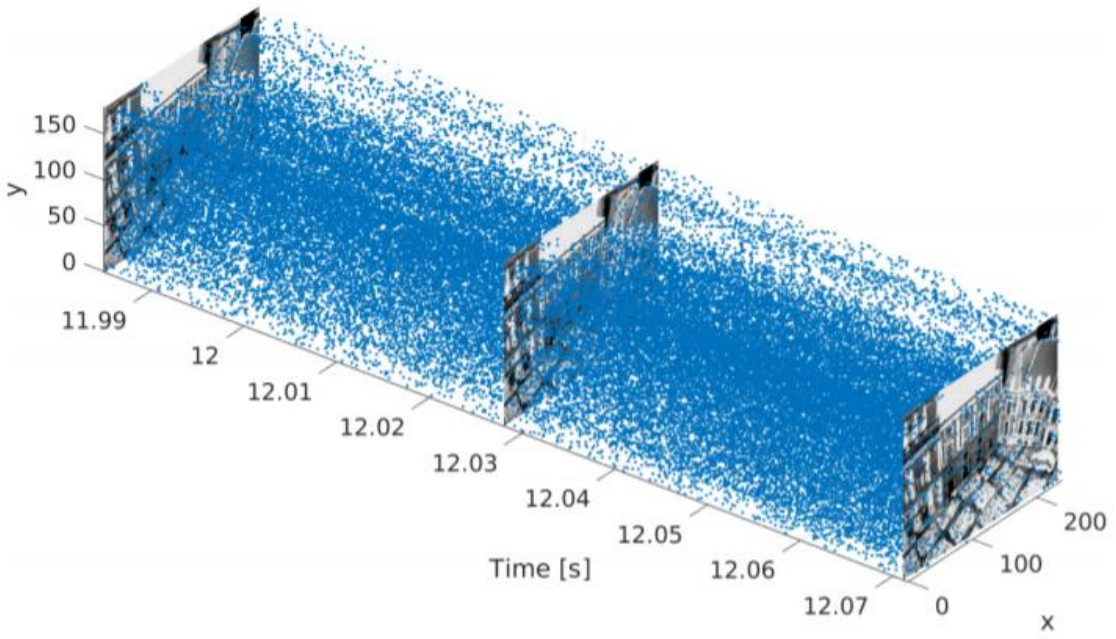


Figure 2.7: Visualization of event and frame data in spatial-temporal domain (picture from Mueggler et al. [170]). The blue dots represent events, which indicate a high temporal resolution in contrast to the frames that are output at a fixed rate.

structured light techniques for depth sensing. These systems project a known infrared light pattern onto the scene and infer pixel-wise depth by analyzing the distortion of the reflected pattern [62]. While effective indoors, sunlight interferes with the detection of the pattern making it inappropriate for use in outdoor scenarios. To address this limitation, more advanced RGB-D cameras employ the Time-of-Flight (ToF) method. This approach obtains the depth by measuring the time taken for light to be emitted, and to travel to a surface and be reflected back to the sensor, which is less affected by sunlight and is more robust for outdoor applications. Alternatively, the Intel Realsense series [107] combines both structured light and a stereo camera to be more robust. The stereo camera works well outdoors where there tends to be more texture.

2.5.3 Event Camera

An event camera is a bio-inspired vision sensor that mimics part of the way an eye works. Unlike conventional frame-based cameras, it senses the pixelwise brightness change and outputs the information asynchronously in the form of an

event (Fig. 2.7). Each event is a tuple and commonly expressed in Address-Event Representation (AER) [148]:

$$\mathbf{e} = \Big\{ t, \quad x, \quad y, \quad p \Big\}, \tag{2.36}$$

where t is the timestamp at which the event is triggered. (x, y) indicates the pixel location where event occurs on the image plane and p is a binary number representing the sign of brightness change (1 for brightness increase and 0 for decrease).

Event cameras offer significant advantages over conventional frame-based cameras in terms of temporal resolution, dynamic range, and power efficiency. While frame-based cameras (Sec. 2.5.1) often struggle under conditions of high-speed motion or poor illumination, event cameras operate with microsecond latency, making them highly responsive to fast dynamic scenes. Moreover, they consume less than 100 mW of power, making them well-suited for power-constrained robotic platforms in contrast to conventional frame-based camera which typically consumes ten to one hundred times more power [224]. With a high dynamic range of over 120 dB, event cameras are also capable of capturing meaningful information in extreme lighting conditions where conventional sensors may fail [71]. However, since the event data is fundamentally different from the images, event output is not directly applicable to the frame-based algorithms. A paradigm shift is required to fully leverage the benefits of event-based sensing.

3
Literature Review

Contents

This chapter reviews algorithms and systems that are closely related to the research conducted in this thesis. The discussion is organized into three key areas aligned with the core contributions of the thesis. Sec. 3.1 provides an overview of single-session, vision-based state estimation and mapping pipelines. Sec. 3.2 focuses on perception in legged robots, specifically about state estimation and terrain. These platforms often operate in the visually challenging environment, which is an application scenario of the work presented in Chapter 5. Finally, Sec. 3.3 surveys existing multi-session mapping and change detection frameworks, which are relevant to the challenges tackled in Chapter 6.

3.1 Vision-based State Estimation and Mapping

This section reviews prior work related to vision-based state estimation and mapping, particularly those relevant to the systems developed in Chapter 4 and Chapter 5. Specifically, it focuses on two core components: odometry and SLAM.

Odometry estimates the relative motion of a robot using onboard sensor measurements, typically without knowledge of a global reference frame (i.e., a map). As all sensors are affected by noise, odometry estimates inevitably accumulate error over time, leading to drift in trajectory estimation [215, 67]. SLAM extends odometry by jointly estimating the robot's trajectory and constructing a map of the environment. A key feature of SLAM systems is their ability to detect loop closures, which recognizes previously visited locations and uses this information to correct accumulated drift in the trajectory estimate [29].

Both odometry and SLAM can be implemented using vision sensors, either as standalone systems or in combination with other modalities such as Inertial Measurement Unit (IMU) or range sensors. The following subsections review relevant methods from two perspectives: frame-dominant approaches, which mainly rely on conventional frame-based cameras, and event-dominant approaches, which leverage the unique characteristics of event cameras.

3.1.1 Frame-based Approaches

Odometry

The use of visual images to estimate robot ego-motion dates back to Moravec's seminal work [164] in the 1980s. In his study, monocular cameras were mounted on planetary rovers to estimate motion by detecting and matching corner features across consecutive images [165]. The term Visual Odometry (VO) was later introduced by Nistér et al. [180], who proposed a monocular and stereo VO pipeline that utilized Random Sample Consensus (RANSAC) [63, 179] for robust outlier feature tracklets rejection. Poses were computed via Perspective-n-Points (PnP) algorithms [178, 88] using the inlier tracklets. This work demonstrated accurate robot pose

estimation in large-scale environments and laid the foundation for subsequent visual odometry systems.

When using a stereo camera setup, metric depth can be recovered for individual 2D features via triangulation [90] (also discussed in Sec. 2.5.1), thereby resolving the scale ambiguity inherent in monocular VO [1]. One canonical stereo VO pipeline, LIBVISO [127, 77], adopts a similar approach as Nister et al.'s stereo VO pipeline [180]. It follows a similar process for feature detection, tracking, and outlier rejection, and estimates motion via the trifocal tensor using matched features across three consecutive images of a static scene, as described by Yu et al. [271]. The LIBVISO pipeline also incorporates a Kalman filter-based backend [115, 227] for state estimation.

Filtering versus Smoothing: The backend of the VO pipeline serves to minimize the uncertainty introduced by the noisy sensor data during estimation and provides refined poses [215]. Two main types of backends exist: filtering-based and smoothing-based. While filtering methods like the Kalman filter update the state incrementally, smoothing-based approaches maintain a history of states and constraints. These are often represented as factor graphs, with optimization performed via non-linear least squares to minimize the overall cost function [113]. Smoothing-based methods were first widely adopted in photogrammetric Bundle Adjustment (BA) problem [247], where the joint optimization of camera poses, landmark positions, and (optionally) camera parameters is performed to refine both motion and structure estimates. Traditional BA was designed for offline processing, but to support real-time applications, sliding window techniques were developed, optimizing only the most recent subset of states [68, 236]. An incremental version of smoothing, known as Incremental Smoothing and Mapping (iSAM), was introduced in [114] and later incorporated into the GTSAM, an open-source library containing various smoothing and mapping (SAM) algorithms useful for robotics and vision applications [42]. GTSAM has become widely adopted in modern real-time odometry and SLAM

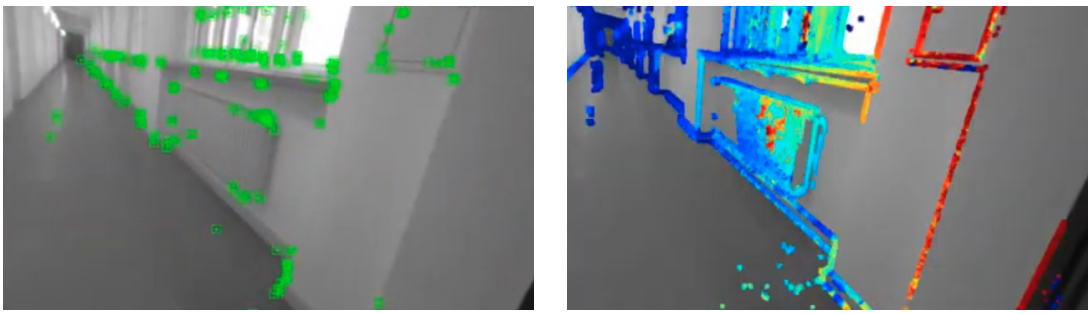


Figure 3.1: Comparison between feature-based and direct frontend. **Left:** Sparse corner features are extracted and tracked between images, and the pose can be estimated by minimizing the reprojection error using the feature tracklets. **Right:** Direct method computes the photometric error of the image intensities between frame for pose estimation.

systems, such as Kimera [208] and VILENS [275, 264], where it enables efficient and scalable optimization with real-time performance.

Direct Methods: The aforementioned VO pipelines rely on sparse feature-based frontends, where local features are detected, matched across frames to form tracklets, and then used to estimate relative camera poses. In contrast, an alternative class of methods known as direct methods estimates camera poses and reconstructs the scene geometry by directly optimizing over image intensities, instead of relying on the extracted features (Fig. 3.1). By utilizing the full image information, direct methods can perform better in low-texture or feature-less environments where traditional feature-based approaches may fail [51]. Early examples of direct methods include DAGM [233], which presents a dense depth map estimation pipeline to construct the scene directly from multiple images using optical flow [272]. DTAM [176] estimates the depth map of the scene by relying on the brightness constancy assumption, which aims to minimize the photometric error of the image captured close to a reference image. It also adopts a regularization term to enforce smoothness in the resultant depth map. Camera tracking is performed by minimizing the whole image photometric cost function between the reference image and tracked image in combination with the estimated depth map.

While these dense approaches utilize all available pixel data, their computational demands are high—typically requiring a Graphics Processing Unit (GPU) for real-time performance. To address this limitation, Engel et al. [55] adopts a semi-dense filtering approach for direct VO and depth map estimation, which only uses pixels with high image gradient for estimation and can run on Central Processing Unit (CPU) in real-time. DSO [53] presents a computationally lighter approach by selecting candidate pixels that are evenly spread across gradient-rich regions in the image to perform direct VO in a sparse manner. Leveraging both feature and direct tracking methods, the semi-direct approach of SVO [65] can also run on resource-constrained platforms such as drones at high frame rate.

Visual Inertial Odometry: An IMU as a different sensor modality provides high-frequency measurements of linear acceleration and angular velocity when a robot moves. Integrating a vision sensor with an IMU therefore bridges the temporal gaps between two consecutive image frames, allowing the capture of rapid motion dynamics of the robot where vision sensors alone may miss, and enables a robust, high-frequency motion estimation even during fast or abrupt movements. This fusion is commonly referred to as Visual-Inertial Odometry (VIO). VIO is particularly well-suited to scenarios involving high-speed or jerky motion, making it ideal for deployment on agile platforms such as drones [226] and autonomous vehicles [143]. Like VO systems, VIO systems can also use a filtering-based or smoothing-based backend.

In filtering-based VIO, state updates are performed incrementally using recursive estimators such as the Extended Kalman Filter (EKF). An example is ROVIO [21], which detects and tracks FAST features [252] across image frames, and uses an iterative EKF [117, 110] to jointly estimate the robot's pose, velocity, IMU biases, and the extrinsic calibration between the camera and IMU. Another filtering-based system, OpenVINS [78], builds on the Multi-State Constraint Kalman Filter (MSCKF) framework [167], incorporating techniques such as stochastic cloning [209] to improve computational efficiency while maintaining competitive accuracy.

Smoothing-based VIO backends optimize a sliding window of past states to achieve high estimation accuracy, but can be more computationally intensive. Depending on the application requirements and available resources, such systems may employ either feature-based frontends (e.g., [141]) or direct frontends (e.g., [234]).

More recently, with the development of machine learning and neural network, the learning-based framework have also been proposed for the odometry task. This can happen in terms of frontend feature detection and tracking [270, 102], or utilizing pre-trained 3D reconstruction prior [279, 174] for better performance. Feeding image sequences into the network to estimate pose in an end-to-end manner has also been investigated in [257, 269].

SLAM and Place Recognition

While accurate short-term pose estimation can be achieved through robust image tracking and backend optimization, estimation errors accumulate over time due to sensor noise and drift. The ability to identify previously visited places and to close loops is required to mitigate for such errors (Fig. 3.2). This process relies on visual place recognition, also referred to as image retrieval, which involves detecting and matching past observations with the current view. Specifically, it aims to find previously recorded keyframes that correspond to roughly the same physical location as the current frame. Classic handcrafted approaches like FAB-MAP [38] and DBoW [73] compute a global image descriptor from local features like SIFT [151] or ORB [211], and query matched images from a database. More recently, learning-based methods have shown significant improvements in recognition accuracy. One notable example is NetVLAD [9], which uses a Convolutional Neural Network (CNN)-based architecture to generate compact global descriptors. These methods consistently outperform traditional handcrafted approaches, particularly in environments with perceptual aliasing or varying illumination conditions [213].

Early real-time monocular visual SLAM systems, such as MonoSLAM [41] and PTAM [128], employ a feature-based frontend and maintain a persistent map of landmarks as part of the state vector. MonoSLAM uses an EKF-based backend,

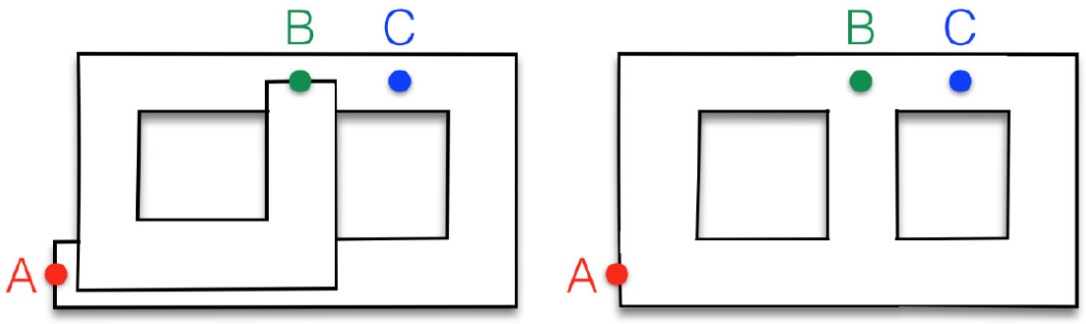


Figure 3.2: Demonstration of state estimation with/without a loop closure module (picture from Cadena et al. [29]). **Left:** Map is built by an odometry system without a loop closure module, where B and C are actually on the same floor. **Right:** Map is built by an SLAM system with loop closure module, where the previously visited A and B places are recognized and loops are closed. This results in a consistent map.

while PTAM adopts a smoothing-based approach. Neither system explicitly performs loop closure, which limits their long-term consistency. Additionally, both are designed for small-scale, room-level environments. ORB-SLAM [172] builds upon this foundation by incorporating explicit loop closure mechanisms and covisibility graph-based keyframe management [232, 156]. It presents a robust monocular SLAM system capable of operating in large-scale environments and correcting drift via loop closure. The system has been further extended to support stereo and visual-inertial configurations in ORB-SLAM2 [173], and supports multi-session localization and map merging in ORB-SLAM3 [30].

SLAM systems using direct visual odometry have also been presented in LSD-SLAM [51], which relies only on a monocular camera and is able to align keyframes by minimizing photometric errors while simultaneously reconstructing the environment as a point cloud. LSD-SLAM optionally integrates FAB-MAP [38] for loop detection. The system has been extended to stereo [52] and omnidirectional camera setups [32]. DSO [53] also supports loop closure through the integration of bag-of-words methods such as DBoW [73], as demonstrated in LDSO [74], which enables pose graph optimization for global consistency.

Traditional SLAM systems typically operate under the assumption of a static world, an assumption that does not hold when deployed over extended periods in

large-scale, dynamic environments. To facilitate long-term autonomy, it is essential to design SLAM systems that can robustly handle outliers arising from false data associations and environmental changes. Beyond front-end outlier rejection methods such as RANSAC [63] and its variants [10, 245, 49], backend robustness can be significantly enhanced by adopting advanced optimization techniques. In particular, incorporating robust loss functions [201, 12] or employing switchable constraints [237] within the factor graph optimization framework are methods which have proven effective. These approaches downweight the influence of measurements with high residual errors, thereby improving system resilience against spurious data and dynamic changes in the environment.

3.1.2 Event-based Approaches

Unlike traditional frame-based cameras, which output images at a fixed frame rate, event cameras produce a continuous stream of events that reflect asynchronous changes in scene brightness (Sec. 2.5.3). Conventional vision-based state estimation and mapping systems typically operate in a discrete-time manner, estimating camera poses only at the times of new measurements [47]. This approach simplifies system design and allows event data to be processed in batches, making it compatible with many classical pipelines. However, it introduces challenges such as high computational load due to the elevated event rates and the need to group events into time windows such that the scene is well-represented by the accumulated events.

An alternative and increasingly favored method is to model the camera trajectory as a continuous-time function. This approach avoids the need to define discrete state updates at every measurement timestamp while still enabling per-event state estimation. It is particularly well-suited for the asynchronous nature of event data, as it allows for more flexible and accurate integration of individual event timings [35].

This section reviews event-based state estimation and mapping systems, categorizing them based on how they handle the inherent asynchronicity of event stream—either by aggregating events into discrete-time representations or by processing them in a fully continuous-time manner.

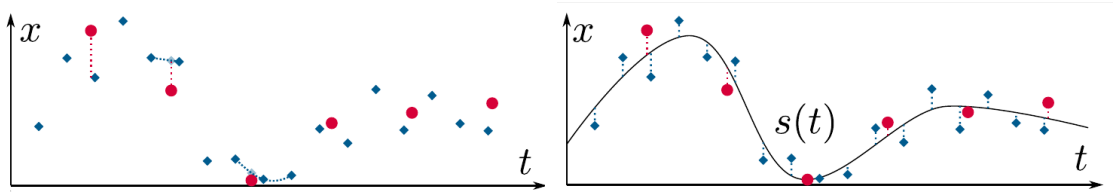


Figure 3.3: Comparison between discrete- and continuous-time state estimation approaches (picture adapted from Cioffi et al. [35]). Here the states, x, at different timesteps, t, are estimated. **Left:** When new measurements (red dots) arrive, in discrete-time state estimation, they need to be associated to the states. If no existing state (blue diamonds) has the same measurement time, new states are created by interpolating between existing states. **Right:** The trajectory is expressed as a continuous function (e.g., a spline, $s(t)$), and new measurements can be directly integrated into the continuous function for estimation.

Discrete-Time Approaches

Initial event-based VO research focused on simplified scenarios, e.g., 2D planar motion [262] or rotation-only motion [122]. Research extended estimation to $SE(3)$ motion by incorporating complementary sensors into the estimation pipeline. Kueng et al. [131] fuse both events and traditional-camera frames to detect features on the image frames and track them using events. Weikersdorfer et al. [261] use both event and RGB-D cameras to provide depth information for each event and create a 3D map for localization. A later work, Ultimate-SLAM [249] combines intensity images, IMU and event data to generate motion-compensated event frames and reformulates the cost function for camera egomotion estimation.

Early pure event-based $SE(3)$ motion estimation adapted the traditional feature-based approach using frames, where events are accumulated and processed into image-like representations. Such a representation can record the event timestamps known as Time Surface (TS) [137] or the number of events falling into the same pixel to construct an intensity image [123]. Tailored event-based feature detectors (e.g., eHarris [248], Arc* [5]) and trackers (e.g., ACE [4], HASTE [3]) are subsequently proposed to operate on the reconstructed image representation based on event data. These event-based feature tracklets allow the direct application of a frame-based pipeline which computes the poses via PnP and optimizes them by minimizing

the reprojection error [86]. However, as the event data is dependent on the scene intensity change and camera motion [71], the feature tracking is not always reliable.

To mitigate unreliable feature tracking on the event data, the direct approach has been investigated, which estimates the poses by aligning the image-like representation created from the event data [277]. One early work [123] interleaves three filters to estimate camera motion, intensity-based frames and scene depth on the event data by assuming constant brightness change of events on the same pixels. This method however requires a GPU for real-time processing. EVO [205] improves computational performance by interleaving geometric semidense mapping [203] and image-to-model alignment for pose estimation. The two works however require gentle motion in the initialization stage. Using a stereo camera, ESVO [277] can be rapidly bootstrapped. It utilizes the spatio-temporal consistency of events across the image plane to estimate the egomotion trajectory and a semidense 3D scene reconstruction in a parallel tracking and mapping manner. The system however has a high computational load with the increase in camera resolution. Its evolved version, ESVO2 [181], samples contour points to better scale with higher resolution event data and uses a motion prior from IMU preintegration to handle degenerated scenarios.

To better construct an image-like representation, events can be warped to a common timestamp via a motion compensation method [72]. With the aid of IMU, the motion compensated event frame can be better constructed, which increases the contrast of the frame representation and yields a more accurate estimation. This can either be performed in a monocular-IMU [280, 204] or stereo-IMU [33] setup.

Continuous-Time Approach

Handling event stream in a batch and solving for discrete-time estimation reduces the amount of event data which needs to be processed and simplifies the problem, but sacrifices the inherent temporal resolution of event-based cameras [169]. Characterizing the camera poses as a continuous-time trajectory offers its benefits in maintaining the temporal fidelity and handling the large throughput of the event

stream. Such representation allows the direct integration of the event data into the estimation pipeline, and allows the sensor pose to be queried at any event time. The trajectory is also described as a parametric model, which only requires a few parameters to estimate, significantly reducing the size of the state space [35].

Early work applied the continuous-time approach to a localization problem where the prior map consisting of 3D line segments is given [169]. The pipeline characterizes the trajectory as a cumulative B-splines function [228], where the control states are optimized by minimizing the reprojection error of predicted event positions against the tracked line segments [70]. The work was subsequently extended to a VIO pipeline, which additionally computes the predicted angular velocity and linear acceleration using a spline representation via [189] against the IMU measurements without the need for preintegrated IMU factors [64]. This evolved work provides a more accurate estimation and recovers the absolute trajectory scale. Using a spline for event-based VO was also presented by Wang et al. [258]. The work uses volumetric contrast maximization for continuous-time estimation. The trajectory is initialized with an Ackermann motion model [101], and globally optimized with a B-spline-based continuous-time estimation framework. This allows the estimator to maintain the native temporal resolution of the event data but the work is only tested on planar motion scenarios.

In contrast to a spline function, representing the trajectory as temporal Gaussian processes is the other variation in continuous-time event-based state estimation. Such an approach commonly adopts a physically founded motion prior (e.g., white-noise-on-acceleration (WNOA) prior [243]), which accounts for real-world kinematics compared to spline-based methods that enforce mathematical smoothness independent of its physical plausibility. Liu et al. [146] estimate continuous-time trajectories with Gaussian process regression and a WNOA motion prior in a monocular VO system. Their estimator runs asynchronously and considers individual event times but, unlike traditional VO pipelines, couples tracklet outlier rejection with the motion estimation which may introduce different sources of error. By incorporating IMU measurements, a different approach to apply Gaussian

processes for continuous-time event-based state estimation was presented in Gentil et al. [138] and Dai et al. [40] using event-based line [131] or corner [5] features. These approaches model inertial measurements using temporal Gaussian processes to form preintegrated measurements [139] that are solved in a smoothing-based framework.

3.2 Perception in Legged Robots

This section focuses on perception systems for legged robots. It begins with an overview of prominent legged robot platforms developed over the past decades by the robotics industry (Sec. 3.2.1). It then reviews perception techniques commonly employed in these systems, with particular attention to two critical components: state estimation (Sec. 3.2.2) and terrain mapping (Sec. 3.2.3). These topics are directly relevant to the main robotic platform, exoskeleton, for *Exosense* system introduced in Chapter 5, which is also a type of legged robots.

3.2.1 Canonical Legged Robot Systems

Legged robots are a class of mobile robots that employ articulated limbs for locomotion [147]. Inspired by the morphology of humans and various multi-legged animals, these systems offer greater mobility and adaptability in unstructured or uneven environments, where wheeled or tracked robots often struggle [265]. The enhanced dexterity and terrain adaptability of legged robots make them well-suited for complex real-world applications.

Legged robots can be broadly categorized by the number of legs they possess. This section focuses on representative systems within the two most prominent categories: bipedal and quadrupedal robots.

Bipedal Robot

Bipedal robots are designed to emulate the locomotion of two-legged creatures, typically humans, and have seen decades of development across academia and industry. In the 1980s, one of the earliest examples in this category was the MIT Planar Biped (Fig. 3.4a), which featured two telescoping legs actuated by hydraulics

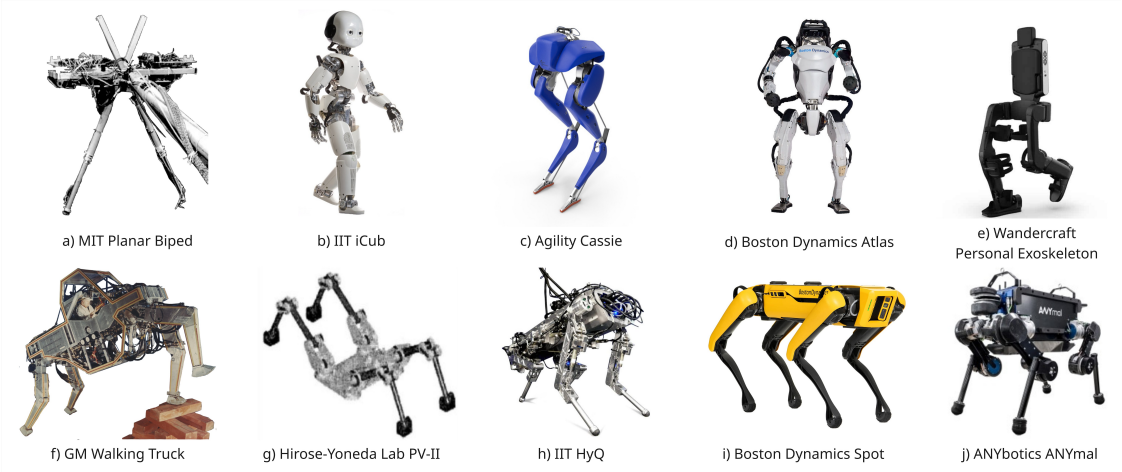


Figure 3.4: A set of existing legged robot systems. **Top:** Bipedal robots. **Bottom:** Quadruped robots.

and connected to a central body [96]. This system demonstrated basic bipedal locomotion such as walking, running, stair climbing, and even flipping—albeit within a controlled laboratory environment. The design was later extended to the 3D Biped, which achieved more dynamic motions, including somersaulting, and demonstrated improved flexibility in natural environments [192].

In the early 2000s, advances in sensing and computer vision enabled more interactive and autonomous humanoids. Honda's ASIMO robot incorporated vision and auditory systems, allowing it to interact with humans and perform receptionist tasks [212]. Around the same period, the Nao robot was introduced with a programmable interface that enabled customized task execution [81]. Nao has since been widely deployed in public demonstrations, including RoboCup soccer competitions and synchronized dances at the Shanghai Expo.

To support research in embodied artificial intelligence, the open-source child-sized humanoid platform iCub was launched in 2004 (Fig. 3.4b) [160]. iCub features rich multimodal sensing, including vision, acoustics, inertial measurement, proprioception, and tactile feedback. It remains under active development, with major hardware upgrades such as a full-body tactile skin system [219] and the integration of event cameras for dynamic vision applications [13], greatly facilitating this line of research and benefiting the robotics community.

Diverging from the anthropomorphic paradigm, Agility Robotics designed Cassie (Fig. 3.4c), a lower-limb-only biped inspired by the biomechanics of flightless birds such as ostriches [17]. Leveraging spring-mass dynamics and compliant leg designs, Cassie achieves agile and fast walking capabilities compared to earlier, plodding humanoids [8].

Aiming to emulate the human range of motion, Boston Dynamics developed the Petman humanoid robot for the testing of chemical protective suits for military use [191]. Petman could perform tasks such as walking, squatting, and push-ups. It later evolved into the more advanced Atlas platform (Fig. 3.4d), a hydraulically powered biped equipped with LiDAR and cameras for state estimation, perception, and motion planning. Atlas has demonstrated complex locomotion and manipulation tasks in uneven terrain, with applications envisioned in search and rescue, disaster response, and autonomous delivery [132].

In the healthcare domain, bipedal robots have been realized as wearable exoskeletons for mobility assistance. The Atalante series from Wandercraft, comprising Atalante and its successor Personal Exoskeleton (Fig. 3.4e), enables individuals with lower-limb disabilities to stand and walk without the aid of crutches. These systems feature self-balancing capabilities and autonomous gait generation, making them well-suited for clinical and personal rehabilitation scenarios [85, 142, 106]. In Chapter 5, a vision-based scene understanding system is developed to facilitate the long-term, safe navigation of the exoskeletons.

Quadruped Robot

While bipedal robots mimic human locomotion by walking upright on two legs—freeing the upper limbs for manipulation and enabling an elevated viewpoint—this design inherently involves periods of single-leg support, requiring sophisticated balance and gait control to maintain stability. In contrast, quadrupedal robots benefit from having multiple ground contact points during motion, offering greater stability, maneuverability, and adaptability to uneven terrain.

Walking Truck [166] (Fig. 3.4f), as one of the early quadruped robots in the 1960s, possessed bulky mechanical structures, with three meter in height and weighted over one ton. The robot can perform level-ground walking and crawling but owing to the heavy structure, it had limited mobility.

Subsequent designs have focused on reducing weight and increasing flexibility. A notable example is PV-II (Fig. 3.4g), developed by the Hirose-Yoneda Lab, which was the first quadruped robot capable of stair climbing. It achieved this through a combination of toe tactile sensors and attitude sensing for feedback-based motion control [95]. This lab later introduced the Titan series [94], which demonstrated enhanced terrain adaptability and was deployed in industrial applications such as borehole drilling [97] and anti-personnel mine detection and removal [93].

To support robust dynamic locomotion, many quadruped systems, such as HyQ (Fig. 3.4h), have employed hydraulic actuation, which offers high torque control and force density [222]. However, hydraulic systems typically suffer from high power consumption and are difficult to scale or maintain in compact platforms. Advances in electric motor design, combined with the adoption of series elastic actuators (SEAs) [194], have enabled the development of fully electric quadrupeds that offer improved energy efficiency and locomotion performance. This design philosophy underpins systems such as StarlETH [104] and MIT Cheetah [223], both of which demonstrated agile, efficient, and compact quadrupedal locomotion.

Modern quadrupeds increasingly integrate perception sensors to enable autonomous operation. The original ANYmal design (Fig. 3.4j) was equipped with a rotating LiDAR for 3D environmental perception [105]. In the DARPA Subterranean Challenge, the ANYmal-C platform carried a multi-modal sensor suite—including LiDARs, cameras, and IMUs—enabling it to traverse complex and dynamic underground environments [246]. Another prominent example is Spot (Fig. 3.4i) from Boston Dynamics, which incorporates five stereo cameras positioned around the body. These sensors support its built-in SLAM system and provide a full 360-degree perception of its surroundings, facilitating advanced navigation and obstacle avoidance [25].

3.2.2 Legged Robot State Estimation

Proprioceptive sensors are sensor modalities that measure the internal state of a robot. Common examples include joint encoder and IMU. These sensors are typically used to estimate the robot's pose, velocity, and orientation in space.

In legged robots, proprioceptive state estimation is often performed using leg odometry derived from kinematic data. When a robot's foot makes contact with the ground, forward kinematics from joint encoder readings and foot force sensors can be used to estimate the robot's body pose [152]. However, such odometry is vulnerable to errors from terrain deformation and foot slippage, which often produce non-Gaussian noise that is difficult to model analytically [58]. On the other hand, solely relying on IMU data leads to rapid drift, especially with low-cost consumer-grade IMUs, which may lose positional accuracy within seconds without additional correction mechanisms [83]. Although increasing the number of IMUs with known relative positions can improve accuracy [256], this requires custom hardware and remains uncommon in commercial systems. Learning-based IMU estimators can perform well on familiar motion patterns [149], but tend to generalize poorly to unseen movements.

A more robust solution involves fusing kinematic and IMU data, commonly under an Extended Kalman Filter (EKF) framework. For instance, Bloesch et al. [20] used an Observability-Constrained EKF (OCEKF) [100] to jointly estimate pose, velocity, and foot contact positions of the quadruped StarlETH. While effective, this approach can become ill-conditioned in certain configurations, leading to unobservable absolute position and yaw. An extension using an Unscented Kalman Filter (UKF) [19, 112] was proposed to increase robustness against outliers, such as foot slippage, though the observability limitations remained. For the successor robot ANYmal, a Two-State Implicit Filter (TSIF) [18] was introduced. Rather than explicitly modeling system dynamics, TSIF leverages residual-based modeling, offering greater flexibility in incorporating kinematic and IMU measurements.

In the domain of self-balancing exoskeletons, EKF-based approaches are also prevalent. Vigne et al. [250] fused multiple IMU and joint encoder signals using

a flexible kinematic model to account for mechanical deformation during walking. This work was extended in MOVIE [251], which adopted a velocity-aided approach to estimate orientation with respect to the gravity. Building upon this, Elnecave et al. [266] developed an EKF-based estimator for full 6-DoF pose and velocity of the exoskeleton body.

While kinematic-IMU fusion yields reliable short-term performance, these methods inherently suffer from drift in unobservable dimensions, such as global position and yaw [18]. Consequently, they are unsuitable for long-term deployment unless augmented with additional sensing modalities [202].

To address this, multi-modal sensor fusion strategies have been developed. Early efforts fused vision with proprioceptive data using filtering-based techniques. Chilian et al. [34] used an indirect information filter to combine IMU, leg odometry, and stereo visual odometry on a hexapod navigating gravel terrain under variable lighting. A similar approach for quadrupeds was presented by Ma et al. [153], which used an EKF to fuse stereo visual odometry, kinematic measurements, and IMU data, with GPS information and Zero-Velocity Update (ZUPT) algorithm [66] incorporated when available.

LiDAR-based fusion has also been explored. Fallon et al. [59] integrated LiDAR with kinematic-IMU measurements, and this was later expanded by Nobili et al. [183] to include vision. These efforts resulted in the open-source estimator Pronto [31], which has been validated on both bipedal and quadrupedal platforms, and is used to provide state feedback for locomotion control. All components in Pronto are loosely coupled and integrated via an EKF framework.

In contrast, tightly-coupled sensor fusion has demonstrated superior accuracy by jointly optimizing all sensor measurements in a unified framework. Hartley et al. [91] introduced a hybrid system combining IMU, LiDAR, vision, and kinematic data within a factor graph, achieving accurate state estimation on the biped Cassie. However, this system was evaluated only in short (2 min) controlled indoor trials. VILENS [264] represents a more mature solution: it uses a tightly-coupled

factor graph to fuse visual, LiDAR, kinematic, and inertial data with custom-designed residual factors. This system has demonstrated robust performance in large-scale, long-term deployments on various legged platforms across field trials and missions, including those in the DARPA SubT Challenge [246], nuclear fusion reactor inspection [229], and active mapping [23].

3.2.3 Terrain Mapping

Accurate terrain modeling is essential for the safe and effective navigation of legged robots in unknown or unstructured environments. While the working principles of common metric map representations are discussed in Sec. 2.4, this section reviews terrain mapping techniques not only at the geometric level but also highlights approaches that incorporate additional semantic and traversability information to support informed navigation.

Early works in robotic mapping [50, 241] adopted the 2D occupancy grid, where each cell encodes the probability of being occupied. Although effective for wheeled robots in structured indoor settings, such representations fall short for legged platforms due to their inability to model terrain elevation and uneven surfaces.

A natural extension of the occupancy grid is the 2.5D elevation map, where each grid cell stores the surface height relative to a reference plane [61]. This representation has been widely adopted for both legged [155, 133] and wheeled [60, 157] robots. The elevation map also supports a multi-layered grid structure to encode additional information, such as semantic labels or terrain traversability. These enhancements enable elevation maps to be extended for semantic-aware planning [56], traversability analysis [161], and terrain property estimation [57], further broadening their utility for autonomous navigation.

Fully 3D representations model free and occupied space explicitly using volumetric approaches. Fixed-resolution voxel grids can become memory-intensive when scaled to large environments, especially at fine resolutions. Octree-based representations, such as OctoMap [98] and Octree-based fusion [273], address this limitation by hierarchically pruning unoccupied space. More recent methods, like

Wavemap [207], compress voxel data using wavelet transforms, enabling multi-resolution mapping that scales efficiently in both size and detail.

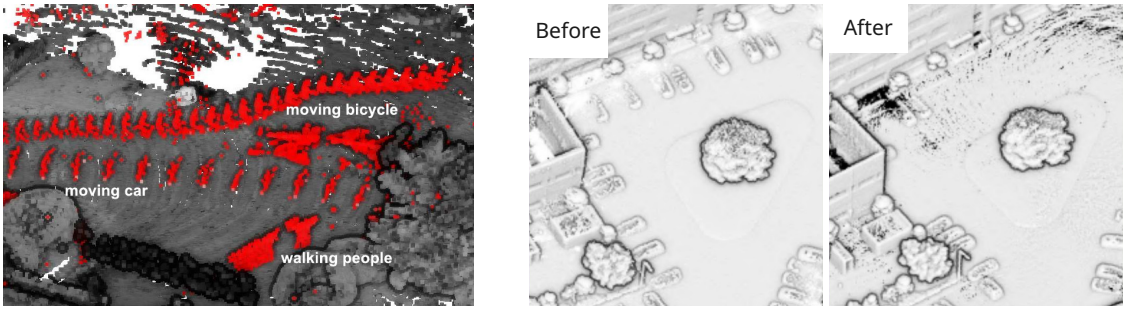
Another popular approach to 3D mapping is the use of implicit surface representations, such as signed distance fields (SDFs) [39]. Voxblox [185] utilizes a Truncated SDF (TSDF) for local terrain modeling and builds an Euclidean SDF (ESDF) on top for motion planning. This system was later extended into C-blox [162], which generates lightweight submaps suitable for large-scale exploration. Voxgraph [206] further refines this by incorporating submaps, odometry, and loop closures into a unified factor graph, achieving globally consistent map optimization. Panoptic Mapping [218] builds on this by representing submaps as semantic entities (e.g., objects, background, free space) and leverages a TSDF representation to model the geometry of each entity. For overlapping submaps, a semantic similarity search is applied to determine whether an environment change occurs.

Beyond geometric reconstruction, terrain mapping is increasingly integrated into high-level scene understanding. Systems like Hydra [103] and S-Graphs [15] introduce hierarchical scene graphs that capture contextual relationships among rooms, objects, and spatial entities. Emerging approaches use neural implicit representations combined with vision-language models (VLMs), such as OpenScene [190], LERF [118], and CLIP-Fields [225]. These systems enable semantic segmentation and interactive querying using natural language, allowing for map interpretation via simple text prompts.

Collectively, these mapping methods provide a rich foundation for legged robot autonomy, ranging from traversable surface modeling to semantic-level scene comprehension, and serve as building blocks for robust operation in real-world environments.

3.3 Towards Lifelong Robot Operation

This section reviews the foundational work on change detection and multi-session SLAM, which are critical components for enabling the long-term operation of robotic systems. These capabilities allow a robot to maintain and update its



a) High Dynamic Changes b) Semi-static Changes

Figure 3.5: Example high dynamic and semi-static changes in the point cloud map (picture adapted from LT-Mapper [119]). **a)** High dynamic changes are commonly caused by active moving objects such as pedestrians and cars, which result in a sequence of "ghost trails" that can affect the downstream tasks utilizing the map for localization and path planning. **b)** Semi-static changes come from scenarios where objects are moved over days, weeks or months. Identifying such changes is a critical task for environmental monitoring and inspection.

understanding of dynamic environments over time, ensuring robust perception and navigation during repeated deployments.

Sec. 3.3.1 presents various approaches to change detection, with an emphasis on techniques that leverage 3D map representations. Sec. 3.3.2 introduces representative frameworks for multi-session mapping from both vision-based and LiDAR-based perspectives. The systems reviewed in this section closely align with the design and objectives of the framework presented in Chapter 6.

3.3.1 Change Detection

With the increasing demand for long-term robotic operation in dynamic environments, the ability to detect changes in the scene over time is essential. Change detection enables robots to adapt their behavior in response to environmental variations and is widely used in applications such as environment monitoring [195], infrastructure inspection [229], and disaster response [184]. These changes may range from highly dynamic (e.g., pedestrians and vehicles) to semi-static alterations that evolve over longer periods (Fig. 3.5).

Change detection can be performed via geometric analysis of map representations. Grid-based structures such as elevation maps [61] and OctoMap [98] support ray-tracing techniques that update occupancy based on sensor ray traversal. Although

accurate, these methods are computationally intensive due to the need to process every cell along each ray. Real-time deployment of this approach often requires hardware acceleration using GPUs [161, 163].

Visibility reasoning simplifies the problem by checking whether a point visible in one scan remains visible from another viewpoint [193]. While efficient, such methods are sensitive to incidence angle ambiguity—especially on ground surfaces—leading to misclassifications [145]. To mitigate this issue, visibility is often encoded as an auxiliary feature in downstream classifiers [193, 120].

Volumetric maps such as Signed Distance Fields (SDFs) and occupancy grids allow online change detection by modeling free space. Systems like Dynablox [217] and DUFOMap [46] detect changes when new sensor measurements violate prior free-space assumptions. For inter-session analysis, approaches like LiSTA [210] and BeautyMap [108] align volumetric maps and perform voxel-level differencing to detect environmental changes.

Despite their effectiveness, geometry-based approaches rely on discretizing the space at a fixed resolution, which limits their ability to detect changes that involve semantic or appearance differences without significant geometric variation [79]. Incorporating semantics through instance- or class-level segmentation can help manage non-rigid object-level changes [218, 216], but such methods struggle in unstructured outdoor settings like construction sites.

Recent advances in learning-based methods offer an alternative through end-to-end segmentation of dynamic elements directly from 3D data [158, 235]. These methods achieve high accuracy but rely heavily on large, labeled datasets for training, limiting their generalization to novel environments or rare classes.

3.3.2 Multi-session SLAM

A typical SLAM algorithm estimates a robot's trajectory while simultaneously constructing a map of the environment, and has become a foundational capability for autonomous systems. However, most conventional SLAM systems assume a single, continuous exploration session. In contrast, multi-session SLAM extends

this framework to support long-term and large-scale operations by incrementally fusing the outputs of multiple SLAM sessions—whether performed by a single robot across different time intervals or by multiple robots collaboratively. This capability enables persistent mapping, robust localization across revisits, and resilience to environmental changes over time.

This section reviews multi-session SLAM approaches from both vision- and LiDAR-based perspectives, covering systems developed for single-agent scenarios as well as collaborative multi-agent settings.

Multi-session Visual SLAM

In multi-session visual SLAM, the system needs to recognize previously visited places across different sessions using visual inputs. This is inherently challenging due to changes in lighting conditions, viewpoint, and scene appearance. Labbe and Michaud [136] present a multi-session visual SLAM framework centered around re-localization, with each individual session built using RTAB-Map [134]. Their work evaluates various visual descriptors for illumination-invariant place recognition and loop closure. Experimental results indicate that learning-based feature detectors and matchers (e.g., SuperPoint [44] and SuperGlue [214]) offer improved robustness to appearance changes, albeit at the cost of increased computation and memory. To mitigate this, the framework incorporates a graph reduction strategy [135] to conserve resources while preserving localization accuracy.

Dedicated multi-session mapping systems like *maplab* [220] provide a tightly integrated pipeline for vision-based SLAM. It uses ROVIO [21] to construct individual sessions, saving pose graphs, keyframes, image features, and associated resources for inter-session place recognition, merging, and reconstruction. The updated system, *maplab 2.0* [37], expands support to heterogeneous sensor modalities and robot platforms, becoming agnostic to odometry sources. It also supports incorporating non-visual data (e.g., LiDAR scans, GPS), enabling more versatile graph optimization constraints for tasks like multi-agent mapping and semantic mapping.

Kimera-Multi [242] investigates multi-session mapping in distributed multi-robot scenarios. Each robot runs Kimera [208], a metric-semantic SLAM system that reconstructs semantically labeled mesh maps locally. These distributed submaps are merged into a globally consistent mesh when communication becomes available, enabling collaborative mapping under bandwidth constraints.

The ability to maintain multiple maps and merge them together when the robot revisits the same place can also be an important application for single-session SLAM. ORB-SLAM3 [30] provides a versatile visual (-inertial) SLAM implementation that supports single-session visual SLAM for various camera modalities and can create new maps when localization fails. These new maps can be merged into the previous map when the robot revisits the existing mapped area. ORB-SLAM3 achieves the multi-map data association while being robust to visually challenging environments.

Multi-session LiDAR SLAM

In contrast to vision-based systems that rely on image-based place recognition, multi-session LiDAR SLAM extracts geometric descriptors directly from point clouds. These descriptors are then matched across sessions to establish correspondences. FRAME [230], for example, uses 3DEG descriptors [231] to identify correspondences between submaps from different sessions. Fast-GICP [130] is used to align matched clouds and enable map merging.

However, registration based solely on pairwise cloud alignment can introduce global inconsistencies when integrating multiple sessions. Pose graph optimization is often used to ensure long-term consistency. LT-Mapper [119] presents a modular multi-session LiDAR mapping framework that uses Scan Context [121] for inter-session place recognition. Corresponding poses are added as edges in a pose graph, enabling alignment of multiple sessions into a shared global frame. It also integrates Removert [120] to remove dynamic points using visibility analysis, yielding a cleaner, temporally updated map.

ELite [79] extends LT-Mapper by introducing an ephemerality score for each point, quantifying how likely a point is to change over time. This allows meaningful

change detection and better rejection of dynamic outliers during map merging. Ephemerality improves robustness by avoiding contamination from dynamic points in merged sessions.

Efficiently managing map growth across sessions is also addressed in MS-Mapping [267], which emphasizes scalable graph construction through keyframe selection based on Wasserstein distance metrics. The system voxelizes the point cloud for compact storage while maintaining spatial accuracy. Additionally, MS-Mapping models the uncertainty of inter-session constraints and adjusts it dynamically, improving the reliability of map registration without manual covariance tuning.

This chapter surveyed a broad range of state estimation and mapping algorithms highlighting both traditional and event-driven methods tailored for robust perception under visually challenging conditions. It further reviewed perception systems in legged robotics, emphasizing state estimation and terrain mapping that are essential for legged robot navigation. It finally summarized change detection and multi-session SLAM frameworks that are critical to realize long-term, adaptive robot operation.

Motivated by these works, the subsequent chapters introduce a novel event-based state estimator capable of producing smooth, continuous-time trajectories in high-speed and high dynamic range scenarios. This is followed by a vision-centric scene understanding system designed to support safe navigation for self-balancing exoskeletons, and its extension to a multi-session mapping framework that enables persistent, lifelong deployment in dynamic environments.

4
Event-based Visual Odometry

Contents

Event cameras offer significant advantages over conventional frame-based cameras in terms of high temporal resolution and high dynamic range. These properties make them well-suited for motion estimation in challenging scenarios, such as high-speed motion or extreme lighting conditions. Unlike frame-based cameras that capture images at fixed intervals, event cameras report pixel-wise brightness changes asynchronously, resulting in a fundamentally different data format. Consequently, event data is not directly compatible with traditional image-based algorithms, and a paradigm shift in algorithm design is required to fully leverage their potential.

A common practice in event-based VO pipelines is to aggregate events over short time intervals to form image-like representations—often referred to as event

frames—on which conventional frame-based motion estimation techniques are applied. While this simplifies integration with existing algorithms, it discards the temporal information encoded in event streams, thereby underutilizing the unique sensing characteristics of event cameras.

To fully exploit the asynchronous nature of event data, a more appropriate approach is to model the camera trajectory as a continuous-time function. This allows pose estimation to be performed at each individual event timestamp without temporal discretization. In this chapter, a stereo event-based visual odometry pipeline is presented, which integrates a physically founded motion prior with continuous-time trajectory estimation. By treating each event as an independent spatiotemporal observation, the proposed method recovers a smooth and high-fidelity estimate of the camera trajectory across the estimation window, with poses available at arbitrary timestamps.

4.1 Event-Based Stereo Visual Odometry With Native Temporal Resolution via Continuous-Time Gaussian Process Regression

The following article was published in the IEEE Robotics and Automation Letters (RA-L) and it was also presented at the IEEE International Conference on Robotics and Automation (ICRA) 2024 [253]. An accompanying video is available online at: https://www.youtube.com/watch?v=lUf8hAB7Dwk.

© 2022 IEEE. Reprinted, with permission, from Jianeng Wang and Jonothan D. Gammell, "Event-Based Stereo Visual Odometry With Native Temporal Resolution via Continuous-Time Gaussian Process Regression," in IEEE Robotics and Automation Letters, 2023.

Event-Based Stereo Visual Odometry With Native Temporal Resolution via Continuous-Time Gaussian Process Regression

Jianeng Wang and Jonathan D. Gammell

***Abstract*—Event-based cameras asynchronously capture individual visual changes in a scene. This makes them more robust than traditional frame-based cameras to highly dynamic motions and poor illumination. It also means that every measurement in a scene can occur at a unique time. Handling these different measurement times is a major challenge of using event-based cameras. It is often addressed in visual odometry (VO) pipelines by approximating temporally close measurements as occurring at one common time. This grouping simplifies the estimation problem but, absent additional sensors, sacrifices the inherent temporal resolution of event-based cameras. This paper instead presents a complete stereo VO pipeline that estimates directly with individual event-measurement times without requiring any grouping or approximation in the estimation state. It uses continuous-time trajectory estimation to maintain the temporal fidelity and asynchronous nature of event-based cameras through Gaussian process regression with a physically motivated prior. Its performance is evaluated on the MVSEC dataset, where it achieves $7.9 \cdot 10^{-3}$ and $5.9 \cdot 10^{-3}$ RMS relative error on two independent sequences, outperforming the existing publicly available event-based stereo VO pipeline by two and four times, respectively.**

***Index Terms*—Event-based Visual Odometry, Vision-Based Navigation, Localization, SLAM.**

I. Introduction

VISUAL Odometry (VO) is a technique to estimate ego-motion in robotics [1], [2], [3], [4], [5]. VO systems using traditional frame-based cameras often struggle in scenarios with high speed motion and poor illumination. In these scenarios, the motion blur and poor image contrast of frame-based cameras result in bad estimation performance.

Event-based cameras perform better than traditional cameras in these challenging scenarios. They detect pixelwise intensity change and report the time at which the change occurs asynchronously. This gives them high temporal resolution and high dynamic range avoiding the limitations of frame-based cameras and providing the potential for more accurate VO systems [6]. Any event-based VO system must address the asynchronous event times. Many pipelines do this by grouping similar feature times to a common time [7], [8], [9]. This allows for the direct application of frame-based VO pipelines but sacrifices the temporal resolution of event cameras.

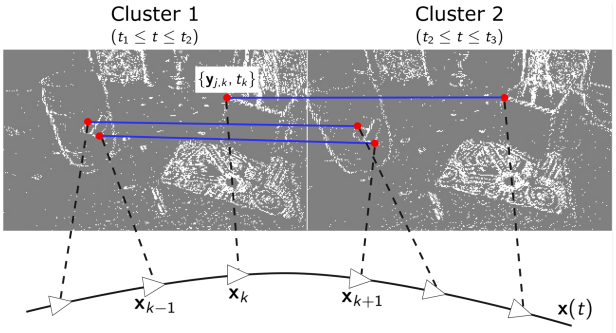


Fig. 1. Illustration of the continuous-time trajectory estimation pipeline. Event clusters are defined by an asynchronous event data stream in discrete windows based on number of events and their times (e.g., $t_1 \leq t \leq t_2$). Features (red) are detected from the resulting clusters and matched with features occurring in other clusters (blue) to create tracklets. Each event feature in the tracklet, $\mathbf{y}_{j,k}$, is a measurement of landmark, $\mathbf{p}_j$, and defines a trajectory state in the estimation problem, $\mathbf{x}_k = \{\mathbf{T}_{k,1}, \boldsymbol{\varpi}_k\}$, at the measurement time, t_k. The camera motion is estimated as a continuous-time trajectory function, $\mathbf{x}(t)$, defined by the discrete states and a physically founded motion prior.

This paper instead presents an event-based VO system that uses the unique asynchronous timestamps directly in the estimation problem without grouping or approximation. It estimates the camera motion as a continuous-time trajectory represented by states at unique feature times and a white-noise-on-acceleration (WNOA) motion prior. The trajectory is estimated using non-parametric Gaussian process regression. This results in a continuous, physically founded trajectory that exploits the temporal resolution of asynchronous event cameras and can estimate complex, real-world motions. (Fig. 1). This paper presents a complete event-based stereo VO pipeline using continuous-time Gaussian process regression. It is compatible with any feature detector and tracker, including frame-based methods for traditional cameras, without reducing the temporal resolution of event-based cameras. It also uses Motion-Compensated RANSAC (MC-RANSAC) [10] to consider the unique measurement times during outlier rejection and independently provide better tracklets and initial conditions for estimation. The resulting continuous-time trajectory provides estimates of camera poses at any and all timestamps in the estimation window.

It is evaluated on the publicly available Multi Vehicle Stereo Event Camera (MVSEC) dataset [11], where it obtains a more accurate and smoother trajectory estimate than the state-of-the-art Event-based Stereo Visual Odometry (ESVO) [9]. It achieves

Manuscript received 1 June 2023; accepted 17 August 2023. Date of publication 4 September 2023; date of current version 12 September 2023. This letter was recommended for publication by Associate Editor L. Zhang and Editor J. Civera upon evaluation of the reviewers' comments. *(Corresponding author: Jianeng Wang.)*

The authors are with the Estimation, Search, and Planning (ESP) Group, Oxford Robotics Institute (ORI), University of Oxford, OX1 4AR Oxford, U.K. (e-mail: jianeng@robots.ox.ac.uk; gammell@robots.ox.ac.uk).

Digital Object Identifier 10.1109/LRA.2023.3311374

2377-3766 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on September 12,2023 at 13:39:40 UTC from IEEE Xplore. Restrictions apply.

$7.9 \cdot 10^{-3}$ and $5.9 \cdot 10^{-3}$ root-mean-squared (RMS) relative error in $SE(3)$ and 5.78% and 4.93% final global translational error as a percent of path length on two independent MVSEC sequences. This outperforms the publicly available ESVO on these sequences, especially in terms of trajectory smoothness and RMS relative error where it is two- and four-times better, respectively. The rest of the paper is organized as follows. Section II summarizes the existing literature on event-based VO. Section III presents the complete pipeline of Gaussian-process continuous-time VO. Section IV evaluates the system and ESVO on the MVSEC dataset and discusses the results. Section V presents the summary of the work.

II. Related Work

Event-based motion estimation techniques can be described by their handling of asynchronous event times as either grouping times into discrete frames (Section II-A) or considering times individually (Section II-B).

A. Grouped-Time Approaches

Grouping event data together creates data frames like traditional cameras. Traditional VO pipelines can then be used to estimate camera poses at the discrete times assigned to the event frames. These frame times reduce the temporal resolution of the data by replacing all the individual events in a frame with a single time.

Initial event-based VO research focuses on simplified scenarios, e.g., 2D planar motion [12] or rotation-only motion [13]. Research extends estimation to $SE(3)$ motion by incorporating complementary sensors into the estimation pipeline. Kueng et al. [14] fuse both events and traditional-camera frames to detect features on the image frames and track them using events. Weikersdorfer et al. [15] use both event and RGB-D cameras to provide depth information for each event and create a 3D map for localization.

Event-based Visual Inertial Odometry (VIO) systems specifically include inertial measurement units (IMUs) and are a popular area of research. Zhu et al. [16] and Rebecq et al. [17] accumulate events in a spatial-temporal window to reconstruct frames for feature tracking. The resulting tracklets are used to minimize reprojection and initial error for pose estimation. Ultimate-SLAM [7] extends [17] to use the IMU to generate motion-compensated event frames and reformulates the cost function for camera egomotion estimation. Chen et al. [18] extract features using an asynchronous feature detector (Arc* [19]) from a stereo pair of event cameras and adopt an estimation pipeline similar to [7]. IMU Dynamic Vision Sensor Odometry using Lines (IDOL) [20] uses an alternative VIO paradigm. It uses Gaussian process regression to preintegrate IMU measurements and associate each event with timely accurate IMU data. This maintains the temporal resolution of the event data but still estimates the trajectory at the discrete states of the event frames.

Kim et al. [21] estimate $SE(3)$ motion using only event-based cameras. They interleave three filters to estimate camera motion, intensity-based frames and scene depth. EVO [8] improves computational performance by interleaving geometric semidense mapping [22] and image-to-model alignment for pose estimation. ESVO [9] uses parallel tracking and mapping to estimate the egomotion trajectory and a semidense 3D scene reconstruction. Hadviger et al. [23] present a feature-based stereo VO pipeline using events, which group events into frames and then adopts a similar estimation framework to a traditional frame-based stereo VO pipeline [3].

Discrete-time event-based VO groups multiple event times into a single time. This is helpful for feature detection and tracking but approximates the time of individual event features and reduces the temporal resolution of the measured data. This approximation is inconsistent with the asynchronous nature of event cameras and introduces potential measurement errors. This paper instead presents a VO pipeline that estimates the camera trajectory from individual event times and maintains the temporal resolution of event cameras. It uses only an event-camera stereo pair and can be implemented with either frame-based or event-based feature detection and tracking.

B. Individual-Time Approaches

Including all the individual, possibly unique event times in the estimation maintains the temporal resolution of event cameras but defines an underconstrained problem. Similar estimation problems are solved for rolling-shutter cameras and scanning lidars using continuous-time estimation [24], [25]. These techniques estimate the camera trajectory as a continuous function where pose can be queried at any time in the estimation window. A comparison of discrete and continuous-time trajectories can be found in [26].

Mueggler et al. [27] use a continuous-time pose estimation framework that uses IMU measurements and represents the trajectory as cumulative cubic B-splines. This avoids grouping event feature times and maintains their temporal resolution, but requires preprocessing to obtain a scene map. Wang et al. [28] use volumetric contrast maximization for continuous-time estimation. The trajectory is initialized with an Ackermann motion model [29], and globally optimized with a B-spline-based continuous-time estimation framework. This allows the estimator to maintain the native temporal resolution of the event data but limits it to planar motion.

Liu et al. [30] also estimate continuous-time trajectories with Gaussian process regression and a WNOA motion prior, but in a monocular VO system. Their estimation runs asynchronously and considers individual event times but, unlike traditional VO pipelines, couples tracklet outlier rejection with the motion estimation which may introduce different sources of error. The evaluation of their algorithm on real-world event datasets is also limited to five-second sequences.

This paper presents a complete continuous-time event-based stereo VO pipeline that maintains individual event times in the trajectory estimation. In contrast to these existing works, it maintains temporal resolution with either frame-based or event-based feature detection and tracking and with a RANSAC formulation [10] that separates outlier rejection from estimation. It uses a WNOA motion prior to estimate a full $SE(3)$ trajectory directly from event tracklets and their unique timestamps. This approach takes full advantage of the asynchronous nature of the event cameras and allows pose to be queried at any time in the estimation window.

III. Methodology

This paper presents an event-based stereo VO system that uses the native temporal resolution of event-based cameras for estimation (Fig. 2). Features are detected by clustering the event streams of each camera while maintaining the unique event times and tracked in a traditional frame-based manner (Section III-A).

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on September 12,2023 at 13:39:40 UTC from IEEE Xplore. Restrictions apply.

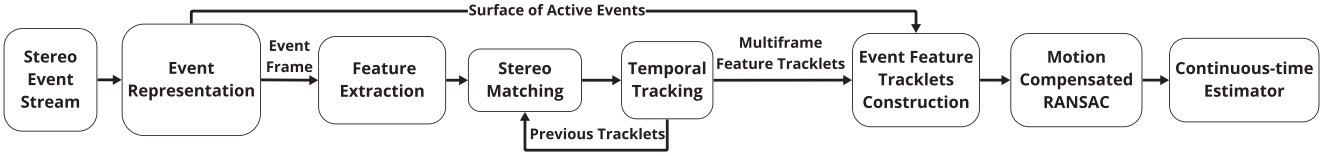


Fig. 2. Overview of the presented VO pipeline. The system takes an asynchronous stereo event stream and clusters the events into event frames and SAEs. Features are detected and tracked in the event frames and each feature is assigned an event time from the SAE. The resulting asynchronous event feature tracklets are filtered with a motion-compensated RANSAC that accounts for their asynchronous times. This gives a consistent inlier tracklet set and motion prior for the continuous-time estimator.

This allows for the use of any frame-based feature detection and tracking method or could be directly replaced by event-based approaches. The resulting asynchronous tracklets are then filtered for outliers with a motion-compensated RANSAC (Section III-B). This removes outliers more accurately than other methods by accounting for the different tracklet times. The camera trajectory is then estimated from all the unique tracklet states using Gaussian process regression with a WNOA prior (Section III-C). This results in a continuous-time VO system that estimates the camera pose at each unique tracklet time and can be queried for the pose at any other time in the estimation window.

A. Event Feature Extraction and Matching

Geometric features are extracted from clustered events (Section III-A1) using traditional frame-based feature detection techniques, and then matched between the stereo pair and through time to construct feature tracklets (Section III-A2). The temporal resolution of event cameras is maintained by assigning (possibly unique) times to each feature from the corresponding event in a Surface of Active Events (SAE) [31]. This allows the pipeline to use any frame-based feature detector and tracker and still create asynchronous tracklets for estimation. It could also be directly replaced with event-based methods, e.g., Arc* [19] or HASTE [32].

1) Event Clustering: The stereo event stream is rectified and clustered to construct new SAEs and new binary event frames (Fig. 2). The left and right event streams are synchronously clustered by registering events within a user-specified time window or until either camera registers more than a user-specified number of events in that time. These thresholds define a minimum effective frame rate when the scene changes slowly (e.g., small motion) and a faster frame rate when it changes rapidly (e.g., high-speed motion), while keeping the left and right frames synchronized for feature matching. The SAE records the most recent event timestamp of each pixel location and is used to maintain the asynchronous nature of the event camera. The binary event frame is a grey image where white pixels denote events, regardless of polarity.

2) Feature Detection and Matching: Features are independently detected in the left and right binary event frames for each stereo pair of clusters (Fig. 2). These features are then matched using a quad-matching scheme. Features in the current left frame are first matched to the current right frame. The matched features are then successively matched from the current right frame to the previous right frame, from the previous right frame to previous left frame and finally from the previous left frame back to the current left frame. Features successfully matched to all these pairs are kept as tracklets. This feature detection and tracking can use any traditional frame-based approach.

The timestamp of each tracklet state is assigned from the nearest event in the associated SAE. This allows traditional frame-based feature detection methods to detect asynchronous tracklets that maintain the original temporal resolution of the event data. Event-based feature detection and tracking algorithms generate these asynchronous tracklets directly and could be used instead.

Tracklets are extended beyond consecutive frames by matching the new tracklets to previously detected tracklets in earlier frames. This is done by independently checking each new tracklet against a user-specified number of previous frames and recording any additional matches.

The tracklets are filtered using user-specified thresholds to discard those that have

1) a large time difference between stereo features,
2) a small disparity between stereo features,
3) a short length, and
4) a short time.

These filtering schemes remove incorrect matches and improve tracklet quality. This feature extraction and matching process is performed independently at different image resolutions to detect and track features of different size. The resulting tracklets are then processed together to remove outliers.

B. Outlier Rejection

Traditional VO pipeline uses Random Sample Consensus (RANSAC) [33] to remove tracklet outliers before estimation. Traditional RANSAC assumes tracklet states occur at common times and uses a discrete transform motion model. This assumption is incorrect for asynchronous event tracklets and this paper instead uses MC-RANSAC [10], which makes no assumptions about common state times and uses a constant-velocity model in $SE(3)$. The fast version of MC-RANSAC (Section III-B1) is used to find an initial inlier set by repeatedly selecting tracklets, calculating the constant-velocity model, and segmenting the tracklets into inliers and outliers based on a user-specified threshold. This process is repeated a user-specified number of times and then the largest inlier set is refined using the full iterative MC-RANSAC (Section III-B2) to find the final inlier and outlier segmentation (Fig. 2). Both versions of MC-RANSAC are compared to traditional RANSAC in [10].

1) Fast MC-RANSAC: The set of tracklets between two event clusters is a number, M_{clst}, of stereo measurements of different landmarks at possibly unique times. MC-RANSAC segments these tracklets into inliers and outliers by finding the most tracklets that can be explained by a single velocity. The constant $SE(3)$ velocity of the sensor, $\boldsymbol{\varpi} \in \mathbb{R}^{6\times 1}$, is

$$\boldsymbol{\varpi} = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix},$$

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on September 12,2023 at 13:39:40 UTC from IEEE Xplore. Restrictions apply.

where $\mathbf{v}, \boldsymbol{\omega} \in \mathbb{R}^{3\times1}$ are linear and angular velocity components, respectively. This constant velocity over a time, Δt, gives the relative $SE(3)$ transformation,

$$\mathbf{T} = \exp\left(\Delta t \boldsymbol{\varpi}^{\wedge}\right),$$

where $\exp(\cdot)$ is the matrix exponential and $(\cdot)^{\wedge}$ is the lifting operator that converts $\mathbb{R}^{6\times1}$ to $\mathbb{R}^{4\times4}$ [34].

A velocity can be calculated from tracklets faster but less accurately by minimizing the error in Euclidean space,

$$J_{\text{fast}}(\boldsymbol{\varpi}) = \frac{1}{2}\sum_{j=1}^{M_{\text{rand}}} \mathbf{e}_{\text{fast},j,k}^{T}\mathbf{e}_{\text{fast},j,k}, \tag{1}$$

where $M_{\text{rand}} \leq M_{\text{clst}}$ is the the number of randomly selected tracklets and $\mathbf{e}_{\text{fast},j,k} \in \mathbb{R}^{3\times1}$ is the motion-model error,

$$\mathbf{e}_{\text{fast},j,k} = \mathbf{P}\left(\mathbf{p}_{k}^{j,k} - \mathbf{z}\left(\mathbf{T}_{k,k'}, \mathbf{p}_{k'}^{j,k'}\right)\right), \tag{2}$$

where $\mathbf{P}$ is the mapping from homogeneous to 3D coordinates, $\mathbf{p}_{k}^{j,k} \in \mathbb{R}^{4\times1}$ is the position of the j^{th} landmark relative to the camera pose at time t_k, $\mathbf{T}_{k,k'}$ is the transformation from timestamp $t_{k'}$ to t_k, $t_{k'} < t_k$ is the earlier time of the tracklet segment, and $\mathbf{z}(\cdot)$ is the constant-velocity motion model,

$$\mathbf{z}\left(\mathbf{T}_{k,k'}, \mathbf{p}_{k'}^{j,k'}\right) = \mathbf{T}_{k,k'}\mathbf{p}_{k'}^{j,k'}$$

Assuming a short duration tracklet, the corresponding small transformation is approximated as

$$\mathbf{T} \approx \mathbf{1} + \boldsymbol{\xi}^{\wedge} = \mathbf{1} + \Delta t \boldsymbol{\varpi}^{\wedge}, \tag{3}$$

where $\mathbf{1} \in \mathbb{R}^{4\times4}$ is the identity matrix, $\boldsymbol{\xi} \in \mathbb{R}^{6\times1}$ is the pose in vector form and Δt is the duration of the tracklet. Substituting (3) into (2) approximates the error term as,

$$\begin{aligned}\mathbf{e}_{\text{fast},j,k} &\approx \mathbf{P}\left(\mathbf{p}_{k}^{j,k} - (\mathbf{1} + \Delta t_{k,k'}\boldsymbol{\varpi}^{\wedge})\mathbf{p}_{k'}^{j,k'}\right)\\ &= \mathbf{d}_{j,k} - \Delta t_{k,k'}\mathbf{D}_{j,k}\boldsymbol{\varpi},\end{aligned} \tag{4}$$

where

$$\begin{aligned}\Delta t_{k,k'} &= t_k - t_{k'},\\ \mathbf{d}_{j,k} &= \mathbf{P}\left(\mathbf{p}_{k}^{j,k} - \mathbf{p}_{k'}^{j,k'}\right),\\ \mathbf{D}_{j,k} &= \mathbf{P}\left(\mathbf{p}_{k'}^{j,k'}\right)^{\odot},\end{aligned}$$

and $(\cdot)^{\odot}$ is the $\mathbb{R}^{4\times1}$ to $\mathbb{R}^{4\times6}$ operator [34]. Substituting (4) into the Euclidean space cost function in (1), differentiating with respect to $\boldsymbol{\varpi}$, and setting the result equal to zero gives the velocity best describing a set of M_{rand} tracklets,

$$\boldsymbol{\varpi} = \left(\sum_{j=1}^{M_{\text{rand}}} \Delta t_{j,k}^{2}\mathbf{D}_{j,k}^{T}\mathbf{D}_{j,k}\right)^{-1}\left(\sum_{j=1}^{M_{\text{rand}}} \Delta t_{j,k}\mathbf{D}_{j,k}^{T}\mathbf{d}_{j,k}\right).$$

This velocity can then be used to segment all M_{clst} tracklets into inliers and outliers by comparing their reprojection error relative to tracklet length,

$$e_{\text{rel}} = \frac{\mathbf{y}_{j,k} - \mathbf{s}\left(\mathbf{z}\left(\exp\left(\Delta t_{k,k'}\boldsymbol{\varpi}^{\wedge}\right), \mathbf{p}_{k'}^{j,k'}\right)\right)}{\|\mathbf{y}_{j,k} - \mathbf{y}_{j,k'}\|}, \tag{5}$$

to a user-specified threshold, where $\mathbf{y}_{j,k}$ is a measurement of the j^{th} landmark at time t_k and $\mathbf{s}(\cdot)$ is the nonlinear camera projection model from a 3D landmark to a 2D image point.

The process of randomly selecting a small number of tracklets, quickly calculating a velocity from them, and then using this velocity to classify all the tracklets as inliers or outliers is repeated a user-specified number of times. The largest inlier set found is used as the initial segmentation for the full iterative MC-RANSAC.

2) Iterative MC-RANSAC: A more accurate iterative MC-RANSAC approach minimizes the reprojection error of each tracklet in image space with the cost function,

$$J_{\text{iter}}(\boldsymbol{\varpi}) = \frac{1}{2}\sum_{j=1}^{M_{\text{fast}}} \mathbf{e}_{\text{iter},j,k}^{T}\mathbf{R}_{j,k}^{-1}\mathbf{e}_{\text{iter},j,k}, \tag{6}$$

where $\mathbf{R}_{j,k}$ is the covariance matrix for the measurements of the j^{th} tracklet and M_{fast} is the number of inliers found by the fast MC-RANSAC. The reprojection error is

$$\mathbf{e}_{\text{iter, j,k}} = \mathbf{y}_{j,k} - \mathbf{s}\left(\mathbf{z}\left(\exp\left(\Delta t_{k,k'}\boldsymbol{\varpi}^{\wedge}\right), \mathbf{p}_{k'}^{j,k'}\right)\right). \tag{7}$$

The error function is linearized by representing the velocity as a nominal value, $\bar{\boldsymbol{\varpi}}$, and a small perturbation, $\delta\boldsymbol{\varpi}$,

$$\boldsymbol{\varpi} = \bar{\boldsymbol{\varpi}} + \delta\boldsymbol{\varpi}. \tag{8}$$

Substituting (7) and (8) into (6), differentiating with respect to $\delta\boldsymbol{\varpi}$, and setting the result equal to zero gives the perturbation that minimizes the linearization,

$$\delta\boldsymbol{\varpi}^{*} = \left(\sum_{j=1}^{M_{\text{fast}}} \mathbf{H}_{j,k}^{T}\mathbf{R}_{j,k}^{-1}\mathbf{H}_{j,k}\right)^{-1}\left(\sum_{j=1}^{M_{\text{fast}}} \mathbf{H}_{j,k}^{T}\mathbf{R}_{j,k}^{-1}\mathbf{e}_{\text{iter},j,k}\right),$$

where $\mathbf{H}_{j,k}$ is the Jacobian of error function in (7),

$$\mathbf{H}_{j,k} = \left.\frac{\partial\mathbf{s}}{\partial\mathbf{z}}\right|_{\bar{\mathbf{z}}} \Delta t_{k,k'}\mathbf{T}_{k,k'}\left(\mathbf{p}_{k'}^{j,k'}\right)^{\odot}\boldsymbol{\mathcal{T}}_{k,k'}^{-1}\boldsymbol{\mathcal{J}}_{k,k'},$$

where the partial derivative of the sensor model is evaluated at the nominal value, $\boldsymbol{\mathcal{T}}_{k,k'}$ is the adjoint of $SE(3)$ and $\boldsymbol{\mathcal{J}}_{k,k'}$ is the left Jacobian of $SE(3)$ [34].

This process is iterated until convergence to find the velocity best describing the initial inlier set found by fast MC-RANSAC,

$$\bar{\boldsymbol{\varpi}} \leftarrow \bar{\boldsymbol{\varpi}} + \delta\boldsymbol{\varpi}^{*}.$$

This velocity is then used to segment all M_{clst} tracklets into inliers and outliers by comparing their reprojection error using (5) to a user-specified threshold. This is more accurate outlier rejection than using fast MC-RANSAC alone.

C. Continuous-Time Trajectory Optimization

Traditional discrete-time estimation requires at least three measurements at every estimation state. This is often achieved in event-based VO by grouping event features to common times, which reduces the temporal resolution of event cameras. Continuous-time trajectory estimation can instead operate directly on asynchronous tracklets, which may result in estimation states with less than three measurements, by incorporating a motion prior or basis function. This allows continuous-time estimation techniques to maintain the temporal resolution of the event cameras.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on September 12,2023 at 13:39:40 UTC from IEEE Xplore. Restrictions apply.

The inlier tracklets from MC-RANSAC are used to define the trajectory optimization problem (Fig. 2). Each unique tracklet timestamp defines a state in the estimation problem. The continuous-time trajectory is estimated from these states using Gaussian process regression with a WNOA prior [24] (Section III-C1). This iterative process uses the velocities found during MC-RANSAC as an initial condition.

The WNOA motion prior is physically founded and accounts for real-world kinematics, unlike other continuous-time parametrizations that enforce mathematical smoothness independent of its physical plausibility. It is compared quantitatively to other estimation techniques in [24], [35]. The resulting trajectory can be queried for the camera pose at any timestamp in the estimation window (Section III-C2).

1) WNOA Estimator: The estimated states are defined as

$$\mathbf{x} = \left\{ \mathbf{T}_{k,1}, \boldsymbol{\varpi}_k, \mathbf{p}_1^{j,1} \right\}_{j=1,\ldots,M, k=1,\ldots,K},$$

where $\mathbf{T}_{k,1} \in SE(3)$ is the pose at the time t_k relative to the initial pose, $\boldsymbol{\varpi}_k$ is the corresponding 6DOF body-centric velocity and $\mathbf{p}_1^{j,1}$ is the position of the j^{th} landmark relative to the initial pose. The camera trajectory is represented by discrete estimated states, $\mathbf{T}_{k,1}$ and $\boldsymbol{\varpi}_k$, which can be denoted with a slight abuse of notation as $\mathbf{x}_k = \{\mathbf{T}_{k,1}, \boldsymbol{\varpi}_k\}$. The acceleration, $\dot{\boldsymbol{\varpi}}$, is assumed to be a zero-mean, white-noise Gaussian process,

$$\dot{\boldsymbol{\varpi}} \sim \mathcal{GP}(\mathbf{0}, \mathbf{Q}_c \delta(t - t')), \tag{9}$$

where $\mathbf{Q}_c \in \mathbb{R}^{6\times 6}$ is a diagonal power spectral density matrix, and $\delta(\cdot)$ is the Dirac delta function.

The WNOA assumption defines a locally constant velocity motion model. A local trajectory state, $\boldsymbol{\gamma}_k$, can be defined as a continuous-time function with respect to the global trajectory state,

$$\boldsymbol{\gamma}_k(\tau) = \begin{bmatrix} \boldsymbol{\xi}_k(\tau) \\ \dot{\boldsymbol{\xi}}_k(\tau) \end{bmatrix} = \begin{bmatrix} \ln(\mathbf{T}_k(\tau)\mathbf{T}_k^{-1})^\vee \\ \boldsymbol{\mathcal{J}}(\ln(\mathbf{T}_k(\tau)\mathbf{T}_k^{-1})^\vee)^{-1}\boldsymbol{\varpi}_k(\tau) \end{bmatrix},$$

where $\mathbf{T}_k(\tau)$ is the pose at time $t_k \le \tau \le t_{k+1}$, $\boldsymbol{\mathcal{J}}(\cdot)^{-1}$ is the inverse left Jacobian function, $\ln(\cdot)$ is the inverse exponential map, and $(\cdot)^\vee$ is the inverse lifting operator [34].

The estimator minimizes a joint cost function,

$$J_{\text{joint}}(\mathbf{x}) = J_{\text{prior}}(\mathbf{x}) + J_{\text{meas}}(\mathbf{x}), \tag{10}$$

where $J_{\text{prior}}(\mathbf{x})$ is the motion prior cost function and $J_{\text{meas}}(\mathbf{x})$ is the measurement cost term. The prior cost function penalizes trajectory states that deviate from WNOA assumption. The prior cost function is

$$J_{\text{prior}}(\mathbf{x}) = \frac{1}{2}\sum_{k=1}^{K-1} \mathbf{e}_{\text{prior},k+1,k}^T \mathbf{Q}_k^{-1}(t_{k+1}) \mathbf{e}_{\text{prior},k+1,k},$$

where the prior error is

$$\mathbf{e}_{\text{prior},k+1,k} = \begin{bmatrix} \ln\left(\mathbf{T}_{k+1,1}\mathbf{T}_{k,1}^{-1}\right)^\vee - (t_{k+1} - t_k)\boldsymbol{\varpi}_k \\ \boldsymbol{\mathcal{J}}\left(\ln\left(\mathbf{T}_{k+1,1}\mathbf{T}_{k,1}^{-1}\right)^\vee\right)^{-1} \boldsymbol{\varpi}_{k+1} - \boldsymbol{\varpi}_k, \end{bmatrix} \tag{11}$$

and the prior covariance matrix at t_τ from t_k is,

$$\mathbf{Q}_k(\tau) = \begin{bmatrix} \frac{1}{3}\Delta t_{\tau,k}^3 \mathbf{Q}_c & \frac{1}{2}\Delta t_{\tau,k}^2 \mathbf{Q}_c \\ \frac{1}{2}\Delta t_{\tau,k}^2 \mathbf{Q}_c & \Delta t_{\tau,k} \mathbf{Q}_c \end{bmatrix}, \tag{12}$$

where $\Delta t_{\tau,k} = \tau - t_k$ and $\mathbf{Q}_c \in \mathbb{R}^{6\times 6}$ is the power spectral density matrix defined in (9).

The measurement cost function that minimizes the feature tracklet reprojection error is

$$J_{\text{meas}}(\mathbf{x}) = \frac{1}{2}\sum_{j,k} \mathbf{e}_{\text{meas},j,k}^T \mathbf{R}_{j,k}^{-1} \mathbf{e}_{\text{meas},j,k},$$

where $\mathbf{R}_{j,k} \in \mathbb{R}^{3\times 3}$ is the measurement covariance matrix of the j^{th} landmark viewed from the k^{th} state and the tracklet reprojection error is

$$\mathbf{e}_{\text{meas},j,k} = \mathbf{y}_{j,k} - \mathbf{s}\left(\mathbf{z}\left(\mathbf{T}_{k,1}, \mathbf{p}_1^{j,1}\right)\right). \tag{13}$$

The optimal camera trajectory is obtained by optimizing the joint cost function,

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \{J_{\text{meas}}(\mathbf{x}) + J_{\text{prior}}(\mathbf{x})\},$$

using the Gauss-Newton method. The states are approximated as operating points, $\mathbf{x}_{\text{op}} = \{\mathbf{T}_{\text{op}}, \boldsymbol{\varpi}_{\text{op}}, \mathbf{p}_{\text{op}}\}$, and perturbations, $\delta\mathbf{x} = \{\delta\boldsymbol{\xi}, \delta\boldsymbol{\varpi}, \delta\boldsymbol{\zeta}\}$, linearizing (10) as

$$J_{\text{joint}}(\mathbf{x}) = J_{\text{joint}}(\mathbf{x}_{\text{op}}) - \mathbf{b}^T\delta\mathbf{x} + \frac{1}{2}\delta\mathbf{x}^T\mathbf{A}\delta\mathbf{x}, \tag{14}$$

where

$$\begin{aligned} \mathbf{A} &= \sum_{j,k} \mathbf{P}_{j,k}^T \mathbf{G}_{j,k}^T \mathbf{R}_{j,k}^{-1} \mathbf{G}_{j,k} \mathbf{P}_{j,k} \\ &\quad + \sum_k \mathbf{P}_k^T \mathbf{E}_k^T \mathbf{Q}_k^{-1}(t_{k+1}) \mathbf{E}_k \mathbf{P}_k, \\ \mathbf{b} &= \sum_{j,k} \mathbf{P}_{j,k}^T \mathbf{G}_{j,k}^T \mathbf{R}_{j,k}^{-1} \mathbf{e}_{\text{meas},j,k} \\ &\quad + \sum_k \mathbf{P}_k^T \mathbf{E}_k^T \mathbf{Q}_k^{-1}(t_{k+1}) \mathbf{e}_{\text{prior},k+1,k}, \end{aligned}$$

where $\mathbf{P}_{j,k}$ and $\mathbf{P}_k$ are matrices to pick the specific components of the total perturbation, $\mathbf{G}_{j,k}$ is the Jacobian of (13),

$$\mathbf{G}_{j,k} = \left.\frac{\partial \mathbf{s}}{\partial \mathbf{z}}\right|_{\bar{\mathbf{z}}} \begin{bmatrix} \left(\mathbf{T}_{\text{op},k,1}\mathbf{p}_{\text{op},1}^{j,1}\right)^\odot & \mathbf{0} & \mathbf{T}_{\text{op},k,1}\begin{bmatrix} \mathbf{1} \\ \mathbf{0}^T \end{bmatrix} \end{bmatrix},$$

and $\mathbf{E}_k$ is the Jacobian of the prior error function in (11),

$$\mathbf{E}_k = \begin{bmatrix} \mathbf{E}_{11} & \Delta t_{k+1,k}\mathbf{1} & \mathbf{E}_{13} & \mathbf{0} \\ \mathbf{E}_{21} & \mathbf{1} & \mathbf{E}_{23} & \mathbf{E}_{24} \end{bmatrix},$$

where

$$\begin{aligned} \mathbf{E}_{11} &= \boldsymbol{\mathcal{J}}_{k+1,k}^{-1}\boldsymbol{\mathcal{T}}_{k+1,k}, & \mathbf{E}_{13} &= \mathbf{E}_{24} = -\boldsymbol{\mathcal{J}}_{k+1,k}^{-1}, \\ \mathbf{E}_{21} &= \frac{1}{2}\boldsymbol{\varpi}_{k+1}^\curlywedge \boldsymbol{\mathcal{J}}_{k+1,k}^{-1}\boldsymbol{\mathcal{T}}_{k+1,k}, & \mathbf{E}_{23} &= -\frac{1}{2}\boldsymbol{\varpi}_{k+1}^\curlywedge \boldsymbol{\mathcal{J}}_{k+1,k}^{-1}, \end{aligned}$$

and $(\cdot)^\curlywedge$ is the $\mathbb{R}^{6\times 1}$ to $\mathbb{R}^{6\times 6}$ operator [34].

Taking the derivative of (14) with respect to $\delta\mathbf{x}^*$, setting the result equal to zero and solving the resulting linear system, $\mathbf{A}\delta\mathbf{x}^* = \mathbf{b}$, gives the perturbation that minimizes the linearization. The estimation states are updated,

$$\begin{aligned} \mathbf{T}_{\text{op},k,1} &\leftarrow \exp(\delta\boldsymbol{\xi}^*)\mathbf{T}_{\text{op},k,1}, \\ \boldsymbol{\varpi}_{\text{op},k} &\leftarrow \boldsymbol{\varpi}_{\text{op},k} + \delta\boldsymbol{\varpi}^*, \\ \mathbf{p}_{\text{op},j} &\leftarrow \mathbf{p}_{\text{op},j} + \delta\boldsymbol{\zeta}^*. \end{aligned}$$

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on September 12,2023 at 13:39:40 UTC from IEEE Xplore. Restrictions apply.

and the process is iterated until convergence. The final estimate is the landmark position and the continuous-time trajectory represented as discrete state poses, discrete local velocities, and the WNOA prior.

2) Querying the Continuous-Time Trajectory: The continuous-time trajectory can be queried for the camera pose at any time during the estimation window, $t_1 \leq \tau \leq t_K$. The pose is interpolated between the states at the two closest times, $t_m \leq \tau \leq t_n$, using the WNOA prior. The local trajectory state, $\boldsymbol{\gamma}_m(\tau)$, is interpolated [24] as

$$\boldsymbol{\gamma}_m(\tau) = \boldsymbol{\Lambda}(\tau)\boldsymbol{\gamma}_m(t_m) + \boldsymbol{\Omega}(\tau)\boldsymbol{\gamma}_n(t_n),$$

where

$$\boldsymbol{\Lambda}(\tau) = \boldsymbol{\Phi}(\tau, t_m) - \boldsymbol{\Omega}(\tau)\boldsymbol{\Phi}(t_n, t_m)$$

$$\boldsymbol{\Omega}(\tau) = \mathbf{Q}_m(\tau)\boldsymbol{\Phi}(t_n, \tau)^T\mathbf{Q}_m(t_n)^{-1}$$

$$\boldsymbol{\Phi}(\tau, t_m) = \begin{bmatrix} \mathbf{1} & (\tau - t_m)\mathbf{1} \\ \mathbf{0} & \mathbf{1} \end{bmatrix},$$

and $\mathbf{Q}_m(\cdot)$ is defined in (12).

This interpolation can also be used to define the estimation states at a subset of the measurement times [24], [35].

IV. Experiments

The presented pipeline is evaluated on the MVSEC dataset [11] and compared against the publicly available ESVO [9], a discrete event-based stereo VO pipeline. MVSEC consists of complex, nonconstant-velocity motion with stereo event camera data and 100 Hz ground truth poses for indoor scenes. The *indoor1* and *indoor3* sequences are used since ESVO provides tuning for these sequences. The performance of both algorithms is evaluated using global and relative error (Section IV-A) and the results are discussed in Section IV-B.

A sliding-window version of the system is implemented in MATLAB using LIBVISO2 [36] for feature detection and tracking. The sliding window width is five and LIBVISO2 is run on both full- and half-resolution images. Tracklets are filtered out if they

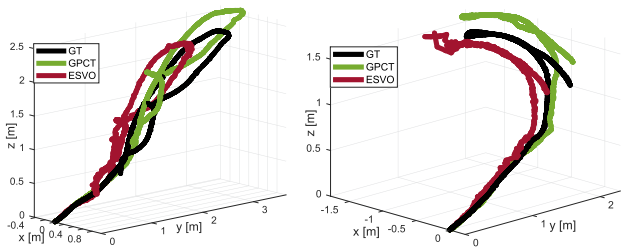
1) have more than 20 ms time difference between stereo features,
2) have less than 2px disparity,
3) have less than 2px length, and
4) last less than 40 ms in time.

Outlier rejection was done with 10000 iterations of fast MC-RANSAC followed with one call of iterative MC-RANSAC, both using an inlier threshold of 5%. The estimator uses the covariances $\mathbf{Q}_c^{-1} = 50\text{diag}(1, 1, 1, 10, 10, 10)$ and $\mathbf{R}_{j,k}^{-1} = 0.1\text{diag}(5, 5, 1)$ and terminates when the cost change between two iterations is less than 1%.

A. Evaluation Metrics

Global error quantifies the estimator accuracy relative to the initial pose. Relative error quantifies the amount of error in each estimate and is often used to calculate aggregate values over the trajectory. Both can be calculated from ground truth using a single general equation [37],

$$\text{err}(t_m, t_n) = \ln\left(\mathbf{T}_{m^{\text{GT}},m}\mathbf{T}_{n,m}\mathbf{T}_{m^{\text{GT}},m}^{-1}\mathbf{T}_{n^{\text{GT}},m^{\text{GT}}}^{-1}\right)^{\vee},$$



(a) MVSEC *indoor1* sequence (b) MVSEC *indoor3* sequence

Fig. 3. Trajectory plots of the presented Gaussian process continuous-time approach (GPCT), ESVO and ground truth (GT) results in 3D space. GPCT performs better in challenging scenarios like rotation and back-and-forth motions. It also has a smoother trajectory due to the WNOA motion prior.

where $\mathbf{T}_{n,m}$ is the estimated transform to the n^{th} frame from the m^{th} frame, $\mathbf{T}_{n^{\text{GT}},m^{\text{GT}}}$ is the ground truth transform to the n^{th} frame from the m^{th} frame, and $\mathbf{T}_{m^{\text{GT}},m}$ is the transform to the ground truth m^{th} frame from the estimate of the m^{th} frame. The global error at a time, t_k, is then defined relative to the initial pose,

$$\text{GE}(t_k) = \text{err}(t_k, t_1),$$

and the relative error is defined relative to the previous pose,

$$\text{RE}(t_k) = \text{err}(t_k, t_{k-1}).$$

The error of the presented continuous-time system is calculated using the timestamps of ESVO. For global error, the continuous-time trajectory is aligned with ESVO's initial pose and then queried at its own state times. For relative error, the continuous-time trajectory is only queried at the discrete state times of ESVO so that both estimators have the same durations between estimates and their relative errors can be compared directly. The high frequency ground truth trajectory is linearly interpolated to the timestamps of the estimated states.

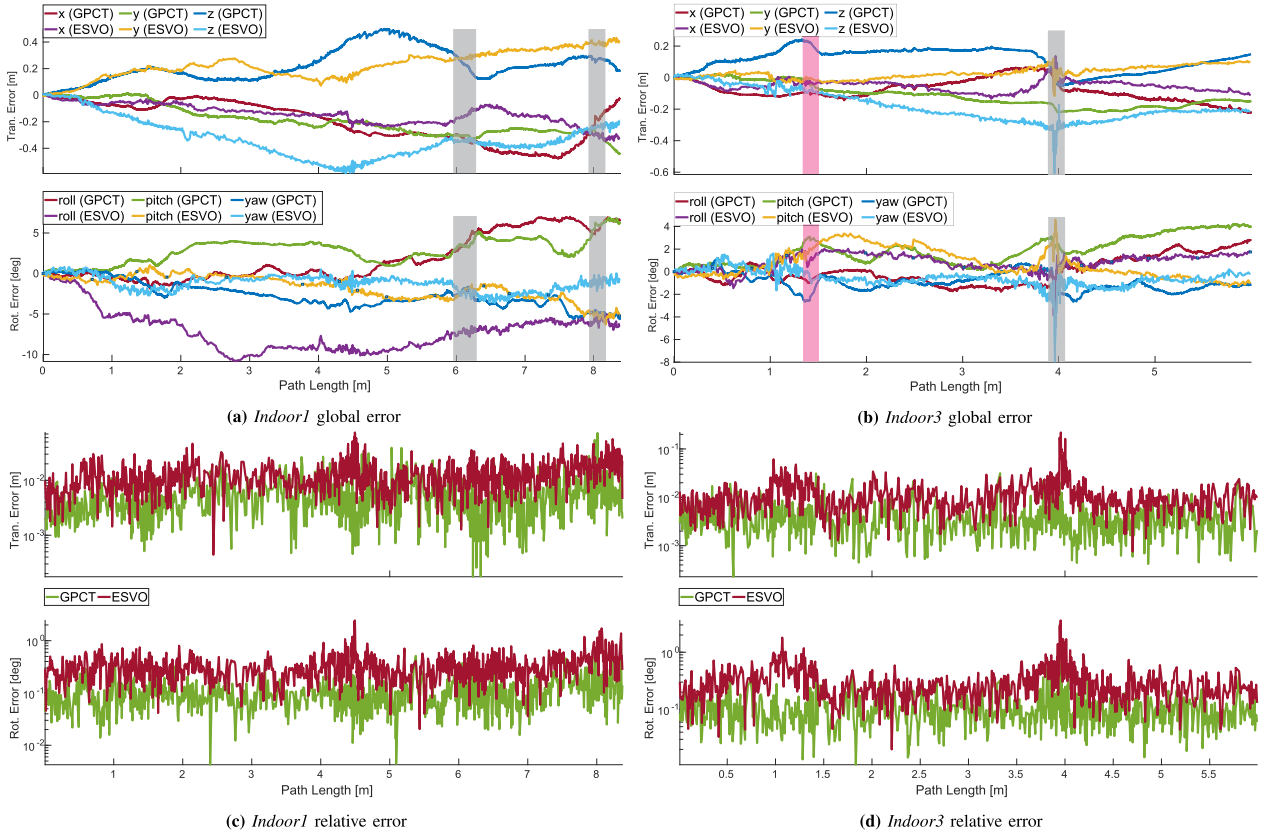
The prototype implementation of the continuous-time system does not run in real time. The system's computational performance should be improved by implementing it in a more efficient language, such as C++, and using keytimes to reduce the number of the estimation states [24], [35].

B. Results

The estimated trajectories are evaluated qualitatively relative to the ground truth in 3D (Fig. 3). They are evaluated quantitatively by calculating the global and relative errors with respect to ground truth (Fig. 4, Tables I and II). The errors are evaluated statistically using root-mean-squared (RMS), standard deviation (St. Dev.) and maximum error (Max). The maximum global error and final global error are also presented as a percentage of the integration of the norm of the relevant ground-truth component (e.g., path length). The qualitative results demonstrate the benefits of the WNOA prior, with the presented system having a smoother estimated trajectory than that of ESVO. The presented system also quantitatively has smaller RMS relative error and better or equivalent maximum global error than ESVO.

1) MVSEC indoor1: The presented system has a similar performance in global error to ESVO (Fig. 4(a)). The relative error is smaller than ESVO (Fig. 4(c)) with an RMS value that is two-times better (Table II). This illustrates the better local

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on September 12,2023 at 13:39:40 UTC from IEEE Xplore. Restrictions apply.



(a) *Indoor1* global error

(b) *Indoor3* global error

(c) *Indoor1* relative error

(d) *Indoor3* relative error

Fig. 4. Global and relative error of the presented Gaussian process continuous-time approach (GPCT) and ESVO as a function of path length on MVSEC *indoor1* and *indoor3*. The translational and rotational global error are plotted separately and the relative error is plotted as a single $SE(3)$ quantity. Note that the relative error y-axis is a logarithmic scale which suppresses spikes. The grey shaded areas represent complex camera motions that result in poor feature quality. The pink areas indicate featureless regions where the presented technique relies solely on the motion prior.

TABLE I
GLOBAL ERROR OF THE PRESENTED GAUSSIAN PROCESS CONTINUOUS-TIME APPROACH (GPCT) AND ESVO

		Indoor1					*Indoor3*				
		Max	Max %	Final %	RMS	St. Dev.	Max	Max %	Final %	RMS	St. Dev.
GPCT	tran.	**0.635**	**7.58%**	**5.73%**	**0.372**	0.192	**0.306**	**5.11%**	5.08%	**0.218**	**0.059**
	rota.	0.195	**7.98%**	7.6%	**0.087**	0.05	**0.088**	**4.23%**	4.13%	0.051	0.023
	$SE(3)$	**0.639**	**7.18%**	**5.78%**	**0.382**	0.196	**0.319**	**4.95%**	4.93%	**0.224**	**0.061**
ESVO	tran.	0.642	7.66%	6.61%	0.477	**0.152**	0.683	11.4%	**4.33%**	0.232	0.104
	rota.	**0.189**	8.87%	**6.8%**	0.139	**0.039**	0.165	8.66%	**2.39%**	**0.038**	**0.02**
	$SE(3)$	0.663	7.55%	6.51%	0.497	**0.155**	0.697	10.9%	**4.12%**	0.235	0.103

TABLE II
RELATIVE ERROR OF THE PRESENTED GAUSSIAN PROCESS CONTINUOUS-TIME APPROACH (GPCT) AND ESVO

		Indoor1			*Indoor3*		
		Max	RMS	St. Dev.	Max	RMS	St. Dev.
GPCT	tran.	$\mathbf{73 \cdot 10^{-3}}$	$\mathbf{7.5 \cdot 10^{-3}}$	$\mathbf{5.3 \cdot 10^{-3}}$	$\mathbf{32 \cdot 10^{-3}}$	$\mathbf{5.5 \cdot 10^{-3}}$	$\mathbf{3.6 \cdot 10^{-3}}$
	rota.	$\mathbf{18 \cdot 10^{-3}}$	$\mathbf{2.4 \cdot 10^{-3}}$	$\mathbf{1.4 \cdot 10^{-3}}$	$\mathbf{13 \cdot 10^{-3}}$	$\mathbf{2.3 \cdot 10^{-3}}$	$\mathbf{1.3 \cdot 10^{-3}}$
	$SE(3)$	$\mathbf{74 \cdot 10^{-3}}$	$\mathbf{7.9 \cdot 10^{-3}}$	$\mathbf{5.3 \cdot 10^{-3}}$	$\mathbf{32 \cdot 10^{-3}}$	$\mathbf{5.9 \cdot 10^{-3}}$	$\mathbf{3.6 \cdot 10^{-3}}$
ESVO	tran.	$75 \cdot 10^{-3}$	$17 \cdot 10^{-3}$	$9.2 \cdot 10^{-3}$	$219 \cdot 10^{-3}$	$24 \cdot 10^{-3}$	$19 \cdot 10^{-3}$
	rota.	$43 \cdot 10^{-3}$	$7.2 \cdot 10^{-3}$	$3.9 \cdot 10^{-3}$	$63 \cdot 10^{-3}$	$8.3 \cdot 10^{-3}$	$5.6 \cdot 10^{-3}$
	$SE(3)$	$86 \cdot 10^{-3}$	$18 \cdot 10^{-3}$	$9.2 \cdot 10^{-3}$	$220 \cdot 10^{-3}$	$25 \cdot 10^{-3}$	$19 \cdot 10^{-3}$

consistency of the trajectory estimate and explains the smooth trajectory plot in Fig. 3(a).

The grey shaded areas in Fig. 4(a) denote complex motions where significant error occurs. When the camera undergoes large motions (e.g., large rotation) the observed scene changes drastically. This reduces the quality and quantity of tracklets found by the clustered feature detection and tracking and as a result the quality of the trajectory estimation. This can be improved with better feature tracking, likely specifically designed for event-based cameras.

2) MVSEC indoor3: The presented system estimates a smoother trajectory than ESVO in the *indoor3* segment. The estimator qualitatively describes the ground truth motion and is locally consistent. It has almost a four-times better RMS relative error than ESVO (Table II)

Challenging camera motions are marked in Fig. 4(b) as pink and grey shaded areas. The grey areas correspond to a back-and-forth camera motion and results in poor performance for both techniques. ESVO loses tracking and has to reinitialize, resulting in large global and relative error spikes (Fig. 4(a) and (d)). The

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on September 12,2023 at 13:39:40 UTC from IEEE Xplore. Restrictions apply.

presented system performs better than ESVO in this area, but feature quality also decreases which increases both global and relative error.

The pink areas correspond to a large rotational motion. Feature tracking fails for the presented system during this motion but the WNOA prior carries the estimate through without causing significant error. This demonstrates the robustness of the presented system to feature tracking failure and the potential to improve performance with ongoing research on event-based feature tracking.

V. Conclusion

This paper presents a complete event-based continuous-time VO pipeline that maintains the temporal resolution of event cameras throughout the estimation. This pipeline can use either traditional frame-based or new event-based feature detectors and trackers to generate asynchronous tracklets. These tracklets are filtered for outliers using a motion-compensated RANSAC that accounts for the unique tracklet times. The pipeline estimates a continuous-time trajectory using nonparametric Gaussian process regression with a physically founded WNOA motion prior that can be queried for the camera pose at any time within the estimation window. The system's performance is evaluated on the publicly available MVSEC dataset where it achieves better performance than the publicly available ESVO pipeline, especially in terms of RMS relative error.

References

[1] H. P. Moravec, *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. Stanford, CA, USA: Stanford Univ., 1980.

[2] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2004.

[3] Y. Cheng, M. Maimone, and L. Matthies, "Visual Odometry on the Mars Exploration Rovers," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2005, pp. 903–910.

[4] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.

[5] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast Semi-Direct Monocular Visual Odometry," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 15–22.

[6] G. Gallego et al., "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, Jan. 2022.

[7] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 994–1001, Apr. 2018.

[8] H. Rebecq, T. Horstschäfer, G. Gallego, and D. Scaramuzza, "EVO: A geometric approach to event-based 6-DoF parallel tracking and mapping in real time," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 593–600, Apr. 2017.

[9] Y. Zhou, G. Gallego, and S. Shen, "Event-based stereo visual odometry," *T-RO*, vol. 37, no. 5, pp. 1433–1450, 2021.

[10] S. Anderson and T. D. Barfoot, "RANSAC for motion-distorted 3D visual sensors," in *Proc. Int. Conf. Intell. Robots Syst.*, 2013, pp. 2093–2099.

[11] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The multivehicle stereo event camera dataset: An event camera dataset for 3D perception," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 2032–2039, Jul. 2018.

[12] D. Weikersdorfer, R. Hoffmann, and J. Conradt, "Simultaneous localization and mapping for event-based vision systems," in *Proc. Int. Conf. Comput. Vis. Syst.*, 2013, pp. 133–142.

[13] H. Kim, A. Handa, R. Benosman, S.-H. Ieng, and A. Davison, "Simultaneous mosaicing and tracking with an event camera," in *Proc. Brit. Mach. Vis. Conf.*, 2014, doi: 10.5244/c.28.26.

[14] B. Kueng, E. Mueggler, G. Gallego, and D. Scaramuzza, "Low-latency visual odometry using event-based feature tracks," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2016, pp. 16–23.

[15] D. Weikersdorfer, D. B. Adrian, D. Cremers, and J. Conradt, "Event-based 3D SLAM with a depth-augmented dynamic vision sensor," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 359–364.

[16] A. Zihao Zhu, N. Atanasov, and K. Daniilidis, "Event-based visual inertial odometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5816–5824.

[17] H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization," in *Proc. Brit. Mach. Vis. Conf.*, 2017, doi: 10.5244/c.31.16.

[18] P. Chen, W. Guan, and P. Lu, "ESVIO: Event-based stereo visual inertial odometry," *IEEE Robot. Automat. Lett.*, vol. 8, no. 6, pp. 3661–3668, Jun. 2022.

[19] I. Alzugaray and M. Chli, "Asynchronous corner detection and tracking for event cameras in real time," *Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3177–3184, Oct. 2018.

[20] C. Le Gentil, F. Tschopp, I. Alzugaray, T. Vidal-Calleja, R. Siegwart, and J. Nieto, "IDOL: A framework for IMU-DVS odometry using lines," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5863–5870.

[21] H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3D reconstruction and 6-DoF tracking with an event camera," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 349–364.

[22] H. Rebecq, G. Gallego, E. Mueggler, and D. Scaramuzza, "EMVS: Event-based multi-view stereo–3D reconstruction with an event camera in real-time," in *Proc. Int. J. Comput. Vis.*, 2018, pp. 1394–1414.

[23] A. Hadviger, I. Cvišić, I. Marković, S. Vražić, and I. Petrović, "Feature-based event stereo visual odometry," in *Proc. Eur. Conf. Mobile Robots*,2021, pp. 1–6.

[24] S. Anderson and T. D. Barfoot, "Full STEAM ahead: Exactly sparse Gaussian process regression for batch continuous-time trajectory estimation on SE(3)," in *Proc. Int. Conf. Intell. Robots Syst.*, 2015, pp. 157–164.

[25] P. Furgale, C. H. Tong, T. D. Barfoot, and G. Sibley, "Continuous-time batch trajectory estimation using temporal basis functions," *IJRR*, vol. 34, no. 14, pp. 2088–2095, 2015.

[26] G. Cioffi, T. Cieslewski, and D. Scaramuzza, "Continuous-time vs. discrete-time vision-based SLAM: A comparative study," *RA-L*, vol. 7, no. 2, pp. 2399–2406, Apr. 2022.

[27] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza, "Continuous-time visual-inertial odometry for event cameras," *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1425–1440, Dec. 2018.

[28] Y. Wang et al., "Visual odometry with an event camera using continuous ray warping and volumetric contrast maximization," *Sensors*, vol. 22, no. 15, 2022, Art. no. 5687.

[29] K. Huang, Y. Wang, and L. Kneip, "Motion estimation of non-holonomic ground vehicles from a single feature correspondence measured over n views," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12706–12715.

[30] D. Liu, A. Parra, Y. Latif, B. Chen, T.-J. Chin, and I. Reid, "Asynchronous optimisation for event-based visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2022, pp. 9432–9438.

[31] R. Benosman, C. Clercq, X. Lagorce, S.-H Ieng, and C. Bartolozzi, "Event-based visual flow," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 407–417, Feb. 2013.

[32] I. Alzugaray and M. Chli, "HASTE: Multi-hypothesis asynchronous speeded-up tracking of events," in *Proc. Brit. Mach. Vis. Conf.*, 2020, Art. no. 744.

[33] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme," in *Proc. IEEE Intell. Veh. Symp.*, 2010, pp. 486–492.

[34] T. D. Barfoot, *State Estimation for Robotics*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[35] C. H. Tong, P. Furgale, and T. D. Barfoot, "Gaussian process gauss–newton for non-parametric simultaneous localization and mapping," *IJRR*, vol. 32, no. 5, pp. 507–525, 2013.

[36] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3D reconstruction in real-time," in *Proc. EEE Intell. Veh. Symp.*, 2011, pp. 963–968.

[37] K. M. Judd and J. D. Gammell, "Multimotion visual odometry (MVO)," 2021, *arXiv:2110.15169*.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on September 12,2023 at 13:39:40 UTC from IEEE Xplore. Restrictions apply.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Event-Based Stereo Visual Odometry With Native Temporal Resolution via Continuous-Time Gaussian Process Regression
Publication Status	Published
Publication Details	J. Wang and J. D. Gammell, "Event-Based Stereo Visual Odometry With Native Temporal Resolution via Continuous-Time Gaussian Process Regression," in IEEE Robotics and Automation Letters, vol. 8, no. 10, pp. 6707-6714, Oct. 2023, doi: 10.1109/LRA.2023.3311374.

Student Confirmation

Student Name:	Jianeng Wang		
Contribution to the Paper	My contributions to the paper were: • Developed the core idea behind the paper with co-authors • Implement the algorithm • Performed the experiments with co-authors • Wrote the paper with co-authors		
Signature		Date	2025/04/03

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Jonathan D. Gammell		
Supervisor comments This paper was written by Jianeng with feedback from me and presents work he developed under my supervision.		
Signature 	Date	2025-06-29

This completed form should be included in the thesis, at the end of the relevant chapter.

4.2 Discussion

This chapter presents an event-based stereo VO pipeline that fully leverages the native temporal resolution of the input event stream. The pipeline is compatible with any feature detector and tracker that preserves individual event timestamps. The frontend incorporates a tailored outlier rejection scheme specifically designed for asynchronous event-based feature tracklets, ensuring consistent inlier selection. The backend performs continuous-time trajectory estimation, allowing the camera state to be queried at any timestamp within the estimation window without compromising the temporal fidelity of the event data.

The pipeline has been evaluated on a publicly available dataset, demonstrating lower relative pose error and smoother trajectory estimates compared to state-of-the-art (SOTA) event-based VO methods. In the following, we would like to discuss the merits, limitations, and future improvements of the existing system in more detail.

4.2.1 Compatibility with Alternative Feature Detector and Tracker

A key merit of the proposed pipeline is its architectural design, which decouples the continuous-time backend from the specific choice of frontend feature detector and tracker. The system backend is designed to operate on a standardized input, a stream of asynchronous feature tracklets, where each tracklet is defined by its spatial coordinates and its start and end timestamps. Those tracklets are generated in the frontend.

In the presented manuscript, this compatibility was demonstrated using a frame-based feature detector and tracker, which generates these asynchronous feature tracklets from clustering events into image-like representations. However, the pipeline can natively support purely event-driven approaches, such as HASTE [3] or Messikommer et al. [159]. The event-based approaches produce the same standardized output format but have advantages in lower latency and higher temporal precision, as they avoid the intermediate step of creating event frames.

In the subsequent outlier rejection stage, the MC-RANSAC scheme, explicitly uses each tracklet's individual time duration to fit a motion model, which ensures consistent inlier tracklet set for backend processing whether they were generated synchronously (from a frame-based method) or asynchronously (from an event-based method).

While not experimentally demonstrated, this theoretical compatibility is a crucial aspect of the design, allowing the backend to benefit from future advances in event-based feature detection and tracking without requiring architectural modification. This way, we claim that theoretically the presented odometry pipeline can work with any feature detector and tracker as long as it can provide both geometric feature position and time duration.

4.2.2 Limitations of Experimental Validation

The quantitative evaluation of our pipeline was performed on two challenging sequences from a widely used public dataset. These sequences were selected because they are established benchmarks in the event-based vision community and contain the high-speed and rapid motions that highlight the advantages of our continuous-time approach. The results successfully demonstrate the core claim that leveraging native temporal resolution can lead to smoother and more accurate trajectory estimates.

However, we acknowledge that this evaluation on two sequences from a single dataset is not exhaustive and constitutes a limitation on the demonstrated generalizability of the work. These experiments do not cover the full spectrum of potential scenarios. For instance, system performance may differ in environments with sparse textures or slow motion, where the low event rate could pose challenges for both the frontend and backend. A more extensive validation across a wider variety of datasets, capturing different motion profiles and scene characteristics, would be required to fully characterise the system's performance and its robustness in these alternative conditions. Such an evaluation remains a vital step for future work.

4.2.3 Computational and Energy Considerations

A critical aspect of any event-based system is its potential for low-latency and low-power operation. While event cameras are highly energy-efficient, the computational pipeline must also be optimized for real-world deployment. The current MATLAB implementation is a proof-of-concept and does not operate in real time, with performance bottlenecks present in both the frontend and backend.

In the frontend, the computational load is heavily influenced by the quality of the feature tracklets. The current frame-based approach can generate a significant number of outlier tracklets, which in turn requires the MC-RANSAC algorithm to run for an excessive number of iterations to find a consistent inlier set. This is a primary bottleneck. The path to improving this involves adopting a high-quality event-based tracker, which would yield a higher inlier ratio and drastically reduce the cost of outlier rejection. Furthermore, integrating an IMU could provide a strong motion prior to initialize MC-RANSAC, further reducing the search space and iteration count.

In the backend, the current continuous-time formulation could lead to a significant number of states in the optimization problem, as each event pair could theoretically define a state. The most effective strategy to mitigate this is to sparsify the problem by introducing keytimes. By estimating trajectory states only at selected timestamps rather than for every event, the size of the Gaussian Process regression problem can be substantially reduced. This approach has been successfully explored in frameworks like STEAM [7] and other Gaussian process-based formulations [244].

Finally, for a practical implementation, the entire pipeline must be ported from MATLAB to a more performant language like C++. The combination of an improved frontend, a sparsified backend, and a C++ implementation would be the key to unlocking the real-time, energy-efficient potential of this event-based VO system.

4.2.4 Future Developments

The field of event-based visual odometry has advanced rapidly, with subsequent work focusing on the key challenges of frontend feature quality and backend computational efficiency. These developments can be categorized into two main branches, frontend innovations and backend fusion strategies.

Frontend Innovations have pursued two primary paths. The first is the development of purely event-based feature detectors and trackers to replace the latency-prone frame-based methods. A robust, low-latency frontend is critical for improving overall system performance. As a contribution in this direction, we developed a novel event-based corner detector that operates directly on the stream of events. This method, presented in detail in Appendix A, leverages the natural tendency of events to form on moving contours. By searching for intersecting edges on the Surface of Active Events (SAE), it achieves promising performance compared to other state-of-the-art detectors. Developing this into a full tracking pipeline remains a key area for future work. The second path has been multi-modal frontends, which improve robustness by combining events with other sensing modalities. This includes methods that track features detected in traditional frames using events [131] or direct formulations that jointly optimize over both data types, such as in EDS [92].

Backend Fusion Strategies have increasingly focused on tighter sensor integration, particularly with the IMU. Recent state-of-the-art estimators [182, 181] now incorporate IMU pre-integration factors as powerful constraints within the backend optimization, significantly enhancing robustness. Beyond improving accuracy, the IMU also offers a path to greater computational efficiency, directly addressing the challenges discussed previously. For example, velocity estimates from the IMU can provide a high-quality initialization for the trajectory optimization. Furthermore, in a sparse *keytime* formulation, IMU data can enable accurate and efficient state interpolation between the estimated keytimes, mitigating the computational burden of a dense, continuous-time approach. These trends toward native event-based frontends and tightly-fused, efficient backends represent the key evolutionary paths in the field, building upon the foundational concepts explored in this chapter.

5

Vision-Based Scene Understanding System For Exoskeletons

Contents

Self-balancing exoskeletons enable individuals with lower-limb disabilities to walk independently without the need for external support such as crutches. Much of the existing research in this domain focuses on hardware development and control strategies, often relying on pre-defined gait trajectories that are manually triggered via an operator control panel. While effective, this control paradigm limits the autonomy of the system.

Integrating a perception module into the exoskeleton platform would enable real-time environmental understanding, allowing for more responsive low-level decision-making. This, in turn, would support partial automation of locomotion tasks and

help to reduce the user's cognitive and physical load. However, for exoskeletons to be used reliably in daily, long-term scenarios, perception systems must go beyond the short-term reactivity of currently published systems. They should be capable of building persistent, reusable representations of the environment—supporting tasks such as localization, re-navigation, and long-term scene understanding.

This chapter presents *Exosense*, a vision-centric scene understanding system designed specifically for walking exoskeletons operating in indoor environments. Built upon a wide field-of-view multi-camera setup, Exosense is capable of generating rich, globally consistent elevation maps that integrate both semantic and terrain traversability information. The system is designed to operate robustly in dynamic and visually challenging scenarios, enabling exoskeletons to better perceive, interpret, and interact with their environment.

5.1 Exosense: A Vision-Based Scene Understanding System for Exoskeletons

The following article was published in the IEEE Robotics and Automation Letters (RA-L) [254]. An accompanying video is available online at: https://www.youtube.com/watch?v=IPPxuW4suqg.

© 2022 IEEE. Reprinted, with permission, from Jianeng Wang, Matias Mattamala, Christina Kassab, Guillaume Burger, Fabio Elnecave, Lintong Zhang, Marine Petriaux and Maurice Fallon, "Exosense: A Vision-Based Scene Understanding System for Exoskeletons," in IEEE Robotics and Automation Letters, 2025.

Exosense: A Vision-Based Scene Understanding System for Exoskeletons

Jianeng Wang, Matias Mattamala, *Member, IEEE*, Christina Kassab, Guillaume Burger, Fabio Elnecave, Lintong Zhang, Marine Petriaux, and Maurice Fallon, *Senior Member, IEEE*

***Abstract*—Self-balancing exoskeletons are a key enabling technology for individuals with mobility impairments. While the current challenges focus on human-compliant hardware and control, unlocking their use for daily activities requires a scene perception system. In this work, we present *Exosense*, a vision-centric scene understanding system for self-balancing exoskeletons. We introduce a multi-sensor visual-inertial mapping device as well as a navigation stack for state estimation, terrain mapping, and long-term operation. We tested Exosense attached to both a human leg and Wandercraft's *Personal Exoskeleton* in real-world indoor scenarios. This enabled us to test the system during typical periodic walking gaits, as well as future uses in multi-story environments. We demonstrate that Exosense can achieve an odometry drift of about 4 cm per meter traveled, and construct terrain maps under 1 cm average reconstruction error. It can also work in a visual localization mode in a previously mapped environment, providing a step towards long-term operation of exoskeletons.**

***Index Terms*—Wearable robotics, prosthetics and exoskeletons, RGB-D perception, mapping.**

I. Introduction

RECENT advances in self-balancing exoskeletons, such as Wandercraft's *Atalante* exoskeleton [1], are enabling individuals with lower-limb disabilities to walk independently without requiring additional support from crutches [2]. These powered exoskeletons are being used in controlled clinical and therapeutic contexts [3]. However, the ultimate goal is to enable users to do everyday activities at home and outdoors.

Exoskeleton development has primarily focused on hardware and control challenges, aiming to design systems that can support and transport individuals while mimicking natural human walking. Many of these approaches employ control schemes with pre-defined gait trajectories [4]. This requires manual activation by an operator using a control panel, which increases the metabolic cost [5]. Integrating perception systems into the loop could reduce metabolic costs by partly automating low-level control tasks, such as switching gait modes, crossing doorways, or climbing stairs.

Vision sensing, because of its rich source of information [6], has been the primary sensor used to achieve this. Cameras have been used to estimate the semantic class of the terrain (namely stairs, ramps, and level ground walking) [7], [8], to determine basic geometric features such as ground inclination and step height [9], as well as to detect potential obstacles in the environment [10]. While these approaches are effective in providing the instantaneous information required for low-level decision making, they do not aim to integrate this information in long-term representations, which could be reused when revisiting environments.

In this work, we take initial steps towards developing long-term operation of self-balancing exoskeletons by presenting *Exosense*, a vision-centric scene understanding system. Exosense aims to generate home-scale, rich scene representations from vision and geometry, capturing terrain structure, semantics, and traversability for localization and future navigation in previously visited environments. Our solution is primarily designed for self-balancing exoskeletons, such as the Wandercraft's *Personal Exoskeleton* (Fig. 1), which aims to be the first self-balancing exoskeleton designed for domestic use. To achieve this, we introduce a versatile multi-sensor unit that can be attached to the exoskeleton's leg or carried by a human as a leg-mounted wearable device. This placement allowed us to rigidly attach the Exosense to the exoskeleton, while also avoiding potential occlusions from the user's body. Additionally, it enabled us to develop and test Exosense in realistic human walking scenarios and to seamlessly transfer the system to the exoskeleton given the similarities we observed in the gait dynamics (Fig. 3).

The key contributions of our work are:

- A versatile, leg-mounted multi-sensor unit that provides wide-angle vision and depth sensing for state estimation, terrain mapping, and localization.
- A scene understanding system that builds local maps embedded with the terrain geometry, room semantics, and traversability of indoor environments.
- A study of the performance of visual odometry for different camera configurations during typical walking patterns,

Received 16 October 2024; accepted 3 February 2025. Date of publication 20 February 2025; date of current version 3 March 2025. This article was recommended for publication by Associate Editor Lukas M. Schmid and Editor Cesar Cadena upon evaluation of the reviewers' comments. This work was supported in part by a Royal Society University Research Fellowship awarded to Maurice Fallon and Christina Kassab, in part by the Horizon Europe project DigiForest under Grant 101070405 awarded to Jianeng Wang, and in part by the EPSRC C2C project under Grant EP/Z531212/1 awarded to Matias Mattamala. *(Corresponding author: Jianeng Wang.)*

Jianeng Wang, Matias Mattamala, Christina Kassab, Lintong Zhang, and Maurice Fallon are with the Department of Engineering Science Oxford Robotics Institute, University of Oxford, OX1 4AR Oxford, U.K. (e-mail: jianeng@robots.ox.ac.uk; matias@robots.ox.ac.uk; christina@robots.ox.ac.uk; lintong@robots.ox.ac.uk; mfallon@robots.ox.ac.uk).

Guillaume Burger, Fabio Elnecave, and Marine Petriaux are with Wandercraft SAS, 75004 Paris, France (e-mail: guillaume.burger@wandercraft.health; fabio.elnecave@wandercraft.health; marine.petriaux@wandercraft.health).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3543971, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3543971

2377-3766 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on March 05,2025 at 12:44:38 UTC from IEEE Xplore. Restrictions apply.

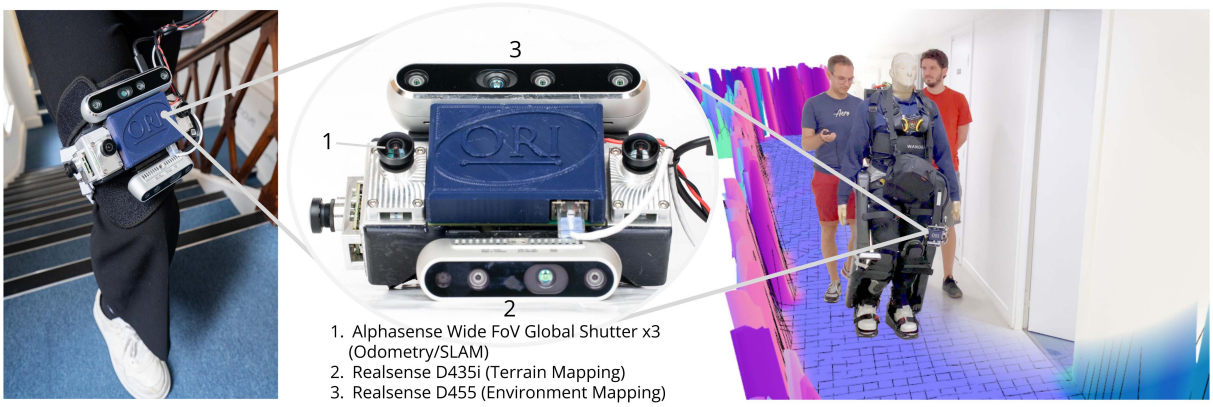


Fig. 1. Exosense is a vision-based hardware and software system for scene understanding by exoskeletons. We developed a specialized multi-sensor unit (center), consisting of three global shutter wide-angle Alphasense cameras and Realsense D435i and D455 RGB-D units to provide 3D terrain and environment measurements. The hardware can be worn as a human-leg-mounted wearable device (left) or as an lower-limb attachment for an exoskeleton, such as the Wandercraft's *Personal Exoskeleton* (right).

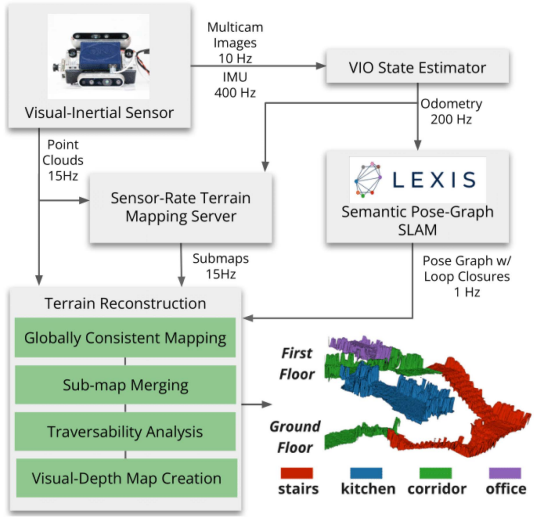


Fig. 2. Exosense scene understanding system. The inputs are RGB images and 3D point clouds from the multi-sensor unit. Different modules provide odometry and sensor-rate terrain maps, which are integrated into a semantic pose graph and processed with a terrain reconstruction module. The final scene representation (bottom right) contains terrain geometry, semantics, and traversability as well as visual localization information to aid long-term operation.

where we obtained 4 cm drift per meter traveled for the selected odometry algorithm.

- Extensive experiments of the Exosense's scene representation, particularly accuracy of the terrain reconstruction and traversable space estimates.
- A real-world demonstration of the Exosense integrated into the Wandercraft's *Personal Exoskeleton* for indoor localization tasks, demonstrating the potential for future long-term operation in home environments.

II. Related Work

We briefly review works on vision systems for wearable devices (Section II-A) and perception for self-balancing exoskeletons (Section II-B), which are relevant to Exosense.

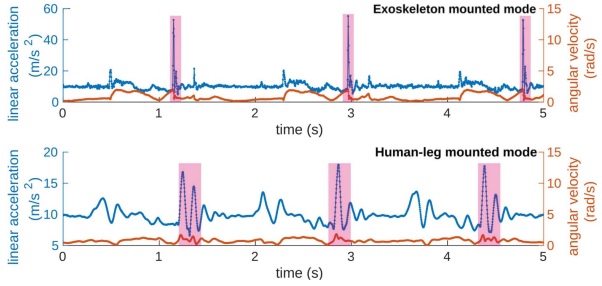


Fig. 3. Sample of linear acceleration and angular rotation rates measured by Exosense in exoskeleton (top) and human-leg-mounted (bottom) configurations. Both modes have a similar gait duration. The highlighted spikes (pink) occur during foot strikes.

A. Wearable Vision-Based Systems

Vision-based systems provide richer information about the users and their surroundings than proprioceptive sensing, drawing increasing interest in wearable robotics research [11]. Integrating computer vision into upper-limb wearable robots has been commonly used in rehabilitation applications to assist object manipulation tasks including determining object dimensions [12] and detecting user intention [13]. These solutions are integrated as external egocentric cameras (e.g., mounted on glasses [14]) or directly attached to a robot [15]. In addition to manipulation tasks, vision-based wearables have enabled independent navigation for visually-impaired people [16]. In industrial settings, similar solutions have provided assistance to alleviate joint stress and to protect users from work-related musculoskeletal disorders [17].

Lower-limb exoskeletons mainly use vision to detect relevant ground features for reliable locomotion. Ramanathan et al. [18] developed a vision-based perception system to enable exoskeletons to change the step size subject to detected obstacles. Follow-up work extended this approach to terrain recognition [19]. Tricomi et al. [20] used RGB images to identify types of terrain and adjust the walking controller of a hip exosuit. Karacan et al. [8] used a depth camera and an IMU mounted on a subject's waist to classify objects (e.g., ramps, staircases, obstacles), and

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on March 05,2025 at 12:44:38 UTC from IEEE Xplore. Restrictions apply.

to predict staircase height. A similar setup has been deployed on a lower-limb prosthesis to recognize features in the environment [21].

In general, the aforementioned vision-based approaches focus on assisting short-horizon low-level decision making, without preserving historical data. With Exosense we instead aim to build a map representation that can be reused for future operations in indoor settings.

B. Perception for Self-Balancing Exoskeletons

Existing self-balancing exoskeletons primarily rely on proprioceptive sensing for state estimation. Vigne et al. [22] incorporated multiple IMUs and robot joint encoder measurements to estimate position and velocity through a flexible kinematic model to account for deformations during exoskeleton walking. MOVIE [23] fused the same sensor measurements and takes a velocity-aided approach to estimate the robot's orientation with respect to gravity. Elnecave et al. [24] built on top of MOVIE to estimate the 6 DoF pose and velocity of the exoskeleton's body using an EKF.

Self-balancing exoskeletons also incorporate proprioceptive sensing into the control scheme. Tian et al. [25] used motor force sensors to estimate the center of mass deformation and physical parameters of the human operator during exoskeleton walking, which were then fed into a joint control framework to help the robot walk stably. Li et al. [26] used IMUs to estimate both robot and human center of mass; these estimates were integrated into a human-in-the-loop cooperative control scheme to adaptively adjust the motion controller to follow the user's intention.

In contrast to previous work that is tightly tailored to the exoskeleton platform, we developed an integrated sensing unit that relies on exteroceptive sensing only, hence being independent of the particular platform, user gait, or control framework.

III. System

The Exosense system overview is shown in Fig. 2. It is a scene understanding system which involves a leg-mounted vision-based sensing unit, and a navigation stack designed for highly dynamic walking motions. The system generates an environment representation encoding the terrain geometry, semantics, and traversability that can enable the continuous deployment of the exoskeleton for localization and navigation. The following sections describe the main components, from the hardware design to the navigation stack.

A. Multi-Sensor Setup

The Exosense hardware consists of a lower-limb-mounted wearable device, shown in Fig. 1. The sensing device includes a Sevensense Alphasense unit with three hardware-synchronized wide-angle global shutter cameras plus an inertial sensor, and two Realsense RGB-D cameras (D435i and D455). The Alphasense unit is used mainly for state estimation (odometry estimation and localization) due to its wide field-of-view (FoV). The Realsense cameras provide 3D sensing for terrain mapping.

The multi-sensor unit can be attached to either a human thigh or an exoskeleton. We developed the sensing system to only require exteroceptive sensing so as to disregard challenges related to leg deformation and bending. On the exoskeleton, the device is rigidly attached to the thigh to avoid occlusions with the rest of the user's body and to facilitate its future integration with the exoskeleton's walking controller.

Because the device can also be attached to a human leg, we could develop the scene understanding system without needing permanent access to an exoskeleton—enabling us to test algorithms that go beyond the current capabilities of the exoskeleton, such as multi-floor navigation. This design decision is supported by the similarities we observed in the walking motions of the human-leg-mounted and exoskeleton-mounted sequences, as illustrated in Fig. 3.

B. Visual-Inertial Odometry

To estimate ego-motion, we use a visual-inertial odometry system to provide high-frequency state estimation and to handle high rotation rates and jerk during the exoskeleton locomotion. We considered OpenVINS [27], VILENS-MC [28] and ORB-SLAM [29]. We evaluated their performance in custom sequences recorded with our multi-sensor unit to assess their performance and reliability under walking patterns, which is discussed in Section IV-Exp B. For our experiments, we chose OpenVINS due to its better balance of estimation accuracy and computational cost.

C. Semantic Pose-Graph SLAM

The odometry estimate serves as input to a visual SLAM system. We used LEXIS [30], as it provides a semantic pose graph representation that can be easily extended with other information sources, such as terrain maps. LEXIS constructs a pose graph representation with evenly spaced nodes. The nodes store corresponding RGB images, which are used for visual localization and loop closure detection. For semantics, LEXIS uses the CLIP visual-language model [31] to obtain visual embeddings, which are compared against text embeddings of a list of room classes (e.g., office, kitchen, corridor), providing potential room labels for each node in the graph.

The room labels associated to each node enable hierarchical place recognition by comparing the predicted room class of the current image to the keyframe labels in the graph. PnP [32] is used as a geometric verification step to propose loop closure candidates, which are jointly optimized in a factor graph to reduce the drift in the graph.

The output of LEXIS is a pose graph encoding odometry and loop closure connectivity with a pose-level room segmentation. In Exosense we extend this representation by adding terrain maps to each node, which can be refined by exploiting the room information. This enables us to produce an *elastic* globally-consistent terrain representation defined by the pose graph, as proposed in the Atlas framework [33].

D. Mapping and Reconstruction

To obtain the local terrain maps, we use the method by Fankhauser et al. [34], which integrates point clouds from both

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on March 05,2025 at 12:44:38 UTC from IEEE Xplore. Restrictions apply.

RGB-D cameras at 15 Hz, to generate a rolling local multi-layered 2.5D map [35] at 2 cm resolution around the Exosense's sensing unit location. We used this method as a server, providing local terrain maps to be attached to LEXIS' pose graph nodes on request.

While this representation enables a lightweight elastic terrain reconstruction, the individual submaps only represent a local region around the corresponding node, which might be suboptimal due to moving objects and partial visibility. Hence, we propose to exploit the semantic room information already stored in the pose graph to refine the terrain estimates, creating single *room-based* terrain maps.

For submaps within the same room, we fuse the height values of overlapping map cells using the median:

$$h_i^{\text{merged}} = \text{median}(\{h_i\}), \tag{1}$$

where h_i^{merged} is the fused height value for the i^{th} cell of terrain map, $\{h_i\}$ is the set of all the valid overlapping height values in cell i. While other submap fusion strategies could be used, here we exploit the room information already provided by LEXIS, which provides a semantic guidance.

E. Terrain Traversability Analysis

Following the room-based fusion step, we wish to estimate the traversability of each room's terrain map. This traversability estimate is computed on a cell basis, characterizing which areas of each room should be accessible by the exoskeleton in a navigation setting. To obtain it, we perform a geometric analysis of the local terrain tailored to the exoskeleton's gait specifications.

Technically, the local terrain analysis module determines a traversability score for a cell i, $t_i \in [0, 1]$, which characterizes how difficult it would be for the exoskeleton to step on the cell (specifically, 0 for untraversable and 1 for traversable). For this, we assume that the exoskeleton has a nominal maximum stride length s^* (size of step forward) and step height h^* (maximum height it can step on). For each cell i with height h_i, we select the neighboring cells j within a radius s^*, denoted by $\mathcal{C}_{s^*}$, and compute the maximum height difference $h_i^{\max}$ in the neighborhood:

$$h_i^{\max} = \max(|h_j - h_i|), j \in \mathcal{C}_{s^*}. \tag{2}$$

We then define the traversability score of a cell as the percentage difference of the maximum height difference $h_i^{\max}$ with respect to the nominal step height h^*:

$$t_i = 1 - \min\left(\frac{h_i^{\max}}{h^*}, 1\right). \tag{3}$$

This traversability score then represents a conservative cell estimate of how safe it would be for the exoskeleton to step into any other cell given this nominal step height. Fig. 8 illustrates how this compares to a geometric approach based on surface normals on a staircase.

F. Localization Within the Scene Representation

Exosense generates a scene representation that includes terrain geometry, semantics, and traversability. We extend it with images and depth maps obtained at each topological map node. This enables us to perform place recognition and metric localization in subsequent missions, enabling the reuse of previously built maps.

Our localization approach uses visual bags of words [36] to obtain place candidates, and PnP [32] to obtain a metric pose estimate from the RGB and depth images. We combine this with the odometry estimate to provide a continuous pose estimate between relocalizations, as well as providing an estimate when the exoskeleton visits an unmapped area. We demonstrate this in Section IV-Exp F.

IV. Experiments

We conducted several experiments to validate the Exosense's multi-sensor unit and navigation pipeline for indoor exoskeleton applications. We collected two datasets using the Exosense multi-sensor unit in two indoor environments:

- *Human* – Four sequences recorded with the Exosense device mounted on a human leg (Fig. 1, left): Sequences with 2-minute duration *H1, H2, H3* were captured using a Vicon motion capture system in a research lab, while *H4* was 10-minute long and recorded in a multi-floor office environment. The objective was to evaluate our design before testing on the exoskeleton, as well as testing *Exosense* mapping capabilities which go beyond the current exoskeleton locomotion capabilities.
- *Exo* – Two 7-minute sequences recorded with the Exosense device attached to the thigh of Wandercraft's *Personal Exoskeleton* (Fig. 1, right): *E1* was used to evaluate the terrain reconstruction, while *E2* was used to demonstrate the localization capabilities of Exosense. The exoskeleton was teleoperated while carrying a dummy during the recordings in a mixed office and lab environment with occasional passers-by. The objective of this dataset was to assess the mapped terrain quality in realistic conditions, as well as the potential of Exosense for indoor navigation.

All data was post-processed with a mid-range laptop, Intel i7 10750H @ 2.60 Hz 12 core laptop, Nvidia GTX 1650Ti GPU. All the algorithms are CPU-based except for the CLIP feature extractor in LEXIS.

Our first two experiments, Section IV-Exp A and Section IV–Exp B, aim to assess our hardware and odometry estimators decisions prior to the deployment of Exosense on the exoskeleton—hence they are demonstrated with the human-leg mounted mode. We additionally show the potential of the full pipeline to operate in multi-story environments in Section IV-Exp C. The last three experiments assess the reconstruction accuracy, traversability quality, and demonstrate a localization use case in the *Exo* sequences.

Exp A. Study of Wide FoV Multi-Camera Systems [Human]

Our first experiment tested the suitability of our multi-camera setup. We achieved this by comparing the performance of a

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on March 05,2025 at 12:44:38 UTC from IEEE Xplore. Restrictions apply.

TABLE I
EXP A - TRANSLATION AND ROTATION RPE AT 1 M, AVERAGED OVER FIVE RUNS UNDER DIFFERENT CAMERA CONFIGURATIONS

Relative Pose Error (RPE) – Translation [m] / Rotation [°]			
FoV (H × V)	No. Camera	Translation RMSE [m]	Rotation RMSE [°]
64° × 90°	2	Fail	Fail
64° × 90°	3	0.61	3.57
92.4° × 126°	2	0.34	**2.38**
92.4° × 126°	3	**0.11**	2.70

visual-inertial odometry system, for different numbers of cameras and different FoV in sequence *H1*. We used VILENS-MC [28], a fixed-lag optimization-based visual-inertial odometry algorithm designed in our group which works with multi-camera systems. The Exosense sensing unit was carried by a human as a wearable device on the person's thigh, while mimicking the exoskeleton walking pattern. Both wide and narrow field-of-view (FoV) images were recorded, and a Vicon motion capture system was used to provide ground truth poses. When evaluating the odometry performance, we report the Root Mean Square Error (RMSE) of the Relative Pose Error (RPE) as our main metric.

Table I reports the main results of this experiment. We observe that using a small FoV camera results in significant drift or even motion tracking failure due to the high accelerations and jerks present in the walking motion. Wider FoV cameras help to mitigate these effects, significantly reducing drift. Adding the lateral camera mitigates situations where no features are detected in front of the device. The lowest drift rates are achieved using both wide FoV cameras and the multi-camera setup with forward and lateral views. This configuration achieved reliable motion tracking, even in scenarios with significant viewpoint changes or occlusions under the jerky walking motion.

Exp B. Comparison of VI Odometry Algorithms [Human]

Next, we extended the evaluation of odometry estimators for Exosense to other open source algorithms, using the wide FoV and 3-camera configuration determined in the previous experiment. The objective was to assess the performance of other methods in these challenging walking conditions. We compared VILENS-MC to ORB-SLAM (optimization-based) [29] and OpenVINS (filtering-based) [27]. We must note that for ORB-SLAM we used the stereo-inertial configuration as it does not support multi-camera setups; we also disabled loop closure mechanisms for a fair comparison with the odometry systems. Further, OpenVINS processes each camera as a monocular input, while VILENS-MC treats the three cameras as a stereo pair and a monocular camera.

We used sequences *H2* and *H3* to test the performance of these systems in new conditions not considered in *H1*. In *H2*, the operator walked at a slow pace in a loop that included a small staircase. The sequence had a peak linear acceleration of 37.6 ms^{-2} and rotation rate of 4.4 rad/s. In *H3* we included periods of abrupt rotation change and occasional occlusions of the front stereo cameras. This sequence had peak acceleration and rotation rates of 55.1 ms^{-2} and 5.0 rad/s respectively. The algorithms were run five times on each sequence. RPE at 1 m and 5 m are presented in Table II.

TABLE II
EXP B - TRANSLATION AND ROTATION RPE (AT 1 M AND 5 M) FOR EACH ODOMETRY ALGORITHM, AVERAGED OVER FIVE RUNS

Relative Pose Error (RPE) – Translation [m] / Rotation [°]					
Dist	Method	*Seq. H2*		*Seq. H3*	
		Translation RMSE	Rotation RMSE	Translation RMSE	Rotation RMSE
1m	ORB-SLAM	0.08	**3.14**	0.19	**2.49**
	OpenVINS	**0.06**	3.45	0.15	2.55
	VILENS-MC	0.10	3.62	**0.14**	2.55
5m	ORB-SLAM	0.27	3.62	0.58	**3.56**
	OpenVINS	**0.20**	3.40	0.40	4.10
	VILENS-MC	0.26	**3.20**	**0.38**	4.25

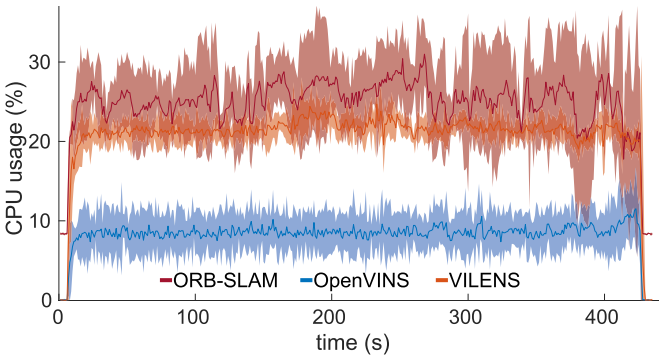


Fig. 4. Exp B – CPU usage over time for the evaluated odometry algorithms over five runs. The darker lines show the mean, while the shaded areas are the 95% confidence interval. OpenVINS is significantly more lightweight—using about half the computation.

Additionally, to provide further insights on the computational budget required by each method, we logged the CPU usage of the algorithms. These results are presented in Fig. 4.

Our odometry evaluation results show that the three tested VIO algorithms are robust and reliable even during walking patterns. This follows our design decision to use wide FoV cameras as discussed in Section IV-Exp A. ORB-SLAM experiences higher drift rates at times as it lacks multi-camera support, which we also noted in a previous paper [28]. While OpenVINS and VILENS-MC achieve comparable odometry accuracy (2 cm RPE at 5 m, i.e. 4 cm drift per meter traveled), OpenVINS uses less computation (i.e., under 10% CPU usage). As a result, we chose OpenVINS as the odometry source for Exosense for our next experiments. In future we envisage these algorithms being run on low power hardware such as an ASIC or FPGA chip.

Exp C. Multi-Story Mapping With Exosense [Human]

The last experiment in the human-leg mounted mode tested the full Exosense navigation pipeline for sequence *H4*, featuring a multi-story building. This sequence shows the potential of Exosense to build multi-floor representations that are beyond the current locomotion capabilities of self-balancing exoskeletons on multi-step staircases.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on March 05,2025 at 12:44:38 UTC from IEEE Xplore. Restrictions apply.

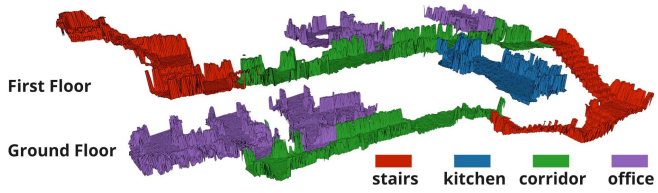


Fig. 5. Exp C – Multi-story mapping of sequence *H4* in the *Human* dataset. Exosense generated a globally consistent multi-floor terrain map. Each room is a single individual submap colored by its type.

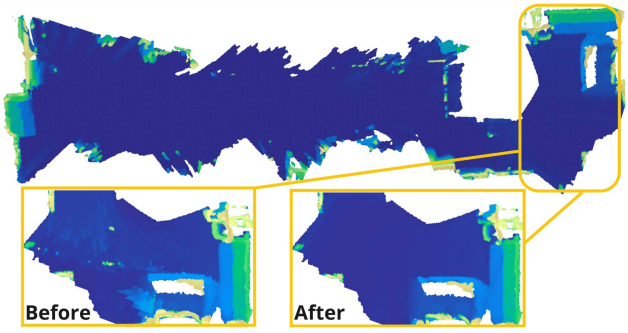


Fig. 6. Exp D – Qualitative mapping result after submap merging of the *Exo* sequence, colored by the elevation. Staircases and part of the ground areas are shown in detail both before and after applying submap merging (bottom). The median-based merging method removed outliers in the terrain submap while preserving the sharp features and edges of terrain geometry.

Fig. 5 demonstrates the terrain reconstruction of a multi-story office environment carrying the Exosense unit on a human leg. The system handles elevation maps in the multi-floor scenario and merges them by semantic room labels, which yields a smooth terrain map. We demonstrate the detailed view of multi-floor mapping in the attached multi-media material and present the quantitative reconstruction evaluation in Section IV-Exp D.

Exp D. Evaluation of Terrain Reconstruction Quality [Both]

We evaluated the Exosense terrain reconstruction quality in both the exoskeleton and human-leg mounted modes. For this experiment we focused on staircases present in sequences *H4* (*Human*) and *E1* (*Exo*), using millimeter-accurate reconstructions from tripod-based laser scanners, Leica RTC360 and Faro Focus 3D-X130, respectively. We ran Exosense with the same setup used in Section IV-Exp C.

Fig. 6 shows the reconstruction of the *Exo* sequence colored by elevation. We also show maps before and after applying the submap fusion strategy introduced in Section III-D. We observed improvements in the terrain flatness, and better crispness at the edges of the staircase steps. Furthermore, we note that outliers from odometry drift and dynamic objects, though not explicitly modeled in the individual submaps, are mitigated by our fusion strategy, resulting in a consistent reconstruction of the terrain.

Further, we performed a quantitative evaluation against the laser scans, by extracting key areas from the elevation map that the exoskeleton could traverse (e.g., staircases). We cropped these regions and converted the terrain maps into meshes to preserve the geometry of the terrain, and then sampled 10000 points per square meter from meshes to compute point-to-point distances to the ground truth scans.

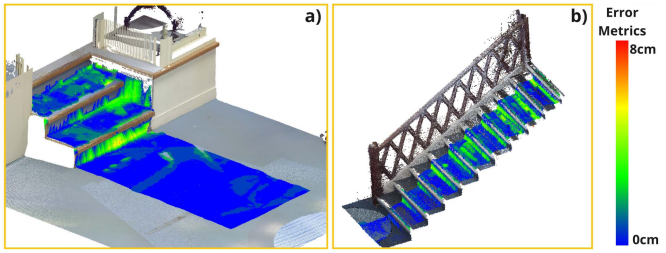


Fig. 7. Exp D – Maps of two areas of interest from the Exosense system in (**a**) exoskeleton mounted and (**b**) human-leg mounted modes. The mapping results are compared with ground truth laser scans and colored by the point-to-point distance. Errors in the near-vertical surfaces of the terrain map should be ignored.

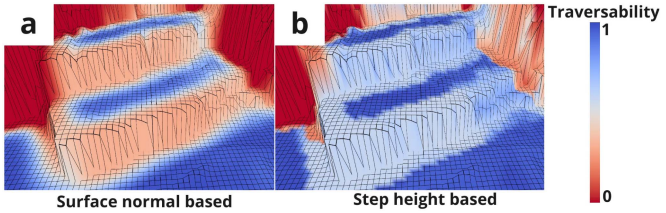


Fig. 8. Exp E – Comparison between traversability analysis obtained based on (**a**) terrain normals, and (**b**) our method based on the exoskeleton's step height. We observe that for the same traversability range, our method assigns a higher traversability score to the complete staircase compared to the normals-based method.

TABLE III
EXP D - TERRAIN RECONSTRUCTION QUALITY

Point-to-point Distance of Staircase Reconstruction				
dataset	res. [cm]	mean [cm]	max [cm]	90% [cm]
Human	2	1.36	8.45	2.84
Exo	2	0.80	6.64	1.85

We measured the point-to-point distance between our terrain reconstruction and the ground truth point cloud scan. We present the mean, max and 90th percentile error to quantify the mapping quality.

We present the results in Table III. The mapping results under both mounting modes showed similar accuracy, indicating the design choices made using the human-leg mounted mode can be successfully transferred to the exoskeleton-mounted mode. For the exoskeleton-mounted mode, Exosense produced an average error under 1 cm with 90th percentile error under 2 cm, indicating a sensible reconstruction quality for the use with the exoskeleton. Fig. 7 additionally shows the error distribution for staircases present in the sequences. The large errors mainly appear on the vertical areas, which is expected from an elevation-based terrain reconstruction method and the chosen evaluation procedure.

Exp E. Evaluation of Terrain Traversability [Exo]

Next, we assessed the traversability estimation result of the Exosense system. In Fig. 8, we present an example of per-cell traversability obtained from the mapping result in sequence *E1:*

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on March 05,2025 at 12:44:38 UTC from IEEE Xplore. Restrictions apply.

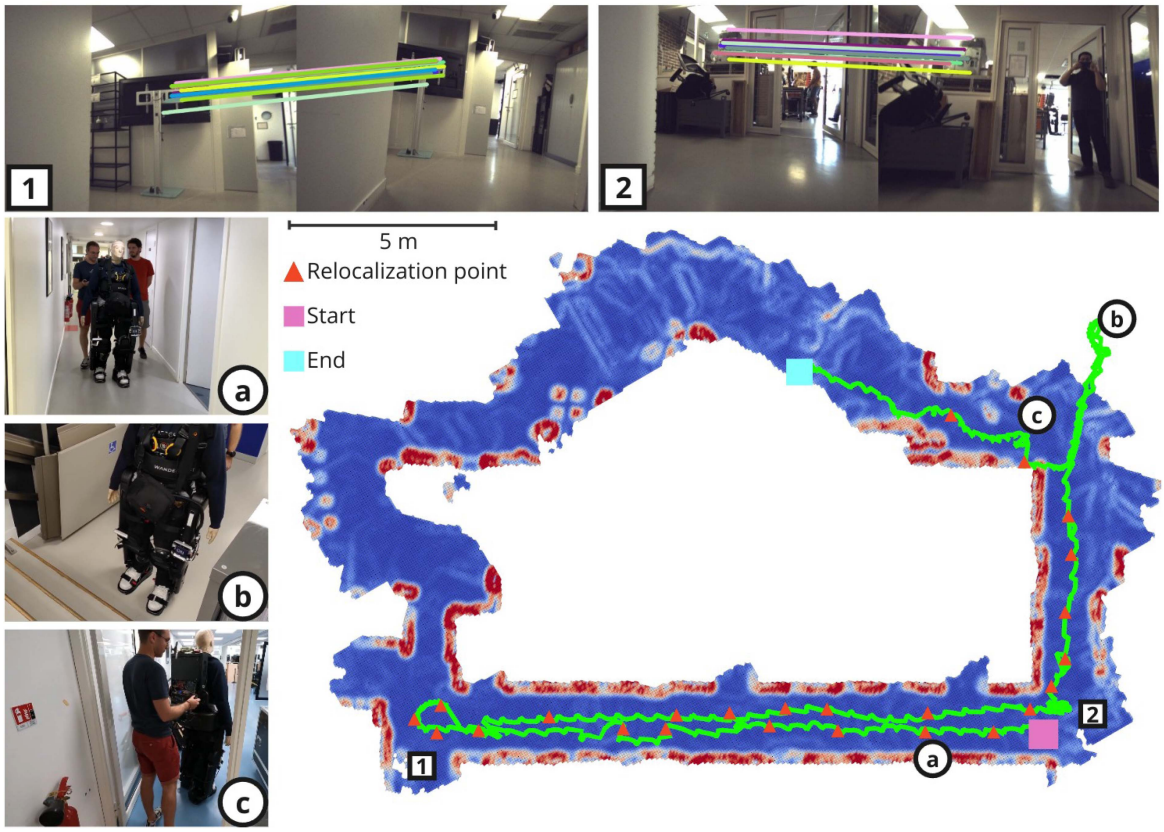


Fig. 9. Exp F – Exoskeleton-mounted localization demonstration within a prior map. The green line indicates the path estimated by *Exosense* using our navigation system in localization mode; the triangles denote a subset of areas where a relocalization fix was achieved. Left images show part of the testing area: ⓐ while crossing a long corridor, ⓑ in an unmapped region, and ⓒ passing through a narrow doorway. The top images show visual matches between the prior map (human-leg-mounted, left), and live exoskeleton stream (right).

On the left, we show a baseline based on surface normals [37], while the right is our proposed step height-based method. We set our method with a maximum stride length of 20 cm and nominal step height of 20 cm. We observed that it correctly assigned high traversability scores to the riser and treads of the staircase, which reflects the effective traversable areas of a walking system. In contrast, methods based on surface normals correctly determine walls to be untraversable, but may assign low traversability scores to the risers—which is undesired for navigation tasks.

To quantitatively show the benefits of our approach, we stored the per-cell traversability predictions and estimated elevation, and hand-labeled the traversable and untraversable areas. We then binarized the traversability scores of the normals-based method and ours with different thresholds. Then we evaluated the quality of the predictions as a classification problem, where the traversable areas were positive and the untraversable areas negative. We computed precision, recall, and F-score values over different thresholds to evaluate the traversability estimation accuracy. Our method obtained F-scores above 0.9 for different threshold values, with an optimal threshold of 0.5 (F-score value 0.93). In contrast, the normals-based method reported lower F-scores (below 0.87) for all threshold values and was more sensitive to changes in the optimal traversability threshold.

Exp F. Localization Demonstration [Exo]

Finally, we demonstrated the ability of the Exosense to visually relocalize within a prior map. To achieve this, we recorded an initial sequence using the leg mounted configuration, and then used the localization mode in the Exo-mounted configuration (sequence *E2*).

Fig. 9 shows the prior map built with Exosense in human-leg-mounted mode. The path followed by the exoskeleton in a subsequent experiment is shown in green. The exoskeleton was teleoperated to walk around 80 m, with our system obtaining about 60 visual relocalizations (about one every 0.8 m on average, shown as orange triangles). Our localization system was able to correct the estimate when the exoskeleton returns back along the corridor (ⓐ) and when passing through the narrow doorway (ⓒ). We observed that the odometry estimate was generally reliable when moving through unmapped regions (ⓑ), and the system was able to relocalize when returning to previously visited areas.

V. Conclusion

We introduced Exosense, a vision-based scene understanding system for self-balancing exoskeletons. Our system consists of a multi-sensor unit and a navigation stack, designed to be independent of the exoskeleton hardware and also usable as a

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on March 05,2025 at 12:44:38 UTC from IEEE Xplore. Restrictions apply.

wearable device. We investigated the hardware and dynamics of the problem, concluding that a visual-inertial unit with wide-angle cameras overcomes most of the challenges of the walking motion. We further introduced a mapping pipeline able to capture accurate terrain structure, semantics and traversability, as well as demonstrated how Exosense can relocalize in previously visited places. This provides input on the advantages of exteroceptive sensing for the eventual deployment of exoskeletons in indoor environments. In future work, we aim to extend the applicability of Exosense for long-term, multi-session localization and mapping applications under environment changes, explore its usage in outdoor scenarios, and further integrate the system into the exoskeleton's navigation and control stack additionally exploiting its proprioceptive sensing.

ACKNOWLEDGMENT

The authors thank Wayne Tubby and Matthew Graham for hardware design support.

REFERENCES

[1] T. Gurriet et al., "Towards restoring locomotion for paraplegics: Realizing dynamically stable walking on exoskeletons," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 2804–2811.

[2] V. Huynh et al., "Versatile dynamic motion generation framework: Demonstration with a crutch-less exoskeleton on real-life obstacles at the Cybathlon 2020 with a complete paraplegic person," *Front. Robot. AI.*, vol. 8, 2021, Art. no. 723780.

[3] D. Tian et al., "Self-balancing exoskeleton robots designed to facilitate multiple rehabilitation training movements," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 32, pp. 293–303, 2024.

[4] T. Yan, M. Cempini, C. M. Oddo, and N. Vitiello, "Review of assistive strategies in powered lower-limb orthoses and exoskeletons," *Robot. Auton. Syst.*, vol. 64, pp. 120–136, 2015.

[5] M. Kim et al., "Visual guidance can help with the use of a robotic exoskeleton during human walking," *Sci. Rep.*, vol. 12, 2022, Art. no. 3881.

[6] A. H. Al-dabbagh and R. Ronsse, "A review of terrain detection systems for applications in locomotion assistance," *Robot. Auton. Syst.*, vol. 133, 2020, Art. no. 103628.

[7] A. G. Kurbis, B. Laschowski, and A. Mihailidis, "Stair recognition for robotic exoskeleton control using computer vision and deep learning," in *Proc. IEEE Int. Conf. Rehabil. Robot.*, 2022, pp. 1–6.

[8] K. Karacan, J. T. Meyer, H. I. Bozma, R. Gassert, and E. Samur, "An environment recognition and parameterization system for shared-control of a powered lower-limb exoskeleton," in *Proc. IEEE/RAS EMBS Int. Conf. Biomed. Robot. Biomechatron.*, 2020, pp. 623–628.

[9] A. H. A. Al-Dabbagh and R. Ronsse, "Depth vision-based terrain detection algorithm during human locomotion," *IEEE Trans. Med. Robot. Bionics.*, vol. 4, no. 4, pp. 1010–1021, Nov. 2022.

[10] D.-X. Liu, J. Xu, C. Chen, X. Long, D. Tao, and X. Wu, "Vision-assisted autonomous lower-limb exoskeleton robot," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 51, no. 6, pp. 3759–3770, Jun. 2021.

[11] L. Gionfrida, D. Kim, D. Scaramuzza, D. Farina, and R. D. Howe, "Wearable robots for the real world need vision," *Sci. Robot.*, vol. 9, no. 90, 2024, Art. no. eadj8812.

[12] C. Hu, D. Kim, S. Luo, and L. Gionfrida, "PointGrasp: Point cloud-based grasping for tendon-driven wearable robotic applications," 2024, *arXiv:2403.12631*.

[13] E. Rho et al., "Multiple hand posture rehabilitation system using vision-based intention detection and soft-robotic glove," *IEEE Trans. Ind. Inform.*, vol. 20, no. 4, pp. 6499–6509, Apr. 2024.

[14] D. Kim et al., "Eyes are faster than hands: A soft wearable robot learns user intention from the egocentric view," *Sci. Robot.*, vol. 4, no. 26, 2019, Art. no. eaav2949.

[15] J. Kuhn, J. Ringwald, M. Schappler, L. Johannsmeier, and S. Haddadin, "Towards semi-autonomous and soft-robotics enabled upper-limb exoprosthetics: First concepts and robot-based emulation prototype," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 9180–9186.

[16] H.-C. Wang, R. K. Katzschmann, S. Teng, B. Araki, L. Giarré, and D. Rus, "Enabling independent navigation for visually impaired people through a wearable vision-based feedback system," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 6533–6540.

[17] F. Missiroli et al., "Integrating computer vision in exosuits for adaptive support and reduced muscle strain in industrial environments," *IEEE Robot. Autom. Lett.*, vol. 9, no. 1, pp. 859–866, Jan. 2024.

[18] M. Ramanathan et al., "Visual environment perception for obstacle detection and crossing of lower-limb exoskeletons," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 12267–12274.

[19] M. Ramanathan, L. Luo, M. J. Foo, C. H. Chiam, W.-Y. Yau, and W. T. Ang, "Heuristic vision based terrain recognition for lower limb exoskeletons," in *Proc. Int. Conv. Rehabil. Eng. Assistive Technol.*, 2024, pp. 25–28.

[20] E. Tricomi et al., "Environment-based assistance modulation for a hip exosuit via computer vision," *IEEE Robot. Autom. Lett.*, vol. 8, no. 5, pp. 2550–2557, May 2023.

[21] K. Zhang et al., "Environmental features recognition for lower limb prostheses toward predictive walking," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 3, pp. 465–476, Mar. 2019.

[22] M. Vigne, A. El Khoury, F. Di Meglio, and N. Petit, "State estimation for a legged robot with multiple flexibilities using IMUs: A kinematic approach," *IEEE Robot. Autom. Lett.*, vol. 5, no. 1, pp. 195–202, Jan. 2020.

[23] M. Vigne, A. E. Khoury, M. Pétriaux, F. D. Meglio, and N. Petit, "MOVIE: A velocity-aided IMU attitude estimator for observing and controlling multiple deformations on legged robots," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3969–3976, Apr. 2022.

[24] F. E. Xavier, G. Burger, M. Pétriaux, J.-E. Deschaud, and F. Goulette, "Multi-IMU proprioceptive state estimator for humanoid robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*, 2023, pp. 10880–10887.

[25] D. Tian et al., "Dual-loop control framework of a self-balancing lower-limb exoskeleton for assisted walking," *IEEE Trans. Instrum. Meas.*, vol. 73, 2024, Art. no. 7506511.

[26] Z. Li, T. Zhang, P. Huang, and G. Li, "Human-in-the-loop cooperative control of a walking exoskeleton for following time-variable human intention," *IEEE Trans. Cybern.*, vol. 54, no. 4, pp. 2142–2154, Apr. 2024.

[27] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 4666–4672.

[28] L. Zhang, D. Wisth, M. Camurri, and M. Fallon, "Balancing the budget: Feature selection and tracking for multi-camera visual-inertial odometry," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1182–1189, Apr. 2022.

[29] C. Campos, R. Elvira, J. J. Gomez, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[30] C. Kassab, M. Mattamala, L. Zhang, and M. Fallon, "Language-EXtended indoor SLAM (LEXIS): A versatile system for real-time visual scene understanding," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2024, pp. 15988–1599.

[31] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[32] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[33] M. Bosse, P. M. Newman, J. J. Leonard, and S. J. Teller, "Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework," *Int. J. Robot. Res.*, vol. 23, no. 12, pp. 1113–1139, 2004.

[34] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3019–3026, Oct. 2018.

[35] P. Fankhauser and M. Hutter, "A universal grid map library: Implementation and use case for rough terrain navigation," in *ROS– The Complete Reference (Volume 1)*. Berlin, Germany: Springer, 2016, pp. 99–120.

[36] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.

[37] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter, "Navigation planning for legged robots in challenging terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 1184–1189.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on March 05,2025 at 12:44:38 UTC from IEEE Xplore. Restrictions apply.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Exosense: A Vision-Based Scene Understanding System for Exoskeletons
Publication Status	Published
Publication Details	J. Wang et al., "Exosense: A Vision-Based Scene Understanding System for Exoskeletons," in IEEE Robotics and Automation Letters, vol. 10, no. 4, pp. 3510-3517, April 2025, doi: 10.1109/LRA.2025.3543971.

Student Confirmation

Student Name:	Jianeng Wang		
Contribution to the Paper	My contributions to the paper were: • Developed the core idea behind the paper with co-authors • Designed the system hardware with co-authors • Implemented the system software • Performed the experiments with co-authors • Wrote the paper with co-authors		
Signature	Jianeng Wang	Date	2025/04/03

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:		
Supervisor comments This work was wholly the work of Jianeng Wang (aside from collaborative contributions from other co-authors). He wrote all the relevant code and carried out the main experiments. It's a key output from his PhD.		
Signature	Date	2025/06/23

This completed form should be included in the thesis, at the end of the relevant chapter.

5.2 Additional Remarks

5.2.1 Evolution of Exosense Hardware Design

While the main manuscript focuses on the mature Exosense system, the final hardware configuration is the result of a rigorous, iterative design process.This section therefore provides an in-depth discussion on how the Exosense system evolves.

The project's origin lies in a collaboration with Wandercraft, manufacturers of the Eve self-balancing exoskeleton (Fig. 5.1). The Eve, while capable of independent walking, is equipped only with proprioceptive sensors (e.g., joint encoders, IMU) and thus lacks any awareness of its external environment. Our primary goal was to develop a vision system to provide this crucial exteroceptive sensing, specifically for mapping and understanding the terrain where the exoskeleton is operating. The initial design constraint was the sensor placement. In order to ensure a clear, non-occluded view in the direction of travel, the exoskeleton's leg was identified as the only viable mounting location. To facilitate rapid, parallel development, a key strategy was to prototype on a human leg, which allowed us to validate the system in realistic walking scenarios.

The first prototype consisted of a single RGB-D camera (Fig. 5.2a). While lightweight and simple, this design failed immediately in practice. During a normal walking gait, the leg swing produces a rapid jerky motion. The limited Field of View (FoV) of the single camera meant that it could not retain a sufficient number of stable feature points in its view, leading to frequent and unrecoverable tracking failures.

To solve the tracking problem, the design was revised to decouple the tasks of state estimation and depth mapping. We introduced three wide-angle, global-shutter cameras dedicated to motion tracking, leveraging their wide FoV and resistance to motion blur to create a robust visual-inertial odometry frontend. In this version (Fig. 5.2b), the single RGB-D camera was oriented vertically to minimize the unit's width. While this design successfully solved the motion tracking problem, it introduced a new failure mode: inadequate terrain coverage. The vertically-oriented

Figure 5.1: The Wandercraft Eve self-balancing exoskeleton. **a)** The base platform and **b)** the system with a dummy. The configuration illustrates that the lower leg is the only suitable mounting location for Exosense to maintain a non-occluded, forward-facing viewpoint when the robot is in use.

depth camera could not capture a wide enough ground plane, resulting in sparse and incomplete maps, which were insufficient for safe navigation.

The final design (Fig. 5.2c), as presented in this thesis, directly addresses the shortcomings of the previous iterations. It retains the three wide-angle tracking cameras, as their overlapping fields of view provide the near-180° horizontal coverage necessary to ensure tracking stability throughout the entire gait cycle. To solve the mapping problem, a dual RGB-D camera setup was introduced. One downward-facing camera is dedicated to capturing high-resolution, near-field terrain for immediate foot placement planning. The forward-facing camera is responsible for mapping medium-range obstacles and the broader environment. This dual-depth configuration resolves the inherent trade-off faced by a single sensor, simultaneously providing the near-field detail and far-field context required for robust and safe navigation. The final design is therefore not arbitrary, but a purpose-built solution

a) Initial Design **b)** Intermediate Design **c)** Final Design

Figure 5.2: The iterative hardware evolution of the Exosense sensor suite. **(a)** The initial prototype concept, illustrated here with a gradienter as the placeholder representing the intended placement of a single RGB-D camera. This design proved insufficient for robust tracking due to its limited field of view during dynamic leg motion. **(b)** The second version solved the tracking issue by adding three wide-angle cameras, but its single, vertically-oriented depth sensor provided inadequate terrain mapping. **(c)** The final design achieves comprehensive scene understanding by using two RGB-D cameras oriented to map both near-field ground terrain and forward-facing obstacles.

derived directly from the lessons learnt from the iterative prototyping.

5.2.2 Deployment in Dynamic Environments

The experiments presented in this manuscript focused on testing Exosense in static indoor environments. However it is important that the system is robust to people or other things moving around in the operating environment. A supporting video demonstrating that Exosense is robust to dynamics such as a pedestrian passing by can be seen in: **link to video**. When pedestrians pass by the robot, they are detected by the depth camera causing anomalous artifacts to appear in the elevation map. However, when pedestrians leave the scene, the local elevation mapping module continues to process new measurements and can correctly reconstruct the floor. The cells that previously contained dynamic points are updated with new measurements from the static background. In this way, those artifacts can be removed from the online map, leaving only the static environment structure.

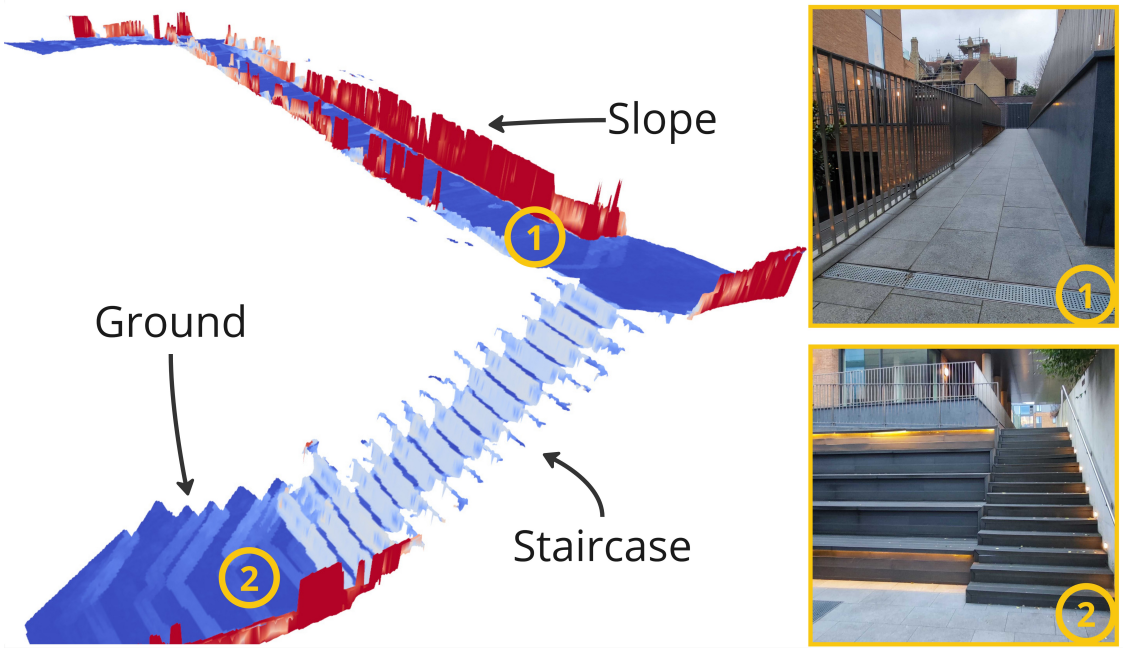


Figure 5.3: Traversability estimation result of the merged terrain map produced by Exosense in an outdoor scene containing both an accessibility slope (①) and a staircase (②).

5.2.3 Deployment in Outdoor Environments

Additional sequences were recorded with the Exosense operating outdoors including on stairs and slopes. These experiments were carried out with the human leg-mounted configuration so as to test the Exosense's adaptability in such scenarios. A video clip is attached to demonstrate the Exosense functioning outdoors (**link to video**). Fig. 5.3 shows the traversability estimation results of the complete terrain map at the end of the experiment.

As demonstrated by this outdoor sequence, *Exosense* can reconstruct the terrain to generate a globally consistent multi-floor elevation map. The traversability estimates on the ground and staircase area are consistent with the results reported in the main manuscript of this chapter. Walkable terrain is correctly classified as traversable while the vertical walls are classified as non-traversable. For the accessibility ramp (marked as ① in the figure), the step-height based traversability estimation method can also correctly determine the traversability of the ground, the flat steps, the walls, and the fencing on the sides.

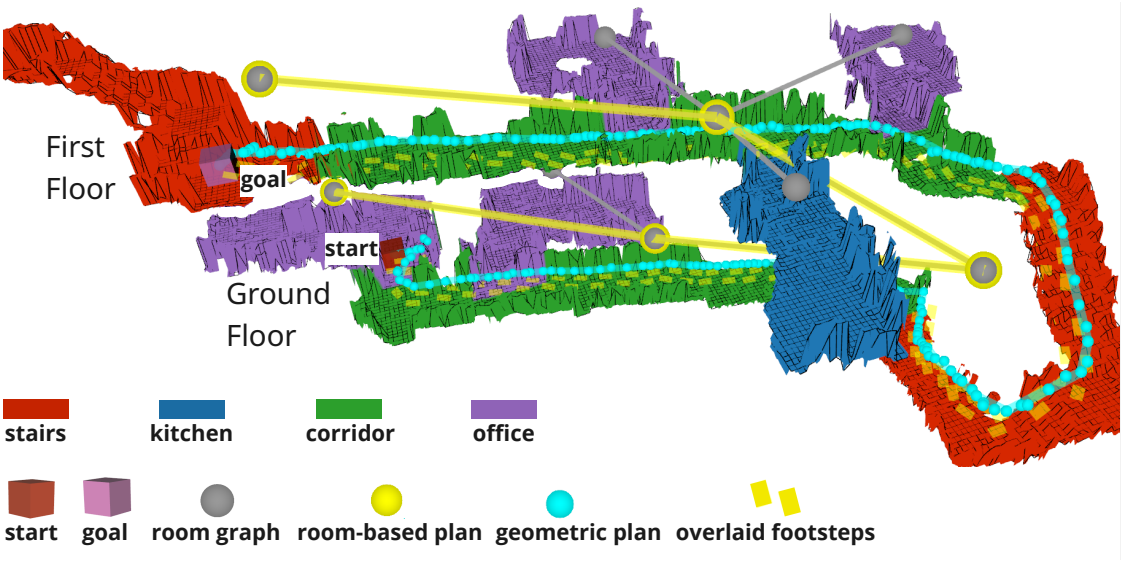


Figure 5.4: Demonstration of hierarchical motion planning on the *Exosense* map. The optimal path starts from a ground-level office, up a staircase, and ends in a first-floor closet.

5.2.4 Path Planning Demonstration

The map representation produced by the *Exosense* system can be used for high-level navigation tasks. Fig. 5.4 demonstrates the Exosense's output used to perform hierarchical, room-based path planning.

The method is based on two stages: room-level planning and geometric motion planning. In the room-level planning stage, a *room graph* from LEXIS' output, which encodes the connectivity of the rooms, is built. Dijkstra's algorithm [45] can then be used to find the sequence of rooms that the exoskeleton needs to travel between to reach the goal location. For geometric planning, a probabilistic roadmap (PRM) [116] is computed for each room, and the user can query the PRM to find a valid geometric path. This demonstration suggests potential applications that *Exosense* map representation offers for exoskeleton navigation.

5.3 Discussion

This chapter presented *Exosense*, a vision-based scene understanding system designed for self-balancing exoskeletons. The hardware features a versatile, leg-mounted multi-sensor unit that offers wide-angle vision and depth sensing to support

state estimation, terrain mapping, and localization. On the software side, the system builds reusable local maps embedded with terrain geometry, room-level semantics, and traversability information, enabling enhanced decision-making and long-term autonomy in indoor environments. The system has been extensively validated in both human-leg- and exoskeleton-mounted scenarios, demonstrating a key step towards robust, long-term exoskeleton operation.

It is important to note that *Exosense* is currently a proof-of-concept system aimed at introducing exteroceptive sensing and environmental representation into the exoskeleton domain. Several promising directions remain for further development.

For wearable applications, both energy efficiency and processing speed are critical. Reducing the system's computational overhead through hardware acceleration or algorithmic simplification is a key objective. For example, the visual-inertial odometry (VIO) module could be replaced with compact, embedded alternatives such as the ASIC-based implementation within the Intel RealSense T265 [84]. Likewise, SLAM and elevation mapping components could be accelerated on mobile GPUs, such as the NVIDIA Jetson platform, where similar systems have demonstrated success [263, 161]. These enhancements would facilitate the deployment of Exosense on-board a self-balancing exoskeleton in real-world settings.

At present, the Exosense system operates independently of the exoskeleton's control stack, and its output is not yet integrated into the locomotion controller. A future extension of this work involves tighter integration, allowing terrain maps and semantic labels produced by Exosense to directly influence locomotion decisions, such as automatic switching between walking modes when encountering flat ground, stairs, or obstacles. Such integration would also enable the study of how different user-induced gait dynamics affect mapping quality. While the visual-inertial estimator is designed to be robust to motion, quantifying the impact of varying knee trajectories on map quality would be valuable for ensuring performance across a diverse range of users.

In addition to single-session use, enabling long-term scene understanding is essential for daily-life deployment. This requires the ability to explore new areas,

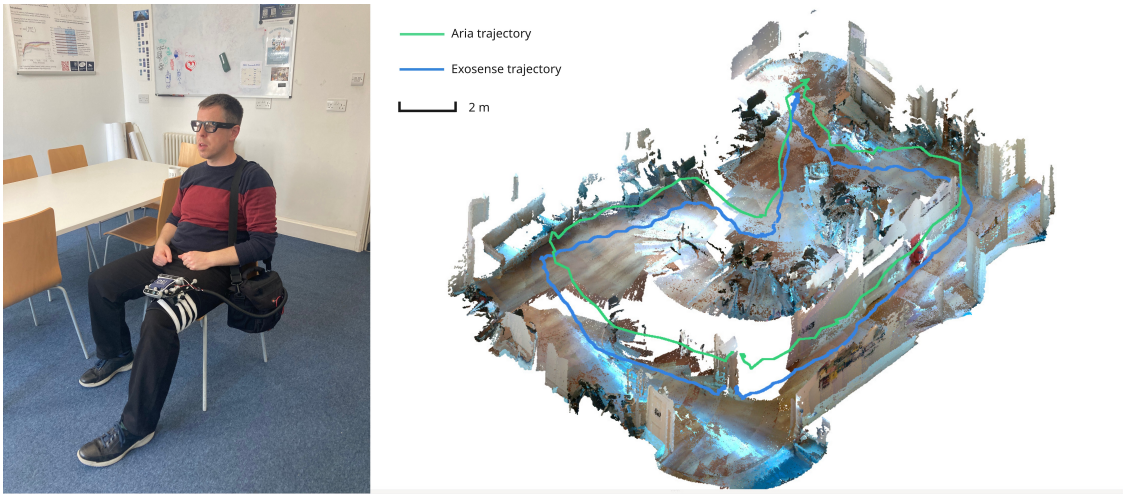


Figure 5.5: A demonstration of jointly using Exosense and Aria glasses. **Left:** Devices setup on a human operator (Left). **Right:** Combined plot of Exosense representation and Aria trajectory. The outputs from the two systems are registered together using AprilTag [186, 255] given the known locations of tags in some arbitrary map frame. The registration method using AprilTag is detailed in Appendix B. Still extensive research would be required for real-time joint SLAM from two different perspectives to obtain their extrinsics, hence building a unified scene representation for scene understanding tasks.

merge multi-session maps, and identify environmental changes. These capabilities are addressed in part in Chapter 6, where multi-session mapping and reusability of maps are explored.

The placement of the Exosense hardware unit was also a critical design consideration, and its modularity invites discussion of alternative configurations. The current above-the-knee, leg-mounted position was deliberately chosen to align the sensor's perspective with the direction of locomotion while minimising self-occlusion from the user's body. This placement also respects the 0.3 m minimum operating range of the downward-facing depth camera (i.e., Realsense D435i), where a lower mounting point would conflict. While adding a sensor to each leg was considered, the wide field-of-view of the current hardware proved sufficient to cover the terrain for the exoskeleton's immediate next steps, making a second unit redundant. Other sensor modalities, such as LiDAR, were ruled out due to prohibitive power consumption for a wearable platform.

Alternative placements could offer complementary information. For example, a

rear-mounted camera could aid in backward motion estimation [150] but cannot serve as the primary sensor for forward locomotion.

Fusing data from multiple perspectives offers great potential for the further development of Exosense. As demonstrated in Fig. 5.5, combining the low-level terrain view from the leg-mounted Exosense with an egocentric, eye-level perspective from a device like the Meta Aria Glasses [54] would create a unified scene representation, significantly enhancing the cooperative interaction between the user, the exoskeleton, and their environment.

Another future direction, which links the work presented in Chapter 4 to the Exosense system, involves augmenting the system with event cameras and adopting the continuous-time estimation framework. The high temporal resolution of an event camera would be particularly advantageous for tracking the rapid, jerky motions during exoskeleton locomotion, which also mitigate the motion blur that can challenge conventional frame-based cameras. Furthermore, the continuous-time trajectory estimator is naturally suited to fuse the asynchronous, high-rate data from an event camera with the synchronous measurements from the multi-camera system. This integration would not only enhance the robustness of the state estimation but also increase its temporal precision, leading to a more accurate and responsive scene understanding system.

Beyond its specific application to exoskeletons, Exosense system offers broader insights for the field of wearable robotic sensing. A key recommendation is that sensor placement must be task-driven. While head-mounting provides a viewpoint ideal for high-level scene understanding, the leg-mounted configuration of Exosense is specifically tailored for the more dynamic challenge of immediate, near-field terrain analysis for safe navigation. For any such dynamic application, a multi-camera, wide-angle system provides essential robustness against motion and occlusion. While vision sensors offer an excellent balance of data richness and low power, alternative sensor modalities like LiDAR can provide more accurate and wider-coverage mapping, assuming the platform's size, weight, and power constraints can accommodate them.

The principles derived from developing Exosense therefore contribute to a general framework for designing perception systems across a range of wearable devices.

6
Multi-session Visual Mapping

Contents

Self-balancing exoskeletons are a promising mobility solution for individuals with lower-limb disabilities, enabling independent walking without external aids like crutches. Equipped with a vision-based navigation system, Exosense, the semantically annotated elevation maps that encode both the geometric layout and terrain traversability of the surrounding environment can be generated for the safe navigation of exoskeletons. This helps to reduce the need for manual operator intervention and supports more autonomous, context-aware decision-making during locomotion.

While the Exosense system provides scene understanding and terrain analysis to support single session safe navigation of the exoskeleton, the long-term deployment of an exoskeleton needs to respond to the dynamics in real-world scenarios, where progressive environment changes are present. To do so, the ability to not only

perform robust single session scene understanding, but also retain, update, and reuse multi-session knowledge over time is required.

This chapter presents *LT-Exosense*, a change-aware, multi-session mapping system designed to address these long-term challenges when deploying an exoskeleton for daily use. LT-Exosense extends the single-session mapping capabilities of Exosense by incrementally merging multiple exploration sessions, detecting environmental changes, and maintaining an up-to-date global map. By integrating this persistent spatial memory with adaptive path planning, LT-Exosense enables the exoskeleton to reroute around newly introduced obstacles and to discover optimal paths as conditions change, demonstrating the long-term autonomy needed for intelligent assistive mobility systems.

6.1 LT-Exosense: A Vision-centric Multi-session Mapping System for Lifelong Safe Navigation of Exoskeletons

The following article was written in a manuscript style which we plan to submit to IEEE Robotics and Automation Letters (RA-L).

LT-Exosense: A Vision-centric Multi-session Mapping System for Lifelong Safe Navigation of Exoskeletons

Authors

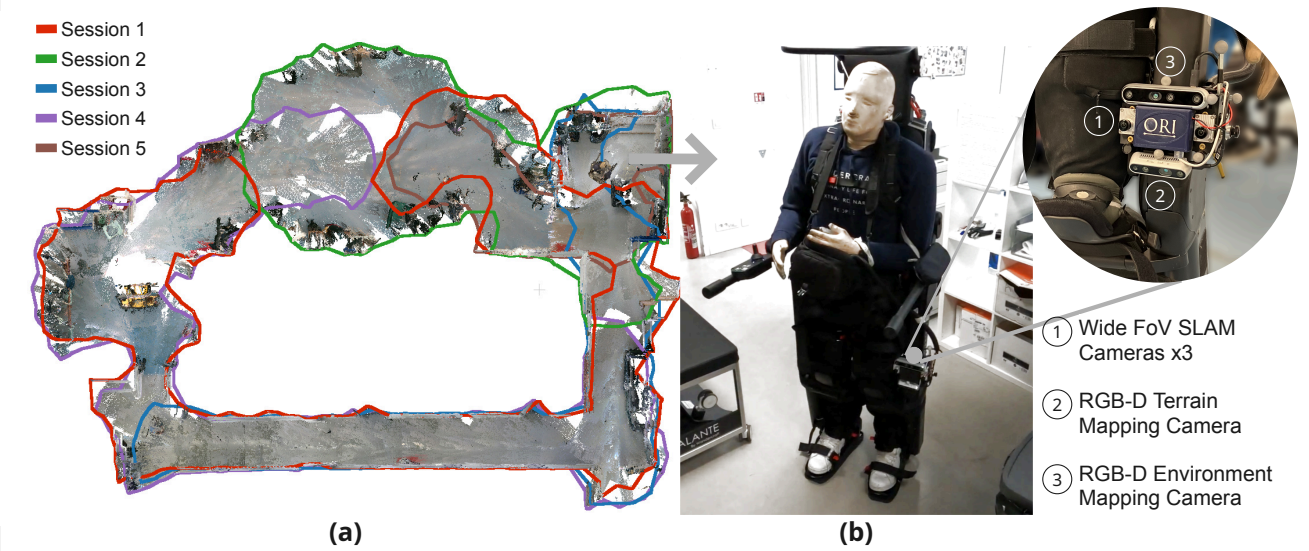


Fig. 1: LT-Exosense is capable of merging multiple sessions generated by a previous work, Exosense, a vision-centric scene understanding system with its sensing unit (**Top-Right**) integrated into a self-balancing exoskeleton (**b**). The merged map (**a**) contains five sessions with colored contour indicating each session's coverage. Such a merged map can be further converted into a navigation map, allowing obstacle-free planning across multiple sessions.

***Abstract*— Self-balancing exoskeletons offer a promising mobility solution for individuals with lower-limb disabilities. For reliable long-term operation, these exoskeletons require a perception system that is effective in changing environments. In this work, we introduce LT-Exosense, a vision-centric, multi-session mapping system designed to support long-term (semi)autonomous navigation for exoskeleton users. LT-Exosense extends single-session mapping capabilities by incrementally fusing spatial knowledge across multiple sessions, detecting environmental changes, and updating a persistent global map. This representation enables intelligent path planning, which can adapt to newly observed obstacles and can recover previous routes when obstructions are removed. We validate LT-Exosense through several real-world experiments, demonstrating a scalable multi-session map that achieves an average point-to-point error below 5 cm compared to ground-truth laser scans, while supporting adaptive path planning in dynamically changing indoor environments.**

***Index Terms*— Multi-session Mapping, Wearable Robotics, Prosthetics and Exoskeletons, RGB-D Perception, Mapping**

I. Introduction

Self-balancing exoskeletons provide a transformative solution enabling mobility-impaired individuals to walk independently, offering an alternative to wheelchairs and crutches. While extensive efforts have been made in human-compliant hardware design and control strategies [1], [2], their deployment remains largely confined to structured clinical and therapeutic contexts [2]. This limits their benefits on patients' daily lives and long-term rehabilitation.

Moving beyond the clinical setting, the daily real-world usage of exoskeletons offers the immense potential of increased independence for the users while also reducing the occurrence of secondary health conditions, hence improving the quality of life for patients with lower-limb disabilities [3], [4]. However, achieving this goal requires not only advancements to hardware and control algorithms but also effective perception to enable reliable navigation and adaptability to dynamic environments. For this, a persistent, long-term global map of the environment is essential. The map should enable the exoskeleton to plan long-distance paths between different rooms or floors, while also relying on local planning for short-term trajectory following and obstacle handling. This type of semi-autonomy is especially important for paretic patients recovering from stroke to enhance their mobility, who may experience both lower-limb impairment and limited upper-limb agility [5]. These individuals may be unable to react quickly to environmental changes or control the exoskeleton manually. Equipping the system with the ability to detect and respond to environmental changes across multiple traversals is therefore critical for their safety concerns.

A recent system, Exosense [6], introduced a vision-based scene understanding method for exoskeletons that could generate rich, home-scale scene representations. The sensing unit is designed to be rigidly attached to the upper leg of the exoskeleton or human, so as to avoid potential occlusions from the user's body. This setup however introduces a jerky walking motion pattern, making accurate motion estimation difficult. While Exosense could enable exoskeleton localization and navigation, it could only operate in a single-session setting and had no capacity to accumulate environmental knowledge over extended periods of time or to respond to spatial changes across mapping sessions.

To improve upon these limitations, we present *LT-Exosense*, a change-aware, multi-session mapping system tailored for the long-term deployment of self-balancing exoskeletons in real-world environments. LT-Exosense can integrate spatial data from multiple exploration sessions to incrementally build a persistent map. It can detect and track environment changes, and update the global map to reflect the latest state of the world. This capability facilitates lifelong, intelligent navigation by allowing the exoskeleton to reuse maps of previously explored areas. It can also adapt to new conditions, and plan safe paths to familiar destinations. LT-Exosense has the potential to improve the experience of an exoskeleton user by providing intuitive mobility assistance.

The key contributions of our work are:

- LT-Exosense, a multi-session mapping system for self-balancing exoskeletons that captures terrain traversability information as well as identifies changes and obstacles in realistic dynamic environments.
- We demonstrate a real-world adaptive path planning pipeline that can re-route around detected obstacles using the updated multi-session map.
- We conduct several experiments to evaluate LT-Exosense's ability to identify object-level change and to carry out multi-session reconstruction in an evolving office environment.

II. Related Work

A. Multi-session Visual SLAM

Traditional SLAM algorithms estimate a robot's trajectory while simultaneously constructing a map of the environment, which makes them a key building block for autonomous systems. However, most conventional SLAM systems assume a single, continuous exploration session. In contrast, *multi-session SLAM* [7] extends this framework to support long-term and large-scale operations by incrementally fusing the outputs of multiple SLAM sessions—whether performed by a single robot across different time intervals or by multiple robots collaboratively. This capability enables persistent mapping, robust localization when revisiting a place, and resilience to environmental changes over time.

In multi-session visual SLAM, the system must recognize previously visited places across different sessions using visual inputs. This is inherently challenging due to changes in lighting conditions, viewpoint, and scene appearance. Labbe and Michaud [8] present a multi-session visual SLAM framework centered around re-localization, with each individual session built using RTAB-Map [9]. Their work evaluates various visual descriptors for illumination-invariant place recognition and loop closure. Experimental results indicate that learning-based feature detectors and matchers (e.g., SuperPoint [10] and SuperGlue [11]) offer improved robustness to appearance changes, albeit at the cost of increased computation and memory. To mitigate this, the framework incorporates a graph reduction strategy [12] to conserve resources while preserving localization accuracy.

Dedicated multi-session mapping systems like *maplab* [13] provide a tightly integrated pipeline for vision-based SLAM. It uses ROVIO [14] to construct individual sessions, saving pose graphs, keyframes, image features, and associated resources for inter-session place recognition, merging, and reconstruction. The updated system, *maplab 2.0* [15], expands support to heterogeneous sensor modalities and robot platforms, becoming agnostic to odometry sources. It also supports storing non-visual data (e.g., LiDAR scans, GPS), enabling more versatile graph optimization constraints for tasks like multi-agent mapping and semantic mapping. This system has been successfully deployed in DARPA Subterranean Challenge [16] to support collaborative mapping and navigation of aerial and legged robots. While maplab itself does not target exoskeletons, which can be viewed as a class of legged robotic systems, its design motivates the type of adaptability we seek in the LT-Exosense system for multi-session exoskeleton mapping.

LT-Exosense adapts multi-session SLAM techniques for assistive mobility, focusing on long-term usability for self-balancing exoskeletons rather than general-purpose mapping. Our system achieves reliable map fusion across multiple sessions. It also incorporates change detection in order to support navigation in evolving environments, thereby bridging the gap between SLAM research and real-world deployment on an exoskeleton system.

B. Change Detection

As mapping matures, there is an increasing demand for long-term autonomy in dynamic environments for which the ability to detect changes in the scene over time is essential. Change detection enables robots to adapt their behavior in response to environmental variations and it is widely used in applications such as environment monitoring [17], infrastructure inspection [18], and disaster response [19]. These changes may range from highly dynamic (e.g., pedestrians and vehicles) to semi-static alterations that evolve over longer periods.

Changes in a scene can be determined via geometric analysis of map representations. Grid-based structures such as elevation maps [20] and OctoMap [21] support ray-tracing techniques that update occupancy based on sensor ray traversal. By continuously integrating sensor measurements, the representation can adapt to changes, but without explicitly modeling the change. Although accurate, these methods are computationally intensive due to the need to process every

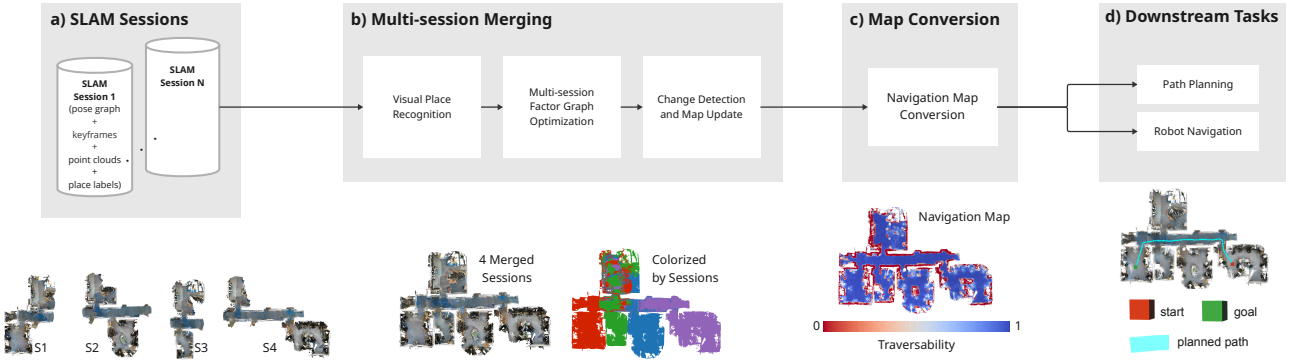


Fig. 2: Overview of the LT-Exosense system. Multiple SLAM sessions with keyframe images and point cloud submaps (**a**) are registered into one common reference frame and reflect the latest environment changes for safe navigation of exoskeletons (**b**). The merged map can be further converted into elevation maps with traversability estimated (**c**), which allows obstacle-free walkable path to be planned on top (**d**).

cell along each ray. Real-time deployment often requires hardware acceleration, such as GPUs [22].

Visibility reasoning simplifies the change detection problem by checking whether a point visible in one scan remains visible from another viewpoint [23]. While efficient, such methods are sensitive to incidence angle ambiguity—especially on ground surfaces—leading to misclassifications [24]. To mitigate this, visibility is often encoded as an auxiliary feature in downstream classifiers [25].

Volumetric maps such as Signed Distance Fields (SDFs) and occupancy grids allow online change detection by modeling free space. Systems like Dynablox [26] and DUFOMap [27] detect changes when new sensor measurements violate prior free-space assumptions. For inter-session analysis, approaches like LiSTA [28] and BeautyMap [29] align volumetric maps and perform voxel-level differencing to detect environmental changes. These approaches typically assume that the compared maps are spatially complete and densely observed, and primarily target LiDAR sensors, which have a wide field-of-view coverage.

LT-Exosense adapts a volumetric change detection method, LiSTA [28], for incremental mapping scenarios with RGB-D sensing, in the context of exoskeleton applications. Furthermore, it handles the issue of non-overlapping areas between multiple sessions and maintains a single lifelong map to reflect the latest environment state to support practical downstream tasks such as adaptive path planning. This unlocks the potential for exoskeletons to navigate dynamic environments safely and efficiently over time.

III. System

The overall architecture of the LT-Exosense system is presented in Fig. 2. Individual SLAM sessions are generated using data from tracking cameras and an RGB-D camera (Sec. III-A). Maps from multiple sessions are aligned and merged to form a unified map that reflects the latest state of the environment (Sec. III-B). The unified map is then converted into an elevation map-based representation encoding the terrain geometry, semantics, and traversability (Sec. III-C), which supports downstream navigation tasks for the exoskeleton (Sec. III-D).

A. Single-session Map Creation

We adapt the session creation pipeline from the *Exosense* scene understanding system [6] to generate vision-centric SLAM sessions for later processing. Exosense uses a multi-camera setup to estimate the robot's state, mapping its surroundings and analyzing the terrain to generate globally consistent elevation maps that integrate both semantic and terrain traversability information. However, we use point clouds as the basic submap representation, as they can be easily converted into OctoMaps [21] for change detection, and elevation maps [20] for terrain-aware navigation.

LT-Exosense takes individual SLAM sessions as input. Each session consists of a pose graph made up of vertices and edges, along with associated resources linked to each vertex. A vertex represents the $SE(3)$ pose represented in the session map frame, $\mathbf{T}_{\mathrm{m,b}}$, where m is the fixed map frame and b is the robot base of the current session. The edges of the graph come from either relative odometry (consecutive vertices), or loop closures (non-consecutive, when the robot revisits the same place). To each vertex we associated resources such as a stereo image pair, a point cloud submap and a semantic place label describing the vertex, such as the type of the room. These resources are later used to merge multiple sessions and to perform change detection to create a lifelong map.

B. Multi-session Merging

1) Visual Place Recognition: For every new SLAM session, we perform visual place recognition using the images associated with a vertex by computing a global descriptor for each query image in the base pose b_q and matching it against existing sessions in the database to find the corresponding base pose b_p. The global descriptors consist of visual bag-of-words descriptors from ORB [30] features and are matched using DBoW [31]. We use these descriptors for both intra- and inter-session similar image search.

2) Multi-session Factor Graph Optimization: For each matched image pair at base frame b_p in one session, we use SIFT [32] to find local feature matches to the query image at base frame b_q in another session, and then estimate the relative transformation using the Perspective-n-Point (PnP) method in a RANSAC scheme [33], $\hat{\mathbf{T}}_{\mathrm{b}_p,\mathrm{b}_q}$.

The visual place recognition module returns multiple matched candidates for each query image. Among those candidates, the match with the highest number of inliers is selected, and the corresponding relative pose is added as an edge constraint to the factor graph. If multiple prior sessions exist, a query image from the new session may match images across multiple sessions. This introduces inter-session constraints that link the new session to multiple existing sessions, promoting better global alignment and consistency in the merged map. The cost function considering the full inter-session matched set $\mathcal{M}$ is written as

$$\mathcal{J}_{\text{inter}} = \sum_{(b_p,b_q)\in\mathcal{M}} \left\| \ln\left(\hat{\mathbf{T}}^{-1}_{\mathrm{b}_p,\mathrm{b}_q} \cdot \left(\mathbf{T}^{-1}_{\mathrm{m},\mathrm{b}_p} \cdot \mathbf{T}_{\mathrm{m},\mathrm{b}_q}\right)\right)^{\vee} \right\|^2_{\Sigma}, \quad (1)$$

where each pair of b_p and b_q form an inter-session matched poses and Σ is the covariance matrix associated with this relative pose estimate.

Once all vertex-associated images are processed, we select the map frame of the first session to anchor all the remainder sessions, and then build the full merged graph by connecting pose graphs from individual sessions via the inter-session edges. The merged graph is then optimized using the Levenberg-Marquardt algorithm [34] with a Cauchy loss function [35] to produce a globally consistent trajectory across sessions.

To manage memory efficiency, each session is stored locally in the file system. During multi-session merging, only the pose graphs are initially loaded into memory. The data-intensive resources of the graphs (e.g., images, point clouds) are only accessed on demand during relevant operations. This effectively restricts the peak memory usage.

3) Change Detection for Latest Map Update: As the robot incrementally explores its environment, the environment may experience change. To maintain an up-to-date representation while preserving unchanged regions, we adopt the volumetric differencing approach presented in LiSTA [28], which incrementally updates the merged map as each new session is integrated.

Each merged session, comprising optimized poses and point cloud submaps, is converted into an octree representation by OctoMap [21] that efficiently partitions the space into occupied and free voxels. We then define a prior map and its corresponding octree $\mathcal{O}_{\mathrm{p}}$, representing the current global map state prior to merging a new session. For the octree of a new session, $\mathcal{O}_{\mathrm{c}}$, we perform a differencing operation between $\mathcal{O}_{\mathrm{p}}$ and $\mathcal{O}_{\mathrm{c}}$ to identify spatial change. This results in a removed octree $\mathcal{O}_{\mathrm{r}}$, containing occupied voxels present in $\mathcal{O}_{\mathrm{p}}$ but absent in $\mathcal{O}_{\mathrm{c}}$, and an added octree $\mathcal{O}_{\mathrm{a}}$, with newly occupied voxels in $\mathcal{O}_{\mathrm{c}}$ that were previously free in $\mathcal{O}_{\mathrm{p}}$. Additionally, we compute the change-free prior octree, $\tilde{\mathcal{O}}_{\mathrm{p}}$, by subtracting the removed nodes from the prior octree to isolate the unchanged structure:

$$\tilde{\mathcal{O}}_{\mathrm{p}} = \mathcal{O}_{\mathrm{p}} - \mathcal{O}_{\mathrm{r}} = \mathcal{O}_{\mathrm{p}} - (\mathcal{O}_{\mathrm{p}} \ominus \mathcal{O}_{\mathrm{c}}), \quad (2)$$

where $\mathcal{A}-\mathcal{B}$ denotes node deletion, and $\mathcal{A}\ominus\mathcal{B}$ denotes octree differencing. These operations return a set of voxels in the overlapping region with these different occupancy states.

The updated octree is produced by combining it with the current session's octree:

$$\mathcal{O}_{\mathrm{l}} = \tilde{\mathcal{O}}_{\mathrm{p}} + \mathcal{O}_{\mathrm{c}}, \quad (3)$$

where $\mathcal{A}+\mathcal{B}$ merges two octrees in the same frame, with occupied nodes thereby overwriting free nodes.

This volumetric differencing strategy not only captures meaningful object-level changes but also removes misaligned or inconsistent point cloud data from prior sessions. As a result, the final merged map remains geometrically coherent and suitable for downstream navigation.

C. Navigation Map Conversion

1) Point cloud map to Elevation map: With the latest map updated through multi-session merging and change detection, we proceed to convert it to elevation maps for robot navigation. This involves generating elevation maps that encode terrain geometry and traversability for downstream planning.

We begin by clustering spatially adjacent vertices in the merged pose graph that share the same place label. For each cluster, we compute the 3D bounding volume of all associated point cloud submaps. This bounding volume is then used to crop the corresponding region from the latest point cloud map, resulting in environment-specific submaps of the scene.

An elevation map is a 2.5D representation, where each cell encodes the height of the terrain. However, overhanging structures such as ceilings may corrupt the map by introducing spurious height values that do not correspond to walkable terrain. To mitigate this effect, we introduce a coarse-to-fine dangling points removal scheme. For each environment-specific point cloud submap, we partition the space into a 2D grid aligned with the x-y plane. Within each grid cell, we cluster nearby points based on their heights and keep only the lowest cluster. This process is repeated over several iterations using progressively finer grid resolutions.

After removing dangling points, the environment-specific point clouds are converted into elevation maps using the method by Jelavic et al. [36]. Each point cloud is projected onto a 2D grid at a predefined resolution, and the height of each cell is computed as the mean z-value of all points falling within it.

2) Traversability Analysis: For each cell i in the elevation map, we define a traversability score, $t_i \in [0,1]$, representing how difficult it would be for the exoskeleton to step onto it, where $t_i = 1$ indicates fully traversable and $t_i = 0$ indicates untraversable. To compute the terrain traversability, we use the same approach in [6], by first selecting the neighborhood $\mathcal{C}_i$ of cell i as the set of all cells within a nominal maximum stride length s^* of the robot. We then compute the maximum elevation difference within this neighborhood as

$$h_i^{\max} = \max\left(|h_j - h_i|\right), \quad j \in \mathcal{C}_i. \quad (4)$$

Given the maximum height height h^* the exoskeleton can step on, the traversability score of a cell is

$$t_i = 1 - \min\left(\frac{h_i^{\max}}{h^*}, 1\right). \tag{5}$$

This score serves as a conservative estimate of how safely the exoskeleton can navigate from the current cell to its neighbors, given its locomotion capabilities.

D. Path Planning

The resultant elevation maps are merged to form a unified representation of the environment. A global probabilistic roadmap (PRM) [37] is then computed on top of this merged map for geometric motion planning. The PRM is built by randomly sampling a set of nodes representing valid robot configurations across the traversable regions of the merged elevation map. Nodes are connected if a path between them is determined to be collision-free. This process results in a graph that approximates the connectivity of the free space for the robot's safe navigation.

Once the start and goal poses are set on the map, the PRM is queried to connect them to nearby nodes in the existing graph. The resulting sequence of nodes constitutes a feasible geometric path from the start to the goal.

IV. Experiments

We conducted a series of experiments to evaluate the performance of LT-Exosense in four areas: trajectory alignment accuracy (Sec. IV-Exp A), change detection performance (Sec. IV-Exp B), multi-session mapping quality (Sec. IV-Exp C), and its applicability to exoskeleton navigation tasks (Sec. IV-Exp D).

We use the EuRoC dataset [38] to assess multi-session trajectory alignment. We also used a custom-collected dataset that includes multiple sessions with varying environmental conditions, recorded with a multi-camera device from Exosense [6] mounted on a person's or exoskeleton's leg.

We denote each dataset by D_a^d, where d indicates the day the dataset was collected and a indicates the area. Sequences from the same day contain no environment changes, while those recorded on different days include object-level changes.

(H) Human. This dataset includes four sequences recorded with the Exosense sensing unit mounted on a human thigh. Two same-day sequences, $H_{a_1}^{d_1}$ and $H_{a_2}^{d_1}$, were captured sequentially without any environmental change and are spatio-temporally contiguous, enabling processing as a single SLAM session. Sequences $H_{a_1}^{d_2}$ and $H_{a_2}^{d_2}$, were recorded on the same areas on a different day with object-level scene changes. This dataset is used for both change detection and multi-session mapping evaluation.

(E) Exo. This dataset consists of five sequences, $E_{a_{i=1\cdots5}}^{d_1}$, collected with the sensing unit mounted on a self-balancing exoskeleton (Fig. 1**b**) navigating a mixed office and lab environment. Each session covers a different portion of the space, with overlapping regions between sessions. This dataset evaluates LT-Exosense in a real-world exoskeleton deployment scenario.

TABLE I: Comparison of multi-session trajectory alignment accuracy between ov_maplab and LT-Exosense in terms of the root mean squard error (RMSE) of the absolute trajectory error (APE) and relative pose error (RPE).

Multi-session Trajectory Alignment Accuracy								
Dataset	Seq.	ov_maplab			LT-Exosense			Length
		ATE	RPE (1 m)	No. Poses	ATE	RPE (1 m)	No. Poses	(m)
V1	V1_01	0.063	0.086	2774	0.056	0.107	126	58.6
	V1_02	0.057	0.034	1598	0.043	0.049	78	75.9
	V1_03	0.06	0.04	1988	0.08	0.053	97	79
V2	V2_01	0.059	0.033	2170	0.04	0.044	94	36.5
	V2_02	0.045	0.024	2234	0.053	0.03	112	83.2
	V2_03	0.103	0.038	1766	0.086	0.048	108	86.1

Ground-truth point cloud maps are provided for all sequences. For the *Human* dataset, ground-truth is obtained using a millimeter-accurate Leica BLK360 terrestrial LiDAR scanner. For the *Exo* dataset, we use a LiDAR-SLAM system [39]. Note that minor background activity occurred during *Exo* recordings, so individual session observations may not perfectly align with the fused ground-truth map.

For all experiments, individual SLAM sessions are generated using an implementation of the Exosense pipeline [6], which employs OpenVINS [40] for visual-inertial odometry and LEXIS [41] to build a pose graph. Point cloud submaps and images are associated with graph vertices and used for subsequent multi-session merging and evaluation. All processing was performed offline on a mid-range laptop (Intel i7-10750H @ 2.60GHz, 12-core CPU, NVIDIA GTX 1650Ti GPU). All components are CPU-based, except for intrasession visual place recognition, which uses a learning-based model on GPU.

Exp A. Multi-session Trajectory Alignment Accuracy

To evaluate the multi-session trajectory alignment accuracy of LT-Exosense, we compare it against *maplab* [13] using a consistent odometry frontend, OpenVINS [40] (referred to as *ov_maplab*). Both systems are tested on the EuRoC dataset, where multiple sequences are aligned to a common frame and compared against the ground-truth trajectory. We report the Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) of the aligned trajectories in Tab. I.

Overall, LT-Exosense demonstrates competitive alignment performance, achieving lower ATE but slightly higher RPE compared to maplab on most sequences. This indicates that LT-Exosense maintains strong global consistency across merged sessions, benefiting from graph-based optimization. However, since it relies on sparse pose graphs produced by external SLAM systems, which omits high-frequency odometry between consecutive keyframes. The local consistency is therefore slightly degraded relative to maplab, which optimizes over high-frequency odometry poses.

Exp B. Change Detection Performance [***Human***]

We evaluated LT-Exosense's object-level change detection performance on the **Human** dataset by merging pairs of sessions that cover the same area but were recorded at different times. The output includes both added and removed

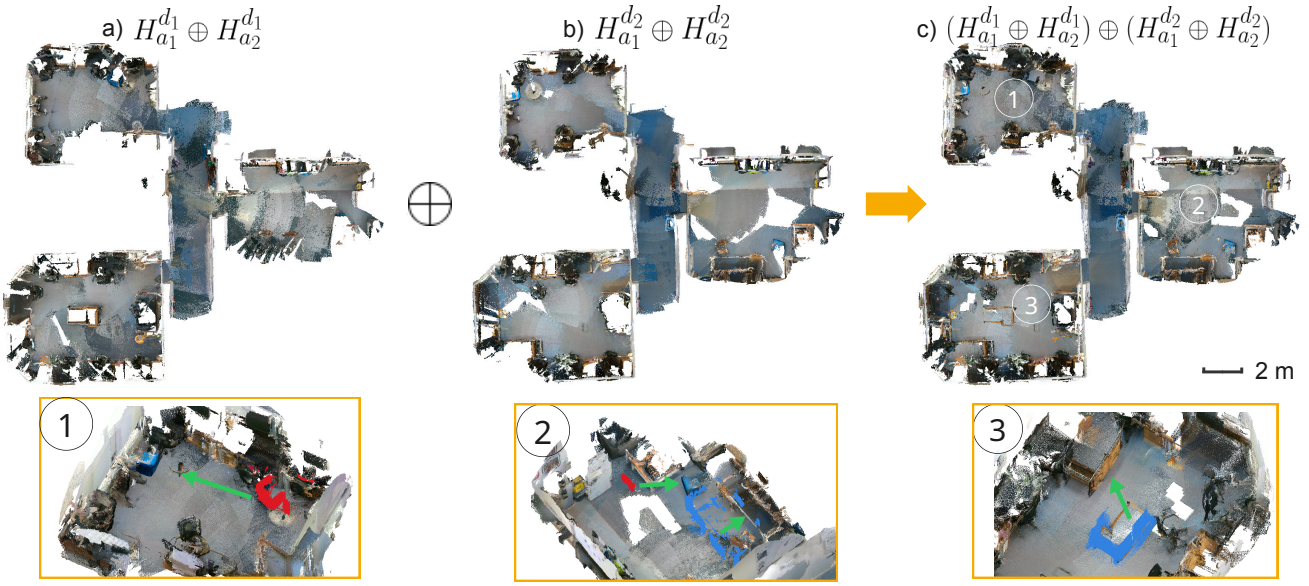


Fig. 3: Multi-session mapping and change detection results sequences ${}^{s1}H_a$ and ${}^{s2}H_b$. Subfigures (a) and (b) show merged maps from sequences recorded on two different days where no environmental changes have occurred within each day. Subfigure (c) shows the result of merging the maps from (a) and (b), where inter-day object-level changes are detected and highlighted in red and blue in the zoomed-in views on the right with green arrows indicating the before and after changes. $\oplus$ here indicates sessions merging followed by change detection and map update.

point clouds, computed using an octree with 5 cm resolution. For ground truth, we manually annotated the changed regions on the corresponding ground truth scans, and aligned the LT-Exosense outputs accordingly. Since the change detection module also removes noisy or misaligned points arising from session merging errors, we restrict the evaluation to areas near true environment changes.

We evaluated the change detection performance as a classification problem. We defined *true positive* (TP) and *false positive* (FP) as the detected changes that are close to the ground truth changed and static points, respectively, while the *false negative* (FN) and *true negative* (TN) are detected static points that are close to the ground truth changed and static points. We used the same 5 cm threshold to associate predicted and ground truth changes and report standard classification metrics—precision, recall and F-score values (Tab. II). Additionally, we compute the Chamfer Distance between the detected and ground truth changes to quantify the discrepancy of the two point clouds.

Our quantitative results show that LT-Exosense achieves high precision, indicating detected changes have few outliers and align well with actual environmental modifications. However, recall is lower, primarily due to a limited sensor field-of-view and incomplete coverage during traversals, which leads to missed detections when ground-truth areas are unobserved. Despite this, the average Chamfer Distance remains low at 4 cm, suggesting that the spatial reconstruction of detected changes is accurate.

*Exp C. Multi-session Mapping Quality [**Human** & **Exo**]*

We next evaluated the reconstruction quality of LT-Exosense using all custom sequences under two conditions:

TABLE II: Change detection performance metrics. For every two sessions of the same area but recorded at different times, we merge them and perform change detection in both directions.

Change Dectection Evaluation				
Comp.	Precision	Recall	F-score	Chamfer Dist. [m]
$H_{a1}^{d_1} \rightarrow H_{a1}^{d_2}$	88.9 %	41.9 %	57.0 %	0.042
$H_{a1}^{d_2} \rightarrow H_{a1}^{d_1}$	88.4 %	48.4 %	62.5 %	0.034
$H_{a2}^{d_1} \rightarrow H_{a2}^{d_2}$	90.9 %	53.4 %	67.3 %	0.043
$H_{a2}^{d_2} \rightarrow H_{a2}^{d_1}$	80.1 %	35.7 %	49.4 %	0.04

1) merging sessions with no environmental changes, and 2) merging sessions that include changes.

For sessions recorded on the same day without changes, we ran the LT-Exosense pipeline (with change detection disabled) to merge them and produce multi-session maps, as illustrated in Fig. 3. Additionally, for the **Human** dataset, sequences recorded on the same days can be concatenated together and processed as a single SLAM session (denoted as $H_{\text{SLAM}}^{d_1}$ and $H_{\text{SLAM}}^{d_2}$)

For sessions that contain inter-session changes, we applied the full LT-Exosense pipeline, including map merging and change detection, to generate updated maps. The quality of the final merged output was evaluated against the ground-truth scans from the newly added sessions (Fig. 3).

We quantified reconstruction quality using point-to-point distances between the merged map and ground truth, with results summarized in Tab. III. In the absence of scene changes, LT-Exosense achieves mapping accuracy comparable to a single-session SLAM pipeline, validating its ability to incrementally build consistent maps even under non-contiguous exploration.

TABLE III: Multi-session Mapping Quality. Using the point-to-point distance between the estimated map against the ground truth scan, we compute the mean, median, max and 90$^{\text{th}}$ percentile error to quantify the multi-session mapping quality. We use $\oplus$ operator to denote multi-session merging operation.

Point-to-point Distance of Multi-session Mapping				
Seq.	mean [m]	median [m]	max [m]	90% [m]
$H^{d_1}_{\text{SLAM}}$	0.0244	0.0166	0.264	0.0496
$H^{d_1}_{a1} \oplus H^{d_1}_{a2}$	0.0281	0.0163	0.325	0.0627
$H^{d_2}_{\text{SLAM}}$	0.031	0.019	0.47	0.067
$H^{d_2}_{a1} \oplus H^{d_2}_{a2}$	0.031	0.021	0.46	0.068
$(H^{d_1}_{a1} \oplus H^{d_1}_{a2}) \oplus (H^{d_2}_{a1} \oplus H^{d_2}_{a2})$	0.032	0.022	0.75	0.069
$E^{d_1}_{a1} \oplus E^{d_1}_{a2} \oplus E^{d_1}_{a3} \oplus E^{d_1}_{a4} \oplus E^{d_1}_{a5}$	0.046	0.027	0.873	0.099

In scenarios involving environmental changes, LT-Exosense shows higher maximum point-to-point errors. In the *Exo* dataset, this is primarily due to background activity during recording, which caused the fused ground truth map to diverge from the environment captured in individual sessions. In the *Human* dataset, higher error arises when geometry from earlier sessions becomes occluded in later traversals. This happens due to the change in the new session occluding those points. Since these outdated geometries are not explicitly removed unless observed again, they may persist in occluded areas. However, these residual elements typically have minimal impact on downstream navigation, as they are not visible or reachable during the latest traversal.

Since the change-aware merging pipeline maintains comparable reconstruction accuracy to merging sessions without change detection, this demonstrates that LT-Exosense preserves mapping quality even during dynamic updates.

Exp D. Path Planning Demonstration

To qualitatively demonstrate the ability of LT-Exosense's path planning module to adapt to environmental changes, we conducted an experiment in a representative indoor environment (Fig. 4a). To ensure realistic collision modeling during planning, we approximated the physical size of a walking exoskeleton (or human operator) using a bounding box of $0.5 \times 0.5 \times 1.8\text{m}^3$.

We designed three mapping sessions, each capturing different regions and environmental states of the same floor:

Session 1. It begins with partial exploration of a meeting room, proceeds to the start of a corridor, traverses through the corridor and enters an office. For the path planning experiment, a path planned from the start in the corridor to the goal in the office is drawn (Fig. 4a).

Session 2. It starts in the office and covers part of the meeting room, which completes the meeting room mapping when it is merged to Session 1. It then traverses the corridor in reverse toward the corridor's start, where an obstacle has been introduced in the corridor. Using the same start and goal, the planner changes its original plan and reroutes a new feasible path: from the corridor's start, detouring through the meeting after merging the newly discovered area, and then entering the office via the rear corridor connection (Fig. 4b).

Session 3. It remaps the corridor area with the obstacles removed. When Session 3 is merged into the existing map, LT-Exosense correctly identifies the updated changes in the environment and automatically recovers the shorter path for the same start and goal as in Session 1 (Fig. 4c).

This experiment highlights the incremental map building, map updating with change detection, and adaptive path planning capabilities of LT-Exosense. By fusing disjoint exploration sessions, the system forms a coherent, evolving spatial memory. It adapts navigation strategies based on current conditions, choosing longer, change-aware detours when necessary and reverting to optimal paths when obstructions are cleared. Such capabilities are critical for deploying exoskeletons in dynamic real-world settings, where persistent spatial understanding and reactivity to environmental change ensure user safety and autonomy.

V. Conclusions

We presented *LT-Exosense*, a change-aware, multi-session mapping system for the long-term deployment of self-balancing exoskeletons in evolving environments. Through experiments, we demonstrated its ability to accurately detect object-level changes, maintain high-quality multi-session maps, and support adaptive path planning in dynamic environments. These results position LT-Exosense as a practical system that helps assistive exoskeletons achieve robust, long-term autonomy. In future work, we will explore tighter integration of the system with navigation modules in the exoskeleton and its extended deployments in home, rehabilitation, and public environments.

References

[1] E. Tricomi, G. Piccolo, F. Russo, *et al.*, "Leveraging geometric modeling-based computer vision for context aware control in a hip exosuit," *IEEE Trans. Robotics*, vol. 41, pp. 3462–3479, 2025.

[2] D. Tian, W. Li, J. Li, *et al.*, "Self-balancing exoskeleton robots designed to facilitate multiple rehabilitation training movements," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 32, pp. 293–303, 2024.

[3] L. E. Miller, A. K. Zimmermann, and W. G. Herbert, "Clinical effectiveness and safety of powered exoskeleton-assisted walking in patients with spinal cord injury: Systematic review with meta-analysis," *Med. Dev. Evid. Res.*, pp. 455–466, 2016.

[4] I. J. van Nes, R. B. van Dijsseldonk, F. H. van Herpen, *et al.*, "Improvement of quality of life after 2-month exoskeleton training in patients with chronic spinal cord injury," *J. Spinal Cord Med.*, vol. 47, no. 3, pp. 354–360, 2024.

[5] J. S. Lora-Millan, F. J. Sanchez-Cuesta, J. P. Romero, *et al.*, "Robotic exoskeleton embodiment in post-stroke hemiparetic patients: An experimental study about the integration of the assistance provided by the reflex knee exoskeleton," *Sci. Rep.*, vol. 13, no. 22908, 2023.

[6] J. Wang, M. Mattamala, C. Kassab, *et al.*, "Exosense: A vision-based scene understanding system for exoskeletons," *IEEE Robot. Autom. Lett.*, vol. 10, no. 4, pp. 3510–3517, 2025.

[7] L. Schmid, J. Martinez Montiel, S. Huang, *et al.*, "Dynamic and deformable SLAM," in *SLAM Handbook — From Localization and Mapping to Spatial Intelligence*, L. Carlone, A. Kim, F. Dellaert, *et al.*, Eds. Cambridge University Press, 2025.

[8] M. Labbé and F. Michaud, "Multi-session visual SLAM for illumination-invariant re-localization in indoor environments," *Front. Robot. AI*, vol. 9, 2022.

[9] M. Labbé and F. Michaud, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *J. Field Robot.*, vol. 36, no. 2, pp. 416–446, 2019.

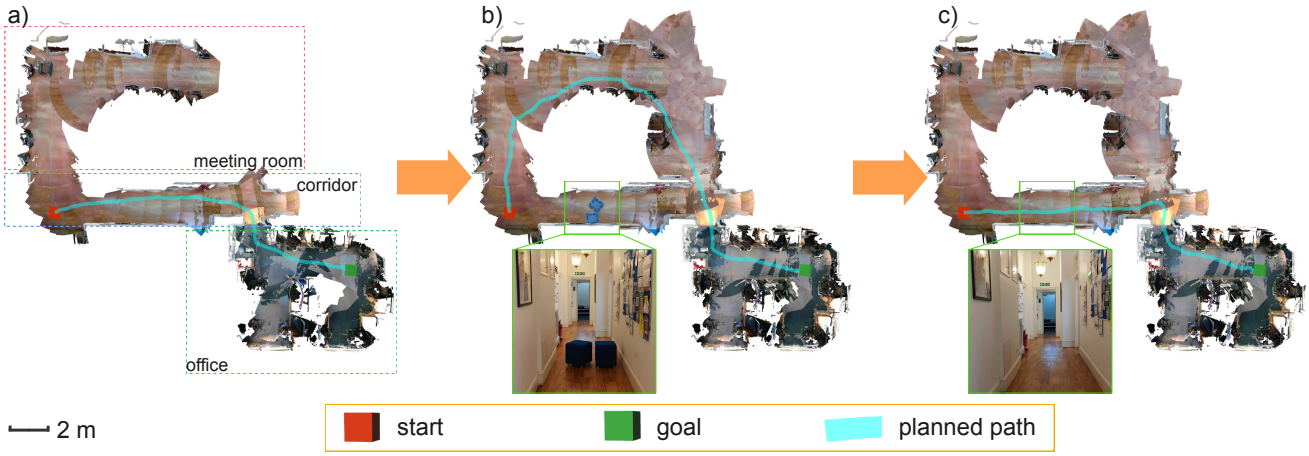


Fig. 4: Change-aware adaptive path planning with LT-Exosense through progressive multi-session mapping. From left to right, each subfigure illustrates the planned path from the same start to the same goal as a new mapping session is incrementally merged. **a)** In Session 1, a direct path through the corridor is planned to reach the goal in the office. **b)** After merging Session 2, the map becomes more complete but two obstacles appear in the middle of the corridor. LT-Exosense re-plans the path to bypass the obstruction by rerouting through the meeting room. **c)** When the obstacle is removed in Session 3, the path planner can recover the shorter route again.

[10] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2018, pp. 337–33 712.

[11] P.-E. Sarlin, D. DeTone, T. Malisiewicz, *et al.*, "SuperGlue: Learning feature matching with graph neural networks," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020.

[12] M. Labbé and F. Michaud, "Long-term online multi-session graph-based splam with memory management," *Auton. Robot.*, vol. 42, pp. 1133–1150, 2018.

[13] T. Schneider, M. T. Dymczyk, M. Fehr, *et al.*, "maplab: An open framework for research in visual-inertial mapping and localization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1418–1425, 2018.

[14] M. Bloesch, S. Omari, M. Hutter, *et al.*, "Robust visual inertial odometry using a direct EKF-based approach," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 298–304.

[15] A. Cramariuc, L. Bernreiter, F. Tschopp, *et al.*, "maplab 2.0 – A modular and multi-modal mapping framework," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 520–527, 2023.

[16] M. Tranzatto, T. Miki, M. Dharmadhikari, *et al.*, "Cerberus in the DARPA subterranean challenge," *Sci. Robot.*, vol. 7, no. 66, 2022.

[17] A. Pretto, S. Aravecchia, W. Burgard, *et al.*, "Building an aerial–ground robotics system for precision farming: An adaptable solution," *IEEE Robot. Autom. Mag.*, vol. 28, no. 3, pp. 29–49, 2021.

[18] M. Staniaszek, T. Flatscher, J. Rowell, *et al.*, "AutoInspect: Toward long-term autonomous inspection and monitoring," *IEEE Trans. Field Robot.*, vol. 2, pp. 529–548, 2025.

[19] K. Ohno, S. Tadokoro, K. Nagatani, *et al.*, "Trials of 3-d map construction using the tele-operated tracked vehicle Kenaf at disaster city," in *IEEE Int. Conf. Robot. Autom.*, 2010, pp. 2864–2870.

[20] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3019–3026, 2018.

[21] A. Hornung, K. M. Wurm, M. Bennewitz, *et al.*, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robot.*, vol. 34, pp. 189–206, 2013.

[22] T. Miki, L. Wellhausen, R. Grandia, *et al.*, "Elevation mapping for locomotion and navigation using gpu," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 2273–2280.

[23] F. Pomerleau, P. Krüsi, F. Colas, *et al.*, "Long-term 3D map maintenance in dynamic environments," in *IEEE Int. Conf. Robot. Autom.*, 2014, pp. 3712–3719.

[24] H. Lim, S. Hwang, and H. Myung, "ERASOR: Egocentric ratio of pseudo occupancy-based dynamic object removal for static 3d point cloud map building," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2272–2279, 2021.

[25] G. Kim and A. Kim, "Remove, then revert: Static point cloud map construction using multiresolution range images," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020.

[26] L. Schmid, O. Andersson, A. Sulser, *et al.*, "Dynablox: Real-time detection of diverse dynamic objects in complex environments," *IEEE Robot. Autom. Lett.*, vol. 8, no. 10, pp. 6259 – 6266, 2023.

[27] D. Duberg, Q. Zhang, M. Jia, *et al.*, "DUFOMap: Efficient dynamic awareness mapping," *IEEE Robot. Autom. Lett.*, vol. 9, no. 6, pp. 5038–5045, 2024.

[28] J. Rowell, L. Zhang, and M. Fallon, "LiSTA: Geometric object-based change detection in cluttered environments," in *IEEE Int. Conf. Robot. Autom.*, 2024, pp. 3632–3638.

[29] M. Jia, Q. Zhang, B. Yang, *et al.*, "BeautyMap: Binary-encoded adaptable ground matrix for dynamic points removal in global maps," *IEEE Robot. Autom. Lett.*, vol. 9, no. 7, pp. 6256–6263, 2024.

[30] E. Rublee, V. Rabaud, K. Konolige, *et al.*, "ORB: An efficient alternative to SIFT or SURF," in *Intl. Conf. on Computer Vision (ICCV)*, 2011, pp. 2564–2571.

[31] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robotics*, vol. 28, no. 5, pp. 1188–1197, October 2012.

[32] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. J. of Computer Vision*, vol. 60, pp. 91–110, 2004.

[33] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, p. 381–395, 1981.

[34] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.

[35] G. H. Lee, F. Fraundorfer, and M. Pollefeys, "Robust pose-graph loop-closures with expectation-maximization," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 556–563.

[36] E. Jelavic, D. Jud, P. Egli, *et al.*, "Towards autonomous robotic precision harvesting: Mapping, localization, planning and control for a legged tree harvester," *Field Robotics*, 2021.

[37] L. Kavraki, P. Svestka, J.-C. Latombe, *et al.*, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[38] M. Burri, J. Nikolic, P. Gohl, *et al.*, "The euroc micro aerial vehicle datasets," *Intl. J. of Robot. Res.*, 2016.

[39] M. Ramezani, G. Tinchev, E. Iuganov, *et al.*, "Online lidar-slam for legged robots with robust registration and deep-learned loop closure," in *IEEE Int. Conf. Robot. Autom.* IEEE, 2020, pp. 4158–4164.

[40] P. Geneva, K. Eckenhoff, W. Lee, *et al.*, "OpenVINS: A research platform for visual-inertial estimation," in *IEEE Int. Conf. Robot. Autom.*, 2020.

[41] C. Kassab, M. Mattamala, L. Zhang, *et al.*, "Language-extended indoor SLAM (LEXIS): A versatile system for real-time visual scene understanding," in *IEEE Int. Conf. Robot. Autom.*, 2024.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	LT-Exosense: A Vision-centric Multi-session Mapping System for Lifelong Safe Navigation of Exoskeletons
Publication Status	Unpublished and unsubmitted work written in a manuscript style
Publication Details	Wang, J., Mattamala, M., Kassab, C., Chebrolu, N., and Fallon, M. (2025) "LT-Exosense: A Vision-centric Multi-session Mapping System for Lifelong Safe Navigation of Exoskeletons" IEEE Robotics and Automation Letters (RA-L) (To Submit)

Student Confirmation

Student Name:	Jianeng Wang		
Contribution to the Paper	My contributions to the paper were: • Developed the core idea behind the paper with co-authors • Implemented the system software • Performed the experiments with co-authors • Wrote the paper with co-authors		
Signature		Date	2025/06/17

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Prof. Maurice Fallon		
Supervisor comments This paper/project was wholly the work of Jianeng (aside from the secondary contribution by the other co-authors). It's a follow-on of the original (published) Exosense paper. It is intended to be submitted to IEEE RA-L during Summer 2025		
Signature 	Date	2025/06/23

This completed form should be included in the thesis, at the end of the relevant chapter.

6.2 Additional Remarks

6.2.1 Compatibility with Other Sensor Suites

The session creation in LT-Exosense is not limited to the specific hardware. It can take odometry, images and point cloud measurements from a multi-sensor setup to generate the SLAM session of the same format. Specifically, vertices are spawned at the time of visual images by interpolating the odometry poses to build the pose graph. A new vertex would only be spawned when its pose difference from the previous vertex reaches the distance threshold. The temporally closest point cloud measurements would then be motion compensated to the vertex base frame. The depth cue of the vertex-associated image is sensor dependent. When a stereo-camera setup is available, we store both stereo images to the vertex. If the vision sensor is an RGB-D camera, the depth image is stored as the depth cue. When the sensor suite has a LiDAR device, we can render a depth image for the vertex-associated visual image from the LiDAR point cloud. Once all vertices have the associated visual images, the semantic terrain label is queried that best describes the image for each vertex given a list of potential terrain classes using CLIP model [198].

In Fig. 6.1, the *Frontier* device (Fig. 6.1a), an alternative multi-sensor suite, is used to record different sequences for multi-session merging. By running VILENS [264] on point cloud data as the odometry source, the pose graph is built, where vertex-associated resources are motion compensated and saved locally. The multi-session merging is executed in the same manner as shown in LT-Exosense by finding inter-session visual place recognition to connect the graph from each session. The resultant merged graph and point cloud are shown in Fig. 6.1b and c.

6.2.2 Ground Plane Segmentation and Registration

When constructing individual SLAM sessions, odometry estimation can be significantly degraded in the presence of high accelerations or jerky motions. This may result in misaligned submaps after motion compensation. Furthermore, since LT-Exosense employs vision-based place recognition, it does not account for the

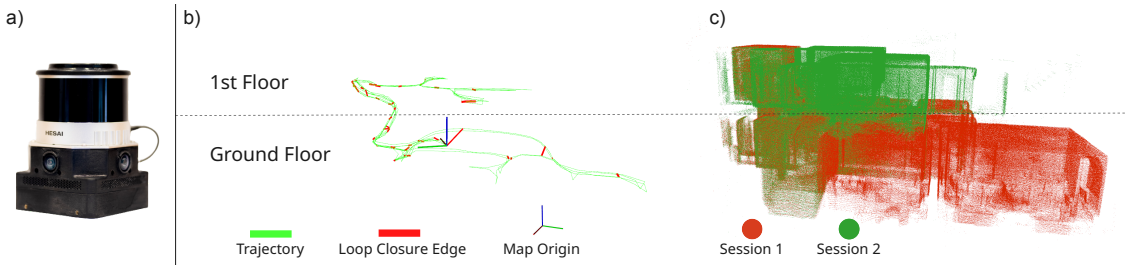


Figure 6.1: Multi-session merging using different sensor suites. **a)** The Frontier sensor suite, which has three cameras, an IMU and a LiDAR. We construct a single SLAM session using the sensor output. **b)** We merge the two SLAM sessions by finding the inter-session visual place recognition and jointly optimize the combined graph. **c)** The resultant merged LiDAR point cloud map. Points from each session are colored separately.

point cloud-level misalignments during loop closure. As a result, when merging multiple sessions, ground-level inconsistencies may persist, which can hinder safe robot navigation (Fig. 6.2b). To address this issue, a ground plane modeling and point cloud registration scheme is designed to improve consistency in the reconstructed ground surface.

Given a merged map composed of several SLAM sessions, pose nodes are first partitioned using a 2D grid structure. Nodes located in the same cell are grouped together. For each group, ground plane segmentation is performed on the associated point cloud, resulting in multiple ground plane estimates in the global map frame (Fig. 6.2a). A refinement step then merges nearby groups with spatially close ground planes. Ground segmentation is repeated using the combined point clouds of the refined groups, and inlier points within a distance threshold from the fitted plane are bookkept and stored relative to each pose's base frame.

For each resulting plane, denoted as $\boldsymbol{\nu} = \begin{bmatrix} a & b & c & d \end{bmatrix}^T$, where the plane normal is $\mathbf{n} = \begin{bmatrix} a & b & c \end{bmatrix}^T$ with $\|\mathbf{n}\| = 1$, each inlier point contributes a unary factor that enforces a point-to-plane distance constraint on its associated pose node. The total cost function is defined as:

$$J_{\text{ground registration}} = \sum_{i,k} \frac{e_{i,k}^2}{\sigma_{i,k}^2}, \tag{6.1}$$

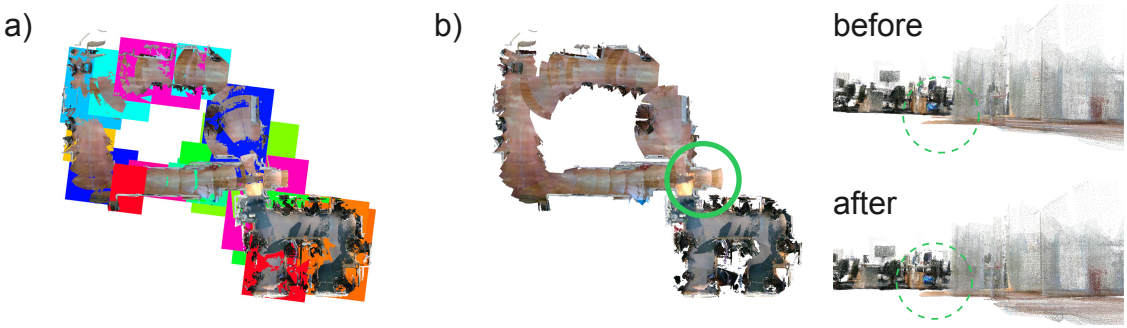


Figure 6.2: Ground plane segmentation and registration. **a)** Ground plane segmentation on a merged map from 3 individual SLAM sessions. **b)** Comparison of the ground point cloud before and after adding ground plane registration factors. The highlight area demonstrates that after ground plane registration, the misalignment of the ground point cloud can be largely mitigated.

where $\sigma_{i,k}^2$ is the variance associated with the point-to-plane distance error, $e_{i,k}$, for pose $\mathbf{T}_{\text{map,base}_i}$ and the inlier point, $\mathbf{p}_{\text{base}_i}^k$,

$$e_{i,k} = \mathbf{n}^T \mathbf{T}_{\text{map,base}_i} \begin{bmatrix} \mathbf{p}_{\text{base}_i}^k \\ 1 \end{bmatrix} + d. \tag{6.2}$$

Only the pose $\mathbf{T}_{\text{map,base}_i}$ is treated as a variable during optimization, while the inlier points are considered as a measurement.

This approach allows misaligned ground surfaces from different sessions to be consistently stitched together, improving map alignment and navigability in merged sessions (Fig. 6.2b). However, the current implementation assumes that each group of poses lies on a single planar surface, which is a strong assumption that does not hold in complex or unstructured terrain. For broader applicability, future work must incorporate more flexible models that account for varying terrain geometries.

6.3 Discussion

This chapter presents *LT-Exosense*, a vision-centric, multi-session mapping system designed to support long-term autonomous navigation for self-balancing exoskeletons. The system operates on single-session maps generated by the *Exosense* system, incrementally fusing them across multiple environment traversals to detect changes and update the global map. These updates enable adaptive path planning that reacts appropriately to evolving environmental conditions. In addition, LT-Exosense

demonstrates compatibility with various sensor configurations, highlighting its potential for broader deployment across different exoskeleton platforms.

While the current system lays the foundation for long-term spatial memory and change-aware planning, several opportunities exist for future improvement.

As all experiments were conducted using a human-leg-mounted mode of the Exosense system, performing LT-Exosense on sessions from a full exoskeleton platform and validating its performance in live navigation scenarios would be a critical next step. Meanwhile, the current implementation does not specifically manage the map resources when more sessions are merged. With increasing number of sessions merged, the size of the pose graph and submaps also grows. Future iterations should incorporate strategies for graph pruning [109], sparsification [175] and efficient map fusion techniques [207] to ensure long-term efficiency.

From the practical system deployment perspective, although LT-Exosense currently operates offline, it could be tightly integrated with the exoskeleton system—for instance, as part of a charging or docking station. In such a setup, the system could automatically retrieve and process session data during idle periods, maintaining an up-to-date map without user intervention.

Finally, the long-term, daily-life deployment of LT-Exosense requires careful consideration of social and ethical responsibilities, particularly regarding privacy. To safeguard the privacy of both the user and bystanders, future development must incorporate specific privacy-preserving features. A crucial software measure would be to process all visual data to detect and blur sensitive information. An open-source tool like EgoBlur [199] could be integrated to automatically blur faces and license plates. This should also be complemented by a hardware-based solution for transparency, such as a visible LED indicator on the sensor unit that clearly signals when the system is capturing images. Together, these measures are essential for building the trust required for the responsible use of the system in public and private spaces.

7
Conclusion and Future Work

Contents

This thesis has presented three vision-centric state estimation and mapping frameworks that improve 3D scene understanding in visually challenging and dynamic environments. These contributions address the challenges associated with asynchronous sensing, wearable robotics, and long-term spatial awareness. This chapter will summarize the contributions and discuss potential future research directions.

7.1 Conclusion

Chapter 4 introduced a visual odometry system that takes advantage of the unique sensing characteristics of event cameras—namely their high temporal resolution and asynchronous output. By modeling the camera motion as a continuous-time trajectory and incorporating a physically-founded motion prior, the system achieves

per-event state estimation that is robust under high-speed motion and extreme lighting conditions. The estimated trajectory supports pose queries at any time in the motion window, offering a finer temporal fidelity than traditional frame-based methods.

Chapter 5 and 6 are centered around enhancing the scene understanding capability of self-balancing exoskeletons. Chapter 5 presents *Exosense*, a vision-based scene understanding system designed for integration on the lower limbs of an exoskeleton. The system provides a navigation stack that builds rich, multi-layer elevation maps encoding terrain geometry, semantic information, and traversability. This enables downstream tasks such as localization and navigation for assistive mobility.

Building on Exosense, Chapter 6 introduces *LT-Exosense*, a change-aware, multi-session mapping system designed to support the long-term deployment of self-balancing exoskeletons in evolving environments. It aligns and merges individual sessions into a globally consistent map while performing change detection and map updates. This system supports persistent and adaptive environment modeling, enabling the exoskeleton to plan paths in response to dynamic changes such as newly introduced obstacles. As a result, the system facilitates adaptive multi-query path planning over time.

7.2 Future Work

7.2.1 Improved Odometry System for Event Camera

The proposed stereo event-based VO system demonstrates the feasibility of continuous-time camera trajectory estimation at high temporal fidelity. Nevertheless, there are several potential directions which could enhance its performance. On the frontend, replacing the clustering-based approach that mimics the traditional frame-based feature detection and tracking with methods specifically designed for event data could yield more reliable feature tracklets. This includes handcrafted methods such as Arc* [6] and HASTE [3], as well as data-driven techniques like SILC [154] and the approach by Messikommer et al. [159].

On the backend, introducing sparse trajectory nodes would reduce the optimization problem size [7], as well as the overall system computation. Moreover, due to the limited contextual awareness of event cameras, fusing asynchronous events with complementary sensor modalities such as standard cameras [131], IMUs [181], or LiDAR [260] could improve robustness across diverse scenarios.

7.2.2 Enhanced Scene Understanding for Self-balancing Exoskeleton

The Exosense system showcases the viability of using leg-mounted, wide field-of-view cameras for terrain-aware perception in exoskeleton applications. Potential future work includes deploying the system on embedded computational platforms such as the NVIDIA Jetson, which would allow real-time inference of more advanced perception models. These may include Large Language Models (LLMs) [187], Vision-Language Models (VLMs) [197], or Vision-Language Navigation (VLN) models [99], to facilitate user-friendly, context-aware scene understanding.

Further, combining Exosense with other wearable vision systems, such as Meta's Aria glasses [54], would result in cooperative, multi-view perception systems. Such a system could build a distributed scene graph with multi-view scene understanding of the environment from both egocentric and limb-centric perspectives. Ultimately, integrating Exosense with the exoskeleton's locomotion control stack would be a key milestone. Such system integration would benefit the autonomy of the self-balancing exoskeleton and help to achieve safe long-term navigation.

7.2.3 Towards Practical Long-Term Multi-Session Mapping System

LT-Exosense takes an important step towards long-term scene representation by merging individual sessions and updating maps based on detected changes. To make this system more suitable for real-world deployment, future iterations should address scalability and automation.

To reduce computational complexity and memory usage, techniques such as graph pruning [109] and spectral sparsification [175] can be used. Additionally,

enabling online variants of session merging [242] and change detection [216] would allow the system to update maps incrementally and in real time. Additionally, incorporating LT-Exosense into a background mapping workflow, such as during idle periods at a docking or charging station, would enable persistent autonomy and seamless map maintenance without requiring user supervision.

Appendices

A

Event-based Corner Detection with Graph Walk

Contents

This appendix introduces an additional and unpublished work. This work presents an event-based corner detector, which leverages a graph walk method to search for corner patterns on the Surface of Active Events (SAE) representation constructed from the event stream.

A.1 Introduction

Over decades of development, the conventional frame-based camera used as sensor input has obtained the dominant position in the machine vision field. Out of the

richness of information provided in the image, the conventional frame-based camera is frequently used for practical machine vision applications like SLAM, where image data can be used to estimate the motion and reconstruct the scene.

Though success has been consistently made under the general scenarios [172, 51], the limitations of frame-based camera grow when the scene shifts to a more extreme case, which specifically refers to images captured at high speed camera motion or under high dynamic range illumination. Under such challenging scenarios, the camera observation is unable to provide sufficient information for those SLAM pipelines, which would result in system failure.

To tackle the limitations in frame-based systems, researchers are actively seeking an alternative sensor modality for the challenging case. The emergence of the bio-inspired event camera has drawn the attention of the community. In contrast to the frame-based camera that outputs images at a fixed rate, the event camera reports the pixelwise brightness change of the scene asynchronously. Whenever the intensity change of an individual pixel reaches a predefined threshold, an event is triggered at that position reporting the pixel location, timestamp and polarity information of that brightness change. Apart from the asynchronous output, the event camera also offers various advantages over the frame-based camera. The high temporal resolution ($< 1\mu s$) and high dynamic range ($> 120dB$) characteristics enable the event camera to detect fast motion under high dynamic range scenes, which is a promising sensor for machine vision tasks in extreme cases. Also, since the event camera only senses brightness change, it does not output events in front of a static scene, effectively reducing the redundant information normally captured by the frame-based camera, which yields a much lighter and low-power system.

Since the event data is fundamentally different from the frame data, event output is not directly applicable to the frame-based algorithms. A paradigm shift in the way of processing events is required. As the event camera would be used in a natural scene with complicated textures, the number of events to process is huge. Effective feature detection is therefore required to compress the event stream, which can be later input to other machine vision systems. Driven by this need,

a novel event-based feature detector is designed to detect corner event features. The system utilizes the nature of the event camera to find corner patterns and operates in an event-by-event manner on the event stream.

A.2 Related Work

The event data for corner detection can be processed in two main manners, in a batch manner or an event-by-event manner [71].

Typical works such as [204, 278] perform corner detection by first accumulating events within a temporal window to generate an event frame, which is an image-like structure analogous to a grayscale frame. Conventional frame-based corner detectors can then be applied to extract corners. A notable example is eHarris [248], an adaptation of the Harris corner detector [89] for event data. In this method, a fixed number of incoming events are registered on a binary patch, where each pixel is set to one if an event occurred there, and zero otherwise. The Harris score is computed over this patch and compared to a threshold to classify an event as a corner.

While such batch-based methods can achieve reasonable corner detection accuracy, they are computationally inefficient due to the overhead of frame construction, which is a key factor in real-time state estimation problem. Moreover, by grouping events, these methods fail to exploit the inherent asynchronicity of event data, which is crucial to enable high-frequency, low-latency applications [4].

Favoring the asynchronicity of event data, several event corner detectors have been proposed to process the event stream in an event-by-event manner. Clady et al. [36] presents a corner detector that firstly registers events on the Surface of Active Events (SAE) [16], a map with the same spatial resolution as the camera. Each element on it records the timestamp of the newest event appearing on that location. A patch is then extracted around the incoming event, and planes in that patch are fitted based on the optical flow calculation [16]. A corner event is classified if the current event appears at the intersection of fitted planes. This method however requires a large amount of computational resource.

To improve speed, Mueggler et al. presented eFAST [168], an event-based adaptation of the FAST corner detector [252]. This method also utilizes the SAE: for each new event, a circle of pixels around its location is evaluated. A continuous arc is searched to identify constituent pixels which have newer timestamps than the rest of the circle. If the arc length falls within a predefined range, the event is classified as a corner. While eFAST is significantly faster than earlier methods, it suffers from limited accuracy and fails to detect corners forming arcs larger than 180°, resulting in missed detections.

To overcome this limitation, Arc* [6] was proposed as an enhancement to eFAST. Arc* considers both the arc and the complementary region around the circular neighborhood, enabling detection of corners with wider arc angles. Although Arc* increases the number of detected corner events, it also raises the rate of false positives, which can degrade overall detection accuracy.

In FA-Harris [144], the authors combine both eHarris and Arc*, where corner candidates are firstly selected by applying Arc* on the event stream then refined by eHarris to output the final corner events. This method slightly improves corner accuracy compared to the FAST-inspired algorithm (i.e., eFAST and Arc*) but increases the overall computational cost.

A.3 Methodology

This section presents the unpublished additional work which aims to develop an improved event-based corner detector.

The detector was motivated by a fundamental property of event cameras: their natural responsiveness to edge motion in a scene [71]. Unlike conventional cameras, event cameras generate asynchronous data based on changes in brightness at individual pixels. According to the general event generation model [188], an event $\mathbf{e} = \{t, x, y, p\}$ is triggered at pixel $\mathbf{x} = (x, y)$ and time t when the change in log intensity exceeds a certain threshold:

$$|\Delta L(\mathbf{x})| > C_k, \tag{A.1}$$

where $\Delta L(\mathbf{x})$ represents the change in log intensity at pixel $\mathbf{x}$, and C_k is a contrast threshold. The polarity $p \in \{-1, +1\}$ indicates whether the brightness increased or decreased.

This model indicates that whenever a physical edge moves across the field of view, the event camera outputs a stream of events that correspond to the edge's projected motion on the image plane. To handle a large number of generated events, each incoming event is registered on the SAE, a 2D temporal map where each pixel stores the timestamp of the most recent event at that location.

For every incoming event, two paths starting from it are extended on SAE. Each path traverses along the newest events that are both spatially and temporally closer to this incoming event's position and timestamp, which effectively connects edge events that represent the edge's most recent motion. If the incoming event is an edge event, the two paths would be parallel. Otherwise, two edges intersect at the incoming event's pixel location, forming a corner pattern, which indicates the event at the intersection point is a corner event (Fig. A.1).

A.3.1 Event preprocessing

Motivated by the insight that two paths alongside the incoming event can effectively mimic the edges in the real scene, two tokens are selected to walk on SAE and hence find paths. Since the quality of the path is dependent on the noise in the event stream, to robustly filter out noise, an event filter [6] is applied before registering the event on SAE. This filter sets a time window to prevent consecutive events triggered by noisy scenes such as the rapid contrast change. For consecutively triggered events on the same pixel, if their timestamp difference is within this time window, it means the events are noisy and rejected from registration. Otherwise, the event registration would be the same as the regular SAE registration. The modified event registration stage is denoted as S^*, and in this work, a fixed time window value of 0.05 s is set.

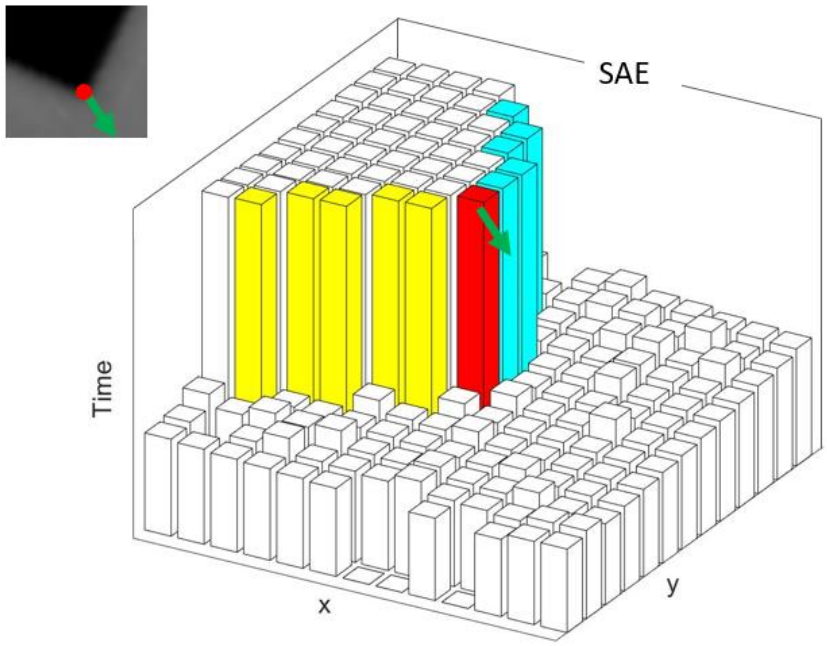


Figure A.1: Visualization of a corner pattern on SAE. The corner event is marked in red, and two paths that connect the most recent events alongside the corner event are shown in cyan and yellow. The green arrow indicates the direction of movement of the corner. The corresponding intensity frame of the corner's movement is shown in the upper left, where the red dot indicates the location of a corner and the green arrow marks the moving direction. In this case, the two paths intersect and form a corner pattern on SAE. The event at the intersection point is therefore classified as a corner event.

A.3.2 Graph walk based event corner detector

For each incoming event registered on S^*, a 3×3 spatial patch centered on the event's pixel location is extracted, encompassing its eight neighboring pixels. Among these, the two most recent events (i.e., the pixels with the highest timestamps) are selected as initial tokens for graph-based traversal. These tokens then begin independent outward walks to extend their respective paths.

Due to imperfections in real-world event generation from sensor noise, hardware limitations, and the inherently stochastic triggering behavior [188], the ideal event distribution predicted by the model in Eq. (A.1) is rarely achieved. As such, the S^* filtering alone is insufficient to guarantee clean signal boundaries. To improve robustness, an adaptive patch expansion strategy is employed during token traversal. In this strategy, the timestamp of the current token is compared

with those of its neighbors. If a neighbor's timestamp is sufficiently close (i.e., within a defined threshold) to that of the current token, the neighborhood of that neighbor is also considered as part of the search space. These newly added neighbors are merged with the surroundings of the current token to form an expanded patch, increasing the likelihood of identifying the next relevant token and improving classification accuracy.

As the graph walk continues, a token may inadvertently step backward and toward previously used regions since those areas may still hold newer events than unvisited regions. To avoid this, previously visited areas are excluded from the next token's search region. Within this pruned region, the pixel with the newest event timestamp is selected as the next token. The walk continues until a predefined distance threshold is reached between the token and the original event, which serves as the stopping condition.

In cases where the two token paths intersect (i.e., the same pixel is selected as the next token for both walks), the overlap is resolved by assigning that pixel to one path, while the second-newest pixel in the other search region is used for the alternate token.

When both tokens are far enough from the original event, the walk terminates and corner classification begins. Two vectors are formed from the current event's pixel location to the final positions of the two tokens. If the angle between these vectors falls within a predefined threshold range, the event is classified as a corner event, indicating that it lies at the intersection of two motion paths (see Fig. A.1).

A.3.3 Algorithm description

The overall algorithm of the presented graph walk based event corner detector is described in Algorithm 1. For every incoming event, $\mathbf{e}_c$, two newest events, denoted as tokens, are selected from its eight neighbor positions on the SAE. The two selected events' positions are then marked as the used region, A_{used} (Line 1). For each following iteration, a patch that contains not only the eight nearest neighbors but also considers the potentially expandable region of each token defines

Algorithm 1 Graph walk based event corner detection

Input: S^*, $\mathbf{e}_c$, Δt, $d_{\max}$, $\theta_{\min}$, $\theta_{\max}$
1: $[\mathbf{w}_1, \mathbf{w}_2, A_{\text{used}}]$=FindInitialTwoTokens($S^*$, $\mathbf{e}_c$)
2: $[d_{\mathbf{w}_1}, d_{\mathbf{w}_2}]$= DistanceCalculation($\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{e}_c$)
3: **while** $d_{\mathbf{w}_1} \leq d_{\max} \| d_{\mathbf{w}_2} \leq d_{\max}$ **do**
4: **if** $d_{\mathbf{w}_1} \leq d_{\max}$ **then**
5: A_1=ObtainNextSearchingRegion($\mathbf{w}_1$, Δt, S^*, A_{used})
6: $[\mathbf{w}_1, A_{\text{used}}]$=FindNextToken($A_1$)
7: **end if**
8: **if** $d_{\mathbf{w}_2} \leq d_{\max}$ **then**
9: A_2=ObtainNextSearchingRegion($\mathbf{w}_2$, Δt, S^*, A_{used})
10: $[\mathbf{w}_2, A_{\text{used}}]$=FindNextToken($A_2$)
11: **end if**
12: $[d_{\mathbf{w}_1}, d_{\mathbf{w}_2}]$=DistanceCalculation($\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{e}_c$)
13: **end while**
14: $[\mathbf{v}_1, \mathbf{v}_2]$=CreateVectors($\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{e}_c$)
15: $\theta_{\mathbf{v}}$=AngleCalculation($\mathbf{v}_1$, $\mathbf{v}_2$)
16: **if** $\theta_{\min} \leq \theta_{\mathbf{v}} \leq \theta_{\max}$ **then**
17: $IsCorner = true$
18: **else**
19: $IsCorner = false$
20: **end if**
Output: $IsCorner$

the next token's search region, which is denoted as A_1 and A_2 for each token's search. A tunable parameter, Δt, is used to determine if the timestamps of eight neighbor element are close to that of the current token, their eight neighbor pixels would also be considered and are merged with the surroundings of the current token to form an expanded patch (Line 5 and 9). Two tokens with the newest timestamps in each searching region are then selected, denoted as $\mathbf{w}_1$ and $\mathbf{w}_2$ respectively. The iteration would run until the distance between the current event and the token, $d_{\mathbf{w}_1}$ and $d_{\mathbf{w}_2}$, respectively, is larger than the distance criteria, $d_{\max}$ on S^* (Line 2-13). After the following of both tokens stops, two vectors, $\mathbf{v}_1$ and $\mathbf{v}_2$, from the location of the current event to the two final selected tokens are created (Line 14) and the angle between the vectors, $\theta_{\mathbf{v}}$, is calculated (Line 15). If the angle is within a pre-defined range, from $\theta_{\min}$ to $\theta_{\max}$, the incoming event is classified as a corner event (Line 16-20).

A.4 Experiments

The proposed corner detector is evaluated on the `shape_6DOF` sequence from the publicly available Event Camera Dataset [171]. This dataset provides both asynchronous event streams and conventional intensity images recorded using a Dynamic and Active-pixel Vision Sensor (DAVIS), which offers a spatial resolution of 240 × 180 pixels [27]. The `shape_6DOF` sequence captures textured scenes of varying complexity, ranging from simple 2D geometric shapes to natural 3D scenes. These sequences are recorded under both standard and high dynamic range (HDR) lighting conditions. For each scene, the camera undergoes motion with increasing degrees of freedom and speed, providing a diverse and challenging set of scenarios for evaluation. Although the corner detector operates exclusively on event data, a similar methodology to that of [6] is employed to obtain corner ground truth. Specifically, corners are extracted from the corresponding intensity frames using a frame-based detector, which are then temporally aligned and spatially compared with the detected event-based corners. This provides a consistent basis for evaluating the detection accuracy of the event-only algorithm.

A.4.1 Ground Truth for corner accuracy evaluation

A detection and tracking paradigm on low-level features for ground truth generation is applied. In this process, the corner features on intensity frames are detected using the Harris corner detector [89], then the BRISK descriptor [140] is applied to each corner and establishes correspondence between detected features in consecutive frames. The tracked features are then exhaustively refined to discard noisy and too short cases, which finally results in multiple sets of 2D tracklets defined as ground truth. For the accuracy-related metrics, the corresponding positions of the detected features on the ground truth tracks are linearly interpolated with the timestamp of the corner event. The minimum distance between the corner position and its corresponding positions is compared with a threshold to determine whether the corner event is true positive.

Given a corner event, $\mathbf{e}_c = \{t_c, x_c, y_c, p\}$, the starting and end points of a tracklet, $\mathbf{P}_s = \{t_s, x_s, y_s\}$ and $\mathbf{P}_e = \{t_e, x_e, y_e\}$ respectively, where $t_s \leq t_c \leq t_e$, the interpolated point, $\mathbf{P}_i = \{t_c, x'_c, y'_c\}$ for this interpolation is shown in Eq. A.2,

$$\begin{aligned} x'_c &= x_s + (t_c - t_s)\frac{x_e - x_s}{t_e - t_s} \\ y'_c &= y_s + (t_c - t_s)\frac{y_e - y_s}{t_e - t_s}. \end{aligned} \tag{A.2}$$

In the previous experiments using intensity-based ground truth [6, 144], the authors only consider events triggered in the neighborhood of the tracklets (up to 5 pixels in the image plane). This consideration does ensure events are actually relevant to the real corners in the scene, but it still has limitations. On the one hand, it is heavily subject to the quality of ground truth. On the other hand, this approach inevitably omits a considerable number of corner events, lowering both the true positive and false positive rates used in their work, which is limited in reflecting the actual performance of the corner detector. To address these limitations, the whole event stream after S^* is considered.

During runtime, high textured scenes may accidentally appear in the field of view, which results in a sudden increase in the number of events output and degrades the quality of ground truth. The sequence therefore stops running at the time when other high textured scenes appear. Meanwhile, as the scene is recorded at an increasing camera moving speed, the images captured at a fixed frame rate may suffer from motion blur, which results in feature tracking failure and unreliable ground truth tracklets. Considering all these factors affecting the quality of ground truth, the first 573 images are selected to generate the ground truth and test the algorithms. The difference between the number of detected and tracked features of each frame is plotted in Fig. A.2, which is used to demonstrate the reliability of ground truth tracklets. The average difference in the tested dataset is 1, and the variance is 2.15 indicating a relatively stable ground truth.

To classify true positive corner events, Fig. A.3 shows the histogram of the distances of every detected corner to its closest tracked corner on the same frame, which indicates the distribution of the minimum distance between detected and

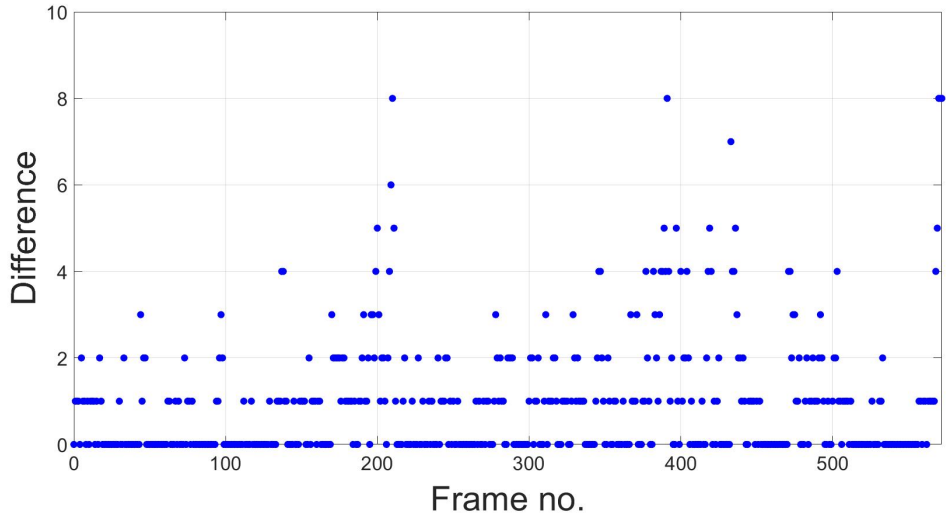


Figure A.2: Difference between the number of detected and tracked features on each frame. The mean difference is 1 and the variance is 2.15, indicating features can be consistently tracked through the selected frames most of the time.

tracked corners on each image. This distribution is assumed to be Gaussian and the 2nd percentile is selected to determine the radius of the valid region (i.e., true positive region) and minimize the number of dependent ground truths based on 68–95–99.7 rule. The radius is obtained by taking half of the selected number to avoid overlap of the valid region for ground truth. In this experiment, the radius to classify true positive events is 2.2 pixels, and every event within the ground truth neighborhood up to 2.2 pixels is considered as a true positive event.

A.4.2 Corner Accuracy

In this experiment, the precision and recall values are reported to evaluate the performance of the corner detectors, where a high precision value indicates fewer false corners detected while a high recall value shows more events are classified as corners. Since all the events within the neighborhood of ground truth tracklets defined by the radius are considered as true positive, all true positive events are denoted as TP_{all} and all true positive corner events are denoted as TP_c. The number of detected corner events is given as Num_c. The precision is calculated as

$$Precision = \frac{TP_c}{Num_c}.$$

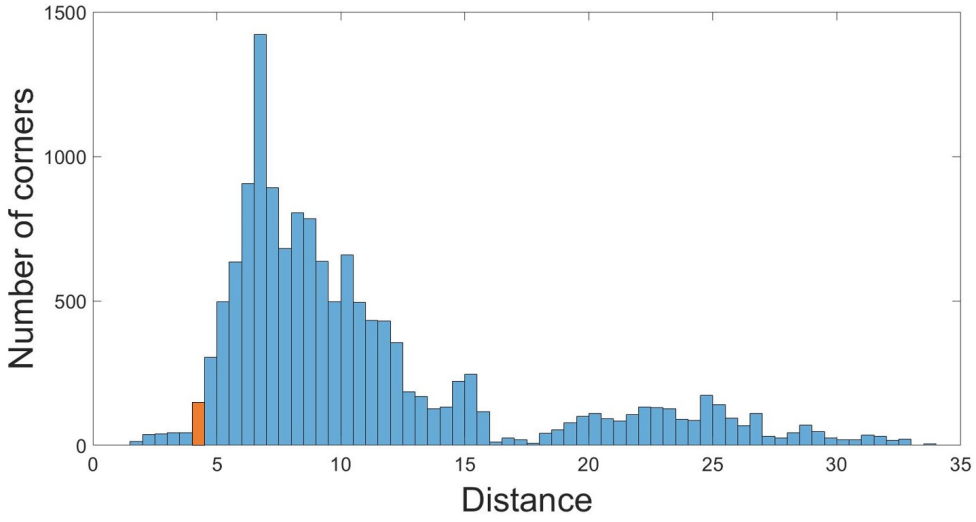


Figure A.3: Distribution of minimum distance between detected and tracked corners on the same image. The region that the second percentile lies in is painted in orange, and the radius value is half of the second percentile to avoid the overlap of true positive regions of adjacent ground truth tracklets. This selection of radius minimizes the number of false associations where a true positive corner is not classified by its nearest ground truth tracklet.

The recall is obtained by

$$Recall = \frac{TP_c}{TP_{all}}.$$

To enable a comprehensive analysis of corner detector performance, the parameter sensitivity test is carried out by varying one tunable parameter of our detector using precision and recall as the metrics. The tunable parameters in this algorithm are the Δt for expanding the potential searching region and the stopping condition, $d_{\max}$. Considering the spatial resolution of the used dataset, either too large or too small stopping condition would significantly degrade the detector performance. The stopping condition used for the detector design is therefore maintained as a constant value, and only the parameter sensitivity test on Δt is conducted. The result of parameter sensitivity test by changing the Δt for expanding the potential searching region is shown in Fig. A.4.

Compared to the operation (i.e., Line 5 and 9 in Algorithm 1) without expanding the potential searching region (i.e., $\Delta t = 0$), the one considers the potential searching region has a higher recall value while maintaining a constant precision value. This

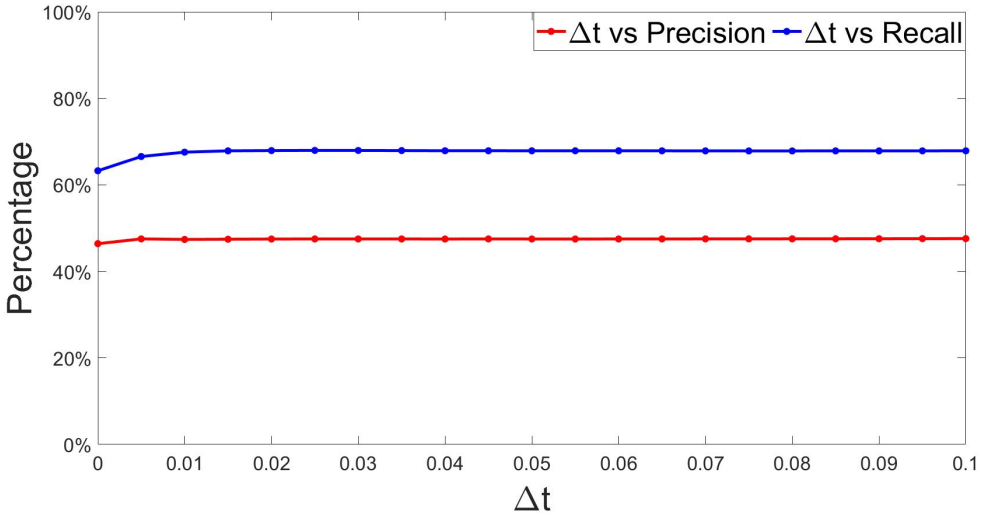


Figure A.4: The precision and recall curves with respect to Δt are marked in red and blue, respectively. The parameter sensitivity test runs Δt values from 0 to 0.1.

indicates the idea of expanding the potential searching region is valid for the event-based corner detector. Because the precision and recall curve is almost flat regardless of the variation of the tunable parameter (except when $\Delta t = 0$), the selection of Δt value has little effect on the detector performance. $\Delta t = 0.1$ is therefore selected to build the detector for performance analysis.

To determine a proper value for stopping the token search (i.e., $d_{\max}$ in Algorithm 1), considering the spatial resolution of the event camera dataset [171] (i.e., 240×180), a fixed distance criterion, $d_{\max} = 5$ is chosen, indicating when the searched token is at least 5 pixels away from the incoming event pixel location, the token search stops and one path is set.

After the token search stops, the angles formed by two vectors connecting from the current event location to the final searched tokens are computed as shown in Line in Algorithm 1. Events with an angle between 30° to 140° are classified as corner events. The maximum angle is set to a value that prevents edge event classification, whose angle would be a sizable obtuse angle close to 180 degrees. The minimum angle is chosen in order to avoid noisy detection, which happens when two paths of search are very close to each other.

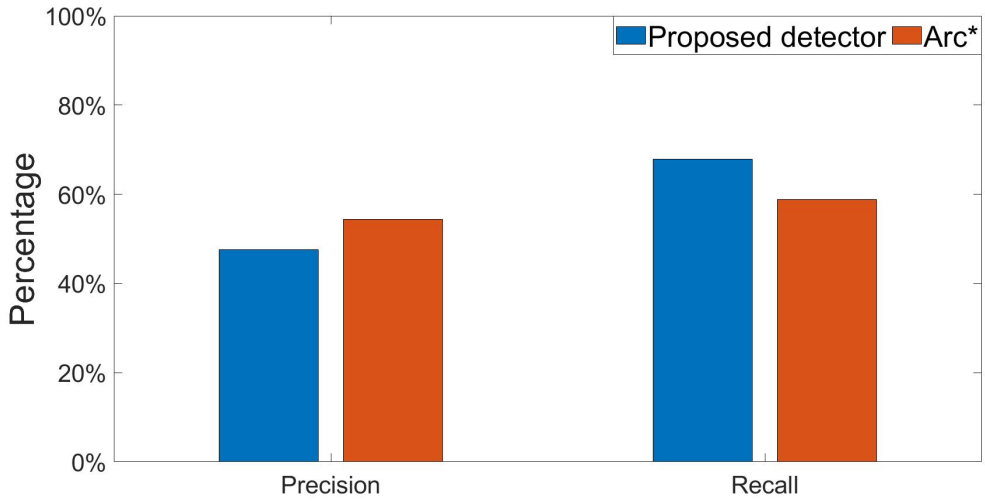


Figure A.5: The precision and recall value of the presented event-based corner detector and Arc* in `shape_6DOF` scene.

A representative peer work, Arc*, is compared with the proposed corner detector for corner accuracy analysis, and the default value provided in their original work [6] is used. For Arc* corner detector, two circles of radius, three and four are extracted around the incoming event, respectively. If the inner arc length is within three to six pixels and the outer arc length is within four to eight pixels, the incoming event is classified as a corner. The comparison of corner detector performance between our algorithm and Arc* is shown in Fig. A.5.

The result indicates that the presented detector has a better recall value, and a similar precision value. It can classify more true positive corners, but in the meantime incur an equivalent increase in the number of false positive corners. This would provide more information for downstream algorithms that require corner feature input, but too many noisy corners may also induce incorrect data association.

A.5 Future Work

As is shown in Section Sec. A.4, the presented detector does not demonstrate significant superiority over existing representative work, Arc*. More investigations are required to improve the detector performance. Because the proposed detector directly operates on SAE with the assumption that the searched paths would extend

along the events triggered by the most recent edge movements, the performance of the presented detector is highly dependent on the quality of SAE registration. In the current work, the values registered on SAE are the monotonically increasing timestamps of events. Since the generation of events is sparse, this simple SAE registration method leads to a large time span in the neighborhood of the incoming event, which may cause the failure of our corner detection. Efficient methods to improve the SAE registration are therefore required. Some existing works propose various approaches to normalize SAE, which can improve our corner detector performance. In Lagorce et al. [137], an exponential decay kernel is applied to SAE in order to normalize the value in the range of zero and one, which is described by Eq. (A.3),

$$T(\mathbf{x}) = \exp(\frac{-(t(\mathbf{x}) - t_{\text{last}}(\mathbf{x}))}{\eta}), \tag{A.3}$$

where $T(\mathbf{x})$ is the value registered on SAE after being processed by the exponential decay kernel. $t(\mathbf{x})$ is the timestamp of incoming event and $t_{\text{last}}(\mathbf{x}))$ is the last event's timestamp on that pixel location. η is a attenuation factor.

By confining the values between zero and one, the method avoids the potential classification failure caused by the monotonically increasing timestamp of the event stream. Applying the exponential decay kernel can also emphasize the most recent events over past events [4], which is aligned with the assumption of building the event-based corner detector by searching for the most recent events that represent the edges' motion.

An alternative approach to register events can also adopt Speed Invariant Time Surface (SITS) proposed in Manderscheid et al. [154] and Glover et al. [80]. The surface restricts the registered values between zero and 255, which mimics the 8-bit intensity images. Whenever an incoming event is triggered, its value on SITS is set to the maximum value (i.e., 255), while other values in its neighborhood are subtracted by one. This event registration method is invariant to the scene speed, which helps suppress the noise and provides a good quality surface for our detector to operate.

B

Seeing in the Dark: Benchmarking Egocentric 3D Vision with the Oxford Day-and-Night Dataset

This appendix attaches a co-authored paper: Seeing in the Dark: Benchmarking Egocentric 3D Vision with the Oxford Day-and-Night Dataset. This work presents a large-scale egocentric dataset recorded with Meta Aria Glass [54] under lighting variation for benchmarking novel view synthesis (NVS) and visual relocalization, supplemented with millimeter-level 3D models of the environments.

My contributions to the paper are the development of a toolkit to generate ground truth trajectories for all sequences within the corresponding ground truth scans and the content writing of Sec. 3.4 3.6, Figure 4 and Table 1.

Seeing in the Dark: Benchmarking Egocentric 3D Vision with the Oxford Day-and-Night Dataset

Zirui Wang[*†] **Wenjing Bian**[*†] **Xinghui Li**[*†]
Yifu Tao[‡] **Jianeng Wang**[‡] **Maurice Fallon**[‡] **Victor Adrian Prisacariu**[†]
[*]Equal Contribution [†]Active Vision Lab [‡]Dynamic Robot Systems Group
University of Oxford

Abstract

We introduce Oxford Day-and-Night, a large-scale, egocentric dataset for novel view synthesis (NVS) and visual relocalisation under challenging lighting conditions. Existing datasets often lack crucial combinations of features such as ground-truth 3D geometry, wide-ranging lighting variation, and full 6DoF motion. Oxford Day-and-Night addresses these gaps by leveraging Meta ARIA glasses to capture egocentric video and applying multi-session SLAM to estimate camera poses, reconstruct 3D point clouds, and align sequences captured under varying lighting conditions, including both day and night. The dataset spans over 30 km of recorded trajectories and covers an area of 40,000 m^2, offering a rich foundation for egocentric 3D vision research. It supports two core benchmarks, NVS and relocalisation, providing a unique platform for evaluating models in realistic and diverse environments. Project page: https://oxdan.active.vision/

1 Introduction

Intelligent wearable devices like smart glasses are gaining traction in the research community. Unlike bulky AR/VR headsets, their compact, lightweight design makes them more suitable for everyday use. To become as essential as smartphones, smart glasses must perform reliably across diverse environments, including challenging ones. A particularly tough scenario is outdoor low-light conditions, which uniquely degrade 3D vision tasks such as reconstruction, novel view synthesis (NVS), and visual localization due to poor signal-to-noise ratios. These tasks are key to interactive 3D experiences, yet current methods struggle in such settings. This highlights the need for a large-scale, egocentric 3D dataset tailored to low-light environments.

Existing 3D datasets, typically captured with handheld or vehicle-mounted cameras, provide diverse imagery but lack the combination of natural head motion, color, and full-day lighting variation, which are keys for all-day-long egocentric applications. Driving datasets like Oxford RoboCar [1] and CMU [2] offer large-scale, varied scenes including night, but are mostly limited to planar motion, unsuitable for agile head movements. Handheld datasets such as Cambridge Landmarks [3] and InLoc [4] offer more pose variation but limited lighting diversity. Aachen Day-Night [5] targets night-time localization but includes few night queries. LaMAR [6] provides egocentric day-night data, but its grayscale headset imagery limits suitability for color-dependent consumer applications.

To overcome limitations in existing datasets, we present Oxford-Day-and-Night, a large-scale video dataset captured across five locations in Oxford at various times of day. Spanning 30 kilometers and 40,000 m^2, it complements current datasets to provide a more comprehensive benchmark for 3D vision. This dataset is enabled by two key components: the Meta ARIA glasses and the Oxford Spires dataset [7].

Meta ARIA glasses are compact, sensor-rich devices equipped with grayscale and RGB cameras, IMUs, GPS, and more, enabling seamless and accurate data collection. Their built-in visual Simultaneous Localization and Mapping (SLAM) system ensures robust, multi-session camera tracking

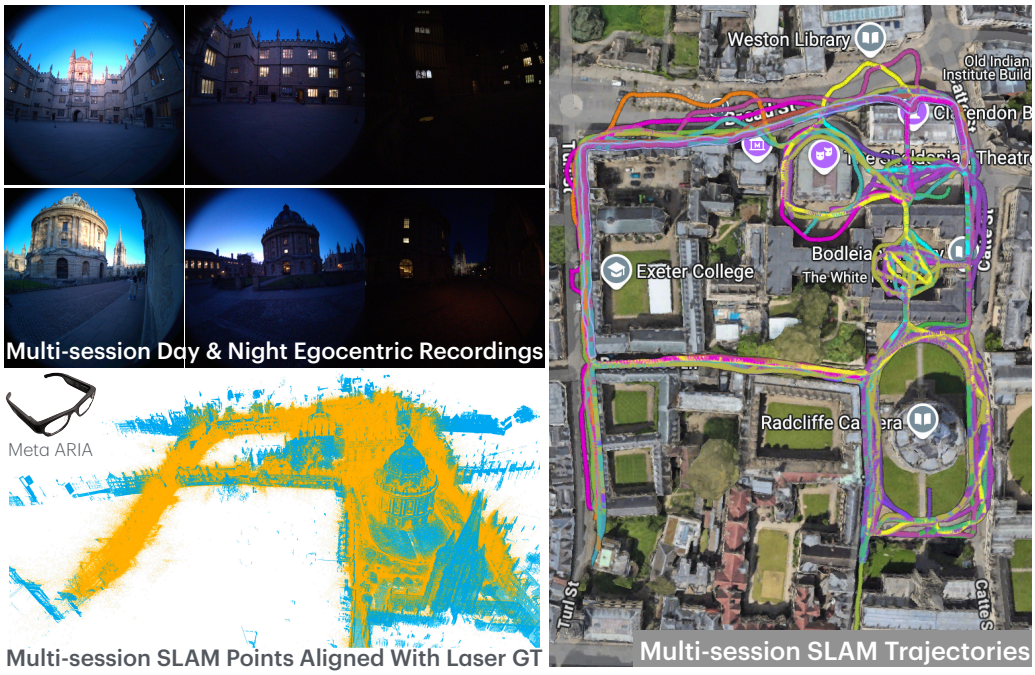


Figure 1: **Overview of the Oxford-Day-and-Night Dataset at Example Scene *Bodleian*.** Our dataset captures egocentric sequences across five locations in Oxford under diverse lighting conditions using Meta ARIA glasses. **Top-left**: Sample fisheye camera views across day and night recordings. **Bottom-left**: multi-session SLAM points aligned with high-quality laser ground truth. **Right**: Multi-session SLAM trajectories visualized on a satellite map, demonstrating consistent camera tracking across varying times of day. The dataset enables testing of challenging benchmarks for novel view synthesis and visual relocalization under extreme illumination changes.

and 3D reconstruction under dramatic lighting changes and city-scale settings. This multi-session SLAM system is the key component in creating our dataset, automating camera pose annotation for challenging night sequences at large scale. As a result, our video recordings cover 30 km and 40,000 m^2 areas in day and night settings, all paired with accurate camera poses and point cloud derived from the SLAM system.

Complementing this multi-session SLAM output, the Oxford Spires [7] dataset offers high-quality 3D laser scans of various Oxford locations. By aligning ARIA recordings with these scans, we both validate the accuracy of the ARIA data and offer reliable 3D geometric ground truth to support downstream tasks and benchmarking.

We benchmark two key 3D vision tasks using our dataset: novel view synthesis (NVS) and visual relocalization. For NVS, Oxford-Day-and-Night presents a challenging, city-scale setting with dramatic lighting variations, while the inclusion of ground-truth point clouds allows for quantitative evaluation of reconstructed geometry. For visual relocalization, the dataset offers a large set of nighttime query images (7197 in total), which is 37$\times$ larger than the Aachen night split (191 in total), enabling rigorous testing of localization pipelines under extreme conditions. Our experiments demonstrate that current state-of-the-art methods struggle on this dataset, exposing their limitations and underscoring the value of our benchmark.

Our contributions are summarized as follows. **First**, We present *Oxford-Day-and-Night*, a large-scale egocentric dataset featuring five urban scenes captured at multiple times of day with extreme illumination changes, along with their corresponding ground-truth point clouds. **Second**, We demonstrate two primary use cases: (i) a NVS benchmark for city-scale scenes with photometric diversity and geometry reference, and (ii) a visual relocalization benchmark featuring extensive night-time queries for testing robustness under challenging conditions. **Last**, We evaluate state-of-the-art NVS and relocalization methods on our dataset, revealing significant performance drops and underscoring the value of our dataset in future research.

Figure 2: **Example Frames Captured at Different Lighting Conditions.** The severe degradation in visual quality from day to night highlights the difficulty of consistent scene understanding, posing significant challenges for both novel view synthesis (NVS) and visual relocalization methods.

2 Related Work

3D Reconstruction Datasets. Evaluating 3D reconstruction algorithms relies on accurate ground truth 3D models, which are typically obtained using methods such as SLAM, Structure-from-Motion (SfM)[8, 9], Terrestrial Laser Scanners (TLS)[10, 11], or through synthetic data [12]. Early multi-view stereo benchmarks like Middlebury [13] and DTU [11] used structured light scanners on robotic arms to capture small objects, while TLS has been employed for large-scale indoor and outdoor environments in datasets such as EuROC [10], ETH3D [14], Tanks and Temples [15], and ScanNet++[16]. Recent SLAM datasets[17, 18, 19] have extended TLS-based ground truth capture to outdoor settings, often integrating lidar for its robustness to lighting variation. These include environments ranging from natural landscapes [20] to structured urban areas [19, 21, 7].

Despite their geometric precision, many existing datasets depend on heavy, bulky, or sensitive equipment, which limits their ability to capture dynamic, agile camera motions, particularly from an egocentric perspective. Our dataset addresses this gap by integrating TLS-derived ground truth from Oxford Spires [7] with lightweight, wearable ARIA glasses. This combination enables high-fidelity 3D geometry alongside rich egocentric video sequences recorded under diverse motion patterns and lighting conditions, offering a valuable resource for advancing reconstruction under realistic and challenging scenarios.

Novel View Synthesis Datasets. NVS relies on datasets with multi-view images and accurate camera poses to enable the synthesis of novel viewpoints. Early datasets such as ShapeNet [22] and DTU [11] focused on object-centric settings, offering clean imagery and precise poses but limited diversity, often through synthetic renderings or controlled captures. As the field progressed toward more realistic scenarios, datasets like Tanks and Temples [15], ScanNet [9], and RealEstate10K [23] introduced real-world indoor and outdoor scenes with greater complexity in geometry and lighting. LLFF [24] and NeRF [25] established canonical benchmarks for neural rendering, with densely sampled forward-facing views, later extended to unbounded 360° captures in Mip-NeRF 360 [26].

More recent efforts have emphasized scale and diversity: CO3D [27] and Objaverse-XL [28] contribute large-scale object-centric data for real and synthetic domains, while scene-level datasets like Phototourism [29], MegaScenes [30], and DL3DV-10K [31] provide broader appearance variation across lighting, weather, and time. However, a consistent limitation remains, datasets with accurate ground-truth geometry are typically synthetic or limited in scale, while those offering visual diversity often lack high-quality geometry and precise camera calibration. Our dataset addresses this gap by combining large-scale real-world scenes, accurate ground-truth geometry, precise camera poses, and broad day-to-night visual variation, supporting the training and evaluation of generalizable NVS models under realistic conditions.

Visual Relocalization Datasets. Visual relocalization estimates a 6-DoF camera pose within a known environment using image data. Existing datasets for this task are typically categorized as indoor or outdoor, but each comes with notable limitations. Early indoor benchmarks such as 7-Scenes [32] and 12-Scenes [33] focus on small, static spaces with RGB-D input, but their constrained geometry and limited spatial coverage have led to performance saturation. Later efforts like InLoc [4], Indoor6 [34], and the Hyundai Department Store dataset [35] introduced more realistic conditions, featuring textureless surfaces, dynamic elements, and moderate illumination changes, but still fall

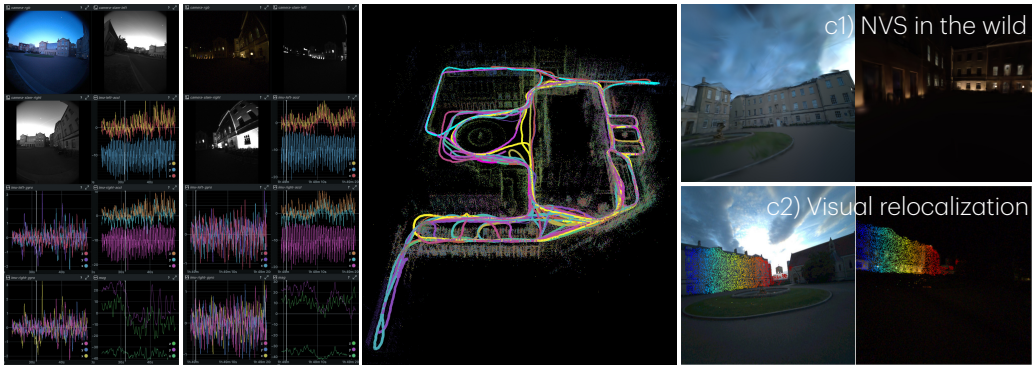


Figure 3: **Data Collection and Processing Pipeline.** At a collection site, our pipeline starts with **a)** capturing 2–10 minute videos using ARIA glasses under varying lighting conditions. These multi-session recordings are processed using **b)** the MPS SLAM system to generate point clouds and camera trajectories in a unified coordinate frame. The colors of the points and trajectories represent different recording sessions; **c)** Leveraging the ARIA data and MPS outputs, we construct two dataset variants for NVS and visual relocalization tasks. Example scene: *Observatory Quarter*.

short in capturing the full variability needed for robust relocalization, particularly under extreme lighting shifts due to their reliance on artificial indoor lighting.

Outdoor datasets offer greater environmental diversity but often compromise in other areas. Vehicle-mounted datasets such as Oxford RoboCar [1], CMU [2], and KITTI [36] span large urban areas and varied conditions across time, weather, and lighting, yet are constrained to road-following, forward-facing viewpoints unsuitable for agile egocentric applications. Handheld datasets like Cambridge Landmarks [3] and InLoc provide more pose variety but limited lighting diversity. Aachen Day-Night [5] introduces night-time scenarios, though with relatively few queries. LaMAR [6] stands out for its egocentric, full-day data collection, but its grayscale headset imagery reduces relevance for color-dependent consumer applications. Overall, existing datasets lack the crucial combination of natural head motion, full-color imagery, and continuous day-long lighting variation required to rigorously evaluate robust, all-day, egocentric visual relocalization systems.

Egocentric Datasets. Popular egocentric datasets [37, 38, 39] have introduced collections of first-person videos in kitchen environments, annotated with fine-grained actions and object interactions. More recent efforts have expanded the scale, diversity, and realism of such data. Ego4D[40] represents a major milestone, offering large-scale, multimodal egocentric video with rich annotations for episodic memory, hand-object interaction, forecasting, and audio-visual understanding. EgoVid-5M[41] supports generative modelling with fine-grained action labels, kinematic data, and textual descriptions tailored for video generation tasks. Meta Project Aria has released several open datasets, including Aria Digital Twin[42], which provides high-fidelity ground truth for objects, environments, and human activities, and Aria Everyday Activities[43], which captures real-world tasks using RGB, stereo IR, IMU, eye-tracking, and audio sensors. EgoExo [44] stands out for offering synchronized egocentric and exocentric video recordings.

While existing datasets support action recognition, question answering, and general video understanding, they often lack 3D geometry, camera motion, and lighting variation, particularly day–night transitions. In contrast, our large-scale dataset targets egocentric 3D vision under varying lighting conditions and includes camera poses and 3D point clouds aligned with ground truth geometry.

3 Oxford Day-and-Night

Our dataset is designed to advance research in egocentric perception under challenging, real-world conditions. It captures large-scale urban environments from a head-mounted, first-person perspective, characterized by natural and agile head movements. A key emphasis is placed on diverse lighting scenarios, with recordings conducted during the day, at dusk, and at night. For each site, the dataset includes high-quality video streams paired with estimated camera poses, along with a semi-dense

point cloud reconstructed via a SLAM system. From these fundamental elements, we derive two dataset variants tailored for NVS and visual relocalization tasks, each optimized for different image and point cloud density requirements.

3.1 Data Collection and Processing

We collect data using Meta ARIA glasses, which record raw sensor streams including IMU, RGB, and grayscale video. To capture varied lighting conditions, day, dusk, and night, sessions are recorded between 4-10pm, covering the natural transition from light to dark. Two individuals wear the glasses casually at each site. Recordings are grouped by location and processed with multi-session Machine Perception Service (MPS) provided by Meta, which estimates per-frame camera poses and semi-dense point clouds unified to a common coordinate frame. Fig. 3 illustrate this data collection process.

Meta ARIA Glasses is a lightweight, sensor-rich device designed for research-grade egocentric data capture. We use recording Profile 2, optimized for RGB video, capturing 20 FPS from both RGB (1408×1408, 110° FOV) and global shutter grayscale SLAM cameras (640×480, 150°×120° FOV), all with fisheye lenses for wide coverage. It also records high-frequency inertial data from dual IMUs (1000Hz and 800Hz). With 2 hours of runtime per charge, ARIA enables efficient, city-scale recording without bulky gear.

MPS is a cloud-based SLAM service that processes grayscale fisheye video and IMU data to generate high-frequency 6-DoF camera trajectories and semi-dense point clouds. It also support multi-session SLAM, which fuses recordings into a single global coordinate frame. This is the core component of our data collection pipeline, ensuring consistent spatial alignment across varying lighting conditions. The resulting globally aligned poses and 3D reconstructions form the backbone of our dataset.

3.2 NVS Dataset Creation

We preprocess video frames, camera poses, and semi-dense point clouds to support NVS tasks through three key steps. **First**, we temporal subsample video frames by $5\times$. As we recorded video at 20 fps, for large-scale scenes like *Bodleian* with 2.8 hours of footage, this results in more than 200,000 frames. While dense image input benefits NVS, such volume demands excessive storage and memory. **Second**, image undistortion, since ARIA uses fisheye lenses and most NVS methods assume a pinhole camera model, we provide both the original and undistorted images. **Third**, point cloud filtering, to improve geometric quality, we filter the semi-dense SLAM point cloud by removing points with high uncertainty, retaining only those with a depth standard deviation below $0.4\ \mathrm{m}$ and inverse depth standard deviation below $0.005\ \mathrm{m}^{-1}$. This results in cleaner geometry suited for NVS systems such as 3DGS [45, 46, 47].

3.3 Visual Relocalization Dataset Creation

We construct our visual relocalization benchmark on top of our NVS dataset. Following established conventions [4, 5], the dataset comprises a set of daytime images with known camera poses (the database) and a separate set of images with unknown poses (the queries). The database images are used either to build a Structure-from-Motion (SfM) model for feature-matching-based relocalization methods [48, 49, 50], or as training data for pose regression-based approaches [51, 52]. Since each scene includes multiple video sequences recorded at different times, many frames depict the same locations from similar viewpoints, leading to redundancy. To promote diversity and reduce overlap, we apply spatial filtering based on the ground-truth camera poses.

We perform spatial filtering by first randomly shuffling all images in a scene and iterating through them to ensure pose diversity. An image is selected if its camera pose lies beyond a spatial radius of θ_{pos} from any previously selected pose; if nearby poses exist, the image is selected only if its orientation differs by at least θ_{ori}. This guarantees diversity in both position and viewpoint. For outdoor scenes (Bodleian Library, H.B. Allen Centre, Keble College, Observatory Quarter), we use thresholds of $\theta_{\text{pos}} = 1.5$ meters and $\theta_{\text{ori}} = 20°$; for the indoor Robotics Institute scene, we adopt stricter thresholds of $\theta_{\text{pos}} = 0.5$ meters and $\theta_{\text{ori}} = 20°$ to reflect its smaller scale.

We apply spatial filtering independently to both daytime and nighttime images. From the filtered daytime set, we construct the visual relocalization benchmark by splitting the images into a database and a daytime query set using a 2:1 ratio. All filtered nighttime images are retained and used solely

Table 1: **Aria MPS Trajectory Quality.** We evaluate the aligned Aria trajectory quality using the point-to-point distance between the aligned MPS point cloud and the ground truth map.

Point Dist. ↓	**Bodleian Library**	**H.B. Allen Centre**	**Keble College**	**Observatory Quarter**	**Robotics Institute**
Mean (cm)	9.7	5.2	9.3	7.1	2.4
Median (cm)	8.0	3.6	7.6	4.6	1.4

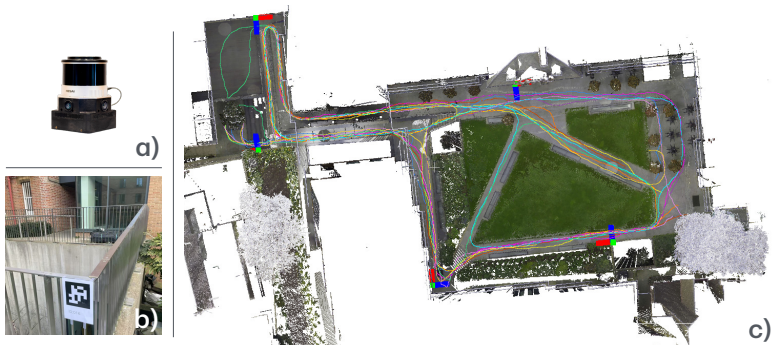


Figure 4: **ARIA MPS Quality Assessment.** We leverage *Frontier* and AprilTag to align ARIA recordings to TLS ground truth map. **a)** The *Frontier* handheld perception unit, equipped with three wide FoV cameras and a 64-channel LiDAR; **b)** A snapshot of an AprilTag; **c)** ARIA trajectories aligned within the ground truth TLS map in the *HBAC* scene. ARIA trajectories colors indicates from different recording sessions.AprilTag poses are highlighted with small colored coordinate frames.

as the nighttime query set, without further splitting. In total, the dataset comprises 5,466 database images, 2,819 daytime query images, and 7,179 nighttime query images. Full details of the filtering procedure are provided in the supplementary material.

3.4 Integration with Ground Truth Map from Oxford Sprires

Oxford Spires [7] is a high-fidelity dataset featuring precisely captured 3D point cloud maps using terrestrial laser scanning (TLS). We complement our Oxford day-and-night dataset with ground truth 3D point clouds from Oxford Spires to provide accurate point cloud reference models as the ground truth maps for benchmarking localization and NVS tasks. These maps were captured with a Leica RTC360 TLS, offering millimeter-level accuracy. We refer readers to [7] for more details.

3.5 ARIA MPS Accuracy Evaluation

To align the Aria world frame with a ground truth map, we developed an automated pipeline. AprilTags [53] are placed along planned paths, and their poses are logged in the ground truth frame using our handheld unit, *Frontier*, which captures images and LiDAR scans: images yield tag poses via AprilTag detection [54], while LiDAR scans are aligned with the map to produce centimeter-accurate trajectories [17]. Using calibrated camera-LiDAR extrinsics [55], all tag poses are expressed in the ground truth frame. We illustrate this process in Fig. 4.

Given the known tag poses in the ground truth map frame, each time a tag appears in the field of view of the Aria glasses, we compute the transformation between the ground truth map frame and the Aria world frame $\mathbf{T}_{\text{map,world}} = \mathbf{T}_{\text{map,tag}}(\mathbf{T}_{\text{aria,tag}})^{-1}(\mathbf{T}_{\text{world,aria}})^{-1}$, where $\mathbf{T}_{\text{map,tag}}$ is a tag pose in the ground truth map frame, and $\mathbf{T}_{\text{aria,tag}}$ is the individual tag detection in the local camera frame of the Aria glasses and $\mathbf{T}_{\text{world,aria}}$ is the corresponding Aria poses at the time of the tag detection in the arbitrary world frame from MPS. We discard detections with poor viewing angles or distances, then average valid transformations to align the closed-loop trajectory and point cloud with the map frame while preserving MPS output consistency.

To further improve MPS-to-GT alignment, the trajectory is refined by registering its associated point cloud to the ground truth map using Iterative Closest Point (ICP) [56]. The resulting trajectory is accurately aligned to the ground truth, with an average point-to-point error of $6.7\,\text{cm}$. Quantitative results are shown in Tab. 1.

3.6 Limitations

The accuracy of our ground-truth camera poses ultimately depends on the multi-session SLAM system provided by Aria MPS, and precisely quantifying SLAM accuracy in a large-scale environment is non-trivial. Traditional methods for obtaining ground-truth poses often rely on additional sensors, such as LiDAR or VICON motion capture systems. However, LiDAR can be unreliable in constrained areas like narrow tunnels, while VICON is impractical for city-scale deployments. Although we use AprilTags localized within our TLS maps for additional reference, both their detection and registration introduce further sources of error into the ground-truth estimation process.

4 Experiments

4.1 Benchmarking Visual Relocalization

Benchmarked Methods. We evaluate a broad range of visual relocalization methods on our dataset, including both *feature matching* (FM) approaches and *scene coordinate regression* (SCR) methods.

Feature Matching Methods. We adopt the HLoc pipeline [48], a widely used benchmark framework for structure-based localization. The pipeline begins by constructing a Structure-from-Motion (SfM) model using the daytime database images, based on pairwise image matching. At test time, the top 50 most visually similar database images are retrieved for each query image using NetVLAD [57], following standard practice. Feature matching is then performed between the query and retrieved images to establish 2D-3D correspondences via triangulated 3D points from the SfM model. Finally, the camera pose of the query image is estimated using PnP-RANSAC.

We evaluate four sparse matching methods within this pipeline: SIFT [58], SuperPoint [64] + SuperGlue [49] (SP+SG), SuperPoint + LightGlue [59] (SP+LG), and DISK [60] + LightGlue (DISK+LG). Additionally, we evaluate three recent dense matching methods: LoFTR [50], RoMA [61], and MASt3R [62], which directly compute dense correspondences between images without requiring keypoint detection.

Scene Coordinate Regression Methods. We also evaluate SCR-based methods, which directly regress 3D scene coordinates from 2D image pixels. Specifically, we test ACE [51], GLACE [52], and R-SCoRe [63]. These methods are trained on our daytime database images to predict per-pixel scene

Table 2: **Visual Relocalization Results on *Day* and *Night* Queries.** We report the percentage of query images correctly localized within three thresholds: (0.25m, 2°), (0.5m, 5°) and (1m, 10°). Results are shown for both feature-matching (FM) and scene coordinate regression (SCR) approaches. For FM approaches, the top 50 images retrieved using NetVLAD [57] are used for matching.

		Visual Relocalization Results on *Day* Queries				
		Bodleian Library	**H.B. Allen Centre**	**Keble College**	**Observatory Quarter**	**Robotics Institute**
FM	SIFT [58]	91.91 / 96.34 / 97.02	75.95 / 81.65 / 82.91	84.98 / 88.78 / 91.06	89.86 / 92.69 / 92.92	70.07 / 73.07 / 74.56
	SP+SG [49]	96.26 / 98.85 / 99.16	96.84 / 98.73 / 99.37	94.68 / 97.34 / 98.10	94.81 / 95.99 / 95.99	89.28 / 90.77 / 91.77
	SP+LG [59]	95.73 / 98.32 / 98.85	96.84 / 98.10 / 98.10	92.78 / 96.20 / 97.15	94.34 / 95.75 / 95.99	88.28 / 89.53 / 90.02
	DISK+LG [60]	94.73 / 97.71 / 98.78	93.67 / 97.47 / 97.47	85.74 / 89.54 / 91.25	92.45 / 95.05 / 95.28	79.80 / 84.79 / 85.79
	LoFTR [50]	96.26 / 98.47 / 99.08	96.84 / 97.47 / 98.10	94.30 / 96.96 / 97.91	94.81 / 95.28 / 95.99	85.04 / 87.03 / 87.53
	RoMA [61]	92.14 / 95.42 / 96.34	87.34 / 93.04 / 94.30	91.83 / 96.20 / 97.15	91.27 / 93.87 / 93.87	85.79 / 87.78 / 88.53
	MASt3R [62]	90.61 / 93.82 / 96.18	94.30 / 98.73 / 99.37	94.68 / 97.91 / 98.86	89.39 / 92.92 / 94.58	84.54 / 90.52 / 94.02
SCR	ACE [51]	0.00 / 0.00 / 0.99	0.63 / 8.86 / 31.65	0.57 / 3.80 / 22.24	0.24 / 8.02 / 25.24	0.00 / 2.24 / 11.72
	GLACE [52]	0.00 / 0.61 / 10.38	0.63 / 4.43 / 34.81	0.19 / 4.18 / 35.93	0.24 / 6.13 / 33.02	0.00 / 0.75 / 29.43
	R-SCoRe [63]	47.71 / 68.32 / 79.62	50.00 / 64.56 / 73.42	60.46 / 75.10 / 85.74	45.52 / 58.02 / 71.23	5.99 / 12.47 / 18.20
		Visual Relocalization Results on *Night* Queries				
		Bodleian Library	**H.B. Allen Centre**	**Keble College**	**Observatory Quarter**	**Robotics Institute**
FM	SIFT [58]	9.70 / 13.72 / 16.09	4.01 / 5.35 / 7.57	0.40 / 0.79 / 1.39	2.38 / 3.35 / 4.55	41.54 / 46.81 / 49.04
	SP+SG [49]	21.63 / 26.55 / 30.78	44.32 / 57.46 / 64.14	10.66 / 13.57 / 17.27	48.14 / 54.40 / 58.05	71.12 / 73.56 / 74.47
	SP+LG [59]	20.46 / 25.28 / 28.78	43.43 / 54.12 / 61.47	9.99 / 14.16 / 18.27	47.91 / 53.50 / 57.68	70.11 / 71.83 / 73.05
	DISK+LG [60]	14.75 / 17.54 / 20.12	9.58 / 11.36 / 14.70	0.53 / 0.79 / 1.06	16.77 / 20.42 / 22.95	53.19 / 57.55 / 60.49
	LoFTR [50]	22.42 / 26.55 / 29.20	41.20 / 52.78 / 58.80	10.39 / 13.63 / 17.01	50.00 / 57.00 / 60.13	66.87 / 70.52 / 72.44
	RoMA [61]	25.24 / 30.98 / 35.18	57.91 / 74.39 / 79.06	14.96 / 22.63 / 30.91	58.94 / 66.92 / 70.79	72.34 / 75.38 / 76.60
	MASt3R [62]	15.65 / 17.54 / 19.98	49.22 / 59.02 / 66.59	12.24 / 16.08 / 19.66	48.66 / 54.47 / 59.91	65.65 / 72.95 / 77.00
SCR	ACE [51]	0.00 / 0.00 / 0.00	0.00 / 0.00 / 0.00	0.00 / 0.00 / 0.00	0.00 / 0.00 / 0.00	0.10 / 0.10 / 0.91
	GLACE [52]	0.00 / 0.00 / 0.03	0.00 / 0.00 / 0.00	0.00 / 0.00 / 0.99	0.00 / 0.00 / 0.00	0.00 / 0.00 / 8.21
	R-SCoRe [63]	2.72 / 7.57 / 13.10	5.57 / 11.58 / 23.61	0.20 / 0.99 / 1.92	3.06 / 7.75 / 13.34	2.13 / 6.08 / 9.52

Table 3: **Accuracy of RoMA on the Visual Relocalization Dataset Using the HLoc Pipeline with Various Image Retrieval Methods on *Night* Queries.** We report the percentage of correctly localized query images within thresholds of (0.25m, 2°), (0.5m, 5°), and (1m, 10°).

	Bodleian Library	H.B. Allen Centre	Keble College	Observatory Quarter	Robotics Institute
RoMA + NetVLAD 50 [57]	25.24 / 30.98 / 35.18	57.91 / 74.39 / 79.06	14.96 / 22.63 / 30.91	58.94 / 66.92 / 70.79	72.34 / 75.38 / 76.60
RoMA + DIR 50 [65, 66]	33.46 / 39.10 / 42.30	55.46 / 72.16 / 81.51	16.48 / 23.36 / 28.33	56.33 / 65.28 / 69.90	74.27 / 78.52 / 79.84
RoMA + OpenIBL 50 [67]	43.95 / 51.24 / 54.50	60.36 / 73.50 / 78.62	18.66 / 27.47 / 35.67	59.09 / 66.32 / 70.19	71.73 / 75.18 / 76.19
RoMA + MegaLoc 50 [68]	70.25 / 79.09 / 82.22	66.37 / 81.51 / 87.31	31.50 / 42.22 / 51.82	72.06 / 80.92 / 84.50	78.22 / 82.27 / 83.69
RoMA + GT Pose 20	80.57 / 89.58 / 92.88	68.82 / 81.51 / 85.75	41.50 / 57.78 / 71.01	80.55 / 87.48 / 90.98	84.50 / 89.36 / 90.48

coordinates. At inference time, they provide dense 2D-3D correspondences for each query image, from which the camera pose is estimated using PnP-RANSAC.

Results. We summarize the results of the evaluation on daytime and nighttime queries in Tab. 2, where we report the percentage of query frames with pose errors of within three thresholds: (0.25m, 2°), (0.5m, 5°) and (1m, 10°). Our experiments are conducted using 48GB NVIDIA RTX A6000 GPUs, with mapping times ranging from a few minutes to several hours, depending on the relocalization method and scene complexity.

Analysis. We observe that feature-matching (FM) methods significantly outperform scene coordinate regression (SCR) approaches on both daytime and nighttime queries, with some SCR methods failing entirely at night. This is consistent with the Aachen Day-Night benchmark [5], where SCR methods generally struggle in large-scale environments and under severe illumination changes. The performance gap between day and night is even more pronounced in our dataset, due to increased lighting variability that makes regressing consistent 3D coordinates especially difficult. FM methods perform well on daytime queries, likely because the query and database trajectories are similar, reducing viewpoint variation, but their performance drops notably at night, highlighting the challenge of low-light conditions. Among them, RoMA achieves the highest overall accuracy and is thus used for a deeper analysis of image retrieval, a factor often overlooked in favor of default choices like NetVLAD in the HLoc pipeline.

To evaluate retrieval quality, we pair RoMA with four retrieval methods: NetVLAD, DIR [65, 66], OpenIBL [67], and MegaLoc [68], retrieving the top 50 database images. A quasi-upper bound is also included using the 20 nearest images based on ground-truth poses. As shown in Tab. 3, more advanced retrieval methods significantly boost performance, with RoMA exceeding 80% accuracy under the strictest threshold (0.25m, 2°) when paired with ground-truth retrieval, indicating that retrieval, not matching, is the main accuracy bottleneck. However, RoMA's major limitation is its runtime: approximately 1 second per image pair, about 30× slower than SuperPoint+LightGlue (~0.03s). These findings point to two important research directions enabled by our dataset: (1) enhancing image retrieval under challenging conditions and (2) accelerating high-accuracy matchers like RoMA, where speed is the limiting factor.

4.2 Benchmarking NVS

Benchmarked Methods. We evaluate two state-of-the-art in-the-wild neural view synthesis (NVS) methods: Splatfacto-W [46] and Gaussian-Wild [47]. To train these models on our NVS dataset, we apply two preprocessing steps. First, we further subsample the image collections for each scene to approximately 2,500 images to ensure manageable CPU memory usage. Second, we perform voxel downsampling of the semi-dense point cloud using a voxel size of $0.1m$ ($0.2m$ for the *Bodleian* scene). Since GPU memory consumption is proportional to the number of initial 3D points, this step helps keep GPU usage under 80GB for these NVS systems.

Result. We follow the standard convention of selecting every 8th image as a test image and report image quality using PSNR and LPIPS metrics. For geometry evaluation, we utilize the ground truth 3D point clouds from Oxford-Spires [7] and measure the point-to-point distance between the centers of the 3DGS Gaussian primitives and the ground truth 3D maps. We compute the point to point distance using CloudCompare. The results are presented in Tab. 4 and Fig. 5.

In this experiment, Splatfacto-W outperforms Gaussian-Wild on the *H.B. Allen Centre* scene but underperforms on the remaining four scenes. However, as indicated by the LPIPS scores, both methods exhibit limited performance across these four scenes. This is primarily due to the large-scale

Table 4: **3DGS In-the-Wild Results.** We report image rendering and geometry quality using the following metrics: PSNR (↑) / LPIPS (↓) / point-to-point distance (↓). The 3DGS geometry is derived by extracting the centers of all Gaussian primitives, with point-to-point distance (meter) computed against the ground truth laser-scanned point cloud. Symbol "-" denotes the system produces a degenerated point cloud (less than 2000 gaussians after training).

Method	**Bodleian Library**	**H.B. Allen Centre**	**Keble College**	**Observatory Quarter**	**Robotics Institute**
Splatfacto-W [46]	25.98 / 0.60 / -	25.65 / 0.59 / 0.75	27.96 / 0.59 / -	25.83 / 0.63 / 0.36	22.73 / 0.61 / 0.42
Gaussian-Wild [47]	28.38 / 0.56 / 1.44	24.94 / 0.59 / 1.48	30.92 / 0.56 / 0.69	28.57 / 0.60 / 0.69	25.05 / 0.57 / 0.76

Figure 5: **NVS In-the-Wild Results in the H.B. Allen Centre (*top*) and Bodleian Library (*bottom*).** Compared to Gaussian-Wild, Splatfacto-W performs better at the H.B. Allen Centre but fails at the Bodleian Library. Although Gaussian-Wild produces some renderings with recognizable content, the overall quality is limited. These results highlight that current state-of-the-art NVS-in-the-wild methods still face significant challenges in large-scale environments with dramatic lighting variations.

nature of the dataset and the extreme lighting variations, ranging from daylight to poorly illuminated night conditions.

Discussion. *First*, further downsampling: as described in Sec. 3.2, we initially downsampled videos and filtered noisy 3D points to create an NVS dataset suitable for *future* use. However, *current* state-of-the-art in-the-wild 3DGS systems still struggle with this data scale. Therefore, we apply more aggressive temporal subsampling and spatial downsampling of the point clouds to ensure feasibility. *Second*, PSNR fails to reflect image quality. In Tab. 4, both methods achieve PSNR > 25 across most scenes, yet high LPIPS values > 0.5 reveal poor visual quality. Similar issues are observed with SSIM, as shown in the supplementary material. *Third*, point-to-point distance may offer a rough indication of NVS performance, but only when basic shapes are preserved and points are not aggressively culled during 3DGS optimization. More details can be found in supplementary.

5 Conclusion

Oxford Day-and-Night fills a crucial gap in egocentric 3D vision research by providing a large-scale, lighting-diverse dataset explicitly designed for challenging outdoor conditions, including nighttime scenarios. Through its combination of rich sensor data, robust multi-session SLAM annotations, and alignment with high-fidelity ground-truth geometry, the dataset enables rigorous benchmarking of novel view synthesis and visual relocalization methods at city scale. Our experiments reveal substantial performance degradation of current state-of-the-art approaches, particularly under extreme lighting changes, underscoring both the difficulty of the tasks and the value of our benchmarks. By exposing these limitations, Oxford Day-and-Night offers a powerful platform to drive progress in robust, all-day egocentric perception systems.

Acknowledgement. This research is supported by multiple funding sources, including an ARIA research gift grant from Meta Reality Lab, a Royal Society University Research Fellowship (Fallon), the EPSRC C2C Grant EP/Z531212/1 (TRO), and a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) under grant number RS 2024 00461409.

References

[1] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *IJRR*, 2017. 1, 4

[2] Hernán Badino, Daniel Huber, and Takeo Kanade. Visual topometric localization. In *IEEE Intelligent Vehicles Symposium*, 2011. 1, 4

[3] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015. 1, 4

[4] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *CVPR*, 2018. 1, 3, 5

[5] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *CVPR*, 2018. 1, 4, 5, 8, 15

[6] Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L. Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, and Marc Pollefeys. LaMAR: Benchmarking localization and mapping for augmented reality. In *ECCV*, 2022. 1, 4

[7] Yifu Tao, Miguel Ángel Muñoz-Bañón, Lintong Zhang, Jiahao Wang, Lanke Frank Tarimo Fu, and Maurice Fallon. The oxford spires dataset: Benchmarking large-scale lidar-visual localisation, reconstruction and radiance field methods. *IJRR*, 2025. 1, 2, 3, 6, 8

[8] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *3DV*, 2017. 3

[9] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017. 3

[10] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The EuRoC micro aerial vehicle datasets. *IJRR*, 2016. 3

[11] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *IJCV*, 2016. 3

[12] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *ICRA*, 2014. 3

[13] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, volume 1, pages 519–528. IEEE, 2006. 3

[14] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017. 3

[15] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and Temples: Benchmarking large-scale scene reconstruction. *ToG*, 2017. 3

[16] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *ICCV*, pages 12–22, 2023. 3

[17] Milad Ramezani, Yiduo Wang, Marco Camurri, David Wisth, Matias Mattamala, and Maurice Fallon. The Newer College dataset: Handheld lidar, inertial and vision with ground truth. In *IROS*, 2020. 3, 6

[18] Lintong Zhang, Michael Helmberger, Lanke Frank Tarimo Fu, David Wisth, Marco Camurri, Davide Scaramuzza, and Maurice Fallon. Hilti-Oxford dataset: A millimeter-accurate benchmark for simultaneous localization and mapping. *RAL*, 2022. 3

[19] Hexiang Wei, Jianhao Jiao, Xiangcheng Hu, Jingwen Yu, Xupeng Xie, Jin Wu, Yilong Zhu, Yuxuan Liu, Lujia Wang, and Ming Liu. FusionPortableV2: A unified multi-sensor dataset for generalized SLAM across diverse platforms and scalable environments. *IJRR*, 2024. 3

[20] Yuanzhi Liu, Yujia Fu, Minghui Qin, Yufeng Xu, Baoxin Xu, Fengdong Chen, Bart Goossens, Poly Z.H. Sun, Hongwei Yu, Chun Liu, Long Chen, Wei Tao, and Hui Zhao. BotanicGarden: A high-quality dataset for robot navigation in unstructured natural environments. *RAL*, 2024. 3

[21] Thien-Minh Nguyen, Shenghai Yuan, Thien Hoang Nguyen, Pengyu Yin, Haozhi Cao, Lihua Xie, Maciej Wozniak, Patric Jensfelt, Marko Thiel, Justin Ziegenbein, and Noel Blunder. MCD: Diverse large-scale multi-campus dataset for robot perception. In *CVPR*, 2024. 3

[22] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 3

[23] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ToG*, 2018. 3

[24] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ToG*, 2019. 3

[25] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 3

[26] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022. 3

[27] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *ICCV*, 2021. 3

[28] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *NeurIPS*, 2023. 3

[29] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *IJCV*, 2021. 3

[30] Joseph Tung, Gene Chou, Ruojin Cai, Guandao Yang, Kai Zhang, Gordon Wetzstein, Bharath Hariharan, and Noah Snavely. Megascenes: Scene-level view synthesis at scale. In *ECCV*, 2024. 3

[31] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*, 2024. 3

[32] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *CVPR*, 2013. 3

[33] Julien Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *3DV*, 2016. 3

[34] Tien Do, Ondrej Miksik, Joseph DeGol, Hyun Soo Park, and Sudipta N Sinha. Learning to detect scene landmarks for camera localization. In *CVPR*, 2022. 3

[35] Donghwan Lee, Soohyun Ryu, Suyong Yeon, Yonghan Lee, Deokhwa Kim, Cheolho Han, Yohann Cabon, Philippe Weinzaepfel, Nicolas Guérin, Gabriela Csurka, et al. Large-scale localization datasets in crowded indoor spaces. In *CVPR*, 2021. 3

[36] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013. 4

[37] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. The epic-kitchens dataset: Collection, challenges and baselines. *TPAMI*, 2020. 4

[38] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *ECCV*, 2018. 4

[39] Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Fan Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, et al. Introducing hot3d: An egocentric dataset for 3d hand and object tracking. In *CVPR*, 2025. 4

[40] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. 4

[41] Xiaofeng Wang, Kang Zhao, Feng Liu, Jiayu Wang, Guosheng Zhao, Xiaoyi Bao, Zheng Zhu, Yingya Zhang, and Xingang Wang. Egovid-5m: A large-scale video-action dataset for egocentric video generation. *arXiv preprint arXiv:2411.08380*, 2024. 4

[42] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *ICCV*, 2023. 4

[43] Zhaoyang Lv, Nicholas Charron, Pierre Moulon, Alexander Gamino, Cheng Peng, Chris Sweeney, Edward Miller, Huixuan Tang, Jeff Meissner, Jing Dong, et al. Aria everyday activities dataset. *arXiv preprint arXiv:2402.13349*, 2024. 4

[44] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *CVPR*, 2024. 4

[45] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ToG*, 2023. 5

[46] Congrong Xu, Justin Kerr, and Angjoo Kanazawa. Splatfacto-w: A nerfstudio implementation of gaussian splatting for unconstrained photo collections. *arXiv preprint arXiv:2407.12306*, 2024. 5, 8, 9, 18

[47] Dongbin Zhang, Chuming Wang, Weitao Wang, Peihao Li, Minghan Qin, and Haoqian Wang. Gaussian in the wild: 3d gaussian splatting for unconstrained image collections. In *ECCV*, 2024. 5, 8, 9, 18

[48] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 5, 7, 16

[49] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 5, 7

[50] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021. 5, 7

[51] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *CVPR*, 2023. 5, 7

[52] Fangjinhua Wang, Xudong Jiang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. Glace: Global local accelerated coordinate encoding. In *CVPR*, 2024. 5, 7

[53] Edwin Olson. AprilTag: A robust and flexible visual fiducial system. In *ICRA*, 2011. 6

[54] John Wang and Edwin Olson. AprilTag 2: Efficient and robust fiducial detection. In *IROS*, 2016. 6

[55] Lanke Frank Tarimo Fu, Nived Chebrolu, and Maurice Fallon. Extrinsic calibration of camera to lidar using a differentiable checkerboard model. In *IROS*, 2023. 6

[56] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *TPAMI*, 1992. 6

[57] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016. 7, 8

[58] David G Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 7

[59] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *CVPR*, 2023. 7

[60] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. In *NeurIPS*, 2020. 7

[61] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *CVPR*, 2024. 7

[62] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *ECCV*, 2024. 7

[63] Xudong Jiang, Fangjinhua Wang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. R-score: Revisiting scene coordinate regression for robust large-scale visual localization. In *CVPR*, 2025. 7

[64] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. 7

[65] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. End-to-end learning of deep visual representations for image retrieval. *IJCV*, 2017. 8

[66] J. Revaud, J. Almazan, R.S. Rezende, and C.R. de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *ICCV*, 2019. 8

[67] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-supervising fine-grained region similarities for large-scale image localization. In *ECCV*, 2020. 8

[68] Gabriele Berton and Carlo Masone. Megaloc: One retrieval to place them all. *arXiv preprint arXiv:2502.17237*, 2025. 8

Seeing in the Dark: Benchmarking Egocentric 3D Vision with the Oxford Day-and-Night Dataset (Supplementary)

https://oxdan.active.vision/

A Full Dataset Statistics

We collected our dataset across five locations in Oxford by walking while wearing ARIA glasses. The data collection took place over the course of one month. During this period, two collectors wore ARIA glasses and walked randomly within each collection site. In total, the walking trajectory spans 30 kilometers, includes 7 hours of walking, and covers an area of 40,000 m^2.

Detailed dataset statistics are provided in Tab. 5. Notably, our dataset offers a well-balanced distribution of day and night recordings, with an approximately 1:1 ratio. The covered areas and walking trajectories are visualized in Figs. 6 and 7.

Table 5: **Dataset Statistics.** We present a summary of the number of frames in the recorded videos, the NVS data variant (obtained by subsampling the video by $5\times$), and the visual relocalization data variant (with additional spatial subsampling and splitting into database, daytime queries, and nighttime queries). We also report the recording durations, trajectory lengths, and mapped area sizes.

Scene	# Video Fr	# NVS Img	# Visual Reloc Img			Duration (hh:mm)			Trajectory Len (m)			Area (m^2)
	D & N	D & N	DB	Day Q	Night Q	Day	Night	D & N	Day	Night	D & N	D & N
Bodleian Lib.	205405	41081	2542	1310	2908	01:32	01:18	02:50	7170	5617	12787	25939
H.B. Allen Cen.	29340	5868	305	158	449	00:13	00:10	00:24	975	765	1740	1271
Keble College	112205	22441	1020	526	1511	00:46	00:46	01:33	3574	3400	6974	5709
Obs. Quarter	87210	17442	821	424	1342	00:34	00:38	01:12	2853	3050	5903	5950
Robotics Inst.	57590	11518	778	401	987	00:25	00:22	00:47	1249	1030	2279	600
Total	491750	98350	5466	2819	7197	03:32	03:16	06:48	15822	13862	29685	39469

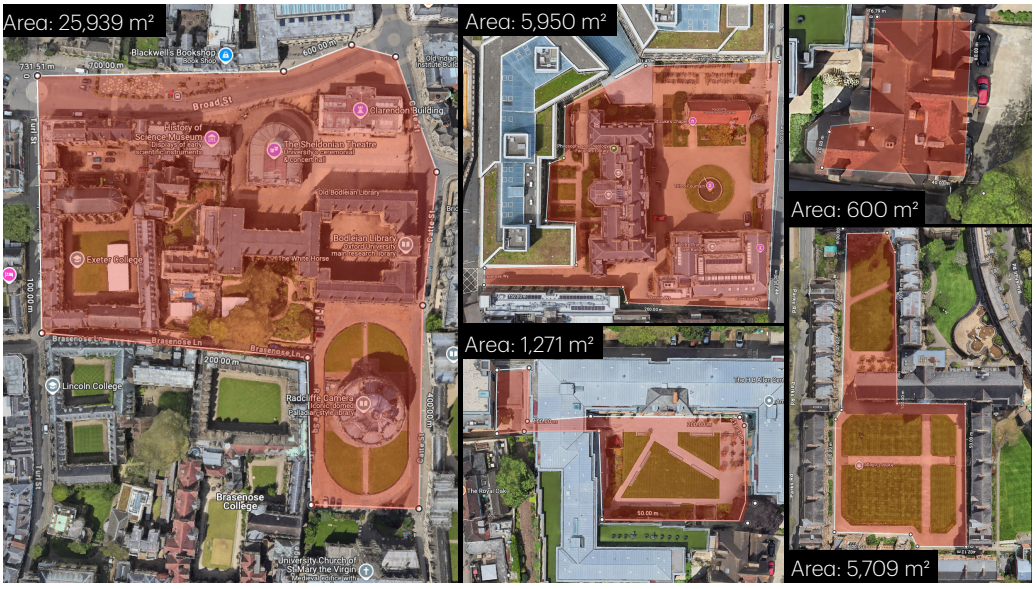


Figure 6: Our dataset covers 40,000 m^2 area.

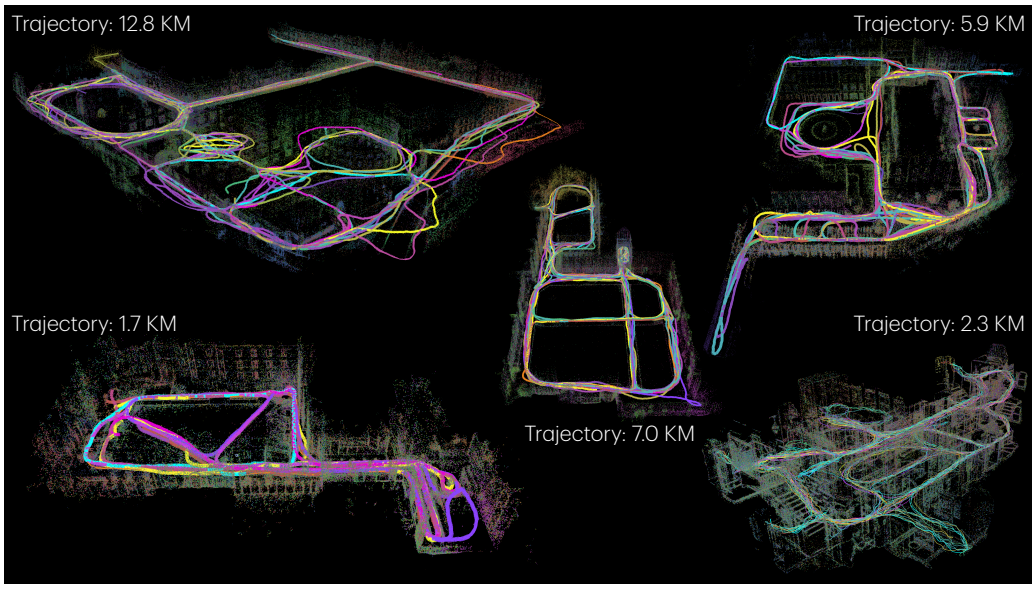


Figure 7: Our dataset spans 30 kilometers of walking trajectory.

B Image Variants

ARIA glasses are equipped with fisheye lenses, resulting in fisheye distortion in the original recordings. To facilitate the use of our dataset, we undistort these images using two different pinhole camera configurations. We provide both the original fisheye images and the undistorted versions. The three image variants are visualized in Fig. 8.

Figure 8: We provide three image types: the original fisheye, a *Max FOV* undistorted version with wider coverage and black borders (with a valid pixel mask provided), and an *All Valid* version with no black borders but a smaller field of view for easier use.

C Additional Details on Visual Relocalization Dataset

Spatial Filtering. We provide additional details about our visual relocalization dataset. In Algorithm 1, we present the pseudo-code for the spatial filtering algorithm used to eliminate redundant images when generating the database, daytime query, and nighttime query splits. For outdoor scenes (Bodleian Library, H.B. Allen Centre, Keble College, Observatory Quarter), we use thresholds of $\theta_{\text{pos}} = 1.5$ meters and $\theta_{\text{ori}} = 20°$; for the indoor Robotics Institute scene, we adopt stricter thresholds of $\theta_{\text{pos}} = 0.5$ meters and $\theta_{\text{ori}} = 20°$ to reflect its smaller scale. Notably, even after applying strong spatial filtering, our dataset includes 7,197 nighttime query images, 37 times more than the 191 nighttime queries in the Aachen Day-Night dataset [5]. Full statistics are summarized in Tab. 5.

Algorithm 1: Spatial Filtering of Camera Poses

Require: Image list I with poses (p_i, R_i), thresholds $\theta_{\text{pos}}, \theta_{\text{ori}}$
Ensure: Filtered image list S

Shuffle I; initialize $S \leftarrow [\,]$, cache $C \leftarrow [\,]$
for each image i in I **do**
  $N \leftarrow \{(p_j, R_j) \in C \mid \|p_i - p_j\| < \theta_{\text{pos}}\}$
  if $N = \emptyset$ or $\forall (p_j, R_j) \in N, \angle(R_i, R_j) > \theta_{\text{ori}}$ **then**
    Append i to S, append (p_i, R_i) to C
  end if
end for
return S

Coverage of Nighttime Queries Across Distance and Rotation Thresholds. To further illustrate the challenge posed by our benchmark, Fig. 9 plots the percentage of nighttime queries that have at least one daytime database image within a specified spatial and angular threshold. Our dataset spans a wide spectrum of difficulty levels, including a particularly challenging subset: nighttime queries that are more than 5 meters and 50° away from any corresponding database image. These difficult cases account for approximately 10% of the nighttime queries. This diversity enables a more fine-grained evaluation of relocalization methods, allowing the community to assess performance across both easy and hard cases.

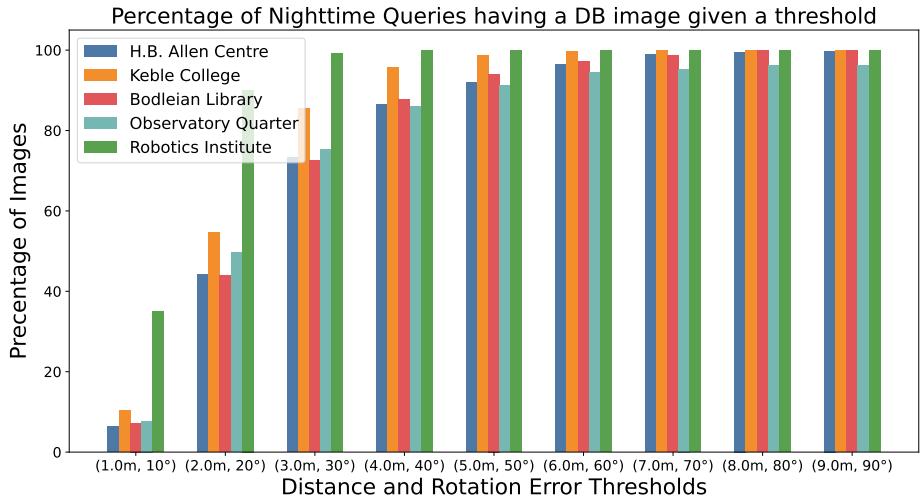
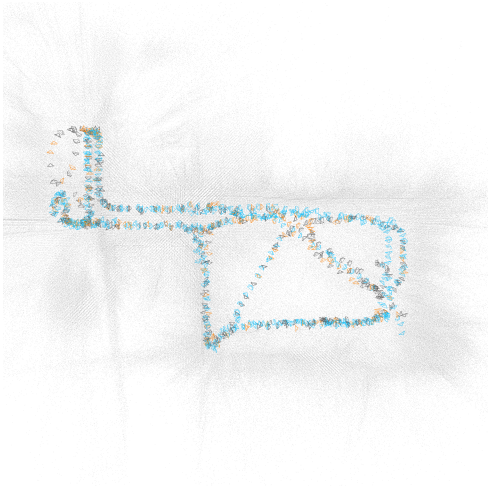
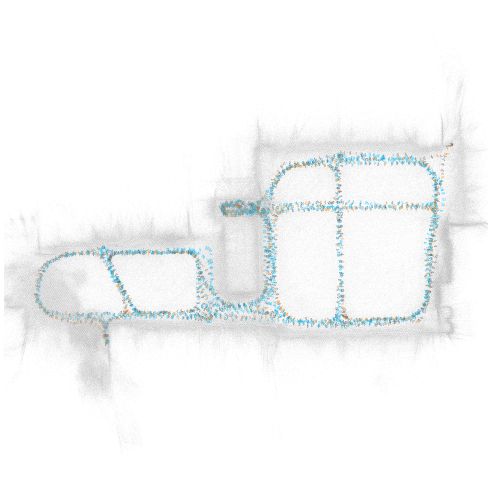


Figure 9: The percentage of nighttime queries that have a database image given a spatial and orientation threshold.

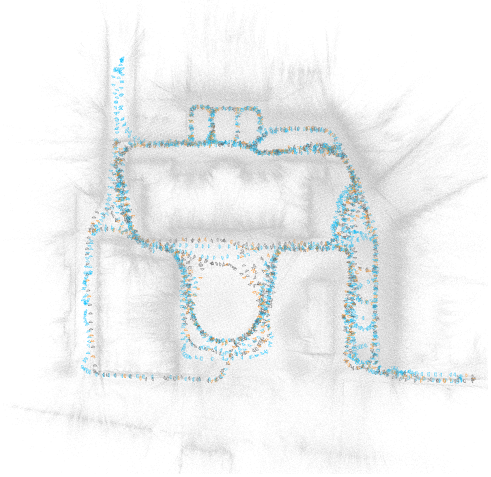
Database Creation, COLMAP, and HLoc. We structure our relocalization dataset using simple image lists, where each split (database, daytime queries, and nighttime queries) corresponds to a text file containing the image filenames relative to the image directory. To facilitate seamless integration with the HLoc Toolbox [48], we also provide a COLMAP model for the database images, generated using ARIA MPS output poses. Specifically, for each database image, we project the 3D point cloud of the scene onto the image plane using the corresponding ground-truth camera pose. We then apply a series of filtering steps to remove invalid projections: depth filtering, image boundary checks, and z-buffer visibility checks. From the valid set of projections, we randomly sample 3,000 2D-3D correspondences per image. Using this information, we construct the `images.bin`, `cameras.bin`, and `points3D.bin` files following COLMAP standard format. Note that our COLMAP model does not incorporate explicit occlusion reasoning. As a result, we do not recommend using it directly for PnP-RANSAC without additional filtering or refinement. However, this limitation does not affect integration with the HLoc Toolbox, as it does not rely on the database point cloud. We provide the visualization of the distribution of database, daytime, and nighttime camera poses in each scene in Fig. 10.



(a) H.B. Allen Centre

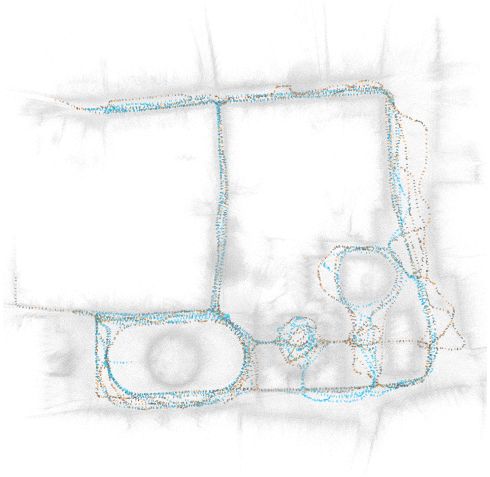


(b) Keble College



(c) Observatory Quarter

(d) Robotics Institute



(e) Bodleian Library

Figure 10: **Camera poses for visual relocalization in each scene.** The cameras of database images are in **black**; the cameras of day query images are in **orange** and the cameras of night query images are in **blue**.

D Additional Results on NVS Dataset

Image Quality. We provide additional NVS evaluation results in Tab. 6 and Fig. 11, which complement the findings presented in Tab. 4 and Fig. 5. Specifically, Table 6 highlights that both 3DGS in-the-wild methods exhibit limited NVS performance on our dataset, as indicated by high LPIPS values. Note that PSNR and SSIM values do not capture this performance degradation.

Table 6: **3DGS In-the-Wild Image Quality.** We report image rendering quality in PSNR (↑) / LPIPS (↓) / SSIM (↑). This table complements Tab. 4 and Fig. 5 by providing additional SSIM scores.

Method	**Bodleian Library**	**H.B. Allen Centre**	**Keble College**	**Observatory Quarter**	**Robotics Institute**
Splatfacto-W [46]	25.98 / 0.60 / 0.79	25.65 / 0.59 / 0.81	27.96 / 0.59 / 0.78	25.83 / 0.63 / 0.78	22.73 / 0.61 / 0.81
Gaussian-Wild [47]	28.38 / 0.56 / 0.86	24.94 / 0.59 / 0.86	30.92 / 0.56 / 0.84	28.57 / 0.60 / 0.86	25.05 / 0.57 / 0.88

Geometry. Figure 11 visualizes the centers of Gaussian primitives after Splatfacto-W training. During this process, the initialized point cloud is culled to a reasonable density in the *H.B. Allen Centre* and *Observatory Quarter*. In contrast, the same culling procedure results in degenerate representations in the *Bodleian Library* and *Keble College* scenes, possibly due to the larger spatial extent of the *Bodleian Library* and the more extreme lighting variations present in *Keble College*.

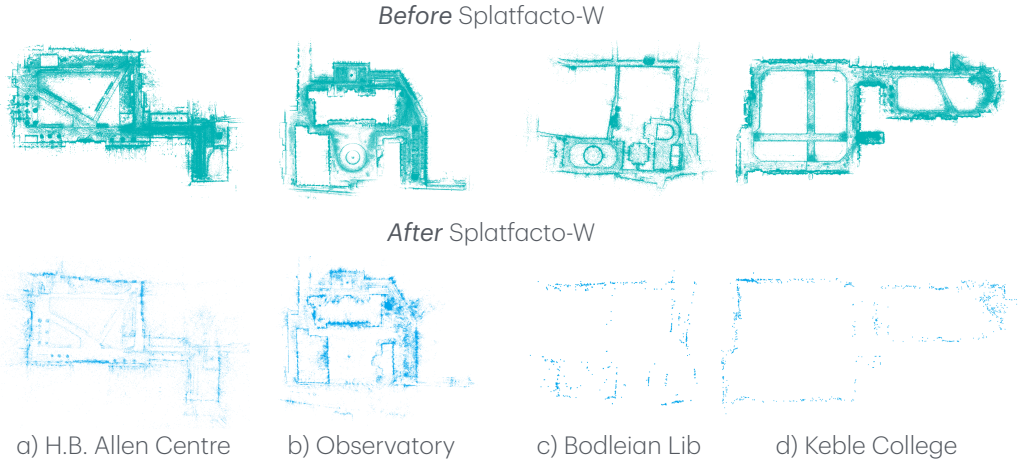


Figure 11: **Visualization of 3D Geometry.** In c) and d), less than 2000 Gaussian primitives remain after the culling process during training. This may be due to limited capability in handling large-scale scenes and dramatic light variations, resulting in a degenerated case for 3DGS rendering.

Overall, our experiments demonstrate that current 3DGS in-the-wild methods continue to face significant challenges in large-scale scenes with dramatic lighting variations.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Seeing in the Dark: Benchmarking Egocentric 3D Vision with the Oxford Day-and-Night Dataset
Publication Status	Submitted for Publication
Publication Details	Wang, Z., Bian, W., Li, X., Tao, Y. Wang, J., Fallon, M, and Prisacariu, V. (2025). "Seeing in the Dark: Benchmarking Egocentric 3D Vision with the Oxford Day-and-Night Dataset". Conference on Neural Information Processing Systems (NeurIPS) (Under Review)

Student Confirmation

Student Name:	Jianeng Wang		
Contribution to the Paper	My contributions to the paper were: • Generated the pseudo ground truth trajectories for all sequences within the corresponding ground truth scan • Wrote the paper with co-authors		
Signature	Jianeng Wang	Date	2025/06/10

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Maurice Fallon		
Supervisor comments Jianeng contributed the system for determining the ground truth location of the sensor system used in this project. This revised a tool he previously developed for his main research project. He did not lead or instigate this project - but this key contribution was valuable and required overall.		
Signature Maurice Fallon	Date	2025/06/23

This completed form should be included in the thesis, at the end of the relevant chapter.

References

[1] M. Agrawal and K. Konolige. "Real-time Localization in Outdoor Environments using Stereo Vision and Inexpensive GPS". In: *18th International Conference on Pattern Recognition (ICPR'06)*. Vol. 3. 2006, pp. 1063–1068. DOI: 10.1109/ICPR.2006.962 (page 31).

[2] Ali H.A. Al-dabbagh and Renaud Ronsse. "A review of terrain detection systems for applications in locomotion assistance". In: *Robot. Auton. Syst.* 133 (2020), p. 103628. ISSN: 0921-8890 (page 2).

[3] I. Alzugaray and M. Chli. "HASTE: multi-Hypothesis Asynchronous Speeded-up Tracking of Events". In: *British Machine Vision Conf. (BMVC)*. 2020. URL: https://www.bmvc2020-conference.com/assets/papers/0744.pdf (pages 37, 64, 104).

[4] Ignacio Alzugaray and Margarita Chli. "ACE: An efficient asynchronous corner tracker for event cameras". In: *IEEE Intl. Conf. on 3D Vision*. 2018, pp. 653–661 (pages 37, 110, 122).

[5] Ignacio Alzugaray and Margarita Chli. "Asynchronous Corner Detection and Tracking for Event Cameras in Real Time". In: *IEEE Robot. Autom. Lett. (RA-L)* 3.4 (2018) (pages 37, 40).

[6] Ignacio Alzugaray and Margarita Chli. "Asynchronous Corner Detection and Tracking for Event Cameras in Real Time". In: *IEEE Robot. Autom. Lett. (RA-L)* 3.4 (2018), pp. 3177–3184. DOI: 10.1109/LRA.2018.2849882 (pages 104, 111, 112, 116, 117, 121).

[7] Sean Anderson and Timothy D. Barfoot. "Full STEAM ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on SE(3)". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2015 (pages 66, 105).

[8] Taylor Apgar, Patrick Clary, Kevin R. Green, Alan Fern, and Jonathan W. Hurst. "Fast Online Trajectory Optimization for the Bipedal Robot Cassie". In: *Robotics: Science and Systems (RSS)* (2018). URL: https://api.semanticscholar.org/CorpusID:46992008 (page 42).

[9] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40.6 (2018), pp. 1437–1451. DOI: 10.1109/TPAMI.2017.2711011 (page 34).

[10] Daniel Barath, Luca Cavalli, and Marc Pollefeys. "Learning to Find Good Models in RANSAC". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 2022, pp. 15723–15732. DOI: 10.1109/CVPR52688.2022.01529 (page 36).

[11] Timothy D. Barfoot. *State Estimation for Robotics*. Cambridge: Cambridge University Press, 2017 (pages 11, 12, 15, 17, 25).

[12] Jonathan T Barron. "A general and adaptive robust loss function". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 2019, pp. 4331–4339 (page 36).

[13] Chiara Bartolozzi, Giacomo Indiveri, and Elisa Donati. "Embodied neuromorphic intelligence". In: *Nature communications* 13.1 (2022), p. 1024 (page 41).

[14] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. "ARKitScenes - A Diverse Real-World Dataset for 3D Indoor Scene Understanding Using Mobile RGB-D Data". In: *Intl. Conf. on Neural Information Processing Systems (NeurIPS)*. 2021. URL: https://openreview.net/forum?id=tjZjv_qh_CE (page 1).

[15] Hriday Bavle, Jose Luis Sanchez-Lopez, Muhammad Shaheer, Javier Civera, and Holger Voos. "S-Graphs+: Real-time localization and mapping leveraging hierarchical representations". In: *IEEE Robot. Autom. Lett. (RA-L)* 8.8 (2023), pp. 4927–4934 (pages 1, 47).

[16] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi. "Event-based visual flow". In: *IEEE Transactions on Neural Networks and Learning Systems* 25.2 (2013), pp. 407–417 (page 110).

[17] Aleksandra V. Birn-Jeffery, Christian M. Hubicki, Yvonne Blum, Daniel Renjewski, Jonathan W. Hurst, and Monica A. Daley. "Don't break a leg: running birds from quail to ostrich prioritise leg safety and economy on uneven terrain". In: *Journal of Experimental Biology* 217.21 (Nov. 2014), pp. 3786–3796. ISSN: 0022-0949. DOI: 10.1242/jeb.102640. eprint: https://journals.biologists.com/jeb/article-pdf/217/21/3786/1887134/3786.pdf. URL: https://doi.org/10.1242/jeb.102640 (page 42).

[18] Michael Bloesch, Michael Burri, Hannes Sommer, Roland Siegwart, and Marco Hutter. "The Two-State Implicit Filter Recursive Estimation for Mobile Robots". In: *IEEE Robot. Autom. Lett. (RA-L)* 3.1 (2018), pp. 573–580. DOI: 10.1109/LRA.2017.2776340 (pages 44, 45).

[19] Michael Bloesch, Christian Gehring, Péter Fankhauser, Marco Hutter, Mark A. Hoepflinger, and Roland Siegwart. "State estimation for legged robots on unstable and slippery terrain". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2013, pp. 6058–6064. DOI: 10.1109/IROS.2013.6697236 (page 44).

[20] Michael Bloesch, Marco Hutter, Mark Hoepflinger, Stefan Leutenegger, Christian Gehring, C. David Remy, and Roland Siegwart. "State Estimation for Legged Robots - Consistent Fusion of Leg Kinematics and IMU". In: *Robotics: Science and Systems (RSS)*. Sydney, Australia, 2012. DOI: 10.15607/RSS.2012.VIII.003 (page 44).

[21] Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. "Robust visual inertial odometry using a direct EKF-based approach". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2015, pp. 298–304. DOI: 10.1109/IROS.2015.7353389 (pages 33, 50).

[22] J. Bloomenthal and C. Bajaj. *Introduction to Implicit Surfaces*. Morgan Kaufmann series in computer graphics and geometric modeling. Elsevier Science, 1997. ISBN: 9781558602335. URL: https://books.google.co.uk/books?id=T3SSqIVnS4YC (page 22).

[23] Rowan Border, Nived Chebrolu, Yifu Tao, Jonathan D Gammell, and Maurice Fallon. "Osprey: Multi-Session Autonomous Aerial Mapping with LiDAR-based SLAM and Next Best View Planning". In: *IEEE Transactions on Field Robotics* (2024) (page 46).

[24] Michael Bosse, Paul M. Newman, John J. Leonard, and Seth J. Teller. "Simultaneous Localization and Map Building in Large-Scale Cyclic Environments Using the Atlas Framework". In: *Intl. J. of Robot. Res.* 23.12 (2004), pp. 1113–1139 (page 22).

[25] Boston Dynamics. *Spot ®- The Agile Mobile Robot*. https://www.bostondynamics.com/products/spot. [Online; accessed 10-Feb-2023]. 2023 (page 43).

[26] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000) (page 26).

[27] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. "A 240 × 180 130 dB 3 µs Latency Global Shutter Spatiotemporal Vision Sensor". In: *IEEE Journal of Solid-State Circuits* 49.10 (2014), pp. 2333–2341. DOI: 10.1109/JSSC.2014.2342715 (page 116).

[28] Duane C Brown. "Decentering distortion of lenses". In: *Photogrammetric Engineering* 32.3 (1966), pp. 444–462 (page 26).

[29] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J. Leonard. "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age". In: *IEEE Trans. Robot.* 32.6 (2016), pp. 1309–1332. DOI: 10.1109/TRO.2016.2624754 (pages 2, 19, 30, 35).

[30] Carlos Campos, Richard Elvira, Juan J. Gomez, Jose M. M. Montiel, and Juan D. Tardos. "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM". In: *IEEE Trans. Robot.* 37.6 (2021), pp. 1874–1890 (pages 2, 35, 51).

[31] Marco Camurri, Milad Ramezani, Simona Nobili, and Maurice Fallon. "Pronto: A Multi-Sensor State Estimator for Legged Robots in Real-World Scenarios". In: *Frontiers in Robotics and AI* 7.68 (2020), pp. 1–18. ISSN: 2296-9144. DOI: 10.3389/frobt.2020.00068. URL: https://www.frontiersin.org/article/10.3389/frobt.2020.00068 (page 45).

[32] D. Caruso, J. Engel, and D. Cremers. "Large-Scale Direct SLAM for Omnidirectional Cameras". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2015 (page 35).

[33] Peiyu Chen, Weipeng Guan, and Peng Lu. "ESVIO: Event-Based Stereo Visual Inertial Odometry". In: *IEEE Robot. Autom. Lett. (RA-L)* 8.6 (2023), pp. 3661–3668. DOI: 10.1109/LRA.2023.3269950 (page 38).

[34] Annett Chilian, Heiko Hirschmüller, and Martin Görner. "Multisensor data fusion for robust pose estimation of a six-legged walking robot". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2011, pp. 2497–2504. DOI: 10.1109/IROS.2011.6094484 (page 45).

[35] Giovanni Cioffi, Titus Cieslewski, and Davide Scaramuzza. "Continuous-Time Vs. Discrete-Time Vision-Based SLAM: A Comparative Study". In: *IEEE Robot. Autom. Lett. (RA-L)* 7.2 (2022). DOI: 10.1109/LRA.2022.3143303 (pages 36, 37, 39).

[36] Xavier Clady, Sio-Hoi Ieng, and Ryad Benosman. "Asynchronous event-based corner detection and matching". In: *Neural Networks* 66 (2015), pp. 91–106 (page 110).

[37] A. Cramariuc, L. Bernreiter, F. Tschopp, M. Fehr, V. Reijgwart, J. Nieto, R. Siegwart, and C. Cadena. "maplab 2.0 – A Modular and Multi-Modal Mapping Framework". In: *IEEE Robot. Autom. Lett. (RA-L)* 8.2 (2023), pp. 520–527. DOI: 10.1109/LRA.2022.3227865 (page 50).

[38] Mark Cummins and Paul Newman. "FAB-MAP: Probabilistic localization and mapping in the space of appearance". In: *Intl. J. of Robot. Res.* 27.6 (2008), pp. 647–665 (pages 34, 35).

[39] Brian Curless and Marc Levoy. "A volumetric method for building complex models from range images". In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 1996, pp. 303–312 (page 47).

[40] Benny Dai, Cedric Le Gentil, and Teresa Vidal-Calleja. "A Tightly-Coupled Event-Inertial Odometry using Exponential Decay and Linear Preintegrated Measurements". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022, pp. 9475–9482. DOI: 10.1109/IROS47612.2022.9981249 (page 40).

[41] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. "MonoSLAM: Real-Time Single Camera SLAM". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 29.6 (2007), pp. 1052–1067. DOI: 10.1109/TPAMI.2007.1049 (page 34).

[42] Frank Dellaert and GTSAM Contributors. *borglab/gtsam*. Version 4.2a8. May 2022. DOI: 10.5281/zenodo.5794541. URL: https://github.com/borglab/gtsam) (pages 19, 31).

[43] Frank Dellaert and Michael Kaess. *Factor Graphs for Robot Perception*. Now Foundations and Trends, 2017. DOI: 10.1561/2300000043 (page 15).

[44] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. "SuperPoint: Self-Supervised Interest Point Detection and Description". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*. 2018, pp. 337–33712. DOI: 10.1109/CVPRW.2018.00060 (page 50).

[45] Edsger W Dijkstra. "A note on two problems in connexion with graphs". In: *Numerische mathematik* 1.1 (1959), pp. 269–271 (page 83).

[46] Daniel Duberg, Qingwen Zhang, MingKai Jia, and Patric Jensfelt. "DUFOMap: Efficient Dynamic Awareness Mapping". In: *IEEE Robot. Autom. Lett. (RA-L)* 9.6 (2024), pp. 1–8. DOI: 10.1109/LRA.2024.3387658 (page 49).

[47] Hugh Durrant-Whyte and Tim Bailey. "Simultaneous localization and mapping: part I". In: *IEEE Robotics & Automation Magazine* 13.2 (2006), pp. 99–110 (page 36).

[48] Ethan Eade. "Gauss-newton/levenberg-marquardt optimization". In: *Tech. Rep.* (2013) (page 17).

[49] Moumen T. El-Melegy. "Random sampler M-estimator algorithm for robust function approximation via feed-forward neural networks". In: *IEEE Intl. Joint Conf. on Neural Networks*. 2011, pp. 3134–3140. DOI: 10.1109/IJCNN.2011.6033636 (page 36).

[50] A. Elfes. "Using occupancy grids for mobile robot perception and navigation". In: *Computer* 22.6 (1989), pp. 46–57. DOI: 10.1109/2.30720 (page 46).

[51] J. Engel, T. Schöps, and D. Cremers. "LSD-SLAM: Large-Scale Direct Monocular SLAM". In: *Eur. Conf. on Computer Vision (ECCV)*. 2014 (pages 32, 35, 109).

[52] J. Engel, J. Stueckler, and D. Cremers. "Large-Scale Direct SLAM with Stereo Cameras". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2015 (page 35).

[53] Jakob Engel, Vladlen Koltun, and Daniel Cremers. "Direct Sparse Odometry". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40.3 (2018), pp. 611–625. DOI: 10.1109/TPAMI.2017.2658577 (pages 17, 33, 35).

[54] Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, et al. "Project aria: A new tool for egocentric multi-modal ai research". In: *arXiv preprint arXiv:2308.13561* (2023) (pages 86, 105, 123).

[55] Jakob Engel, Jürgen Sturm, and Daniel Cremers. "Semi-dense Visual Odometry for a Monocular Camera". In: *Intl. Conf. on Computer Vision (ICCV)*. 2013, pp. 1449–1456. DOI: 10.1109/ICCV.2013.183 (page 33).

[56] Gian Erni, Jonas Frey, Takahiro Miki, Matías Mattamala, and Marco Hutter. "MEM: Multi-Modal Elevation Mapping for Robotics and Learning". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2023, pp. 11011–11018 (page 46).

[57] Parker Ewen, Adam Li, Yuxin Chen, Steven Hong, and Ram Vasudevan. "These Maps are Made for Walking: Real-Time Terrain Property Estimation for Mobile Robots". In: *IEEE Robot. Autom. Lett. (RA-L)* 7.3 (2022), pp. 7083–7090. DOI: 10.1109/LRA.2022.3180439 (page 46).

[58] Shamel Fahmi, Geoff Fink, and Claudio Semini. "On state estimation for legged locomotion over soft terrain". In: *IEEE Sensors Letters* 5.1 (2021), pp. 1–4 (page 44).

[59] Maurice F. Fallon, Matthew Antone, Nicholas Roy, and Seth Teller. "Drift-free humanoid state estimation fusing kinematic, inertial and LIDAR sensing". In: *2014 IEEE-RAS International Conference on Humanoid Robots*. 2014, pp. 112–119. DOI: 10.1109/HUMANOIDS.2014.7041346 (page 45).

[60] David D. Fan, Kyohei Otsu, Yuki Kubo, Anushri Dixit, Joel Burdick, and Ali-akbar Agha-mohammadi. "STEP: Stochastic Traversability Evaluation and Planning for Risk-Aware Off-road Navigation". In: *Robotics: Science and Systems (RSS)*. 2021 (page 46).

[61] Péter Fankhauser, Michael Bloesch, and Marco Hutter. "Probabilistic Terrain Mapping for Mobile Robots With Uncertain Localization". In: *IEEE Robot. Autom. Lett. (RA-L)* 3.4 (2018), pp. 3019–3026 (pages 46, 48).

[62] Péter Fankhauser, Michael Bloesch, Diego Rodriguez, Ralf Kaestner, Marco Hutter, and Roland Siegwart. "Kinect v2 for mobile robot navigation: Evaluation and modeling". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2015, pp. 388–394. DOI: 10.1109/ICAR.2015.7251485 (page 27).

[63] Martin A Fischler and Robert C Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Communications of the ACM* 24.6 (1981), pp. 381–395 (pages 30, 36).

[64] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. "On-Manifold Preintegration for Real-Time Visual–Inertial Odometry". In: *IEEE Trans. Robot.* 33.1 (2017), pp. 1–21. DOI: 10.1109/TRO.2016.2597321 (page 39).

[65] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. "SVO: Fast semi-direct monocular visual odometry". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE. 2014, pp. 15–22 (page 33).

[66] E. Foxlin. "Pedestrian tracking with shoe-mounted inertial sensors". In: *IEEE Computer Graphics and Applications* 25.6 (2005), pp. 38–46. DOI: 10.1109/MCG.2005.140 (page 45).

[67] Friedrich Fraundorfer and Davide Scaramuzza. "Visual Odometry : Part II: Matching, Robustness, Optimization, and Applications". In: *IEEE Robotics & Automation Magazine* 19.2 (2012), pp. 78–90. DOI: 10.1109/MRA.2012.2182810 (page 30).

[68] Friedrich Fraundorfer, Davide Scaramuzza, and Marc Pollefeys. "A constricted bundle adjustment parameterization for relative scale estimation in visual odometry". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2010, pp. 1899–1904. DOI: 10.1109/ROBOT.2010.5509733 (page 31).

[69] Paul Furgale, Joern Rehder, and Roland Siegwart. "Unified temporal and spatial calibration for multi-sensor systems". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2013, pp. 1280–1286. DOI: 10.1109/IROS.2013.6696514 (page 26).

[70] Paul Furgale, Chi Hay Tong, Timothy D Barfoot, and Gabe Sibley. "Continuous-time batch trajectory estimation using temporal basis functions". In: *Intl. J. of Robot. Res.* 34.14 (2015) (page 39).

[71] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. "Event-based vision: A survey". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 44.1 (2020) (pages 2, 23, 28, 38, 110, 111).

[72] Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. "A Unifying Contrast Maximization Framework for Event Cameras, with Applications to Motion, Depth, and Optical Flow Estimation". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 2018, pp. 3867–3876. DOI: 10.1109/CVPR.2018.00407 (page 38).

[73] Dorian Galvez-López and Juan D. Tardos. "Bags of Binary Words for Fast Place Recognition in Image Sequences". In: *IEEE Trans. Robot.* 28.5 (2012), pp. 1188–1197. DOI: 10.1109/TRO.2012.2197158 (pages 34, 35).

[74] X. Gao, R. Wang, N. Demmel, and D. Cremers. "LDSO: Direct Sparse Odometry with Loop Closure". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2018 (page 35).

[75] Stefano Gasperini, Nils Morbitzer, HyunJun Jung, Nassir Navab, and Federico Tombari. "Robust Monocular Depth Estimation under Challenging Conditions". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 2023, pp. 8177–8186 (page 1).

[76] Henri P Gavin. "The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems". In: *Department of Civil and Environmental Engineering Duke University August* 3 (2019), pp. 1–23 (pages 17, 18).

[77] Andreas Geiger, Julius Ziegler, and Christoph Stiller. "StereoScan: Dense 3D Reconstruction in Real-time". In: *IEEE Intell. Veh. Symp. (IV)*. 2011 (page 31).

[78] Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. "OpenVINS: A Research Platform for Visual-Inertial Estimation". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2020, pp. 4666–4672 (page 33).

[79] Hyeonjae Gil, Dongjae Lee, Giseop Kim, and Ayoung Kim. "Ephemerality meets LiDAR-based Lifelong Mapping". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. Atlanta, 2025 (pages 49, 51).

[80] Arren Glover, Aiko Dinale, Leandro De Souza Rosa, Simeon Bamford, and Chiara Bartolozzi. "luvharris: A practical corner detector for event-cameras". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 44.12 (2021), pp. 10087–10098 (page 122).

[81] David Gouaillier, Vincent Hugel, Pierre Blazevic, Chris Kilner, Jérôme Monceaux, Pascal Lafourcade, Brice Marnier, Julien Serre, and Bruno Maisonnier. "Mechatronic design of NAO humanoid". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE. 2009, pp. 769–774 (page 41).

[82] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, Udo Frese, and Christoph Hertzberg. "Hierarchical optimization on manifolds for online 2D and 3D mapping". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2010, pp. 273–278. DOI: 10.1109/ROBOT.2010.5509407 (page 19).

[83] Paul D. Groves. "Navigation using inertial sensors [Tutorial]". In: *IEEE Aerospace and Electronic Systems Magazine* 30.2 (2015), pp. 42–69. DOI: 10.1109/MAES.2014.130191 (page 44).

[84] A Grunnet-Jepsen, M Harville, B Fulkerson, D Piro, S Brook, and J Radford. "Introduction to Intel Realsense Visual SLAM and the T265 Tracking Camera". In: *Product Documentation* (2019) (page 84).

[85] Thomas Gurriet, Sylvain Finet, Guilhem Boeris, Alexis Duburcq, Ayonga Hereid, Omar Harib, Matthieu Masselin, Jessy Grizzle, and Aaron D. Ames. "Towards Restoring Locomotion for Paraplegics: Realizing Dynamically Stable Walking on Exoskeletons". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2018, pp. 2804–2811. DOI: 10.1109/ICRA.2018.8460647 (page 42).

[86] Antea Hadviger, Igor Cvišić, Ivan Marković, Sacha Vražić, and Ivan Petrović. "Feature-based event stereo visual odometry". In: *arXiv preprint arXiv:2107.04921* (2021) (page 38).

[87] Brian C Hall. *Lie groups, Lie algebras, and representations*. Springer, 2013 (page 10).

[88] R.M. Haralick, D. Lee, K. Ottenburg, and M. Nolle. "Analysis and solutions of the three point perspective pose estimation problem". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 1991, pp. 592–598. DOI: 10.1109/CVPR.1991.139759 (page 30).

[89] Chris Harris and Mike Stephens. "A combined corner and edge detector". In: *Alvey vision conference*. Vol. 15. 50. 1988, pp. 10–5244 (pages 110, 116).

[90] Richard I Hartley and Peter Sturm. "Triangulation". In: *Computer vision and image understanding* 68.2 (1997), pp. 146–157 (page 31).

[91] Ross Hartley, Maani Ghaffari Jadidi, Lu Gan, Jiunn-Kai Huang, Jessy W. Grizzle, and Ryan M. Eustice. "Hybrid Contact Preintegration for Visual-Inertial-Contact State Estimation Using Factor Graphs". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2018, pp. 3783–3790. DOI: 10.1109/IROS.2018.8593801 (page 45).

[92] Javier Hidalgo-Carrió, Guillermo Gallego, and Davide Scaramuzza. "Event-Aided Direct Sparse Odometry". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 2022, pp. 5781–5790 (page 67).

[93] S. Hirose and K. Kato. "Development of quadruped walking robot with the mission of mine detection and removal-proposal of shape-feedback master-slave arm". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. Vol. 2. 1998, 1713–1718 vol.2. DOI: 10.1109/ROBOT.1998.677410 (page 43).

[94] S. Hirose and K. Kato. "Study on quadruped walking robot in Tokyo Institute of Technology-past, present and future". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. Vol. 1. 2000, 414–419 vol.1. DOI: 10.1109/ROBOT.2000.844091 (page 43).

[95] Shigeo Hirose. "A study of design and control of a quadruped walking vehicle". In: *Intl. J. of Robot. Res.* 3.2 (1984), pp. 113–133 (page 43).

[96] J. Hodgins. "Legged robots on rough terrain: experiments in adjusting step length". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 1988, 824–826 vol.2. DOI: 10.1109/ROBOT.1988.12162 (page 41).

[97] R. Hodoshima, T. Doi, Y. Fukuda, S. Hirose, T. Okamoto, and J. Mori. "Development of TITAN XI: a quadruped walking robot to work on slopes". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. Vol. 1. 2004, 792–797 vol.1. DOI: 10.1109/IROS.2004.1389449 (page 43).

[98] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. "OctoMap: An efficient probabilistic 3D mapping framework based on octrees". In: *Auton. Robots* 34 (2013), pp. 189–206 (pages 1, 21, 46, 48).

[99] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. "Multimodal Spatial Language Maps for Robot Navigation and Manipulation". In: (2025) (page 105).

[100] Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. "Observability-based Rules for Designing Consistent EKF SLAM Estimators". In: *Intl. J. of Robot. Res.* 29.5 (2010), pp. 502–528. DOI: 10.1177/0278364909353640. eprint: https://doi.org/10.1177/0278364909353640. URL: https://doi.org/10.1177/0278364909353640 (page 44).

[101] Kun Huang, Yifu Wang, and Laurent Kneip. "Motion estimation of non-holonomic ground vehicles from a single feature correspondence measured over n views". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 2019 (page 39).

[102] N. Hughes, Y. Chang, and L. Carlone. "Hydra: A Real-time Spatial Perception System for 3D Scene Graph Construction and Optimization". In: (2022) (page 34).

[103] Nathan Hughes, Yun Chang, and Luca Carlone. "Hydra: A real-time spatial perception system for 3D scene graph construction and optimization". In: *Robotics: Science and Systems (RSS)*. 2022 (pages 1, 47).

[104] Marco Hutter, Christian Gehring, Michael Bloesch, Mark A Hoepflinger, C David Remy, and Roland Siegwart. "StarlETH: A compliant quadrupedal robot for fast, efficient, and versatile locomotion". In: *Adaptive mobile robotics*. World Scientific, 2012, pp. 483–490 (page 43).

[105] Marco Hutter, Christian Gehring, Dominic Jud, Andreas Lauber, C. Dario Bellicoso, Vassilios Tsounis, Jemin Hwangbo, Karen Bodie, Peter Fankhauser, Michael Bloesch, Remo Diethelm, Samuel Bachmann, Amir Melzer, and Mark Hoepflinger. "ANYmal - a highly mobile and dynamic quadrupedal robot". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, pp. 38–44. DOI: 10.1109/IROS.2016.7758092 (page 43).

[106] Vaiyee Huynh, Guillaume Burger, Quoc Viet Dang, Raphaël Pelgé, Guilhem Boéris, Jessy W. Grizzle, Aaron D. Ames, and Matthieu Masselin. "Versatile Dynamic Motion Generation Framework: Demonstration With a Crutch-Less Exoskeleton on Real-Life Obstacles at the Cybathlon 2020 With a Complete Paraplegic Person". In: *Frontiers in Robotics and AI* Volume 8 - 2021 (2021). ISSN: 2296-9144. DOI: 10.3389/frobt.2021.723780. URL: https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2021.723780 (page 42).

[107] *Intel RealSense Depth Camera D435i*. https://www.intelrealsense.com/depth-camera-d435i/. 2025 (pages 23, 26, 27).

[108] Mingkai Jia, Qingwen Zhang, Bowen Yang, Jin Wu, Ming Liu, and Patric Jensfelt. "BeautyMap: Binary-Encoded Adaptable Ground Matrix for Dynamic Points Removal in Global Maps". In: *IEEE Robotics and Automation Letters* 9.7 (2024), pp. 6256–6263. DOI: 10.1109/LRA.2024.3402625 (page 49).

[109] Hordur Johannsson, Michael Kaess, Maurice Fallon, and John J. Leonard. "Temporally scalable visual SLAM using a reduced pose graph". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2013, pp. 54–61. DOI: 10.1109/ICRA.2013.6630556 (pages 102, 105).

[110] Eagle S. Jones and Stefano Soatto. "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach". In: *Intl. J. of Robot. Res.* 30.4 (2011), pp. 407–430. DOI: 10.1177/0278364910388963 (page 33).

[111] Kevin M Judd and Jonathan D Gammell. "Multimotion visual odometry". In: *Intl. J. of Robot. Res.* 43.8 (2024), pp. 1250–1278 (page 1).

[112] S.J. Julier and J.K. Uhlmann. "Unscented filtering and nonlinear estimation". In: *Proceedings of the IEEE* 92.3 (2004), pp. 401–422. DOI: 10.1109/JPROC.2003.823141 (page 44).

[113] M. Kaess. "Incremental Smoothing and Mapping". Ph.D. Georgia Institute of Technology, Dec. 2008 (page 31).

[114] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. "iSAM: Fast Incremental Smoothing and Mapping with Efficient Data Association". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2007, pp. 1670–1677. DOI: 10.1109/ROBOT.2007.363563 (pages 19, 31).

[115] Rudolph Emil Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Transactions of the ASME–Journal of Basic Engineering* 82.Series D (1960), pp. 35–45 (page 31).

[116] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. "Probabilistic Roadmaps for Path Planning in High-dimensional Configuration Spaces". In: *IEEE Trans. Robotics and Automation* 12.4 (1996), pp. 566–580. DOI: 10.1109/70.508439 (page 83).

[117] Jonathan Kelly and Gaurav S Sukhatme. "Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration". In: *Intl. J. of Robot. Res.* 30.1 (2011), pp. 56–79. DOI: 10.1177/0278364910382802 (page 33).

[118] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. "LERF: Language Embedded Radiance Fields". In: *Intl. Conf. on Computer Vision (ICCV)*. 2023 (page 47).

[119] Giseop Kim and Ayoung Kim. "Lt-mapper: A modular framework for lidar-based lifelong mapping". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE. 2022, pp. 7995–8002 (pages 48, 51).

[120] Giseop Kim and Ayoung Kim. "Remove, then Revert: Static Point cloud Map Construction using Multiresolution Range Images". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2020, pp. 10758–10765. DOI: 10.1109/IROS45743.2020.9340856 (pages 49, 51).

[121] Giseop Kim and Ayoung Kim. "Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2018, pp. 4802–4809. DOI: 10.1109/IROS.2018.8593953 (page 51).

[122] Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew Davison. "Simultaneous Mosaicing and Tracking with an Event Camera". In: *British Machine Vision Conf. (BMVC)*. 2014 (page 37).

[123] Hanme Kim, Stefan Leutenegger, and Andrew J Davison. "Real-time 3D reconstruction and 6-DoF tracking with an event camera". In: *ECCV*. 2016 (pages 37, 38).

[124] Alexander Kirillov Jr. *An Introduction to Lie Groups and Lie Algebras*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2008 (page 15).

[125] LARRY D. Kirkpatrick and GREGORY F. Wheeler. *Physics: A World View*. Harcourt College Publishers, 1994. ISBN: 9780030006036. URL: https://books.google.co.uk/books?id=-eepNg2Aec4C (page 24).

[126] Bernd Kitt, Andreas Geiger, and Henning Lategahn. "Visual Odometry based on Stereo Image Sequences with RANSAC-based Outlier Rejection Scheme". In: *IEEE Intell. Veh. Symp. (IV)*. 2010 (page 17).

[127] Bernd Kitt, Andreas Geiger, and Henning Lategahn. "Visual Odometry based on Stereo Image Sequences with RANSAC-based Outlier Rejection Scheme". In: *IEEE Intell. Veh. Symp. (IV)*. 2010 (page 31).

[128] Georg Klein and David Murray. "Parallel Tracking and Mapping for Small AR Workspaces". In: *IEEE/ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR)*. 2007, pp. 225–234. DOI: 10.1109/ISMAR.2007.4538852 (page 34).

[129] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. "G2o: A general framework for graph optimization". In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 3607–3613. DOI: 10.1109/ICRA.2011.5979949 (page 19).

[130] Kenji Koide, Masashi Yokozuka, Shuji Oishi, and Atsuhiko Banno. "Voxelized GICP for Fast and Accurate 3D Point Cloud Registration". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 11054–11059. DOI: 10.1109/ICRA48506.2021.9560835 (page 51).

[131] Beat Kueng, Elias Mueggler, Guillermo Gallego, and Davide Scaramuzza. "Low-latency visual odometry using event-based feature tracks". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2016 (pages 37, 40, 67, 105).

[132] Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake. "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot". In: *Autonomous robots* 40 (2016), pp. 429–455 (page 42).

[133] Scott Kuindersma, Robin Deits, Maurice F. Fallon, Andres Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake. "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot". In: *Auton. Robots* 40.3 (2016), pp. 429–455 (page 46).

[134] Mathieu Labbé and François Michaud. "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation". In: *J. Field Robot.* 36.2 (2019), pp. 416–446. DOI: https://doi.org/10.1002/rob.21831 (page 50).

[135] Mathieu Labbé and François Michaud. "Long-term online multi-session graph-based SPLAM with memory management". In: *Auton. Robots* 42 (2018), pp. 1133–1150 (page 50).

[136] Mathieu Labbé and François Michaud. "Multi-session visual slam for illumination-invariant re-localization in indoor environments". In: *Frontiers in Robotics and AI* 9 (2022), p. 801886 (page 50).

[137] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E. Shi, and Ryad B. Benosman. "HOTS: A Hierarchy of Event-Based Time-Surfaces for Pattern Recognition". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 39.7 (2017), pp. 1346–1359. DOI: 10.1109/TPAMI.2016.2574707 (pages 37, 122).

[138] Cedric Le Gentil, Florian Tschopp, Ignacio Alzugaray, Teresa Vidal-Calleja, Roland Siegwart, and Juan Nieto. "IDOL: A Framework for IMU-DVS Odometry using Lines". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2020 (page 40).

[139] Cedric Le Gentil and Teresa Vidal-Calleja. "Continuous Latent State Preintegration for Inertial-Aided Systems". In: *Intl. J. of Robot. Res.* (2023). DOI: 10.1177/02783649231199537 (page 40).

[140] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. "BRISK: Binary robust invariant scalable keypoints". In: *Intl. Conf. on Computer Vision (ICCV)*. 2011, pp. 2548–2555 (page 116).

[141] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. "Keyframe-based visual–inertial odometry using nonlinear optimization". In: *Intl. J. of Robot. Res.* 34.3 (2015), pp. 314–334 (page 34).

[142] Kejun Li, Maegan Tucker, Erdem Bıyık, Ellen Novoseller, Joel W Burdick, Yanan Sui, Dorsa Sadigh, Yisong Yue, and Aaron D Ames. "Roial: Region of interest active learning for characterizing exoskeleton gait preference landscapes". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE. 2021, pp. 3212–3218 (page 42).

[143] Mingyang Li and Anastasios I. Mourikis. "Improving the accuracy of EKF-based visual-inertial odometry". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2012, pp. 828–835. DOI: 10.1109/ICRA.2012.6225229 (page 33).

[144] Ruoxiang Li, Dianxi Shi, Yongjun Zhang, Kaiyue Li, and Ruihao Li. "FA-Harris: A fast and asynchronous corner detector for event cameras". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2019, pp. 6223–6229. DOI: 10.1109/IROS40897.2019.8968491 (pages 111, 117).

[145] Hyungtae Lim, Sungwon Hwang, and Hyun Myung. "ERASOR: Egocentric Ratio of Pseudo Occupancy-Based Dynamic Object Removal for Static 3D Point Cloud Map Building". In: *IEEE Robot. Autom. Lett. (RA-L)* 6.2 (2021), pp. 2272–2279. DOI: 10.1109/LRA.2021.3061363 (page 49).

[146] Daqi Liu, Alvaro Parra, Yasir Latif, Bo Chen, Tat-Jun Chin, and Ian Reid. "Asynchronous Optimisation for Event-based Visual Odometry". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2022 (page 39).

[147] Jing Liu, Min Tan, and Xiaoguang Zhao. "Legged robots — an overview". In: *Transactions of the Institute of Measurement and Control* 29.2 (2007), pp. 185–202. DOI: 10.1177/0142331207075610 (page 40).

[148] Shih-Chii Liu, Tobi Delbruck, Giacomo Indiveri, Adrian Whatley, and Rodney Douglas. *Event-based neuromorphic systems*. John Wiley & Sons, 2014 (page 28).

[149] Wenxin Liu, David Caruso, Eddy Ilg, Jing Dong, Anastasios I. Mourikis, Kostas Daniilidis, Vijay Kumar, and Jakob Engel. "TLIO: Tight Learned Inertial Odometry". In: *IEEE Robot. Autom. Lett. (RA-L)* 5.4 (2020), pp. 5653–5660. DOI: 10.1109/LRA.2020.3007421 (page 44).

[150] Steven Lovegrove, Andrew J. Davison, and Javier Ibañez-Guzmán. "Accurate visual odometry from a rear parking camera". In: *IEEE Intell. Veh. Symp. (IV)*. 2011, pp. 788–793. DOI: 10.1109/IVS.2011.5940546 (page 86).

[151] D.G. Lowe. "Object recognition from local scale-invariant features". In: *Intl. Conf. on Computer Vision (ICCV)*. Vol. 2. 1999, 1150–1157 vol.2. DOI: 10.1109/ICCV.1999.790410 (page 34).

[152] Kevin M Lynch and Frank C Park. *Modern robotics*. Cambridge University Press, 2017 (page 44).

[153] Jeremy Ma, Max Bajracharya, Sara Susca, Larry Matthies, and Matt Malchano. "Real-time pose estimation of a dynamic quadruped in GPS-denied environments for 24-hour operation". In: *Intl. J. of Robot. Res.* 35.6 (2016), pp. 631–653. DOI: 10.1177/0278364915587333. eprint: https://doi.org/10.1177/0278364915587333. URL: https://doi.org/10.1177/0278364915587333 (page 45).

[154] Jacques Manderscheid, Amos Sironi, Nicolas Bourdis, Davide Migliore, and Vincent Lepetit. "Speed Invariant Time Surface for Learning to Detect Corner Points With Event-Based Cameras". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 2019 (pages 104, 122).

[155] Matias Mattamala, Nived Chebrolu, and Maurice Fallon. "An Efficient Locally Reactive Controller for Safe Navigation in Visual Teach and Repeat Missions". In: *IEEE Robot. Autom. Lett. (RA-L)* 7.2 (2022), pp. 2353–2360. DOI: 10.1109/LRA.2022.3143196 (page 46).

[156] Christopher Mei, Gabe Sibley, and Paul Newman. "Closing loops without places". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2010, pp. 3738–3744. DOI: 10.1109/IROS.2010.5652266 (page 35).

[157] Xiangyun Meng, Nathan Hatch, Alexander Lambert, Anqi Li, Nolan Wagener, Matthew Schmittle, Joonho Lee, Wentao Yuan, Zoey Qiuyu Chen, Sameul Deng, Greg Okopal, Dieter Fox, Byron Boots, and Amirreza Shaban. "TerrainNet: Visual Modeling of Complex Terrain for High-speed, Off-road Navigation". In: *Robotics: Science and Systems (RSS)*. 2023 (page 46).

[158] B. Mersch, X. Chen, I. Vizzo, L. Nunes, J. Behley, and C. Stachniss. "Receding Moving Object Segmentation in 3D LiDAR Data Using Sparse 4D Convolutions". In: *IEEE Robot. Autom. Lett. (RA-L)* 7.3 (2022), pp. 7503–7510 (page 49).

[159] Nico Messikommer, Carter Fang, Mathias Gehrig, Giovanni Cioffi, and Davide Scaramuzza. "Data-driven Feature Tracking for Event Cameras with and without Frames". In: *IEEE Trans. Pattern Anal. Mach. Intell.* (2025) (pages 64, 104).

[160] Giorgio Metta, Lorenzo Natale, Francesco Nori, Giulio Sandini, David Vernon, Luciano Fadiga, Claes von Hofsten, Kerstin Rosander, Manuel Lopes, José Santos-Victor, Alexandre Bernardino, and Luis Montesano. "The iCub humanoid robot: An open-systems platform for research in cognitive development". In: *Neural Networks* 23.8 (2010). Social Cognition: From Babies to Robots, pp. 1125–1134. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2010.08.010. URL: https://www.sciencedirect.com/science/article/pii/S0893608010001619 (page 41).

[161] Takahiro Miki, Lorenz Wellhausen, Ruben Grandia, Fabian Jenelten, Timon Homberger, and Marco Hutter. "Elevation Mapping for Locomotion and Navigation using GPU". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2022, pp. 2273–2280. DOI: 10.1109/IROS47612.2022.9981507 (pages 46, 49, 84).

[162] Alexander Millane, Zachary Taylor, Helen Oleynikova, Juan Nieto, Roland Siegwart, and César Cadena. "C-blox: A Scalable and Consistent TSDF-based Dense Mapping Approach". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2018 (page 47).

[163] Heajung Min, Kyung Min Han, and Young J. Kim. "OctoMap-RT: Fast Probabilistic Volumetric Mapping Using Ray-Tracing GPUs". In: *IEEE Robot. Autom. Lett. (RA-L)* 8.9 (2023), pp. 5696–5703. DOI: 10.1109/LRA.2023.3300227 (page 49).

[164] Hans P. Moravec. "Obstacle avoidance and navigation in the real world by a seeing robot rover". PhD thesis. Stanford University, 1980 (page 30).

[165] Hans P. Morevec. "Towards automatic visual obstacle avoidance". In: *Intl. Joint Conf. on Artificial Intelligence*. IJCAI'77. Cambridge, USA: Morgan Kaufmann Publishers Inc., 1977, p. 584 (page 30).

[166] Ralph S Mosher. "Exploring the potential of a quadruped". In: *SAE Transactions* (1969), pp. 836–843 (page 43).

[167] Anastasios I. Mourikis and Stergios I. Roumeliotis. "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2007, pp. 3565–3572. DOI: 10.1109/ROBOT.2007.364024 (page 33).

[168] Elias Mueggler, Chiara Bartolozzi, and Davide Scaramuzza. "Fast event-based corner detection". In: *British Machine Vision Conf. (BMVC)*. 2017 (page 111).

[169] Elias Mueggler, Guillermo Gallego, and Davide Scaramuzza. "Continuous-time trajectory estimation for event-based vision sensors". In: *Robotics: Science and Systems (RSS)*. 2015 (pages 38, 39).

[170] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM". In: *Intl. J. of Robot. Res.* 36.2 (2017), pp. 142–149 (page 27).

[171] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM". In: *Intl. J. of Robot. Res.* 36.2 (2017), pp. 142–149 (pages 116, 120).

[172] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. "ORB-SLAM: a Versatile and Accurate Monocular SLAM System". In: *IEEE Trans. Robot.* 31.5 (2015), pp. 1147–1163. DOI: 10.1109/TRO.2015.2463671 (pages 35, 109).

[173] Raul Mur-Artal and Juan D. Tardos. "ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras". In: *IEEE Trans. Robot.* 33.5 (2017), pp. 1255–1262. DOI: 10.1109/TRO.2017.2705103 (page 35).

[174] Riku Murai, Eric Dexheimer, and Andrew J. Davison. "MASt3R-SLAM: Real-Time Dense SLAM with 3D Reconstruction Priors". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)* (2024) (page 34).

[175] Jiyeon Nam, Soojeong Hyeon, Youngjun Joo, DongKi Noh, and Hyungbo Shim. "Spectral Trade-Off for Measurement Sparsification of Pose-Graph SLAM". In: *IEEE Robot. Autom. Lett. (RA-L)* 9.1 (2024), pp. 723–730. DOI: 10.1109/LRA.2023.3337590 (pages 102, 105).

[176] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. "DTAM: Dense tracking and mapping in real-time". In: *Intl. Conf. on Computer Vision (ICCV)*. 2011, pp. 2320–2327. DOI: 10.1109/ICCV.2011.6126513 (page 32).

[177] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. "Real-time 3D reconstruction at scale using voxel hashing". In: *ACM Trans. Graph.* 32.6 (Nov. 2013). ISSN: 0730-0301. DOI: 10.1145/2508363.2508374. URL: https://doi.org/10.1145/2508363.2508374 (page 21).

[178] D. Nister. "An efficient solution to the five-point relative pose problem". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 26.6 (2004), pp. 756–770. DOI: 10.1109/TPAMI.2004.17 (page 30).

[179] D. Nister. "Preemptive RANSAC for live structure and motion estimation". In: *Intl. Conf. on Computer Vision (ICCV)*. 2003, 199–206 vol.1. DOI: 10.1109/ICCV.2003.1238341 (page 30).

[180] D. Nister, O. Naroditsky, and J. Bergen. "Visual odometry". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Vol. 1. 2004, pp. I–I. DOI: 10.1109/CVPR.2004.1315094 (pages 30, 31).

[181] Junkai Niu, Sheng Zhong, Xiuyuan Lu, Shaojie Shen, Guillermo Gallego, and Yi Zhou. "ESVO2: Direct Visual-Inertial Odometry With Stereo Event Cameras". In: *IEEE Trans. Robot.* 41 (2025), pp. 2164–2183. DOI: 10.1109/TRO.2025.3548523 (pages 38, 67, 105).

[182] Junkai Niu, Sheng Zhong, and Yi Zhou. "Imu-aided event-based stereo visual odometry". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE. 2024, pp. 11977–11983 (page 67).

[183] Simona Nobili, Marco Camurri, Victor Barasuol, Michele Focchi, Darwin G Caldwell, Claudio Semini, and Maurice F Fallon. "Heterogeneous Sensor Fusion for Accurate State Estimation of Dynamic Legged Robots". In: *Robotics: Science and Systems (RSS)*. 2017 (page 45).

[184] Kazunori Ohno, Satoshi Tadokoro, Keiji Nagatani, Eiji Koyanagi, and Tomoaki Yoshida. "Trials of 3-D map construction using the tele-operated tracked vehicle kenaf at disaster city". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2010, pp. 2864–2870. DOI: 10.1109/ROBOT.2010.5509722 (page 48).

[185] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. "Voxblox: Incremental 3D Euclidean Signed Distance Fields for On-Board MAV Planning". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2017 (pages 22, 47).

[186] Edwin Olson. "AprilTag: A robust and flexible visual fiducial system". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2011, pp. 3400–3407 (page 85).

[187] ACHIAM J OPENAI, S ADLER, et al. "GPT-4 technical report [EB]". In: *arXiv preprint arXiv:2303.08774* (2023) (page 105).

[188] Lichtsteiner Patrick, Christoph Posch, and Tobi Delbruck. "A 128x 128 120 db 15μ s latency asynchronous temporal contrast vision sensor". In: *IEEE Journal of Solid-State Circuits* 43 (2008), pp. 566–576 (pages 111, 113).

[189] Alonso Patron-Perez, Steven Lovegrove, and Gabe Sibley. "A spline-based trajectory representation for sensor fusion and rolling shutter cameras". In: *Intl. J. of Comput. Vision* 113.3 (2015), pp. 208–219 (page 39).

[190] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. "OpenScene: 3D Scene Understanding with Open Vocabularies". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 2023 (pages 2, 47).

[191] *PETMAN (Protection Ensemble Test Mannequin) Humanoid Military Robot*. https://www.army-technology.com/projects/petman/. Accessed: 2025-05-22 (page 42).

[192] R.R. Playter and M.H. Raibert. "Control Of A Biped Somersault In 3D". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. Vol. 1. 1992, pp. 582–589. DOI: 10.1109/IROS.1992.587396 (page 41).

[193] Francois Pomerleau, Philipp Krüsi, Francis Colas, Paul Furgale, and Roland Siegwart. "Long-term 3D map maintenance in dynamic environments". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2014, pp. 3712–3719. DOI: 10.1109/ICRA.2014.6907397 (page 49).

[194] G.A. Pratt and M.M. Williamson. "Series elastic actuators". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. Vol. 1. 1995, 399–406 vol.1. DOI: 10.1109/IROS.1995.525827 (page 43).

[195] Alberto Pretto et al. "Building an Aerial–Ground Robotics System for Precision Farming: An Adaptable Solution". In: *IEEE Robotics & Automation Magazine* 28.3 (2021), pp. 29–49. DOI: 10.1109/MRA.2020.3012492 (page 48).

[196] *Prophesee Metavision Technologies*. https://www.prophesee.ai/. 2025 (page 23).

[197] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. "Learning Transferable Visual Models From Natural Language Supervision". In: *CoRR* abs/2103.00020 (2021) (page 105).

[198] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. "Learning transferable visual models from natural language supervision". In: *Intl. Conf. on Machine Learning (ICML)*. PmLR. 2021, pp. 8748–8763 (page 99).

[199] Nikhil Raina, Guruprasad Somasundaram, Kang Zheng, Sagar Miglani, Steve Saarinen, Jeff Meissner, Mark Schwesinger, Luis Pesqueira, Ishita Prasad, Edward Miller, Prince Gupta, Mingfei Yan, Richard Newcombe, Carl Ren, and Omkar Parkhi. *EgoBlur Model*. 2023. arXiv: 2308.13093 [cs.CV] (page 102).

[200] Manoj Ramanathan, Lincong Luo, Jie Kai Er, Ming Jeat Foo, Chye Hsia Chiam, Lei Li, Wei Yun Yau, and Wei Tech Ang. "Visual Environment perception for obstacle detection and crossing of lower-limb exoskeletons". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2022. DOI: 10.1109/IROS47612.2022.9981412 (page 2).

[201] Milad Ramezani, Matias Mattamala, and Maurice Fallon. "AEROS: AdaptivE RObust Least-Squares for Graph-Based SLAM". In: *Frontiers in Robotics and AI* Volume 9 - 2022 (2022). ISSN: 2296-9144. DOI: 10.3389/frobt.2022.789444. URL: https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2022.789444 (page 36).

[202] Milad Ramezani, Georgi Tinchev, Egor Iuganov, and Maurice Fallon. "Online LiDAR-SLAM for Legged Robots with Robust Registration and Deep-Learned Loop Closure". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2020, pp. 4158–4164. DOI: 10.1109/ICRA40945.2020.9196769 (page 45).

[203] Henri Rebecq, Guillermo Gallego, Elias Mueggler, and Davide Scaramuzza. "EMVS: Event-based multi-view stereo—3D reconstruction with an event camera in real-time". In: *Intl. J. of Comput. Vision* 126.12 (2018) (page 38).

[204] Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. "Real-time Visual-Inertial Odometry for Event Cameras using Keyframe-based Nonlinear Optimization". In: *British Machine Vision Conf. (BMVC)*. 2017 (pages 38, 110).

[205] Henri Rebecq, Timo Horstschäfer, Guillermo Gallego, and Davide Scaramuzza. "EVO: A geometric approach to event-based 6-dof parallel tracking and mapping in real time". In: *IEEE Robot. Autom. Lett. (RA-L)* 2.2 (2016) (page 38).

[206] V. Reijgwart, A. Millane, H. Oleynikova, R. Siegwart, C. Cadena, and J. Nieto. "Voxgraph: Globally Consistent, Volumetric Mapping Using Signed Distance Function Submaps". In: *IEEE Robot. Autom. Lett. (RA-L)* (2020) (page 47).

[207] Victor Reijgwart, Cesar Cadena, Roland Siegwart, and Lionel Ott. "Efficient volumetric mapping of multi-scale environments using wavelet-based compression". In: *Robotics: Science and Systems (RSS)*. 2023 (pages 1, 21, 47, 102).

[208] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. "Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2020, pp. 1689–1696. DOI: 10.1109/ICRA40945.2020.9196885 (pages 32, 51).

[209] S.I. Roumeliotis and J.W. Burdick. "Stochastic cloning: a generalized framework for processing relative state measurements". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. Vol. 2. 2002, 1788–1795 vol.2. DOI: 10.1109/ROBOT.2002.1014801 (page 33).

[210] Joseph Rowell, Lintong Zhang, and Maurice Fallon. "LiSTA: Geometric Object-Based Change Detection in Cluttered Environments". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE. 2024, pp. 3632–3638 (page 49).

[211] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. "ORB: An efficient alternative to SIFT or SURF". In: *Intl. Conf. on Computer Vision (ICCV)*. 2011, pp. 2564–2571. DOI: 10.1109/ICCV.2011.6126544 (page 34).

[212] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura. "The intelligent ASIMO: system overview and integration". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 3. 2002, 2478–2483 vol.3. DOI: 10.1109/IRDS.2002.1041641 (page 41).

[213] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. "From Coarse to Fine: Robust Hierarchical Localization at Large Scale". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 2019 (page 34).

[214] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. "SuperGlue: Learning Feature Matching with Graph Neural Networks". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 2020. URL: https://arxiv.org/abs/1911.11763 (page 50).

[215] Davide Scaramuzza and Friedrich Fraundorfer. "Visual Odometry [Tutorial]". In: *IEEE Robotics & Automation Magazine* 18.4 (2011), pp. 80–92. DOI: 10.1109/MRA.2011.943233 (pages 30, 31).

[216] Lukas Schmid, Marcus Abate, Yun Chang, and Luca Carlone. "Khronos: A Unified Approach for Spatio-Temporal Metric-Semantic SLAM in Dynamic Environments". In: *Robotics: Science and Systems (RSS)*. Delft, Netherlands, 2024. DOI: 10.15607/RSS.2024.XX.081 (pages 49, 106).

[217] Lukas Schmid, Olov Andersson, Aurelio Sulser, Patrick Pfreundschuh, and Roland Siegwart. "Dynablox: Real-time Detection of Diverse Dynamic Objects in Complex Environments". In: 8.10 (2023), pp. 6259 –6266. DOI: 10.1109/LRA.2023.3305239 (page 49).

[218] Lukas Schmid, Jeffrey Delmerico, Johannes Schönberger, Juan Nieto, Marc Pollefeys, Roland Siegwart, and Cesar Cadena. "Panoptic Multi-TSDFs: a Flexible Representation for Online Multi-resolution Volumetric Mapping and Long-term Dynamic Scene Consistency". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2022, pp. 8018–8024. DOI: 10.1109/ICRA46639.2022.9811877 (pages 47, 49).

[219] Alexander Schmitz, Perla Maiolino, Marco Maggiali, Lorenzo Natale, Giorgio Cannata, and Giorgio Metta. "Methods and Technologies for the Implementation of Large-Scale Robot Tactile Sensors". In: *IEEE Trans. Robot.* 27.3 (2011), pp. 389–400. DOI: 10.1109/TRO.2011.2132930 (page 41).

[220] T. Schneider, M. T. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart. "maplab: An Open Framework for Research in Visual-inertial Mapping and Localization". In: *IEEE Robot. Autom. Lett. (RA-L)* 3.3 (2018), pp. 1418–1425. DOI: 10.1109/LRA.2018.2800113 (page 50).

[221] Johannes Lutz Schönberger and Jan-Michael Frahm. "Structure-from-Motion Revisited". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (page 2).

[222] C Semini, N G Tsagarakis, E Guglielmino, M Focchi, F Cannella, and D G Caldwell. "Design of HyQ – a hydraulically and electrically actuated quadruped robot". In: *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 225.6 (2011), pp. 831–849. DOI: 10.1177/0959651811402275. eprint:

https://doi.org/10.1177/0959651811402275. URL: https://doi.org/10.1177/0959651811402275 (page 43).

[223] Sangok Seok, Albert Wang, Meng Yee Chuah, David Otten, Jeffrey Lang, and Sangbae Kim. "Design principles for highly efficient quadrupeds and implementation on the MIT Cheetah robot". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2013, pp. 3307–3312. DOI: 10.1109/ICRA.2013.6631038 (page 43).

[224] *Sevensense Robotics AG*. https://www.sevensense.ai/. 2025 (pages 23, 26, 28).

[225] Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. "CLIP-Fields: Weakly Supervised Semantic Fields for Robotic Memory". In: *Robotics: Science and Systems (RSS)*. 2023 (page 47).

[226] Shaojie Shen, Nathan Michael, and Vijay Kumar. "Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2015, pp. 5303–5310. DOI: 10.1109/ICRA.2015.7139939 (page 33).

[227] Gabe Sibley, G. Sukhatme, and L. Matthies. "The Iterated Sigma Point Kalman Filter with Applications to Long Range Stereo". In: *Robotics: Science and Systems (RSS)*. Vol. 8. Jan. 2006, pp. 235–244 (page 31).

[228] Christiane Sommer, Vladyslav Usenko, David Schubert, Nikolaus Demmel, and Daniel Cremers. "Efficient derivative computation for cumulative b-splines on lie groups". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 2020, pp. 11148–11156 (page 39).

[229] Michal Staniaszek, Tobit Flatscher, Joseph Rowell, Hanlin Niu, Wenxing Liu, Yang You, Robert Skilton, Maurice Fallon, and Nick Hawes. "AutoInspect: Towards Long-Term Autonomous Industrial Inspection". In: *arXiv preprint arXiv:2404.12785* (2024) (pages 46, 48).

[230] Nikolaos Stathoulopoulos, Anton Koval, Ali-akbar Agha-mohammadi, and George Nikolakopoulos. "Frame: Fast and robust autonomous 3d point cloud map-merging for egocentric multi-robot exploration". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE. 2023, pp. 3483–3489 (page 51).

[231] Nikolaos Stathoulopoulos, Anton Koval, and George Nikolakopoulos. "3DEG: Data-Driven Descriptor Extraction for Global re-localization in subterranean environments". In: *Expert Systems with Applications* 237 (2024), p. 121508 (page 51).

[232] Hauke Strasdat, Andrew J. Davison, J.M.M. Montiel, and Kurt Konolige. "Double window optimisation for constant time visual SLAM". In: *Intl. Conf. on Computer Vision (ICCV)*. 2011, pp. 2352–2359. DOI: 10.1109/ICCV.2011.6126517 (page 35).

[233] Jan Stühmer, Stefan Gumhold, and Daniel Cremers. "Real-Time Dense Geometry from a Handheld Camera". In: *Pattern Recognition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 11–20. ISBN: 978-3-642-15986-2 (page 32).

[234] L. von Stumberg and D. Cremers. "DM-VIO: Delayed Marginalization Visual-Inertial Odometry". In: *IEEE Robot. Autom. Lett. (RA-L)* 7.2 (2022), pp. 1408–1415. DOI: 10.1109/LRA.2021.3140129 (page 34).

[235] Jiadai Sun, Yuchao Dai, Xianjing Zhang, Jintao Xu, Rui Ai, Weihao Gu, and Xieyuanli Chen. "Efficient Spatial-Temporal Information Fusion for LiDAR-Based 3D Moving Object Segmentation". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. IEEE. 2022 (page 49).

[236] Niko Sünderhauf, Kurt Konolige, Simon Lacroix, and Peter Protzel. "Visual Odometry Using Sparse Bundle Adjustment on an Autonomous Outdoor Vehicle". In: *Autonome Mobile Systeme 2005*. Ed. by Paul Levi, Michael Schanz, Reinhard Lafrenz, and Viktor Avrutin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 157–163. ISBN: 978-3-540-30292-6 (page 31).

[237] Niko Sünderhauf and Peter Protzel. "Switchable constraints for robust pose graph SLAM". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. IEEE. 2012, pp. 1879–1884 (page 36).

[238] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022 (page 26).

[239] Clark Taylor and Jason Gross. "Factor Graphs for Navigation Applications: A Tutorial". In: *NAVIGATION: Journal of the Institute of Navigation* 71.3 (2024). ISSN: 0028-1522. DOI: 10.33012/navi.653. eprint: https://navi.ion.org/content/71/3/navi.653.full.pdf. URL: https://navi.ion.org/content/71/3/navi.653 (page 18).

[240] Sebastian Thrun. "Robotic mapping: a survey". In: *Exploring Artificial Intelligence in the New Millennium*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, 1–35. ISBN: 1558608117 (page 19).

[241] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005. ISBN: 0262201623. DOI: 10.1162/artl.2008.14.2.227 (pages 15, 46).

[242] Yulun Tian, Yun Chang, Fernando Herrera Arias, Carlos Nieto-Granda, Jonathan P. How, and Luca Carlone. "Kimera-Multi: Robust, Distributed, Dense Metric-Semantic SLAM for Multi-Robot Systems". In: *IEEE Trans. Robot.* 38.4 (2022), pp. 2022–2038. DOI: 10.1109/TRO.2021.3137751 (pages 51, 106).

[243] Chi Hay Tong, Paul Furgale, and Timothy D. Barfoot. "Gaussian Process Gauss-Newton: Non-Parametric State Estimation". In: *2012 Ninth Conference on Computer and Robot Vision*. 2012, pp. 206–213. DOI: 10.1109/CRV.2012.35 (page 39).

[244] Chi Hay Tong, Paul Furgale, and Timothy D. Barfoot. "Gaussian Process Gauss–Newton for non-parametric simultaneous localization and mapping". In: *Intl. J. of Robot. Res.* 32.5 (2013) (page 66).

[245] P.H.S. Torr and A. Zisserman. "MLESAC: A New Robust Estimator with Application to Estimating Image Geometry". In: *Computer Vision and Image Understanding* 78.1 (2000), pp. 138–156. ISSN: 1077-3142. DOI: https://doi.org/10.1006/cviu.1999.0832. URL: https://www.sciencedirect.com/science/article/pii/S1077314299908329 (page 36).

[246] Marco Tranzatto, Takahiro Miki, Mihir Dharmadhikari, Lukas Bernreiter, Mihir Kulkarni, Frank Mascarich, Olov Andersson, Shehryar Khattak, Marco Hutter, Roland Siegwart, et al. "Cerberus in the darpa subterranean challenge". In: *Science Robotics* 7.66 (2022), eabp9742 (pages 43, 46).

[247] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. "Bundle adjustment—a modern synthesis". In: *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*. Springer. 2000, pp. 298–372 (page 31).

[248] Valentina Vasco, Arren Glover, and Chiara Bartolozzi. "Fast event-based Harris corner detection exploiting the advantages of event-driven cameras". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2016, pp. 4144–4149. DOI: 10.1109/IROS.2016.7759610 (pages 37, 110).

[249] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. "Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios". In: *IEEE Robot. Autom. Lett. (RA-L)* 3.2 (2018) (page 37).

[250] Matthieu Vigne, Antonio El Khoury, Florent Di Meglio, and Nicolas Petit. "State Estimation for a Legged Robot With Multiple Flexibilities Using IMUs: A Kinematic Approach". In: *IEEE Robot. Autom. Lett. (RA-L)* 5.1 (2020), pp. 195–202. DOI: 10.1109/LRA.2019.2953006 (page 44).

[251] Matthieu Vigne, Antonio El Khoury, Marine Pétriaux, Florent Di Meglio, and Nicolas Petit. "MOVIE: A Velocity-Aided IMU Attitude Estimator for Observing and Controlling Multiple Deformations on Legged Robots". In: *IEEE Robot. Autom. Lett. (RA-L)* 7.2 (2022), pp. 3969–3976. DOI: 10.1109/LRA.2022.3149025 (page 45).

[252] Deepak Geetha Viswanathan. "Features from accelerated segment test (fast)". In: *Proceedings of the 10th workshop on image analysis for multimedia interactive services, London, UK*. 2009, pp. 6–8 (pages 33, 111).

[253] Jianeng Wang and Jonathan D. Gammell. "Event-Based Stereo Visual Odometry With Native Temporal Resolution via Continuous-Time Gaussian Process Regression". In: *IEEE Robot. Autom. Lett. (RA-L)* 8.10 (2023), pp. 6707–6714. DOI: 10.1109/LRA.2023.3311374 (pages 5, 54).

[254] Jianeng Wang, Matias Mattamala, Christina Kassab, Guillaume Burger, Fabio Elnecave, Lintong Zhang, Marine Petriaux, and Maurice Fallon. "Exosense: A Vision-Based Scene Understanding System for Exoskeletons". In: *IEEE Robot. Autom. Lett. (RA-L)* 10.4 (2025), pp. 3510–3517. DOI: 10.1109/LRA.2025.3543971 (pages 5, 69).

[255] John Wang and Edwin Olson. "AprilTag 2: Efficient and robust fiducial detection". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2016 (page 85).

[256] Liqiang Wang, Hailiang Tang, Tisheng Zhang, Qijin Chen, Jinwei Shi, and Xiaoji Niu. "Improving the Navigation Performance of the MEMS IMU Array by Precise Calibration". In: *IEEE Sensors Journal* 21.22 (2021), pp. 26050–26058. DOI: 10.1109/JSEN.2021.3118455 (page 44).

[257] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE. 2017, pp. 2043–2050 (page 34).

[258] Yifu Wang, Jiaqi Yang, Xin Peng, Peng Wu, Ling Gao, Kun Huang, Jiaben Chen, and Laurent Kneip. "Visual odometry with an event camera using continuous ray warping and volumetric contrast maximization". In: *Sensors* 22.15 (2022) (page 39).

[259] Zirui Wang, Wenjing Bian, Xinghui Li, Yifu Tao, Jianeng Wang, Maurice Fallon, and Victor Adrian Prisacariu. "Seeing in the Dark: Benchmarking Egocentric 3D Vision with the Oxford Day-and-Night Dataset". In: *arXiv preprint arXiv:2506.04224* (2025) (page 5).

[260] Zirui Wang, Yangtao Ge, Kewei Dong, I-Ming Chen, and Jing Wu. "FAST-LIEO: Fast and Real-Time LiDAR-Inertial-Event-Visual Odometry". In: *IEEE Robot. Autom. Lett. (RA-L)* 10.2 (2025), pp. 1680–1687. DOI: 10.1109/LRA.2024.3522843 (page 105).

[261] David Weikersdorfer, David B Adrian, Daniel Cremers, and Jörg Conradt. "Event-based 3D SLAM with a depth-augmented dynamic vision sensor". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2014 (page 37).

[262] David Weikersdorfer, Raoul Hoffmann, and Jörg Conradt. "Simultaneous Localization and Mapping for Event-based Vision Systems". In: *ICVS*. 2013 (page 37).

[263] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. "ElasticFusion: Real-time dense SLAM and light source estimation". In: *Intl. J. of Robot. Res.* 35.14 (2016), pp. 1697–1716. DOI: 10.1177/0278364916669237 (page 84).

[264] David Wisth, Marco Camurri, and Maurice Fallon. "Vilens: Visual, inertial, lidar, and leg odometry for all-terrain legged robots". In: *IEEE Trans. Robot.* 39.1 (2022), pp. 309–326 (pages 32, 45, 99).

[265] Zhenyu Wu, Kun Zheng, Zhiyang Ding, and Hongbo Gao. "A survey on legged robots: Advances, technologies and applications". In: *Engineering Applications of Artificial Intelligence* 138 (2024), p. 109418. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2024.109418. URL: https://www.sciencedirect.com/science/article/pii/S0952197624015768 (page 40).

[266] Fabio Elnecave Xavier, Guillaume Burger, Marine Pétriaux, Jean-Emmanuel Deschaud, and François Goulette. "Multi-IMU Proprioceptive State Estimator for Humanoid Robots". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2023. DOI: 10.1109/IROS55552.2023.10341849 (page 45).

[267] Hu Xiangcheng, Wu Jin, Jiao Jianhao, Jiang Binqian, Zhang Wei, Wang Wenshuo, and Tan Ping. *MS-Mapping: An Uncertainty-Aware Large-Scale Multi-Session LiDAR Mapping System*. 2024. arXiv: 2408.03723 [cs.RO]. URL: https://arxiv.org/abs/2408.03723 (page 52).

[268] Yalin Xiong and Kenneth Turkowski. "Creating image-based VR using a self-calibrating fisheye lens". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 1997, pp. 237–243. DOI: 10.1109/CVPR.1997.609326 (page 26).

[269] N. Yang, L. von Stumberg, R. Wang, and D. Cremers. "D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 2020. arXiv: 2003.01060 [cs.CV] (page 34).

[270] Zhenyu Yin, Dan Feng, Chao Fan, Chengen Ju, and Feiqing Zhang. "SP-VSLAM:Monocular Visual-SLAM Algorithm Based on SuperPoint Network". In: *IEEE International Conference on Communication Software and Networks (ICCSN)*. 2023, pp. 456–459. DOI: 10.1109/ICCSN57992.2023.10297407 (page 34).

[271] Y.K. Yu, K.H. Wong, M.M.Y. Chang, and S.H. Or. "Recursive Camera-Motion Estimation With the Trifocal Tensor". In: *IEEE Trans. Syst. Man Cybern. Syst.* 36.5 (2006), pp. 1081–1090. DOI: 10.1109/TSMCB.2006.874133 (page 31).

[272] C. Zach, T. Pock, and H. Bischof. "A Duality Based Approach for Realtime TV-L1 Optical Flow". In: *Pattern Recognition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 214–223. ISBN: 978-3-540-74936-3 (page 32).

[273] Ming Zeng, Fukai Zhao, Jiaxiang Zheng, and Xinguo Liu. "Octree-based fusion for realtime 3D reconstruction". In: *Graphical Models* 75.3 (2013). Computational Visual Media Conference 2012, pp. 126–136. ISSN: 1524-0703. DOI: https://doi.org/10.1016/j.gmod.2012.09.002. URL: https://www.sciencedirect.com/science/article/pii/S1524070312000768 (page 46).

[274] Ji Zhang and Sanjiv Singh. "LOAM: Lidar Odometry and Mapping in Real-time". In: *Robotics: Science and Systems (RSS)*. 2014 (page 2).

[275] Lintong Zhang, David Wisth, Marco Camurri, and Maurice Fallon. "Balancing the Budget: Feature Selection and Tracking for Multi-Camera Visual-Inertial Odometry". In: *IEEE Robot. Autom. Lett. (RA-L)* 7.2 (2022), pp. 1182–1189. DOI: 10.1109/LRA.2021.3137910 (pages 2, 32).

[276] Zichao Zhang, Henri Rebecq, Christian Forster, and Davide Scaramuzza. "Benefit of large field-of-view cameras for visual odometry". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2016, pp. 801–808. DOI: 10.1109/ICRA.2016.7487210 (page 2).

[277] Yi Zhou, Guillermo Gallego, and Shaojie Shen. "Event-based stereo visual odometry". In: *IEEE Trans. Robot.* 37.5 (2021) (page 38).

[278] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. "Event-based feature tracking with probabilistic data association". In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2017, pp. 4465–4470 (page 110).

[279] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. "NICE-SLAM: Neural Implicit Scalable Encoding for SLAM". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 2022 (page 34).

[280] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. "Event-based visual inertial odometry". In: *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 2017 (page 38).