Multi-Sensor and Multi-Modal Localization in Indoor Environments on Robotic Platforms

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurswissenschaften

von der KIT-Fakultät für Informatik des
Karlsruher Instituts für Technologie (KIT)
genehmigte
Dissertation

von
Marco Sewtz
geboren am 04. November 1992 in Herne

Tag der mündlichen Prüfung: 12. Februar 2025
Erster Gutachter: Prof. Rudolph Triebel
Zweiter Gutachterin: Prof. Teresa Vidal Calleja

Abstract

This dissertation presents a detailed exploration of multi-sensor visual odometry systems, with a focus on enhancing robustness in indoor environments. The core contribution lies in the development of an advanced ego-state estimation framework, wherein visual odometry is bolstered by multiple visual sensors to overcome common challenges such as occlusions and textureless surfaces. By addressing frequent loss-of-tracking (LoT) events, the system ensures continuous, reliable localization in complex, cluttered indoor settings, such as households and elderly care facilities.

To further augment situational awareness, sound source localization (SSL) is integrated as a complementary modality. Its fusion with visual data significantly enhances the robot's perception of the environment, enabling the detection and identification of objects and events that may be visually occluded or otherwise undetectable. This multi-modal fusion provides a more holistic understanding of the robot's surroundings, contributing to improved operational reliability in dynamic, human-centered environments.

A key feature of this research is the introduction of IndoorMCD, a novel multi-sensor benchmark specifically designed to evaluate localization performance in indoor environments. Additionally, this work introduces URSim, an online real-time visual simulation framework that enables rigorous testing of multi-sensor localization systems under various conditions. Extensive experimental validation, using both real-world scenarios and simulated environments, demonstrates the robustness and fault tolerance of the proposed system.

This research advances the state-of-the-art in robotic perception and indoor localization by providing a multi-modal, fault-tolerant approach to localization, offering valuable contributions to both theoretical understanding and practical application in robotics.

Zusammenfassung

Diese Dissertation präsentiert eine detaillierte Untersuchung von Multi-Sensor-Visual-Odometrie-Systemen, mit einem Fokus auf die Erhöhung der Robustheit in Innenräumen. Der zentrale Beitrag liegt in der Entwicklung eines Frameworks zur Ego-Zustandsschätzung, bei dem die visuelle Odometrie durch mehrere visuelle Sensoren gestärkt wird, um häufige Herausforderungen wie Verdeckungen und texturlose Oberflächen zu überwinden. Durch die Bewältigung häufiger Loss-of-Tracking (LoT) Events gewährleistet das System eine kontinuierliche und zuverlässige Lokalisierung in komplexen Innenumgebungen, wie zum Beispiel Haushalten und Pflegeeinrichtungen.

Zur weiteren Steigerung des Situationsbewusstseins wird die Schallquellenlokalisierung (SSL) als komplementäre Modalität integriert. Die Fusion von SSL mit visuellen Daten verbessert die Wahrnehmung des Roboters erheblich und ermöglicht die Erkennung und Identifikation von Objekten und Ereignissen, die visuell verdeckt oder auf andere Weise nicht wahrnehmbar sind. Diese multi-modale Fusion bietet ein ganzheitlicheres Verständnis der Umgebung des Roboters und trägt zur verbesserten Betriebssicherheit in dynamischen, menschenzentrierten Umgebungen bei.

Ein zentraler Bestandteil dieser Forschung ist die Einführung von IndoorMCD, eines neuartigen Multi-Sensor-Benchmarks, der speziell zur Bewertung der Lokalisierungsleistung in Innenräumen entwickelt wurde. Darüber hinaus wird URSim, eine Online-Echtzeit-Visual-Simulation, vorgestellt, der eine Entwiclung von Multi-Sensor-Lokalisierungssystemen unter verschiedenen Bedingungen ermöglicht. Umfangreiche experimentelle Validierungen, sowohl in realen Szenarien als auch in simulierten Umgebungen, zeigen die Robustheit und Fehlertoleranz des vorgeschlagenen Systems.

Diese Arbeit treibt den Stand der Technik in der Wahrnehmung und Innenlokalisierung voran, indem sie einen multi-modalen, fehlertoleranten Ansatz für die Lokalisierung bietet und Beiträge sowohl zum theoretischen Verständnis als auch zur praktischen Anwendung in der Robotik leistet.

Contents

Abstract . . . . **i**

Zusammenfassung . . . . **iii**

List of Figures . . . . **ix**

List of Tables . . . . **xi**

List of Acronyms . . . . **xiii**

Disclosure of Artificial Intelligence Tool Usage . . . . **xvii**

1. Introduction **1**
- 1.1. A Brief History of Mobile Robotic Platforms 1
- 1.2. Robotic Assistance 2
- 1.3. The Importance of Perception in Robotic Assistance 4
- 1.4. Problem Statement 5
- 1.5. Contributions 8
- 1.6. Outline 9

2. Related Work **13**
- 2.1. Visual Odometry 14
- 2.2. Audio Localization 17
- 2.3. Robotic Proprioception 20
- 2.4. Combined Audio-Visual Approaches 21
- 2.5. Literature Discussion 23

3. Multi-Modal Perception of Indoor Environment **25**
- 3.1. Visual Perception 26
- 3.2. Audio Perception 34
- 3.3. Robotic Proprioception 41

4. Audio-Visual Architecture **47**
4.1. Measurement Data Abstraction Layer 48
4.1.1. Visual Sensor Drivers 48
4.1.2. Microphone Array Drivers 51
4.2. Robot Configuration 57
4.3. Benchmarks and Evaluation Data 60
4.3.1. Dataset 61
4.3.2. Simulation 65
4.4. Summary 68

5. Ego-State Estimation **69**
5.1. Visual Odometry 70
5.2. Visual Odometry Fusion 79
5.3. Mapping 84

6. Landmark Estimation **89**
6.1. Landmarks 89
6.1.1. Visual Landmarks 90
6.1.2. Sound Sources 91
6.2. Landmark Pose Batch Optimization 96
6.3. Audio-Visual Information Fusion 98

7. Evaluation **101**
7.1. Multi-Sensor Approaches 101
7.2. Indoor Robot Audition 107
7.3. Audio-Visual Localization 111

8. Conclusion **117**
8.1. Summary 117
8.2. Outlook 119

Bibliography **123**

A. Publications **137**
1. IndoorMCD: A Benchmark for Low-Cost Multi-Camera SLAM in Indoor Environments 138
2. Representing Uncertain Spatial Transformations in Robotic Applications in a Structured Framework Leveraging Lie Algebra 148
3. Robust MUSIC-Based Sound Source Localization in Reverberant and Echoic Environments 158
4. Robust Approaches for Localization on Multi-Camera Systems in Dynamic Environments 166
5. URSim - A Versatile Robot Simulator for Extra-Terrestrial Exploration 172
6. Audio Perception in Robotic Assistance for Human Space Exploration: A Feasibility Study 188
7. A Structured Approach for Uncertain Transformations Trees 200
8. Loosely-Coupled Multi-Sensor Visual Odometry: An Asynchronous Approach for Robust Household Robotics 206
9. Design of a Microphone Array for Rollin' Justin 216
10. Sound Source Localization for Robotic Applications 220

List of Figures

1.1. Proposed architecture 10

3.1. Camera obscura 27
3.2. Pinhole model 28
3.3. Real apartments 34
3.4. Microphone array 37
3.5. Rollin Justin microphone array 39
3.6. Audio shadow source 40
3.7. Kinematic tree of Rollin' Justin 43
3.8. Bending TINA arm 46

4.1. Measurement data abstraction layer architecture 47
4.2. Ego-noise estimation 55
4.3. Coordinate frames 58
4.4. Capture devices in the dataset 62
4.5. Dataset panoramic views 65
4.6. Simulated Martian surface 66
4.7. URSim software architecture 67
4.8. Simulated indoor environment 68

5.1. Ego-state estimation module architecture 69
5.2. Feature suppression results 73
5.3. Feature merging 76
5.4. Illustration of trajectory estimation 81
5.5. Illustration of track fusion 83
5.6. Typical pose graph 86
5.7. Map transformation estimation 87
5.8. Feature maps 88

6.1. Landmark estimation module architecture 89
6.2. Exemplary AprilTags 90

6.3. MUSIC pseudospectrum 93
6.4. Adaptive frequency bin selection 95
6.5. Batch optimization for landmark estimation 97
6.6. Visibility projection 98

7.1. Multi-sensor map example 103
7.2. Loss-of-tracking scenarios 103
7.3. Recovery after loss-of-tracking events 105
7.4. Number of observed features for the multi-sensor approach . . . 106
7.5. Experimental audio setup 107
7.6. Lecture hall results 109
7.7. Entrance hall results 110
7.8. Speaker segmentation 110
7.9. Segmentation evaluation results 111
7.10. Instrument maintenance example 112
7.11. Exemplary evaluation result for the audio source detection . . . 113
7.12. Spectral analysis of audio event 115

List of Tables

7.1. Robustness evaluation on IndoorMCD 105
7.2. Room reverberation times 108
7.3. Direction-of-Arrival evaluation results 108

List of Acronyms

Slerp Spherical Linear Interpolation

PTP Precision Time Protocol

NTP Network Time Protocol

LRU Lightweight Rover Unit

IMU inertial measurement unit

ToF time of flight

FoV field-of-view

CTC Connectionist Temporal Classification

Seq2Seq Sequence-to-Sequence

ORB Oriented FAST and Rotated BRIEF

SIFT The Scale-Invariant Feature Transform

SURF Speeded-Up Robust Features

AGAST Adaptive and Generic Accelerated Segment Test

BRIEF Binary Robust Independent Elementary Features

FLANN Fast Library for Approximate Nearest Neighbors

SAM smoothing and mapping

DFT Discrete Fourier Transform

SFTF Short-Time Fourier Transform

LTSE long-term spectral envelope

LTSD long-term spectral divergence

IPD interaural phase difference

IID interaural intensity difference

HRTF head-related transfer function

DoA direction of arrival

SNR signal-to-noise ratio

SSL sound source localization

PSD power spectral density

LTSE Long-Term Spectral Envelope

LTSD Long-Term Spectral Divergence

GSVD Generalized Singular Value Decomposition

AFRF Active Frequency Range Filtering

MME Motion-Model Enhanced MUSIC

SNR signal-to-noise ratio

DoA direction of arrival

TDOA time delay of arrival

MUSIC multiple signal classification

ESPRIT estimation of signal parameters via rotational invariant techniques

VO visual odometry

LoT loss-of-tracking

SLAM simultaneous localization and mapping

FAST Features from Accelerated Segment Test

RT AMNS range tree adaptive non-maximal suppression

SotA state-of-the-art

MDAL measurement data abstraction layer

ESE ego-state estimation

LE landmark estimation

CAD computer-aided design

SiL Software-in-the-Loop

SVD singular value decomposition

VCS visible cross-section

URDF Unified Robot Description Format

Disclosure of Artificial Intelligence Tool Usage

Artificial intelligence (AI) tools were employed in this dissertation for the purposes of spell checking, grammar correction, and typesetting. These tools were used exclusively to enhance the clarity and accuracy of the written text. No artificial intelligence was involved in the development of the research methodology, code, or supplementary materials. All substantive content, including theoretical frameworks and technical contributions, was independently developed.

1. Introduction

1.1. A Brief History of Mobile Robotic Platforms

The development of mobile robotic platforms has been pivotal in advancing autonomous systems. Early designs focused on simple tasks such as navigating predefined paths in controlled environments.

The 1960s saw the results of Shakey, developed by the Stanford Research Institute. Shakey was capable of perceiving and interacting with its environment using a combination of a television camera, laser rangefinder, and bump sensors. Shakey's ability to navigate and perform tasks autonomously marked a significant milestone in robotics research [99].

In the 1980s, the Stanford Cart, developed at Stanford University, implemented basic obstacle avoidance and path planning. The cart used a single camera to build a three-dimensional model of its surroundings, allowing it to navigate through simple environments. The Stanford Cart's contribution to robotics included early implementations of computer vision and autonomous navigation [87].

The evolution continued with the development of more sophisticated systems, such as NASA's Sojourner rover in the 1990s. Sojourner demonstrated semi-autonomous navigation on the Martian surface using a combination of stereoscopic cameras and hazard detection algorithms. This mission provided valuable insights into the challenges of operating robots in extraterrestrial environments and highlighted the importance of reliable sensor data processing and autonomous decision-making [80].

The 21st century has seen significant advancements in sensor technology, computational power, and control algorithms, leading to highly capable mobile robots that can perform complex tasks in dynamic environments. For instance, Boston Dynamics' Spot robot integrates advanced perception systems, robust navigation algorithms, and dynamic balancing capabilities to operate in diverse and challenging terrains. Similarly, the PackBot by iRobot has been deployed in military applications, showcasing the robustness and versatility of modern mobile robotic platforms [34, 55].

These developments have laid the foundation for the current generation of mobile robotic platforms, which integrate advanced perception, navigation, and control capabilities. Modern robots are equipped with a wide array of sensors and utilize cutting-edge algorithms to achieve high levels of autonomy and reliability in various applications. As these technologies continue to mature, mobile robotic platforms are increasingly being utilized in robotic assistance roles, supporting humans in tasks ranging from industrial automation to healthcare.

1.2. Robotic Assistance

Robotic assistance includes a broad range of applications where robots aid humans in various tasks. The history of robotic assistance can be traced back to the early 20th century with the introduction of simple mechanical aids and automated machines in industrial settings. In the 1960s and 1970s, industrial robots such as Unimate were introduced into manufacturing environments. Unimate, the first industrial robot, was deployed on a General Motors assembly line in 1961 to handle tasks such as welding and material handling. This marked the beginning of the automation revolution in manufacturing, leading to increased efficiency and safety by taking over repetitive and hazardous tasks from human workers [31].

The 1980s and 1990s saw the emergence of service robots designed to assist in non-industrial environments. Robots like HelpMate were developed for use in hospitals to transport medical supplies and equipment, demonstrating the potential for robots to improve efficiency and reduce the workload on hospital staff [115]. During this period, domestic robots also began to appear, with devices like the Electrolux Trilobite, an early robotic vacuum cleaner, entering the market in the late 1990s [37].

The early 21st century witnessed the rise of collaborative robots, or cobots, designed to work alongside humans in shared spaces. Cobots like lightweight robot by the German Aerospace Center (DLR) [4] or the Baxter robot by

Rethink Robotics [40] are equipped with advanced safety features and user-friendly interfaces, making them suitable for a variety of tasks in manufacturing, logistics, and other industries. These robots enhance productivity by assisting with tasks that require precision, strength, or endurance [106].

In recent years, there has been significant interest in developing household robots that can assist with everyday tasks. Robotic vacuum cleaners, such as those from iRobot's Roomba series, have gained widespread popularity due to their ability to autonomously clean floors. However, despite these advancements, the adoption of household robots beyond cleaning remains limited. Robots capable of performing complex household chores, such as cooking, laundry, and organizing, are still in the experimental stage and face numerous technical challenges.

One notable example of household robotic assistance is the Jibo robot, designed as a social companion to interact with family members, provide reminders, and control smart home devices. Although Jibo showcased the potential for social robots in domestic settings, it faced market challenges and was discontinued in 2019 [54]. Another example is the Aido robot, which integrates smart home control, entertainment, and personal assistance features but has struggled to achieve mainstream adoption [35].

The limited presence of advanced household robots can be attributed to several factors, including the complexity of domestic environments, high costs, and the need for robust perception algorithms. While industrial robots have thrived in structured and predictable settings, household environments present diverse and dynamic challenges that require sophisticated perception and decision-making capabilities. Moreover, the economic feasibility of deploying such robots in homes remains a significant barrier.

Despite these challenges, ongoing research and development efforts continue to explore the potential of household robots. Projects like the Smile2Gether [45] or Toyota Human Support Robot [27] aim to assist elderly and disabled individuals with daily activities, such as fetching objects and opening doors, demonstrating the potential for robots to improve quality of life in domestic settings.

1.3. The Importance of Perception in Robotic Assistance

Perception is a critical component for the reliability and success of robotic assistance, particularly in household environments. Robots must be able to accurately sense and interpret their surroundings to perform tasks effectively. This involves the integration of various sensors, such as cameras and ultrasonic sensors, to gather data about the environment. Advanced perception algorithms process this sensory data to enable robots to recognize objects, detect obstacles, and navigate through complex and dynamic spaces [136].

In industrial settings, robots operate in structured and robot-friendly environments where tasks are repetitive and predictable. These environments are specifically designed and prepared for robots, with controlled layouts and minimal unexpected changes. However, household environments are solely designed for humans, making them inherently unstructured and dynamic, which presents unique challenges for robotic perception. For instance, a robot in a home must be able to identify and interact with a wide range of objects, from furniture to kitchen utensils, and adapt to changes in the environment, such as people moving objects around [49].

Robust perception approaches are necessary to ensure that robots can handle these challenges. This includes the development of algorithms that can fuse data from multiple sensors to create a comprehensive understanding of the environment. Techniques such as simultaneous localization and mapping (SLAM) allow robots to build and update maps of their surroundings in real-time while keeping track of their own location. Vision, using cameras, and audio perception are particularly important modalities for household robots. Vision enables detailed recognition and classification of objects, while audio helps in identifying and locating sound sources, which is fundamental for tasks like responding to voice commands or detecting unknown sound profiles [19].

The reliability of robotic assistance in households depends on the ability to perceive and respond to the environment accurately. For example, a cleaning robot must detect and avoid obstacles like furniture and toys while efficiently navigating the space. Similarly, a social robot must recognize human emotions and adapt its interactions accordingly. Ensuring robust perception capabilities is crucial for the widespread adoption and success of household robots [13].

As research and development in perception technologies advance, the capabilities of household robots are expected to improve significantly. Enhanced perception will enable robots to perform a broader range of tasks, from simple chores to complex caregiving activities, thereby increasing their utility and value in domestic settings. The continuous refinement of perception algorithms and sensor technologies will play an important role in the future of robotic assistance in homes.

1.4. Problem Statement

This work will focus on visual odometry techniques aimed at achieving robust ego-motion estimation in indoor scenarios. The primary goal is to enhance perception accuracy in environments with varying visual characteristics, such as homes and elderly care facilities. Robust motion estimation is crucial in these settings due to the presence of dynamic and cluttered elements like furniture and people, as well as varying lighting conditions. Accurate motion tracking ensures that the robot can navigate safely and efficiently, reducing the risk of collisions and ensuring smooth operation in complex, changing environments.

Additionally, this thesis will explore how integrating multiple sensor modalities, such as visual and audio data, can provide a comprehensive understanding of the robot's surroundings. By combining data from different sensors, the system can generate a more detailed environmental representation, addressing limitations posed by using a single modality. In human-centered environments like elderly care, accurate motion estimation becomes even more essential, as errors could lead to safety risks or decreased trust in robotic systems. Ensuring robust performance helps the robot adapt effectively to its surroundings and the movements of people within the space.

The research will also focus on the development of approaches to improve robustness, which is critical in human-centered environments. Failures in such settings could lead to injuries or rejection of robotic systems, necessitating a strong emphasis on fault tolerance and reliability. Robots operating in elderly care facilities, for instance, must rely on accurate motion estimation to avoid accidents, especially in environments that may contain moving objects or individuals.

Lastly, the specific constraints of the thesis revolve around designing solutions for indoor environments. The robotic system must be capable of operating effectively within households and apartments, without the need to significantly modify the environment. This includes ensuring functionality in confined, cluttered spaces and adapting to modern interior designs, such as textureless or repetitive surfaces. The use of multiple sensors enhances robustness by ensuring that the failure of a single sensor does not result in the failure of the entire system. In indoor environments, tracking loss often occurs due to view-dependent factors, such as occlusions or limited fields of view. A multi-sensor approach with multiple views effectively mitigates this issue by providing complementary perspectives, reducing the likelihood of tracking failures and ensuring continuous, reliable operation in complex spaces.

The considered challenges of this thesis are as follows:

Indoor Environment

The target domain in this work is the indoor environment, particularly households and apartments. This poses unique challenges, as the robot must seamlessly operate without requiring modifications to the existing space. The system must be capable of functioning in confined, cluttered areas while integrating the modern design elements typical of indoor environments, such as minimalistic decor and textureless surfaces. These characteristics often complicate perception tasks, as textureless or repetitive surfaces provide fewer visual cues for reliable localization and mapping. Therefore, the perception architecture must be developed to handle these constraints effectively, ensuring robust operation even in spaces where visual features are sparse or ambiguous.

Audio-Visual Perception System

Visual perception provides rich and detailed information about the robot's surroundings, such as spatial structure, object recognition, and motion tracking. It forms the foundation for many key tasks in robotic navigation and interaction. However, visual data alone may not capture all aspects of the environment, especially in dynamic and multi-source settings, where complementary data from other modalities can greatly enhance overall perception.

Audio perception offers an important supplement to visual sensing, as it adds an additional dimension to the robot's understanding of its environment. Unlike vision, audio can detect sound sources regardless of their visibility, enabling the detection of events and objects that may be outside the robot's field of view or temporarily occluded. For instance, the ability to localize sounds, such as a person speaking or a machine operating, provides spatial and contextual cues that complement visual information. This multimodal integration allows the robot to track and interact with multiple sources of information, creating a more robust and adaptive perception system.

By combining audio and visual data, the system benefits from the strengths of both modalities. Audio extends the operational range of perception by capturing environmental cues that are not available through vision alone, especially in cases where visual information might be ambiguous or incomplete. This fusion results in a more holistic understanding of the environment, ensuring that the robot can reliably perceive and respond to its surroundings in real time.

Multi-Sensor Framework

The integration of multiple sensors and modalities plays a pivotal role in enhancing system robustness and reliability. By using multiple sensors, both within the same modality and across different types, the system gains redundancy that can safeguard against single sensor failures. For instance, if one visual sensor fails or provides erroneous data, the remaining sensors can continue to provide input, allowing the robot to maintain its tasks without interruption.

However, implementing a multi-sensor framework introduces additional complexity. It requires precise synchronization of sensor data and accurate fusion of information from sensors positioned at different locations on the robot. This is especially critical in scenarios involving non-rigid transformations between sensors, where the current configuration of the robot must be estimated in real time. Ensuring that the data streams are synchronized and correctly aligned allows the system to create a coherent and accurate representation of the environment, supporting robust decision-making and control.

Robust Operation in Proximity of Obstacles

For a robotic system operating in indoor environments, especially in households or elderly care settings, robust operation near obstacles is crucial. The perception system must reliably navigate close to furniture, walls, and other objects without losing track of its surroundings or making dangerous miscalculations. Proximity to obstacles often limits the availability of visual information, and reflections from shiny surfaces can confuse the system, leading to incorrect motion estimates.

In such challenging conditions, the system must maintain continuous operation, even in the presence of minor sensor failures or occlusions. This requires the ability to detect and correct for errors dynamically, allowing the robot to resume tasks without requiring human intervention. Additionally, the system must remain resilient to environmental changes and capable of adapting to unforeseen situations, ensuring reliable operation in real-world, dynamic environments where obstacles and human interaction are frequent.

1.5. Contributions

This thesis makes several contributions to the field of multi-modal and multi-sensor localization in indoor environments. First, a robust multi-sensor visual localization system was developed, which integrates multiple sensors to ensure accurate and continuous localization in cluttered and confined indoor spaces. This system mitigates frequent loss-of-tracking (LoT) events and preserves localization accuracy, addressing the critical need for robust motion estimation. An overview of this system's architecture is presented in Figure 1.1, highlighting the components involved in the data acquisition and ego-state estimation. This contribution directly supports safe and efficient navigation in dynamic household and elderly care environments, as highlighted in the problem statement in Section 1.4 on page 5.

Secondly, this work proposes the integration of audio and visual modalities for localization. A real-time, motion-aware sound source localization (SSL) system, using adaptive frequency selection, was developed to complement visual perception. This system operates effectively in reverberant and occluded environments, enhancing the robot's understanding of its surroundings. By combining visual and auditory data, the system addresses the challenge of multi-modal perception, improving robustness and situational awareness in complex indoor settings. Figure 1.1 illustrates how the audio and visual sensor drivers feed into the landmark estimation module to support this multi-modal localization.

A third key contribution is the creation of the first multi-sensor benchmark for visual odometry (VO) and simultaneous localization and mapping (SLAM) systems specifically designed for indoor environments. This benchmark enables rigorous evaluation of multi-sensor localization systems, ensuring they meet the unique challenges posed by indoor applications. It directly supports the need to evaluate multi-sensor performance, particularly in environments that are cluttered or lack distinct visual features.

Additionally, a multi-sensor simulation framework was developed and published, providing tools for testing and evaluating perception systems in a variety of indoor environments, using Unreal Engine 4. This framework allows for robust system development in spaces with minimalistic decor or textureless surfaces, addressing the challenge of operating in indoor environments with sparse or ambiguous visual cues. The framework is represented in the measurement data abstraction layer shown in Figure 1.1, where the system supports both live and pre-recorded datasets for evaluation.

Finally, extensive experimental evaluations of the proposed system were conducted, demonstrating the robustness of audio-visual fusion for ego-state estimation and sound source localization. These experiments showcase the system's ability to maintain continuous perception and operation in real-world indoor environments, even when facing obstacles or sensor failures. As seen in Figure 1.1, the architecture facilitates real-time processing of visual and audio data for robust ego-state estimation and landmark localization. The results demonstrate the system's ability to overcome challenges related to obstacle proximity and multi-sensor integration, ensuring reliable and fault-tolerant operation in complex household environments.

1.6. Outline

Chapter 2 on page 13 reviews the related work in the fields of visual perception, audio perception, robotic proprioception, and combined audio-visual approaches. This chapter provides an overview of existing methodologies

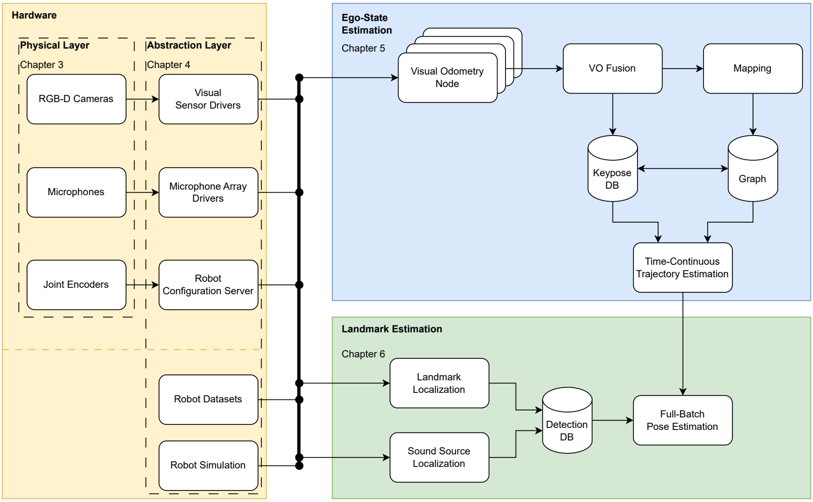


Figure 1.1.: Overview of the different sub-components of the proposed audio-visual perception system. On the left-hand side, the measurement data abstraction layer in yellow, the ego-state estimation in blue, and landmark estimation in green.

and discusses their relevance to multi-sensor and multi-modal localization in indoor environments.

Chapter 3 on page 25 focuses on the machine perception components of the system. It covers the visual, audio, and proprioceptive perception modules, detailing their individual roles and how they contribute to the overall perception architecture.

Chapter 4 on page 47 presents the audio-visual architecture developed in this work. This includes the measurement data abstraction layer (MDAL), the robot configuration, and the hardware emulation components. The integration of these elements ensures the synchronization and processing of multi-sensor data.

Chapter 5 on page 69 discusses the ego-state estimation process, with a focus on the visual odometry and mapping techniques used to maintain accurate localization in indoor environments. The fusion of multi-sensor data is also covered, ensuring the system's robustness in dynamic and cluttered spaces.

Chapter 6 on page 89 addresses landmark estimation, explaining the methods used for detecting and localizing visual and audio landmarks. The process of fusing these data types to improve environmental modeling is also presented.

Chapter 7 on page 101 provides the evaluation of the proposed system. Various experiments are conducted to assess the performance of multi-sensor approaches, including the system's ability to handle loss-of-tracking events and its robustness in audio-visual localization tasks.

Finally, Chapter 8 on page 117 concludes the thesis, summarizing the contributions and presenting avenues for future work. This chapter reflects on the findings and discusses potential improvements and extensions for future research.

2. Related Work

Localization in indoor environments requires a detailed understanding of various perception methods. In this chapter, the related work across multiple modalities will be discussed, starting with visual perception, followed by audio perception, proprioception, and finally, multi-modal and multi-sensor approaches. Each modality presents distinct advantages and limitations, which have been addressed in various studies.

For visual perception, the focus will be on visual odometry (VO) techniques. However, VO is often related to a full SLAM system, therefore a lot of related work is part of more complex mapping approaches as well. These methods rely on visual data to estimate the position and orientation of the robot, often in environments where clear and consistent visual landmarks are essential.

Audio perception has gained attention for its ability to localize sound sources, particularly in environments where visual data is insufficient or unavailable. By leveraging directional audio cues, robots can enhance their understanding of the surroundings and improve interaction capabilities, especially in dynamic and cluttered indoor settings.

Proprioception plays a critical role when fusing audio and visual data, particularly in scenarios where sensors are mounted at different locations on the robotic platform. The robot's kinematic structure introduces dynamic transformations between the sensor frames, making it necessary to have accurate internal sensing. Information from joint encoders enable the system to track these transformations in real-time, ensuring that data from different modalities can be correctly aligned.

Finally, multi-modal approaches combine data from visual, auditory, and proprioceptive sensors, creating a more robust system for localization and mapping. These approaches aim to overcome the limitations of single-modality systems, offering improved performance in complex and variable environments.

The following sections will review the state-of-the-art methods and systems that address the challenges of localization and perception across these modalities, with a focus on their application to indoor robotic platforms.

2.1. Visual Odometry

Visual odometry is a foundational technique in robotic perception, aimed at estimating a robot's motion by analyzing visual input. VO systems infer the camera's trajectory by tracking changes in images over time, which is particularly useful when a globally consistent map is not required, or when mapping is handled by separate processes. Compared to simultaneous localization and mapping, VO focuses exclusively on accurate motion tracking, making it a lighter-weight solution for real-time applications.

Feature-based methods have been central to VO development. These methods track distinct points in the environment, such as corners or edges, to compute the camera's movement. One of the most widely adopted approaches is ORB-SLAM, introduced by Mur-Artal et al. [91], which builds on ORB features for robust, efficient tracking. This algorithm achieves high accuracy while maintaining computational efficiency, a key requirement for real-time systems. Feature-based VO methods remain popular due to their relatively low computational complexity and adaptability in diverse environments, from structured indoor spaces to outdoor settings.

To enhance the robustness and accuracy of VO, significant research has focused on addressing visual ambiguities, such as repetitive patterns or dynamic objects in the scene. For example, Klein et al. [63] introduced parallel processing for separating the tasks of tracking and mapping. By assigning one thread to track camera pose and another to build a 3D map, they improved the real-time performance of VO, particularly for mobile applications. They further proposed the Keyframe approach, where only selected frames are used for optimization, reducing the computational load while maintaining the accuracy of pose estimates. The concept of Keyframe has since become integral to many VO and SLAM systems, as it balances real-time performance with map optimization.

However, feature-based approaches encounter limitations in environments with low texture or poor lighting, where distinctive visual features are scarce.

This led to the development of dense VO methods, which utilize the entire image rather than isolated features. Dense methods directly exploit pixel intensities to estimate motion, providing more precise tracking in textureless regions. For instance, Steinbrucker et al. [126] and Kerl et al. [61] were among the first to demonstrate dense approaches for visual odometry. Their algorithms minimize photometric error between frames, improving robustness in environments where feature-based techniques struggle. Engel et al. [39] advanced this further by introducing a fully direct VO method that tracks every pixel, enabling highly accurate motion estimation even in scenes with few identifiable features.

The computational demands of dense methods are a key challenge, especially for mobile systems where processing power and battery life are limited. While dense VO methods provide greater accuracy, they often require hardware acceleration, such as GPUs, to run in real-time. This limitation was addressed by Whelan et al. [146], who introduced a room-scale dense VO algorithm that reduces reliance on GPUs while maintaining high performance in larger environments. Despite these advancements, feature-based approaches like ORB-SLAM continue to be favored for systems with constrained resources due to their efficiency and ability to operate without specialized hardware.

Another major area of research in VO involves multi-sensor fusion. By combining visual data with other sensor modalities, such as inertial measurement units (IMUs), it is possible to improve robustness and mitigate the weaknesses of purely visual approaches. For example, incorporating IMU data helps counteract the effects of rapid motion or low-texture environments, where visual tracking alone may fail. Forster et al. [42] introduced a method that integrates IMU data with visual information into a manifold representation, significantly improving the accuracy of motion estimation, especially in challenging conditions.

In multi-sensor setups, VO systems can track motion across multiple cameras or integrate data from cameras mounted at different locations on a robotic platform. Müller et al. [93] proposed a system that asynchronously samples Keyframe from different cameras, ensuring robust tracking even when the viewpoints are significantly different or not synchronized. This approach is particularly valuable in dynamic environments where different sensors may encounter varying visual conditions. Furthermore, Zhao et al. [153] and Meng et al. [82] developed tightly coupled multi-sensor systems that fuse data from a central sensor with additional visual inputs, enhancing the system's ability to track motion reliably in cluttered or dynamic environments.

These advancements in visual perception have made VO a critical tool for robotic systems, particularly for mobile platforms that need to operate in real-time while conserving computational resources. Innovations such as Keyframe-based optimization, dense photometric tracking, and multi-sensor fusion have expanded the capabilities of VO, making it suitable for a wide range of applications, from autonomous navigation to human-robot interaction in dynamic indoor environments. By focusing on efficient motion estimation, VO systems continue to play a central role in visual perception for robots, especially in scenarios where lightweight, accurate, and robust solutions are required.

Previous research in visual odometry has largely focused on single-camera systems, with methods such as proposed by Mur-Artal et al. and Forster el al., or dense approaches, as proposed by Steinbrucker et al. and Engel et al., aiming to improve robustness under challenging conditions. These systems typically rely on feature-based methods or dense pixel tracking, optimized for environments with sufficient texture and favorable lighting. However, they often face significant limitations in robustness when applied to cluttered, or low-texture environments, especially in single-camera setups.

The approach presented in this work introduces a multi-sensor visual odometry system that integrates keyframe-based techniques with feature extraction to optimize performance and enhance robustness. Unlike single-camera systems, which are more prone to loss of tracking in occluded or low-texture scenarios, this system leverages sensor fusion, combining data from multiple visual sensors to ensure reliable operation even in complex environments.

A key distinguishing feature is its ability to handle individual sensor failures without compromising overall motion estimation. Previous work in single-camera visual odometry often experiences significant performance degradation or complete loss of tracking in the event of occlusions or sensor malfunctions, requiring manual resets or intervention. In contrast, the system presented here utilizes a loosely coupled architecture, where each sensor operates semi-independently. If a sensor fails, the remaining sensors continue to provide accurate data, allowing seamless reintegration of the failed sensor once it recovers. This capability, demonstrated through testing with the IndoorMCD dataset [118], ensures continuous localization and navigation even in the presence of sensor dropouts, offering greater fault tolerance compared to prior single-camera approaches aimed at enhancing robustness.

2.2. Audio Localization

Early research in sound source localization focused on imitating binaural audio perception in humans and animals. Jeffress [57] proposed a theory based on the interaural phase difference (IPD) of sound waves reaching both ears. Huang et al. [53] designed auditory systems for robots, emphasizing sound localization and separation using interaural intensity difference (IID). Nakadai et al. [96] developed methods for real-time tracking of multiple objects using auditory and visual inputs in humanoid robots, and later discussed enhancing sound source localization with scattering theory [95].

The inclusion of the head-related transfer function (HRTF) and environmental reverberation modeling further increased robustness. MacDonald [75] introduced a localization algorithm utilizing HRTFs. Keyrouz and Naous [62] proposed a method for three-dimensional sound localization using binaural hearing and HRTFs. Kossyk et al. [66] discussed tracking stationary sound sources in reverberant environments. These approaches require accurate calibration, where deviations in environmental modeling significantly impact results.

Successive work focused on the direction of arrival (DoA) estimation. Valin et al. [141] developed robust methods for sound source localization using a microphone array on a mobile robot. Later, they presented a method for localizing multiple moving sound sources using a frequency-domain steered beamformer [140]. These approaches estimate direction using time delays between sensor inputs but face challenges in low signal-to-noise ratio (SNR) environments.

Deep learning approaches promise to overcome these issues but require specific training data or large datasets for generalization. Mumolo et al. [90] presented algorithms for acoustic localization in service robotics. Roden et al. [107] used deep neural networks for speech signal localization. Adavanne et al. [1] proposed a convolutional recurrent neural network for DoA estimation. Xiao et al. [147] introduced a learning-based approach to DoA estimation in noisy environments, and Takeda and Komatani [134] proposed discriminative multiple sound source localization using deep neural networks.

Recent research has shifted towards subspace-based approaches like multiple signal classification (MUSIC) and estimation of signal parameters via rotational invariant techniques (ESPRIT). Schmidt [114] introduced the MUSIC algorithm for emitter location and signal parameter estimation. Roy and Kailath [109] developed the ESPRIT technique for signal parameter estimation via rotational invariance. These methods offer increased robustness and angular resolution, overcoming sampling frequency limitations. Argentieri and Danes [5] discussed broadband variations of the MUSIC method for robotics. Asono et al. [6] addressed sound source localization and signal separation for office robots. Ishi et al. [56] evaluated the real-time application of MUSIC in noisy environments.

Acoustic monitoring is well-established in ecological research, especially for ornithology. Sugai et al. [131] provided a roadmap for terrestrial acoustic monitoring. Later they [132] reviewed perspectives on terrestrial passive acoustic monitoring. Semi-automated analysis is used for temporal and spatial estimation of bird behavior, developed for detecting and monitoring audio events. Llusia et al. [70] discussed terrestrial sound monitoring systems and quantitative calibration. Kasten et al. [59] introduced the Remote Environmental Assessment Laboratory's acoustic library. Kojima et al. [65] proposed semi-automatic bird song analysis integrating detection, localization, separation, and identification. Full-automation methods offer an unsupervised approach but require extensive training. Digby et al. [32] compared manual and autonomous acoustic monitoring methods. Suzuki et al. [133] developed HARKBird for exploring acoustic interactions in bird communities. Astaras et al. [7] used passive acoustic monitoring for law enforcement in Afrotropical rainforests. Ulloa et al. [139] applied unsupervised multiresolution analysis for estimating animal acoustic diversity in tropical environments.

These methods have also been applied to factory and technical applications for process monitoring in additive manufacturing. Koester et al. [64] researched acoustic monitoring for damage and process condition determination in additive manufacturing. Hossain and Taheri [52] explored in situ process monitoring using acoustic techniques. Additionally, convolutional neural networks have been employed for detecting the degradation state of robotic systems [18]. Bynum and Lattanzi [18] combined convolutional neural networks with unsupervised learning for acoustic monitoring in robotic manufacturing facilities. Unknown spectral profiles or signals with high variances remain problematic.

The literature on sound source localization and acoustic monitoring reveals significant advancements in reducing computational complexity for mobile systems, increasing robustness in indoor scenarios, and applying these technologies to robotics and environmental monitoring. Early research established foundational methods such as IPD and IID for binaural audio perception, which were further enhanced by incorporating HRTFs and environmental reverberation modeling. Methods like MUSIC and ESPRIT have provided robust, high-resolution localization, while deep learning approaches have shown promise in complex, low SNR environments despite their data requirements. These developments have been crucial in implementing efficient sound localization on mobile robotic systems, ensuring reliable performance in dynamic indoor environments. Additionally, the integration of acoustic monitoring in ecological research and industrial applications highlights the versatility and practical significance of these technologies. The progress in reducing computational demands has made real-time processing feasible, broadening the scope of applications in both robotics and environmental monitoring.

In contrast to previous methods, which primarily focused on traditional sound localization techniques such as interaural phase difference (IPD), interaural intensity difference (IID), and subspace-based approaches like MUSIC, the approach developed here addresses several key limitations. Earlier works, such as those by Nakadai et al. and Valin et al., concentrated on microphone array setups and direction of arrival (DoA) estimation but often encountered significant challenges in highly reverberant indoor environments and low signal-to-noise ratio (SNR) scenarios. While these methods have contributed to advancements in sound source localization, their performance tends to degrade in environments with complex acoustic properties, such as reflective surfaces.

The approach presented here introduces an improvement specifically tailored for indoor environments. By focusing on real-time processing of reverberation and echo effects, the system more accurately distinguishes between direct and reflected sound sources, mitigating one of the major challenges of prior works. Additionally, rather than relying solely on static frequency selection, this system dynamically adapts its processing to the current auditory scene, allowing for more robust localization in complex acoustic environments.

These advancements enhance its reliability in real-world indoor settings, outperforming traditional methods in terms of robustness to environmental variability.

2.3. Robotic Proprioception

Early work in formulating kinematic frameworks, such as those by Kennedy [60] and Calvert [21], laid the groundwork. Denavit and Hartenberg's methodology [30] was pivotal, providing a structured approach to describing robotic arm transformations. Subsequent advancements, like the use of Lie Algebra for spatial transformations by Murray [105], and conformal geometric algebra by Löw and Calinon [74], further enriched the field. Modern approaches, including neural networks and deep reinforcement learning by Lu et al. [72] and Malik et al. [79], have enhanced the precision and efficiency of solving inverse kinematics problems. Burkhard et al. [84] introduce a probabilistic kinematic model accounting for joint position inaccuracies, mechanical stress-induced deformations, and gravitational influences.

The interaction of robots with objects under uncertainty has been explored by Su et al. [130]. Progress in modeling perception uncertainties includes classical approaches by Stoiber et al. [128] and deep-learning-based methods by Meyer et al. [83]. Recent efforts focus on sparse iterative approaches [127] to enhance robustness in uncertain environments. Hand-eye calibration introduces additional transformation uncertainties, as discussed by Nguyen et al. [98]. Recent studies propose methods to enhance calibration accuracy and robustness [112, 38]. Addressing these inaccuracies is vital for vision-guided robotic systems.

Various robotics sub-fields deal with spatial transformations subject to errors modeled as uncertainties. Systematic approaches, particularly in virtual reality by Carlsson et al. [23] and Tramberend [138], and robotic simulators by Browning et al. [17, 33], utilize scene graphs to represent spatial relationships. The current state-of-the-art framework is *tf* in ROS [41]. However, little work has interconnected different robotics realms to account for spatial information uncertainties. Initial efforts like those by Coelho et al. [25] acknowledged uncertainty within the scene graph but often failed to model error propagation using Lie Algebra. Some approaches resort to sampling-based methods, such as Ruehr's [111], but with computational costs.

Lie Algebra effectively acknowledges the manifold character of spatial relationships and propagates uncertainty along transformation chains. An introduction to this approach, particularly for robotic navigation, is provided by Barfoot et al. [9]. Lie-Algebra-based concepts for error propagation within robotic manipulators are also explored by Wang et al. [150, 84]. Despite its widespread use in uncertainty estimation, no existing approach has integrated Lie Algebra-based uncertainty propagation into a robotics scene graph.

Building on Lie Algebra and previous work that represents robot kinematics as a tree of transformations, this work proposes a novel approach that combines these two concepts to model kinematics with uncertainties. By integrating the mathematical framework of Lie Algebra into the transformation tree, the proposed method allows for more accurate and robust representation of the robot's pose and motion, accounting for internal sensor inaccuracies and environmental uncertainties. This approach provides a structured and scalable way to handle kinematic errors, improving the overall precision and reliability of robotic systems operating in dynamic environments.

2.4. Combined Audio-Visual Approaches

The first combination of audio and visual localization in robotics was conducted by Nakadai et al. [94] to enable more accurate localization of sound sources. Using vision systems to enhance auditory processing helps in resolving ambiguities in sound source direction. An active audition system was introduced to optimize the capture of sound sources by moving the array perpendicular to the source. Building on this, Nakamura et al. [97] proposed a SSL system for dynamic environments. This system involves developing robust and adaptable methods for identifying sound sources amid varying noise conditions and movement, achieved by moving the robot's head to focus audio capture towards the source.

Viciana-Abad et al. [142] formulated a bio-inspired approach that combines audio and visual inputs to localize and track speakers. This fusion leverages Bayesian inference to process sensory data and improve detection accuracy. An adaptive filter with a short training phase was implemented at the beginning to adjust for different environments.

Another common approach is to combine audio and visual data to enhance speech recognition. Gabbay et al. [44] proposed an end-to-end neural network that integrates audio and visual data. A shared representation improved the model's ability to distinguish between target speech and background noise. Similarly, Chung et al. [24] utilized cross-modal biometric learning to link facial appearance with voice characteristics, enabling effective speech separation without prior speaker enrollment. Their robust fusion was achieved by concatenating speech and visual features, allowing the network to utilize speaker identity information for better separation.

Majumder et al. [78] proposed a reinforcement learning agent that learns motion policies to control its camera and microphone, optimizing its movement to enhance audio separation. It utilizes egocentric audio-visual observations to make decisions, allowing the agent to actively navigate and isolate the target sound source in a dynamic environment. Focusing on lip reading and combining it with auditory clues, Afouras et al. [2] presented two transformer models, one using Connectionist Temporal Classification (CTC) loss and one using Sequence-to-Sequence (Seq2Seq) loss. Both models are built on the transformer self-attention architecture, enabling a direct comparison of the advantages and disadvantages of each loss type.

Expanding the focus beyond speech to understanding the whole acoustic scene, Owens and Efros [101] introduced a self-supervised learning approach. A neural network is trained to detect temporal misalignment between audio and visual streams. This task forces the network to learn a fused representation that captures the correlation between visual motions and corresponding sounds. They applied the learned features to classify actions in videos, achieving significant performance improvements over existing self-supervised methods. Finally, Majumder et al. [77] proposed using egocentric audio-visual observations to inform the agent's locomotion and navigation system.

Acoustic monitoring is well established in ecological research, especially for ornithology [131, 132]. Semi-automated analysis, as introduced by Llusia et al. [70], is utilized for the temporal and spatial estimation of bird behavior, which has been developed to detect and monitor audio events. However, expert knowledge is necessary to label received audio fragments. Fully automated methods by Digby et al. [32] and Ulloa et al. [139] offer an unsupervised approach based on training in simulation and datasets. These methods have been applied to factory and technical applications for process monitoring in additive manufacturing [64, 52].

The approach presented in this work differs from previous combined audio-visual methods primarily in its integration of dynamic and multi-source settings, particularly within complex and cluttered environments. Earlier systems such as those by Nakadai et al. and Nakamura et al. focused on static or controlled conditions with emphasis on specific sound source localization through auditory processing supported by vision systems. These approaches, while effective for resolving ambiguities in source direction, struggled in environments with high levels of noise, movement, or reverberation.

In contrast, this work introduces a more adaptive audio-visual perception system that not only localizes sound sources but also integrates real-time visual tracking in dynamic and unpredictable environments. It improves upon the robustness of sound detection by using an enhanced motion model and active frequency filtering to handle reverberation and occlusion challenges. Furthermore, the system excels in multi-source environments by dynamically adapting to changes in the auditory and visual scene, a limitation in earlier frameworks that heavily relied on static frequency selection or microphone array setups.

2.5. Literature Discussion

Most previous work focuses on a single modality, particularly visual data, within perception frameworks. Attempts to integrate additional sensors often lead to increased computational complexity, memory usage, or system requirements, such as synchronization circuitry for acquisition triggers. This work proposes an approach utilizing independent sensors, which are fused using loosely-coupled methods and uncertainty-aware sensor position models. Furthermore, the extension to audio information is designed specifically for indoor environments, as opposed to outdoor or specialized cases in ornithology. Finally, the methods in this thesis advocate for the use of robust architectures to prevent system failure in the event of a single sensor malfunction.

3. Multi-Modal Perception of Indoor Environment

Developing advanced machine perception systems presents significant challenges. Visual perception in machines can be hindered by textureless or reflective surfaces, occlusions, and the need for real-time processing. Similarly, auditory perception must overcome difficulties such as background noise, sound source localization, and the integration of audio data with other sensory inputs. Proprioception, which provides information about the robot's body position and movement, is crucial for tasks requiring precise interaction and coordination. These challenges highlight the complexity of designing sensory systems capable of reliable perception in diverse and dynamic environments.

In robotics, it is often necessary to combine information from various sensory sources to build a more robust perception of the environment. This section will explore the specific difficulties associated with visual, auditory, and proprioceptive perception in robots and how combining these modalities can enhance overall system performance.

Definitions

To support the understanding of these systems, it is essential to define two important terms: *multi-sensor* and *multi-modal.*

A **multi-sensor** system refers to the use of multiple sensors, often of the same type, to gather data from the environment. The core idea behind a multi-sensor setup is redundancy: by relying on several sensors, the system can compensate for the failure of any single sensor. This redundancy is particularly valuable in indoor environments, where sensor reliability can fluctuate due to various factors like occlusions or the appearance of the environment.

Since each sensor operates independently, the failure of one does not render the system blind. The remaining sensors continue to provide valuable data, ensuring continuity of perception. While multi-sensor systems can also involve different types of sensors (e.g., visual and audio), their primary strength lies in improving reliability and robustness by increasing the number of independent data sources.

A **multi-modal** system, on the other hand, involves the integration of different types of sensory inputs, or modalities, into a unified perception framework. In this work, the primary modalities include vision, audio, and proprioception, though other possible modalities, such as tactile or thermal perception, are used in other applications. The advantage of multi-modal systems lies in their ability to combine fundamentally different types of data, allowing for a more comprehensive and nuanced understanding of the environment. For example, while vision provides detailed spatial information, audio offers insight into events outside the line of sight, and proprioception ensures the system is aware of its own movement and configuration. By combining these heterogeneous data streams, multi-modal systems can overcome limitations that arise from relying on a single type of input, enhancing both accuracy and adaptability in complex environments.

3.1. Visual Perception

Visual perception in robotic systems is fundamental for enabling robots to interpret and interact with their environment. Cameras are the primary sensors used for this purpose, capturing images and videos that serve as the basis for further processing. These visual inputs allow robots to perform critical tasks such as navigation, localization, and mapping of their surroundings. The capability to accurately perceive the environment using camera-based systems is essential for various applications, from autonomous vehicles to industrial automation and service robots.

The process of visual perception in robots involves several stages, starting with the acquisition of raw images from cameras. These images undergo preprocessing to enhance quality and remove noise, followed by feature extraction to identify critical elements within the scene. The system then interprets spatial relationships, estimates distances, and constructs detailed

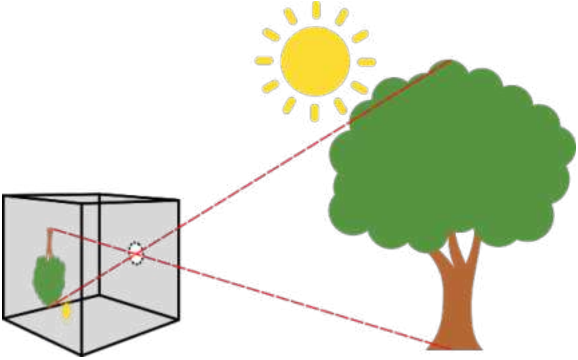


Figure 3.1.: Model of the camera obscura that first used the pinhole camera model to project 3D objects onto a 2D image plane.

maps of the environment. Despite these advancements, visual perception systems still face significant challenges in achieving the robustness and reliability needed for complex tasks.

One of the main challenges in camera-based visual perception is dealing with diverse and unpredictable environmental conditions. Textureless or reflective surfaces can hinder the ability of cameras to capture useful features, while occlusions can obscure important parts of the scene. Additionally, real-time processing of high-resolution visual data requires substantial computational resources, necessitating efficient algorithms and powerful hardware. Addressing these challenges is crucial for improving the effectiveness of visual perception systems in robotics.

From the World into the Robot's Memory

Visual perception in robotic systems relies heavily on the ability to capture and interpret the three-dimensional (3D) world using two-dimensional (2D) images. Understanding how this transformation occurs is essential for various applications, including navigation, localization, and mapping. One of the fundamental principles that underpin this transformation is the pinhole camera model, which provides a simplified yet effective way to describe how a camera projects the 3D world onto a 2D image plane. Figure 3.1 shows a camera obscura, which illustrates this model.

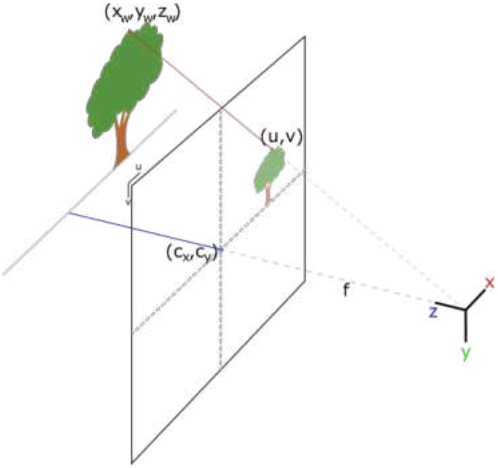


Figure 3.2.: Mathematical representation of the pinhole model.

The pinhole camera model is a foundational concept in computer vision and robotics, used to describe the projection process. In this model, a camera is simplified to a single point, known as the pinhole, through which light rays pass and project an image onto an image plane located behind the pinhole. This setup forms an inverted image of the scene. The basic components of the pinhole camera model include the camera's optical center (the pinhole), the image plane, and the focal length, which is the distance between the pinhole and the image plane.

In the pinhole camera model, the projection of a 3D point onto the 2D image plane involves geometric transformations. Consider a point in the 3D world with coordinates (X, Y, Z). A light ray from this point passes through the pinhole and intersects the image plane, forming an image at coordinates (u, v) on the plane as seen in Figure 3.2. The relationship between the 3D coordinates and the 2D coordinates can be described using similar triangles, where the focal length f is the key parameter. Mathematically, the projection can be expressed as:

$$u = \frac{fX}{Z} \quad \text{and} \quad v = \frac{fY}{Z} \tag{3.1}$$

Here, (u, v) are the coordinates on the image plane, and f is the focal length. This simple yet powerful model allows us to understand how spatial information is captured and represented in a 2D format.

Parameters of the Pinhole Model

In the pinhole camera model [48], the intrinsic parameters define the internal characteristics of the camera, including the focal length, the principal point, and distortion coefficients. These parameters can be encapsulated in a matrix known as the intrinsic matrix, which transforms 3D world coordinates into 2D image coordinates.

The intrinsic matrix, often denoted as **K**, is a 3×3 matrix that includes the focal lengths and the principal point coordinates. The focal lengths in the x and y directions are denoted as f_x and f_y, respectively. The principal point coordinates are (c_x, c_y). The intrinsic matrix is defined as:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

All 3D points $[X, Y, Z]^T$ that are along the ray which is casted from the origin of the camera (see Figure 3.2), are projected onto the same point on the image plane. Therefore, all these points can be normalized by $1/Z$ to obtain a general representation. The transformation from the 3D world coordinates to the 2D image coordinates $[u, v]$ using the intrinsic matrix **K** can be then expressed as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}\frac{1}{Z}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \\ 1 \end{bmatrix} \tag{3.3}$$

Simplifying this equation, we obtain a direct mapping:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_x \frac{X}{Z} + c_x \\ f_y \frac{Y}{Z} + c_y \end{bmatrix} \tag{3.4}$$

Thus, the intrinsic matrix **K** encapsulates the camera's internal parameters, allowing the transformation from 3D world coordinates to 2D image coordinates. To fully describe the transformation from 3D world coordinates to 2D image coordinates, we must also consider the extrinsic parameters. The extrinsic parameters define the position and orientation of the camera in the world and include the rotation matrix **R** and the translation vector **t**. The rotation matrix **R** is a 3×3 matrix that describes the orientation of the camera, and the translation vector **t** is a 3 × 1 vector that describes the position of the camera.

The extrinsic parameters can be combined into a single matrix $[\mathbf{R}|\mathbf{t}]$, where **R** is the rotation matrix and **t** is the translation vector. This matrix transforms points from the world coordinate system to the camera coordinate system.

Given a 3D point $[X_w, Y_w, Z_w]^T$ in the world coordinate system, its coordinates in the camera coordinate system $[X_c, Y_c, Z_c]^T$ can be obtained as:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \mathbf{R} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \mathbf{t} \tag{3.5}$$

In homogeneous coordinates, this transformation can be written as:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = [\mathbf{R}|\mathbf{t}] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{3.6}$$

Combining the intrinsic and extrinsic parameters, we get the projection matrix P, which maps 3D world coordinates directly to 2D image coordinates. The projection $\pi(\cdot)$ is defined as:

$$\pi := \mathbf{K}\frac{1}{Z}[\mathbf{R}|\mathbf{t}] \tag{3.7}$$

Therefore, the complete transformation from 3D world coordinates (X_w, Y_w, Z_w) to 2D image coordinates (u, v) is given by:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \pi\left(\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}\right) \tag{3.8}$$

This projection $\pi(\cdot)$ encapsulates both the intrinsic and extrinsic parameters, providing a comprehensive model of the camera's imaging process. This can now be used to project points from 3D space to the image plane and vice versa in Chapter 5 (pp. 69).

Depth Perception Using Stereo Vision

Depth perception in robotic systems can be achieved using stereo vision, which involves two cameras capturing the same scene from slightly different viewpoints. This setup mimics human binocular vision and enables the robot to perceive depth by comparing the images from both cameras. The two cameras, typically positioned parallel to each other and separated by a fixed baseline distance, form a stereo pair.

The key to depth perception using stereo vision lies in identifying corresponding features in the images captured by the left and right cameras. These corresponding features are points in the scene that appear in both images. The difference in the positions of these corresponding points in the two images is known as the disparity. The disparity is inversely proportional to the depth of the points in the scene: points that are closer to the cameras have a larger disparity, while points that are farther away have a smaller disparity.

Before calculating the disparity, the images from both cameras must be rectified, especially if the focal lengths or other intrinsic parameters differ. Image rectification involves transforming the images so that corresponding points lie on the same horizontal lines. This alignment simplifies the process of finding correspondences and ensures accurate depth estimation. Rectification can be achieved using a combination of camera calibration and homography transformations. The calibration process estimates the intrinsic and extrinsic parameters of both cameras, which are then used to compute rectification transformations that warp the images into a common coordinate system.

To compute the depth of a point in the scene, we first identify its corresponding points in the left and right rectified images. Let $[x_L, y_L]^T$ be the coordinates of a feature point in the left image, and $[x_R, y_R]^T$ be the coordinates of the corresponding point in the right image. Since the images are rectified, $y_L = y_R$, and the disparity d is given by: [50]

$$d = x_L - x_R \tag{3.9}$$

Using the disparity and the known baseline distance B between the two cameras, we can calculate the depth Z of a point based on Equation (3.7) as follows:

$$Z = \frac{fB}{d} \tag{3.10}$$

The depth perception method described here is specific to stereo pairs of images, where the cameras are calibrated, and the images are rectified such that the epipolar lines are horizontal. This simplification allows for a straightforward calculation of depth using the disparity between corresponding points. Depth estimation will become important when features merge together in the VO process described in Section 5.1 on pages 70 or triangulation for feature position estimations.

Challenges in the Indoor Domain

Visual perception in indoor environments presents numerous challenges that impact the performance and reliability of robotic systems. One of the primary issues is the variability of light conditions. Indoor spaces often suffer from insufficient lighting, which hinders the ability of visual sensors to capture detailed images. While adding artificial lighting could mitigate this problem, solutions like flashlights are not always practical or desirable, especially in applications involving human-robot interaction, such as assistant robotics (see Section 1.2, p. 2). Moreover, artificial lighting can introduce additional shadows and reflections, complicating the perception task further.

When light is insufficient, one way to mitigate this issue is to increase the exposure time of the cameras to detect more photons. However, this increased time for measuring the environment means that movements during the exposure period will result in blurred projections, known as motion blur. Motion blur can significantly degrade the quality of the captured images, making it difficult for the robot to perceive details and accurately interpret the environment. This is particularly problematic in dynamic settings where both the robot and objects within its surroundings are constantly moving.

Reflective surfaces present a different kind of problem for visual perception systems. Mirrors, glossy furniture, and other reflective materials can create misleading visual cues. Light reflected off these surfaces originates from different parts of the room, making it difficult for depth estimation algorithms to function correctly. These reflections can cause the robot to perceive false obstacles or misjudge distances, which is particularly problematic in navigation and object manipulation tasks.

Textureless and repetitive environments add another layer of complexity. Modern interior designs often favor minimalist aesthetics with smooth, featureless surfaces and repetitive patterns. Such environments, as depicted in Figure 3.3, provide few visual cues, making it challenging for robots to distinguish between different areas or objects. This lack of distinguishing features complicates tasks like localization and mapping, where the robot relies on visual landmarks to understand its position and navigate the space effectively.

In indoor environments, robots operate near obstacles. The previously described depth perception using stereo vision depends on detecting point correspondences in both images. In such situations, both cameras may not be able to view the point, causing depth perception to fail. Most of these problems depend on the sensor's view of the scene, such as lacking features or close-to-obstacle blindness, and the position of the sensor on the robotic platform. Finding an optimal position is always a trade-off and may only reduce the impact of a single source in specific scenarios. Therefore, using

Figure 3.3.: Photos from real apartments illustrating the modern interior style

multiple sensors facing different directions helps further reduce these impacts, as individual factors can be better counteracted which will be the major contribution of the upcoming Chapter 5 (pp. 69).

3.2. Audio Perception

While visual perception provides critical spatial and temporal information about the environment, it is limited to the line of sight and affected by lighting conditions. Audio perception, on the other hand, complements visual perception by offering a 360-degree sensory capability that is not constrained by visual barriers. Sound waves can travel around obstacles and provide important cues about objects and events that are not directly visible.

In the context of robotics, inspiration is drawn from human auditory perception to develop systems capable of capturing and interpreting auditory information. This involves sensors and processing techniques to accurately measure and understand the auditory environment.

Measuring a Vibrating Medium

Audio perception fundamentally differs from visual perception in that it involves the detection of vibrating molecules rather than the travel of photons. While light perception relies on the interaction of photons with receptors like the human retina or photosensitive components of a camera, audio perception depends on the detection of pressure waves traveling through a medium, such as air, water, or solid materials. These pressure waves cause molecules in the medium to vibrate, creating sound.

When an object vibrates, it disturbs the molecules in the surrounding medium, creating compressions and rarefactions. These alternating high and low-pressure regions travel outward from the source as sound waves. The frequency of these vibrations determines the pitch of the sound, while the amplitude of the vibrations determines the volume. Higher frequency vibrations result in higher-pitched sounds, and larger amplitude vibrations produce louder sounds.

Acoustic waves are longitudinal waves, meaning the displacement of the medium's molecules occurs in the same direction as the wave travels. This is different from transverse waves, like those on a water surface, where the displacement is perpendicular to the wave's direction. In longitudinal waves, such as sound, molecules move back and forth in the direction of the wave, creating areas of compression and rarefaction [102, 149].

In auditory systems, these vibrations are captured by sensors designed to measure the changes in pressure caused by sound waves. Microphones work by converting the mechanical energy of sound waves into electrical signals. This is typically achieved using a diaphragm that vibrates in response to sound waves, causing changes in an electrical circuit that can be measured and analyzed. For example, when a drum is struck, the drumhead vibrates, creating sound waves that travel through the air and are detected by our ears or microphones [102, 12].

The speed of light is approximately 299 792 km/s in a vacuum, but it slows slightly to around 299 702 km/s when traveling through air due to the refractive index of air being about 1.0003 [36]. In contrast, the speed of sound is much slower, approximately 343 m/s in air at room temperature. This vast difference in speed has several implications:

- **Speed Variability:** The speed of sound is significantly influenced by environmental conditions such as temperature. As temperature increases, the speed of sound also increases because warmer air causes molecules to move faster, facilitating quicker transmission of sound waves.

  To speed of sound can be derived from the Newton-Laplace equation

  $$v_0 = \sqrt{\frac{K_s}{\rho}} = \sqrt{\frac{\gamma_{air} P}{\rho}} \tag{3.11}$$

  where:

  - v_0 is the speed of sound in m/s,
  - K_s is the isentropic bulk modulus of air in Pa,
  - ρ is the density of air in kg/m^3,
  - γ_{air} is the adiabatic index of air (≈ 1.4),
  - P is the air pressure Pa.

  Substituting the ideal gas law $P = \rho RT$, the equation simplifies to

  $$v_0 = \sqrt{\gamma_{air} R_{air} T} \tag{3.12}$$

  where:

  - R_{air} is the specific gas constant of dry air ($\approx 287.05\,\mathrm{Jkg^{-1}K^{-1}}$)
  - T is the temperature in K

  Interestingly, the speed of sound is not directly dependent on air pressure, as it is proportional to temperature. Humidity affects the composition of air by adding water molecules, which decreases the specific gas constant. However, previous research in our target domain shows that humidity changes the speed of sound at $T = 20\,^{\circ}\mathrm{C}$ by a maximum of 0.075 %. Therefore, this influence is negligible and will not be considered in the rest of this work [103, 15].

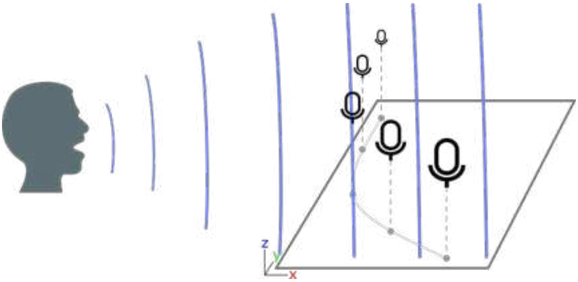


Figure 3.4.: Audio wave propagation in the frame of a 3D microphone array.

- **Measurement Implications:** Taking an image with light-based systems is almost instantaneous due to the high speed of light. In contrast, sound measurements are inherently slower and are collected over time. This temporal aspect of sound requires continuous sampling and processing to capture the dynamics of auditory events accurately. For instance, capturing an audio signal involves recording the changes in air pressure over time, necessitating sophisticated time-based analysis techniques.

Microphone Arrays

Microphone arrays are an essential component in modern audio perception systems, playing an important role in a variety of applications ranging from speech recognition to environmental sound analysis. These arrays consist of multiple microphones arranged in a specific spatial configuration. This setup allows for the simultaneous capture and analysis of sound from different directions.. The fundamental advantage of using microphone arrays lies in their ability to estimate the direction of arrival (DoA) of sound waves, a process that is critical for localizing sound sources and enhancing audio quality in complex environments.

One of the key principles behind microphone arrays is the concept of time delay of arrival (TDOA), illustrated in Figure 3.4. As sound travels through the air, it reaches each microphone at slightly different times. By precisely measuring these time differences, it is possible to infer the direction from which the sound originated. This capability is particularly useful in applications such as beamforming, where the array can focus on sounds coming from a specific direction while attenuating noise from other directions. This technique enhances the clarity and intelligibility of speech signals in noisy environments, making it invaluable in fields such as teleconferencing, hearing aids [11], and as in the case of this work, robotic audition and the localization of sound sources which will be described in Section 6.1.2 (pp. 91).[1]

Assume a microphone array with sensors arranged in 3D space. If the distance to a given source is large enough, we can assume the wavefront is planar. This condition is known as the far-field. The TDOA at each sensor position can be calculated as follows. Let

- $\mathbf{r}_i$ be the position vector of the i-th microphone,
- $\mathbf{r}_j$ be the position vector of the j-th microphone,
- $\mathbf{k}$ be the unit vector in the direction of the incoming sound wave,
- v_0 be the speed of sound in the medium,
- Δt_{ij} be the time difference of arrival between microphone i and microphone j.

The separation vector $\mathbf{d}_{ij}$ between the microphones which represents the Euclidean distance $\|\cdot\|_2$ is given by:

$$\mathbf{d}_{ij} = \|\mathbf{r}_j - \mathbf{r}_i\|_2 \tag{3.13}$$

The TDoA Δt_{ij} can be calculated as:

$$\Delta t_{ij} = \frac{\mathbf{d}_{ij} \cdot \mathbf{k}}{v_0} \tag{3.14}$$

Here, $\mathbf{d}_{ij} \cdot \mathbf{k}$ represents the dot product of the separation vector and the direction vector, indicating the effective path difference that the wavefront travels between the two microphones. This approach leverages the three-dimensional positions of the microphones to accurately determine the direction of the incoming sound wave assuming planar wavefronts.

---

[1] Publication 9

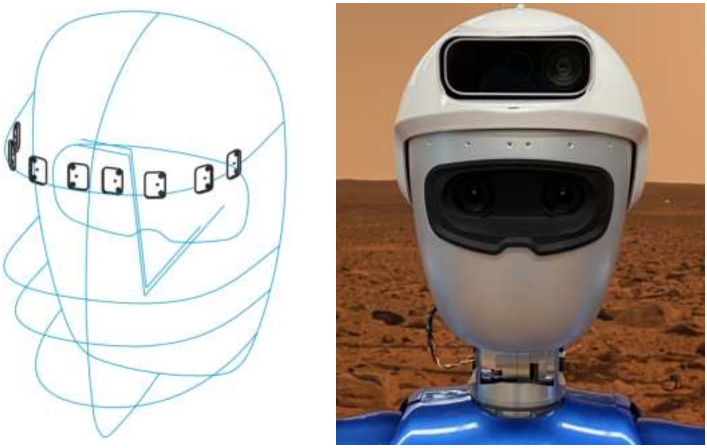


Figure 3.5.: Microphone placement on the forehead of the robot Rollin' Justin.

The spatial arrangement of microphones in an array can vary significantly depending on the application and desired performance characteristics. Common configurations include linear, circular, and spherical arrays, each offering unique advantages. For instance, linear arrays are simple to implement and effective for applications requiring directional sensitivity in one dimension. Circular and spherical arrays, on the other hand, provide more comprehensive coverage and are capable of three-dimensional sound localization, making them ideal for autonomous assistant platforms [81].

Figure 3.5 shows a custom microphone array designed for the humanoid system Rollin' Justin. The array's shape is optimized for human speech frequencies while being discreetly integrated to ensure the robot's face does not appear intimidating. This design consideration is crucial in the field of assistant robotics, where human interaction with the system is common, and the robot's design often prioritizes aesthetics over function [100, 16, 116].

Challenges in the Indoor Domain

In indoor environments, the contained nature of spaces with walls, floors, and ceilings presents significant challenges for audio perception systems. One of the primary issues is the presence of echoes and reverberations. Echoes occur

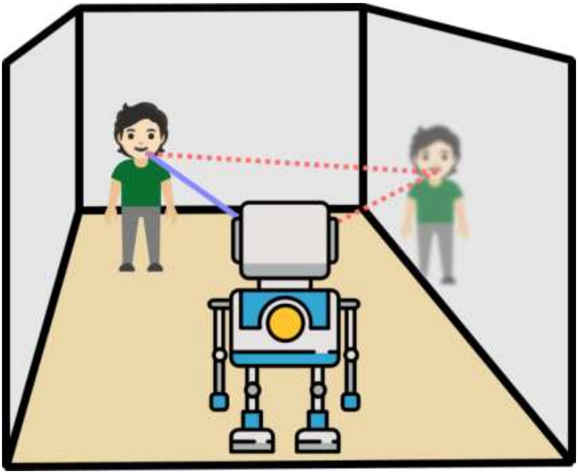


Figure 3.6.: Illustration of a shadow source introduced by echo.

when sound waves reflect off surfaces such as walls and return to the source, creating distinct delayed repetitions of the original sound. This phenomenon is easily observed when shouting in a hall and hearing the shout repeated moments later. These echoes can create "shadow" sources (see Figure 3.6) that confuse audio perception systems, making it difficult to accurately identify and localize the original sound source.

Reverberation, on the other hand, results from the accumulation of multiple small reflections of sound waves within an enclosed space. Unlike distinct echoes, reverberation produces a continuous background noise that lacks a specific direction. This can be likened to the lingering sound in a church after an organ stops playing. The omnipresent nature of reverberation reduces the SNR, significantly impacting the clarity of audio signals and complicating the process of distinguishing individual sound sources. This ambient noise can mask important audio cues, making it challenging for systems to accurately process and respond to sounds.

Another challenge is the variability of sound propagation due to the diverse materials and objects found indoors. Different materials such as carpets, curtains, furniture, and walls have varying acoustic properties, absorbing or reflecting sound waves to different extents depending on the wavelength. This variability can distort audio signals, affecting the accuracy of sound localization and recognition. For instance, sound waves may be dampened by soft furnishings, leading to weaker signals, while hard surfaces may cause excessive reflections, complicating the perception process.

Using an adaptive filter approach that focuses on the dominant frequencies of directed sound, rather than background noise or weakened frequencies, effectively models the environmental situation. Additionally, modeling the audio path and incorporating the Precedence Effect [69], which prefers the first detected direction of arrival over others, reduces distraction from shadow sources or irritation due to low SNR environments, thereby improving estimation accuracy. This approach is integrated as a motion-model into the methods described in Chapter 6 (pp. 89), particularly Section 6.1.2 (pp. 91).

3.3. Robotic Proprioception

In robotics, proprioception—the sense of the relative position of the robot's joints—enables precise interactions with the environment. This capability allows robots to understand and react to their own configuration and motion. This is essential for tasks requiring accurate positioning and movement. Additionally, it involves integrating measurements and observations from different sensors in a complex robotic system. This chapter looks into the basics of robotic proprioception, emphasizing its importance in the broader context of robotic sensory perception.

Understanding the robot's configuration begins with the kinematic analysis of its joints and links. Each joint and link in a robotic system plays a specific role in determining the overall movement and flexibility of the robot. Joints can be categorized based on their movement characteristics: linear, rotary, or a combination of both. Commonly used joints in robotics include prismatic, revolute, spherical, cylindrical, and other specialized joints [124, 143, 123].

- **Prismatic Joints:** Prismatic joints allow sliding motion along a single axis, similar to how a drawer operates. These joints are commonly used in industrial robots to extend and retract robotic arms, allowing for reach without requiring a larger base.
- **Slider Joints:** These joints allow two plates to slide over each other in a plane, providing motion in two dimensions. They are often found in X-Y tables used in 3D printers and CNC machines.

- **Revolute Joints:** Also known as hinge joints, revolute joints allow rotation around a single axis. They are fundamental in robotic arms and hands, enabling tasks such as assembly, packaging, and machining. These joints function similarly to human elbows and knees.
- **Twisting Joints:** Twisting joints, another type of rotary joint, allow rotation around an axis perpendicular to the axis of the connected links. These are often used in robotic wrists to provide additional degrees of freedom for tasks requiring complex movements.
- **Revolving Joints:** These joints enable rotational motion where one link is perpendicular to the axis of rotation, while the other link is parallel. This configuration is used in more complex robotic systems to achieve intricate movements.
- **Spherical Joints:** Spherical joints allow for multi-axial movement, providing rotation, swiveling, and pivoting in various directions. They are highly versatile and are used in robotic arms for maneuvering payloads in tight or complex space.
- **Cylindrical Joints:** These joints combine rotational and linear sliding motions, making them suitable for applications like robotic fingers or legs that require both movements simultaneously.

Understanding the Robotic Configuration

To model the complex kinematics of modern robotic systems efficiently, the configuration of links and joints is often represented as a tree of transformations, as illustrated in Figure 3.7. In this hierarchical structure, each joint and link pair is defined by a transformation that relates the position and orientation of one link to the preceding joint and link. This approach simplifies the calculation of the robot's sensor position and orientation in space by systematically applying transformations from the base to the sensor.[2]

These transformations are typically represented using homogeneous transformation matrices. Each matrix describes a rotation and translation from one coordinate frame to another. If $\mathbf{T}_i$ represents the transformation from frame

---

[2] Publications 7 and 2

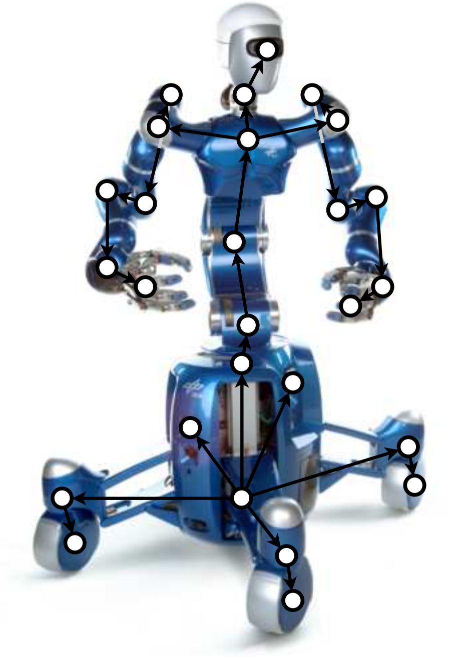


Figure 3.7.: The kinematic tree displayed for the robotic platform Rollin' Justin.

$i - 1$ to frame i, the overall transformation from the base frame to the sensor frame, $\mathbf{T}_{sensor}$, is obtained by chaining these individual transformations together:

$$\mathbf{T}_{sensor} = \mathbf{T}_1 \mathbf{T}_2 \cdots \mathbf{T}_n \tag{3.15}$$

where n is the total number of joints. Each transformation matrix T_i can be expressed as:

$$\mathbf{T}_i = \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ \mathbf{0} & 1 \end{bmatrix} \tag{3.16}$$

Here, $\mathbf{R}_i$ is a 3×3 rotation matrix describing the orientation of frame i relative to frame $i-1$, and $\mathbf{t}_i$ is a 3×1 vector representing the position of the origin of frame i relative to frame $i-1$.

Accounting for Real-World Errors

However, this simplification must consider real-world sources of error, such as flexibility or bending in the links and sensor inaccuracies, which can accumulate along the kinematic chain, leading to deviations in the robot's actual position and orientation compared to its calculated configuration. For accurate modeling and control, these errors need to be accounted for and compensated in the kinematic equations.

The positions of joints in a robotic system are typically measured using sensors such as encoders, potentiometers, and resolvers. Each of these sensors has unique characteristics and methods of operation:

Encoders are widely used for measuring the position of joints. They convert the angular position or motion of a shaft into an analog or digital signal. There are two main types of encoders:

- **Incremental Encoders**: These provide information about position changes (increments) rather than the absolute position. They generate pulses as the shaft rotates, and the position is determined by counting these pulses from a known reference point.
- **Absolute Encoders**: These provide a unique position value for every angular position of the shaft. They can be optical or magnetic and are capable of giving the precise position even if the system is powered down and restarted.

Common sources of errors in encoders include:

- **Quantization Error**: Incremental encoders may suffer from quantization error due to the discrete nature of pulse counting. This can lead to inaccuracies in position measurement, especially in high-precision applications.
- **Environmental Interference**: Optical encoders can be affected by dust, dirt, or other contaminants that obscure the optical path, leading to incorrect readings.

- **Signal Noise**: Electrical noise can interfere with the encoder signals, causing errors in the detected position.

Potentiometers are variable resistors used to measure angular position by varying electrical resistance. As the joint moves, the position of a wiper on a resistive element changes, altering the output voltage proportionally to the angular position. Potentiometers are simple and cost-effective but can suffer from certain inaccuracies:

- **Wear and Tear**: The physical contact between the wiper and the resistive element can lead to wear over time, resulting in increased resistance and reduced accuracy.
- **Temperature Sensitivity**: Changes in temperature can affect the resistive material, altering the voltage output and leading to position measurement errors.
- **Mechanical Play**: Any mechanical looseness in the potentiometer can cause deviations in the position readings.

Resolvers are electromechanical devices that convert the angular position of a shaft to analog signals. They work on the principle of electromagnetic induction and are highly reliable, especially in harsh environments. Resolvers are often used in applications requiring high precision and robustness, such as aerospace and robotics.

Challenges on Robotic Platforms

It is important to note that sensor inaccuracies are not the only sources of errors in robotic kinematics. Mechanical issues such as bending due to acceleration forces, including gravity, can also introduce significant errors. When a robot accelerates or decelerates, the forces involved can cause its links to flex and bend. This bending alters the actual position and orientation of the links, leading to discrepancies between the expected and real positions of the robot's components. These deformations are particularly problematic in high-speed or heavy-load applications, where the forces exerted on the links are substantial, or on lightweight systems that are designed for aerospace applications like the robotic arm TINA displayed in Figure 3.8.

These mechanical deformations are not accounted for by the simple model of rigid links and ideal joints typically used in kinematic analyses. Traditional

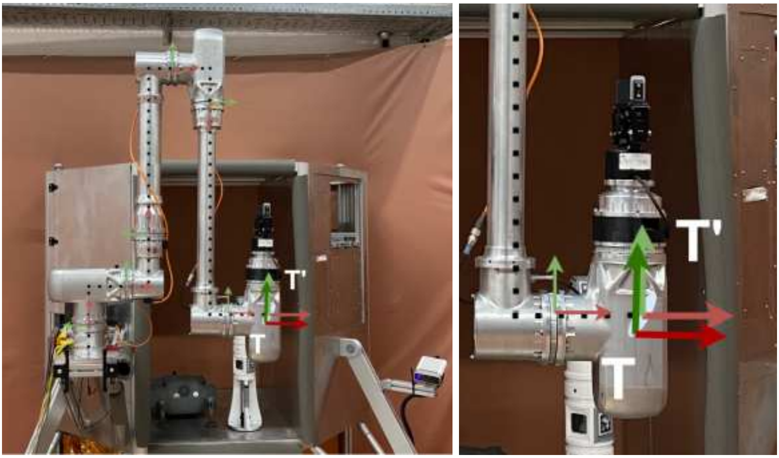


Figure 3.8.: TINA arm bending due to gravity. The computed position, designated as $\mathbf{T}'$, represents the theoretical location without accounting for uncertainties.

kinematic models assume that links are perfectly rigid and joints are ideal, with no backlash. However, in practical applications, these assumptions do not hold true. Factors such as material properties, load distribution, and the physical design of the robot can all contribute to bending and flexing of the links. This discrepancy necessitates the inclusion of more sophisticated modeling techniques that account for uncertainties in robotic systems that will be presented in Section 4.2 on page 57.

4. Audio-Visual Architecture

This chapter focuses on the overall architecture for an audio-visual approach and introduces the measurement data abstraction layer (MDAL) within the proposed audio-visual perception system. The measurement data abstraction layer (MDAL) manages the acquisition, timestamping, and preprocessing of diverse sensor data, including visual (RGB, depth), audio (microphones), and joint encoders. The chapter also covers the robot's configuration and hardware emulation, including dataset recordings and robot simulation. These

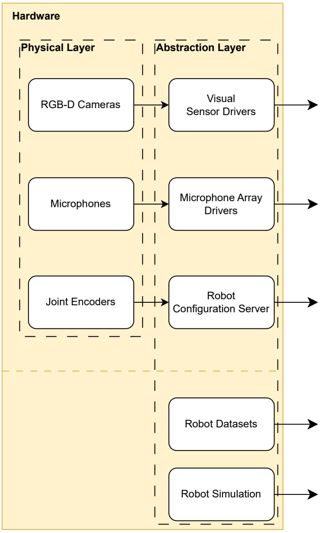


Figure 4.1.: Overview of the measurement data abstraction layer. The physical layer represents actual sensors mounted on the robotic platform and was previously described in Chapter 3 on page 25. The abstraction layer processes sensor information and distributes the data across the system. It may also distribute simulated or recorded data.

components are important for testing and validating the system in controlled environments, ensuring robustness and adaptability.

An overview of all components in this sub-module is given in Figure 4.1.

4.1. Measurement Data Abstraction Layer

4.1.1. Visual Sensor Drivers

Processes on a robotic platform can only gain exclusive access to connected hardware. Therefore, a specialized driver must connect to each camera sensor and provide the acquired data as network streams This interaction is facilitated by dedicated drivers that manage the exchange of data from sensors such as RGB-D cameras. The term RGB-D refers to the combination of RGB (color) images and D (depth) information, both of which are critical for 3D perception.

Depth estimation in robotic systems can be achieved through various methods. One common technique is stereo vision, where two cameras capture images from slightly different perspectives. By identifying corresponding points between the two images and measuring the disparity, the system can estimate the depth of objects in the scene. Another widely used method involves structured light, in which a known pattern is projected into the environment, and the deformation of this pattern is analyzed in the captured image to infer depth. Cameras like Intel RealSense rely on this approach for depth estimation.

An alternative approach is time of flight (ToF) technology, used in systems like Microsoft's Kinect Azure. ToF cameras emit modulated light and measure the time it takes for the light to return after reflecting off objects. This time delay is then converted into depth information. The depth map generated by these methods is often post-processed to align with the RGB images, ensuring spatial consistency between the color and depth data.

This sensor setup is foundational for tasks such as localization, mapping, and interaction within the robot's environment. However, to ensure the accuracy and consistency of the data provided by these sensors, calibration must be performed to fine-tune both the intrinsic and extrinsic parameters of the system.

Calibration of Cameras

The calibration process plays a critical role in ensuring that sensor data, particularly from cameras and other perception systems, is accurate and well-aligned with the robot's coordinate system. This process can be divided into two primary categories: intrinsic calibration and extrinsic calibration.

Intrinsic calibration focuses on determining and correcting the internal parameters of a sensor, such as a camera. For cameras, these parameters include focal length and optical center. The calibration process commonly uses a well-known pattern, such as a checkerboard grid [152]. The corners of the checkerboard provide precise points whose positions in the world are known. By projecting these known 3D points into the image plane and identifying where they appear on the camera's 2D image, the calibration process can estimate the intrinsic parameters that caused this projection.

In detail, the corners of the checkerboard serve as world points, which are mathematically modeled as being positioned in a known coordinate system. The projection of a 3D world point $[X_i, Y_i, Z_i]^\top$ onto the 2D image plane is governed by the pinhole camera model:

$$s \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{K}\,[\mathbf{R} \mid \mathbf{t}] \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \tag{4.1}$$

where s is a scaling factor, and $\mathbf{K}$ (see Section 3.1 on page 29) is the intrinsic camera matrix:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

In this matrix, f_x and f_y are the focal lengths in pixel units, and (c_x, c_y) is the optical center (principal point). The matrices $\mathbf{R}$ and $\mathbf{t}$ represent the rotation matrix and translation vector that transform points from the world coordinate system to the camera coordinate system.

By observing where these points appear within the image, the intrinsic parameters of the camera can be estimated. This calibration allows for corrections that adjust the focal length and optical center to match the expected real-world scale.

Mathematically, this process minimizes the reprojection error E, which is the sum of the squared differences between the observed positions of the checkerboard corners in the image $[u_i, v_i]^\top$ and the calculated projections $\hat{\mathbf{p}}_i$ based on the estimated intrinsic parameters [129]:

$$E = \sum_{i=1}^{N} \left\| \begin{bmatrix} u_i \\ v_i \end{bmatrix} - \hat{\mathbf{p}}_i \right\|^2 . \tag{4.3}$$

This optimization is typically performed over both the intrinsic parameters and the extrinsic parameters (**R** and **t**) for each calibration image, ensuring that subsequent image data from the camera is geometrically accurate. This is important for tasks like 3D reconstruction and mapping as later discussed in Chapter 5.

Extrinsic calibration aligns the spatial relationships between multiple sensors on the robot platform, such as cameras, IMUs or microphone arrays. Extrinsic calibration involves determining the translation $\mathbf{t}$ and rotation $\mathbf{R}$ between sensors relative to a common coordinate frame, which is typically defined by a reference sensor (such as a primary camera or an IMU). This step is essential when fusing data from multiple sensors to ensure that all measurements are correctly positioned and oriented within the same spatial reference frame.

The methods used for extrinsic calibration differ depending on whether the calibration is static, remaining consistent over time, or dynamic, where the calibration is only valid for a short period, such as during online registration. For static calibration, high-fidelity calibration targets like the checkerboard are used. By identifying corresponding corners from both views, the extrinsic parameters are calculated by minimizing the reprojection error, similar to intrinsic calibration. In dynamic calibration, instead of using a calibration target, artificial markers such as AprilTags [144] or visual features in the environment are utilized (see Section 5.1, pp. 74). These methods allow for real-time estimation of extrinsic parameters.

Synchronization of data streams from multiple sensors is essential, especially in real-time systems where even slight temporal misalignment can result in inaccurate sensor fusion. Timestamp acquisition plays a key role in this synchronization, and it can occur at different stages during the data capture process.

Timestamps can be acquired either at the beginning, middle, or end of an acquisition process, depending on the system setup. The choice of when the timestamp is recorded depends on the hardware and the precision requirements of the system. For cameras, it is common to timestamp the image frame at the moment when the exposure begins. However, in some cases, particularly when more precise synchronization is required, the timestamp may be recorded at the middle of the sensor acquisition window to better represent the timing of the measurement.

Accurate timestamping allows sensor data, such as camera frames, audio captures, or other measurements, to be properly aligned in time. This is especially important in dynamic environments where objects move, or the robot itself is in motion. Time synchronization ensures that each sensor's data corresponds to the same moment in time, reducing errors in tasks like sensor fusion and visual odometry. A suitable method for synchronizing timestamps is through networked systems, such as using the industry standards Network Time Protocol (NTP) and Precision Time Protocol (PTP), or hardware triggers to ensure that all sensors begin acquiring data simultaneously.

4.1.2. Microphone Array Drivers

Contrary to light, audio propagation is relatively slow. Hence, the acquisition of audio samples is not an instantaneous process but a measurement over time. Specifically, it involves measuring the air pressure level over time. This continuous measurement process captures the variations in air pressure that constitute sound waves.

An array of microphones (typically 4 to 8 sensors) is placed on the outside of the robot. Depending on the application, typical placements include 1D or 2D configurations, favoring one direction for audio acquisition. For instance, a linear array (1D) may be used for directional sound capture, while a planar array (2D) can provide more comprehensive spatial information. Sound pressure levels are sampled simultaneously at all sensors.

To accurately capture audio signals, the sampling frequency must adhere to Nyquist's theorem. This theorem states that the sampling frequency should be at least twice the frequency of the highest expected frequency in the source signal. For speech and indoor applications, this typically means a sampling frequency of 44 kHz. This ensures that the audio signal is accurately represented without aliasing.[1]

Continuous measurements are typically grouped together for better data management. Only the first sample in each group needs to be timestamped, as all consecutive samples can be defined by the sampling rate. These groups of samples are called frames. For an n-sensor microphone array, this results in an $n \times m$ frame, where m represents the number of consecutive samples. Common values for m are 1024 or 4096 [117], providing a balance between data granularity and manageable frame sizes. Each frame provides a snapshot of the audio environment over a brief period. The size of m determines the temporal resolution of the frames, with larger m values encompassing longer time spans. This framing process facilitates efficient storage, transmission, and processing of audio data, enabling real-time or near-real-time analysis. By timestamping only the first sample, the system reduces the overhead associated with timestamping every sample individually, thereby optimizing data handling. The output can be used for DoA estimation as described in Section 6.1.2 (pp. 91).[2]

Frequency Calibration of Microphone Arrays

In practical applications, microphone arrays must be calibrated to account for frequency-dependent variations in each microphone's response. The frequency response $H_m(f)$ of microphone m is defined in the continuous-time domain as the ratio of the Fourier Transform [43] of the microphone's recorded output $y_m(t)$ to that of the known input signal $x(t)$:

$$H_m(f) = \frac{Y_m(f)}{X(f)} = \frac{\displaystyle\int_{-\infty}^{\infty} y_m(t)\, e^{-j2\pi f t}\, dt}{\displaystyle\int_{-\infty}^{\infty} x(t)\, e^{-j2\pi f t}\, dt}. \tag{4.4}$$

This continuous formulation allows for precise characterization of the microphone's behavior across all frequencies, capturing both amplitude and

---

[1] Publication 10

[2] Publication 3

phase shifts introduced by the microphone. By measuring $H_m(f)$ using a known input signal—such as white noise with a flat spectral density—we can identify and compensate for frequency-dependent sensitivity variations in each microphone, ensuring that the recorded signals accurately reflect the true acoustic environment.

In digital signal processing, signals are sampled at discrete time intervals $t = nt_s$, where $t_s = 1/f_s$ is the sampling period and f_s is the sampling frequency. The continuous-time signals $x(t)$ and $y_m(t)$ become discrete-time signals $x[n] = x(nt_s)$ and $y_m[n] = y_m(nt_s)$. The Discrete Fourier Transform (DFT) [26] is then used to compute their frequency-domain representations:

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j2\pi \frac{kn}{N}}, \tag{4.5}$$

$$Y_m[k] = \sum_{n=0}^{N-1} y_m[n]\, e^{-j2\pi \frac{kn}{N}}, \tag{4.6}$$

where N is the number of samples and $k = 0, 1, \ldots, N-1$ corresponds to discrete frequency bins $f_k = \frac{k}{N} f_s$.

The discrete frequency response $H_m[k]$ extends the continuous case to practical, implementable calculations. It is defined as:

$$H_m[k] = \frac{Y_m[k]}{X[k]}. \tag{4.7}$$

This discrete formulation accounts for sampling and allows for digital compensation of the microphone's frequency characteristics. By calibrating each microphone using $H_m[k]$, we correct for frequency-dependent variations, enhancing the accuracy of applications like sound source localization that rely on precise acoustic measurements.

To improve the spectral analysis and mitigate artifacts such as spectral leakage, windowing functions are applied to the discrete-time signals before computing the DFT. We use the Hamming window [47] $w[n]$, defined as:

$$
\begin{aligned}
w[n] = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), \\
n \in \{0, 1, \ldots, N-1\}\,.
\end{aligned}
\tag{4.8}
$$

Multiplying the signals by the Hamming window smooths the discontinuities at the boundaries of the sampled data, reducing leakage of spectral energy into adjacent frequency bins. The windowed signals are then:

$$x_w[n] = x[n] \cdot w[n], \tag{4.9}$$

$$y_{m,w}[n] = y_m[n] \cdot w[n]. \tag{4.10}$$

Using the windowed signals $x_w[n]$ and $y_{m,w}[n]$, we compute the DFTs and subsequently the discrete frequency response $H_m[k]$ as before. The application of the Hamming window enhances the frequency resolution and accuracy of the calibration, leading to more precise compensation for each microphone's frequency response.

Ego-Noise Reduction

In robotic applications, ego-noise, generated by the robot itself (e.g., motor and actuator noise), can interfere with audio signals captured by the microphone array. This noise complicates the accurate processing and analysis of audio data.

A common first step in ego-noise reduction is initial noise estimation. The robot estimates its noise spectrum $S_{nn}(f)$ by capturing audio when only the robot's operational sounds are present, with no external sound sources. These periods can occur when the robot is stationary or performing routine tasks without external audio inputs. During these times, the microphones record noise from the robot's components. The noise spectrum can also be captured using calibrated audio devices, as shown in Figure 4.2.[3].

---

[3] Publication 6, 3 and 10

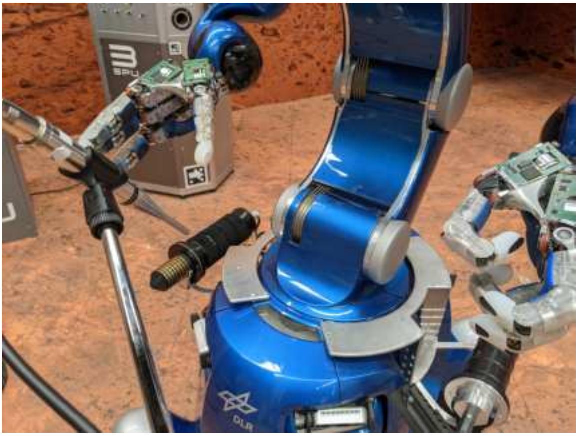


Figure 4.2.: Ego-noise estimation using a calibrated audio probe. The generated sound from motors, fans or other active parts of the systems are recorded and their frequency spectrum is determined. This can be later used to separate ego-noise from actual sound in the environment.

The collected audio data is analyzed to understand the statistical properties of the ego-noise, including calculating the power spectral density (PSD) of the noise using a Fourier Transform, which represents how the power of the noise signal is distributed across different frequencies. The PSD provides information about the frequency components of the ego-noise, helping distinguish it from potential external audio signals. This initial estimation phase sets a baseline noise profile that the filtering process uses to differentiate between noise and desired signals.

Once the initial noise profile is established, the robot employs adaptive filtering techniques to reduce ego-noise during operation. An adaptive Wiener filter [68] is commonly used. The filter dynamically adjusts its parameters based on real-time audio input, continuously estimating the noise characteristics and separating them from the desired signal.

The observation model considers the observed signal $y(t)$ as a combination of the desired signal $s(t)$ and the noise $n(t)$:

$$y(t) = s(t) + n(t). \tag{4.11}$$

This model forms the foundation of Wiener filtering by defining the relationship between the signals involved.

Next, the PSDs of the signals are determined. The power spectral density of the observed signal $S_{yy}(f)$ is estimated directly from the data, and the known power spectral density of the noise $S_{nn}(f)$ is utilized. These spectral densities provide the statistical information needed for the filter design.

The Wiener filter's transfer function in the frequency domain is derived based on these power spectral densities. Ideally, the transfer function is:

$$H(f) = \frac{S_{ss}(f)}{S_{ss}(f) + S_{nn}(f)}, \tag{4.12}$$

where $S_{ss}(f)$ represents the PSD of the desired signal. Since $S_{ss}(f)$ is not directly known, it can be estimated using the relationship of source and noise spectral profiles $S_{yy}(f) = S_{ss}(f) + S_{nn}(f)$:

$$H(f) = \frac{S_{yy}(f) - S_{nn}(f)}{S_{yy}(f)}. \tag{4.13}$$

This formulation allows the filter to balance the contributions of the desired signal and the noise.

The next step involves processing the signal in the frequency domain. The Fourier Transform of the observed signal $Y(f)$ is computed, transforming the signal from the time domain to the frequency domain, making it suitable for applying the Wiener filter. The filter is then applied to the transformed signal using the derived transfer function:

$$\hat{S}(f) = H(f) \cdot Y(f). \tag{4.14}$$

This operation effectively attenuates the noise components while preserving the desired signal components.

Finally, the filtered signal $\hat{S}(f)$ is transformed back to the time domain using the inverse Fourier Transform. This step converts the frequency domain representation of the filtered signal back into a time domain signal, yielding the estimated desired signal $\hat{s}(t)$. This process provides a signal that is a cleaner version of the original observation with reduced noise that can be used for state estimation of the audio environment.

4.2. Robot Configuration

As described in Section 3.3, understanding the current robotic configuration, or proprioception, is essential.

Accurate assessment of a robot's configuration is crucial for various applications, especially when non-static components with perception sensors are involved. Precise positional data ensures effective operation. Registering cameras on robotic manipulators to the robot's origin integrates spatial information within the correct coordinate framework.

Knowing the system's distance to the environment is critical for collision avoidance, especially in confined spaces. Observing and organizing joint positions into a transformation tree helps illustrate the coordinate framework from the root frame. This tree provides an accurate estimate of the robot's spatial volume and movement range.

Neglecting inherent measurement uncertainties and non-static characteristics due to mechanical stress and gravitational forces can lead to erroneous state estimations. These factors impact the reliability of the robot's operation in dynamic environments.

Transformation Tree

Deriving transformations between coordinate frames is a pivotal task in robotics. A common approach models the system as a hierarchical tree of frame transformations as shown in Figure 4.3. To get the transformation between $\mathbf{T}_{cam}$ and $\mathbf{T}_{tcp}$, the entire path involving multiple transformations must be calculated.

Uncaptured deviations in kinematics from the real world can be observed when the manipulator bends due to gravitational forces, causing $\mathbf{T}_{tcp}$ to differ from the expected position. This discrepancy highlights the importance of accounting for real-world factors in kinematic modeling.

Using a hierarchical tree structure has significant advantages, including direct derivation from computer-aided design (CAD) models, which inherently use the same representation. CAD models are typically organized into a hierarchy of parts and subassemblies, mirroring the transformation tree. This

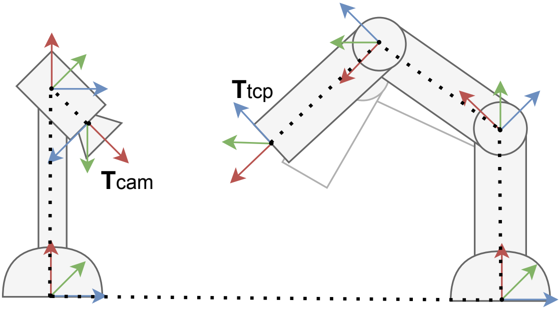


Figure 4.3.: Illustration of different coordinate frames in robot's kinematic system. On the left side, the camera frame T_{cam} and on the right side, the tool center point T_{tcp}.

correlation allows for seamless integration and accurate transfer of geometric data from design to implementation.

Retrieving the direct transformation between any two arbitrary frames involves traversing the path within the structured tree. This systematic approach ensures a clear procedure for obtaining specific transformation information.

Organizing transformations into a hierarchical tree simplifies complex kinematic chains into manageable sub-problems. This reduces computational burden, makes the system scalable and adaptable, aids in debugging, and enhances the modularity of kinematic analysis.[4]

Transformations and Uncertainty

The treatment of uncertainties follows previous work by Burkhard et al. on probabilistic robot kinematics [84], building on Barfoot [9] and Wang [151].

Lie Algebra provides a mathematical framework for describing Lie groups, which are smooth manifolds. This framework is useful in robotics for representing rotations and rigid body transformations, forming the basis of many kinematic and dynamic calculations.

---

[4] Publications 7 and 2

A pose $\boldsymbol{T}_{AB} \in SE(3)$ describes the position and orientation of an object B with respect to a reference frame A. The Special Euclidean group $SE(3)$ includes both rotations and translations in three-dimensional space. A pose can be described locally by its linear tangent space representation $\boldsymbol{\xi} = [\boldsymbol{\rho}\,\boldsymbol{\theta}]^T \in \mathbb{R}^6$, related by the exponential map [125]:

$$\boldsymbol{T} = \mathrm{Exp}(\boldsymbol{\xi}). \tag{4.15}$$

Here, $\boldsymbol{\rho}$ denotes the translational component and $\boldsymbol{\theta}$ the rotational component of the tangent space element. The exponential map converts between the tangent space (Lie Algebra) and the manifold (Lie group).

In Lie Algebra, the tangent space at the identity element of a Lie group forms a vector space called the Lie Algebra of the group. For $SE(3)$, this tangent space is a six-dimensional vector comprising three translational and three rotational components. The adjoint representation maps local tangent space quantities between different coordinate frames.

Local tangent space quantities can be mapped between two local spaces using the *adjoint matrix* **Ad**:

$$^{A}\xi = \mathbf{Ad}(\mathbf{T}_{AB})\,^{B}\xi, \tag{4.16}$$

with

$$\mathbf{Ad} = \begin{bmatrix} \mathbf{R} & [\mathbf{t}]_\times \mathbf{R} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \in \mathbb{R}^{6\times 6}, \tag{4.17}$$

where $\mathbf{R}$ is the rotation matrix of $\mathbf{T}$ and $[\mathbf{t}]_\times$ is the skew-symmetric matrix formed by the translation vector. The term $[\mathbf{t}]_\times \mathbf{R}$ illustrates how local rotation errors create translation errors further down a chain of transformations, depending on the distance from the original error's location.

Any rotation in three-dimensional space can be represented as an element of the $SO(3)$ group, dealing with rotation matrices. Similarly, $SE(3)$ includes translations. The Lie algebra of $SO(3)$ consists of skew-symmetric matrices representing infinitesimal rotations, while the Lie algebra of $SE(3)$ includes both infinitesimal rotations and translations.

The error of a pose can be described as a local deviation $\xi_{\mathrm{B,err}}$ of a nominal pose $\mathbf{T}_{AB}$ in the tangent space of the reference frame B. The corresponding covariance matrix $\Sigma_{AB} = \mathbb{E}\left[\xi_{\mathrm{B,err}}\xi_{\mathrm{B,err}}^T\right] \in \mathbb{R}^{6\times 6}$ is a locally defined tangent space quantity. This covariance matrix encapsulates the uncertainty in both the translational and rotational components of the pose.

Two essential mathematical operations on poses needed for the scene graph are concatenation and inversion. The *concatenation* operation combines two transformations, such as $\mathbf{T}_{AB}$ and $\mathbf{T}_{BC}$, to yield the transformation from A to C:

$$\mathbf{T}_{AC} = \mathbf{T}_{AB} * \mathbf{T}_{BC}, \tag{4.18}$$

$$\Sigma_{AC} = \mathbf{Ad}_{\mathbf{T}_{BC}^{-1}} \Sigma_{AB} \mathbf{Ad}_{\mathbf{T}_{BC}^{-1}}^{T} + \Sigma_{BC}. \tag{4.19}$$

Here, the covariance matrices are transformed into the common reference frame C using the adjoint matrix, allowing them to be added due to the linearity of the tangent space.

The *inverse* operation calculates the transformation from B to A given the transformation from A to B:

$$\mathbf{T}_{BA} = \mathbf{T}_{AB}^{-1}, \tag{4.20}$$

$$\Sigma_{BA} = \mathbf{Ad}_{\mathbf{T}_{AB}} \Sigma_{AB} \mathbf{Ad}_{\mathbf{T}_{AB}}^{T}, \tag{4.21}$$

This shifts the uncertainty from the tangent space of B to the tangent space of A. This representation can implicitly consider *exact* transformations, as zero-covariances simply vanish in Equations (4.19) and (4.21).

Using this uncertainty aware representation of the robot's configuration state helps to fuse data from different sources and different times as discussed later in Section 6.2 on page 96.

4.3. Benchmarks and Evaluation Data

Hardware emulation plays a significant role in the development and validation of sensor-dependent systems. It allows for the replication of real-world conditions within a controlled environment, enabling comprehensive testing and refinement of software. This section discusses the use of datasets and simulation in hardware emulation to ensure robust and reliable system performance.

The use of datasets in hardware emulation is essential for verifying and validating software changes. By employing pre-recorded data streams, developers can test specific edge cases and scenarios repeatedly, which is important for identifying potential issues before deployment. Datasets allow for direct comparisons between different software versions, ensuring that modifications perform as expected under consistent conditions. The repeated use of datasets helps in refining the system's capabilities and enhancing its overall resilience and performance.

Simulation, on the other hand, provides a dynamic and interactive platform for testing. Real-time data streams can be incorporated into the testing framework to simulate a wide range of environmental conditions and scenarios, including events that would be challenging (e.g same environment at different times of the day) or unsafe (e.g. operation close to hazards) to replicate in real life. This approach enables the observation of the system's behavior in real-time, allowing for immediate adjustments and improvements. Simulation is particularly effective for testing the system's adaptability to changing conditions, ensuring that it can handle the complexities of dynamic environments. The integration of both simulated and real data streams in the testing process strengthens the system's robustness, contributing to a more reliable and efficient solution for hardware emulation in sensor-dependent applications.

4.3.1. Dataset

A dataset has been created for this research, comprising synchronized multimodal sensor data collected using a multi-camera setup mounted on a mobile robotic platform. The primary sensors include the RealSense D435i, which integrates an RGB camera, two infrared cameras for depth estimation, and an inertial measurement unit. This dataset is designed to provide comprehensive coverage of various indoor scenarios, capturing a wide range of environmental conditions and robot movements.[5]

A key feature of the dataset is the use of synchronized data streams. The cameras operate at a resolution of 640x480 pixels and a frame rate of 15 Hz, while the IMU captures acceleration data at 250 Hz and angular velocity at 400 Hz. This high-frequency data acquisition ensures precise temporal alignment of visual and inertial data, crucial for accurate sensor fusion and robot navigation tasks. The dataset includes both individual sensor streams

---

[5] Publication 1

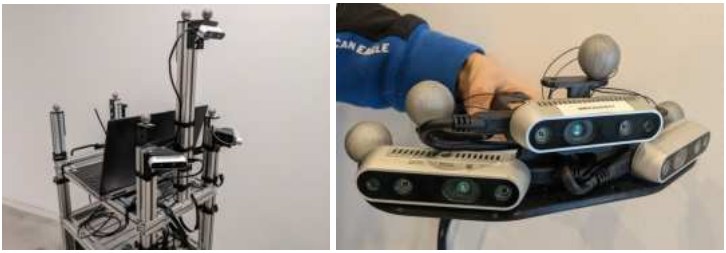


Figure 4.4.: Capture devices in the dataset. Left: A robotic mockup system to mimic the motion and the sensor configuration of a real platform. Right: A handheld device for more complex 6D motions and trajectories close to obstacles.

and a fused stream that interpolates the IMU data to match the timing of the camera frames. Data has been recorded on a robotic mockup and a handheld device, both shown in Figure 4.4.

This dataset is notable for being the first of its kind to focus on multi-sensor setups in the indoor domain. This is important for conducting research on multi-camera systems within indoor environments like the examples in Figure 4.5. It includes typical indoor scenarios such as kitchens, living rooms, and office areas. These environments feature low-texture surfaces and reflective materials, which pose significant challenges for visual perception systems.

Additionally, the dataset captures dynamic changes within these environments. This includes furniture movement, such as chairs and tables being relocated, as well as objects appearing and disappearing, like books and vegetables. It also accounts for changes in the appearance of objects, such as doors being opened or closed and plants changing over time. These variations provide a realistic and challenging setting for evaluating the robustness and adaptability of SLAM and other localization algorithms.

Ground Truth Pose Information

Ground truth data plays a critical role in evaluating the performance of localization systems by providing a benchmark for comparison. In this context, the pose information must exhibit significantly higher accuracy than the system being tested. We utilize a Vicon MX T40 motion capture system, which provides pose updates at 100Hz with a positional accuracy of 1mm. This precision is notably superior to the best-case anticipated performance of 1cm for VO or SLAM systems in indoor scenarios, ensuring that any performance limitations are correctly attributed to the localization system under evaluation.

Moreover, it is essential that the ground truth data be synchronized with the camera data used in the localization process. To achieve this, the system is moved in front of a checkerboard, allowing us to use the camera frames and the pattern on the board to estimate the camera's trajectory. The estimated trajectory from the vision-based approach is then compared to the output of the Vicon tracking system. To ensure temporal alignment, the timestamps from the tracking system are adjusted by applying a temporal offset, denoted as Δt. This Δt is optimized until the relative trajectory error between the estimated trajectory and the externally recorded trajectory is minimized, ensuring precise time synchronization.

Another critical consideration is the use of external markers to track the pose of the device during testing. These markers must be carefully positioned and measured to account for their offset from the test device's reference frame. For instance, the tracking markers for both the wheeled mockup system Marvin and the handheld camera setup, visible in Figure 4.4, are important for accurate tracking. If this offset is not accurately determined, the resulting ground truth pose information may introduce errors into the analysis, reducing the fidelity of the performance evaluation. The high accuracy of the Vicon system enhances confidence in marker placement, but the calibration process remains crucial.

For each benchmark test run, the cameras of the Vicon tracking system were individually arranged to cover the specific test trajectory. This ensured full coverage of the testing area without any breaks in tracking and allowed for maximum accuracy throughout the run. The careful placement of the Vicon cameras was crucial to capturing the full extent of the test device's motion while maintaining high precision across the entire trajectory.

Separately, the calibration of the test device's camera to the external markers is necessary for accurate localization. Similar to multi-sensor camera setups, external calibration ensures consistency between the sensor's measurements and the ground truth pose. In localization tasks, precise calibration of both the sensors and tracking markers, supported by the high accuracy of the Vicon system, is fundamental for obtaining reliable ground truth data. This directly impacts the accuracy of the final performance assessment, as errors introduced during calibration could obscure the true capabilities of the localization system being tested.

Benchmark Environment

The benchmark environment was designed to provide a controlled yet realistic setting for evaluating the performance of localization systems. While testing in an actual apartment would offer realistic scenarios, such environments often lack the necessary interfaces for tracking systems and limit the ability to rapidly modify conditions for multiple test setups. As a result, the benchmark combines controlled lab environments with a real apartment scenario, ensuring a comprehensive evaluation of localization methods under diverse conditions.

Five distinct environments were created for the benchmark (some shown in Figure 4.5), four of which were developed in laboratory settings. These lab environments simulate modern interior designs, reflecting minimalistic aesthetics with textureless surfaces that present a significant challenge for visual localization systems as discussed before in Figure 4.5 on page 65. The lab setups include specific functional areas such as a kitchen, an office, and a living room. Each area is enclosed by 360-degree walls, replicating typical room layouts. Some of these environments feature separated spaces connected by doors, while others adopt an open-plan design, where different areas such as the kitchen and living room merge into one another without clear boundaries. This variability in layout introduces diverse visual and spatial challenges, testing the robustness of the systems being evaluated.

In addition to the lab environments, the benchmark includes a real-world scenario captured in an actual apartment. This setting represents a typical living room and provides a more unmodified, authentic evaluation scenario. The inclusion of this real apartment allows for testing the systems in a space that lacks the strict control of the lab, offering a more practical perspective on the system's performance in everyday environments.

To further challenge the localization systems, both the lab and real-world environments were subjected to dynamic modifications. Objects such as books, pens, toys, and chairs were deliberately placed, moved, or removed during different tests to simulate a dynamic, lived-in space. Additionally,

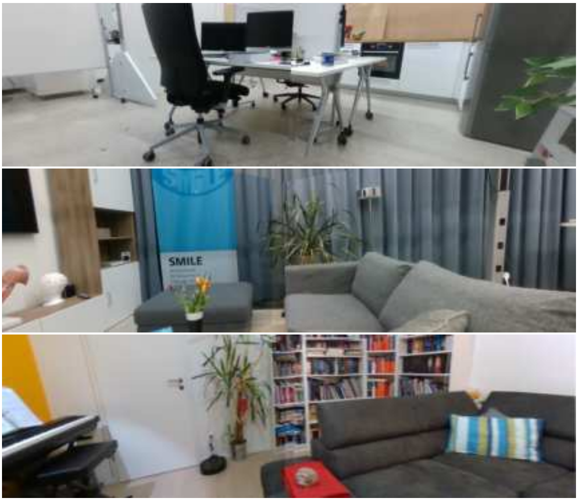


Figure 4.5.: Panoramic views of the environments in the dataset. All sensor images are stiched together. Top image shows an office area in the foreground and a kitchen in the background. The middle image shows a living room scene. The bottom image is taken from the real world apartment scenario.

more significant changes were made to the environment between test runs, such as rearranging furniture like tables, chairs, and doors, or altering the decor with different plants, paintings, and other decorative elements. These modifications introduced variability and complexity, forcing the tested systems to adapt to new or altered visual cues and requiring them to maintain robustness despite the environmental changes.

4.3.2. Simulation

The URSim platform is a comprehensive Software-in-the-Loop (SiL) simulator developed for testing robotic systems, particularly those designed for planetary exploration. It facilitates the integration of various robotic systems and sensors into photo-realistic environments, offering a robust framework for evaluating high-level software components. URSim utilizes the Unreal Engine 4 developed by Epic Games for real-time rendering and physics simulation, which is essential for replicating complex mission scenarios on extraterrestrial surfaces. Its flexible interface and adaptable architecture allow for

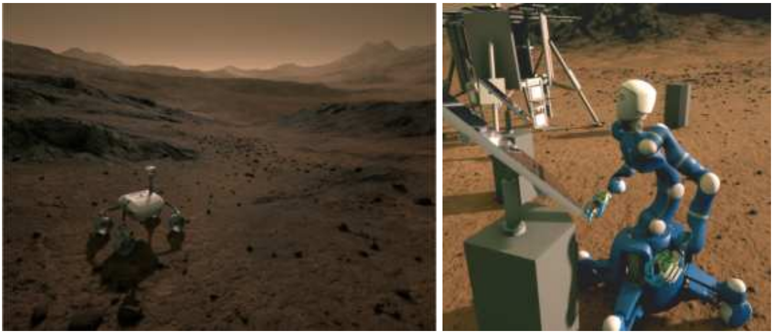


Figure 4.6.: Simulated robotic platforms on the Martian surface. Left: the DLR LRU rover in an unstructured Martian landscape. Right: Rollin' Justin in a scientific camp setup performing maintenance tasks during a simulated mission.

diverse setups, supporting continuous testing without the need for physical hardware.[6]

A key feature of URSim is its ability to simulate multiple robotic systems in detailed mission environments. The platform supports various robots, including the Lightweight Rover Unit (LRU), hexacopter ARDEA, and humanoid robot Rollin' Justin. Each robotic system is described using markup languages, primarily the widely-used Unified Robot Description Format (URDF), which has been extended with sensor definitions. This textual description allows for easy loading and creation of robotic systems at simulation startup, enabling rapid testing of different sensor configurations and setups. The use of URDF also supports version control through systems like Git, ensuring that robotic system definitions can be tracked and managed effectively. By emulating sensor data and offering interfaces that closely match those of actual systems, URSim enables thorough testing of navigation, mapping, and mission execution pipelines. For instance, the complete navigation and mapping pipeline of the LRU can be integrated and tested in simulated Martian and Lunar environments, ensuring reliable performance under mission-relevant conditions. An example is shown in Figure 4.6.

The modular architecture (see Figure 4.7) of URSim is designed to accommodate various robotic systems, environments, and infrastructures. It includes

---

[6] Publication 5

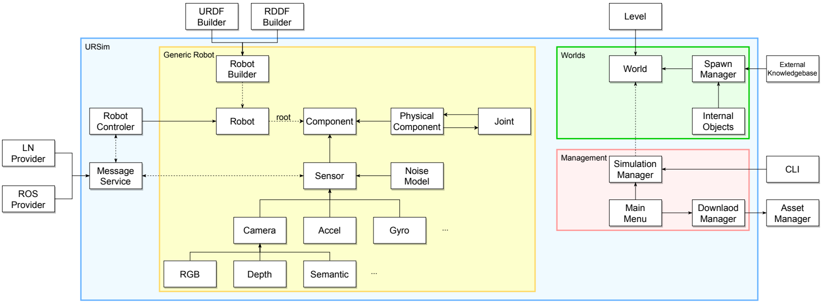


Figure 4.7.: The software architecture of the simulator URSim. The modularity of this framework assures that the software can be easily adapted for future use-cases and different applications. It also supports different input like URDF and output formats like ROS or the DLR framework links_and_nodes (LN).

management, world, and robot modules, each with distinct responsibilities. The management module handles initialization, scenario customization, and feature management. The world module manages the physical environment, and the robot module simulates the robotic platforms. Users can specify properties such as the map, robot type, and additional assets, which can be loaded from an external asset store, facilitating efficient collaboration and development across teams.

URSim ensures the generation of high-quality, synchronized sensor data by simulating various sensors such as visual sensors (RGB and depth cameras), inertial measurement unit, and other physical sensors. These sensors are simulated in real time, producing data streams that closely mimic those from actual missions. This data fidelity is critical for developing perception-action control loops, which are essential for autonomous navigation and exploration tasks on extraterrestrial surfaces.

Additionally, URSim provides dynamic and interactive testing environments. The simulator can populate worlds with both static and dynamic objects, which can move along predefined trajectories or respond to physics simulations. This capability enables the evaluation of robotic systems in evolving environments, ensuring comprehensive testing of autonomous systems. Although primarily focused on space exploration, URSim can be adapted for indoor scenarios, such as testing the humanoid robot Rollin' Justin in household or office environments, as shown in Figure 4.8.

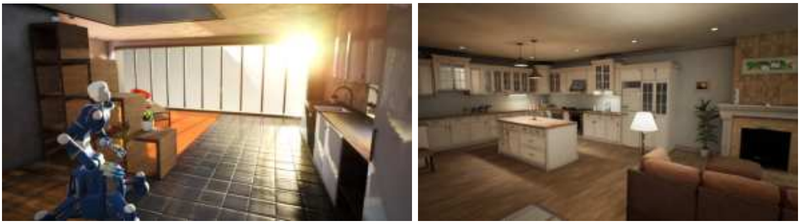


Figure 4.8.: Simulated indoor environments including the robotic system Rollin' Justin. Left image shows a modern apartment with many reflecting surfaces. Right images shows a typical western open-space living room with wooden furniture. All images are captured in the simulation environment.

4.4. Summary

The introduced MDAL module provides several data streams that can be used for further processing. Section 4.1.1 focused on the acquisition of visual data and how 3D objects are projected onto a 2D plane and how depth estimations can be calculated based on known feature correspondences in a given stereo setup. Section 4.1.2 introduced the acquisition of acoustic signals and the state formulation for a given constellation of microphones in 3D space. Section 4.2 discussed the importance of estimation the configuration state of the system itself to obtain information of the position and orientation of sensors. Finally, Section 4.3 showed two alternatives to online data acquisition, a pre-recorded dataset which is specifically designed for the indoor environment, and a photo-realistic simulation.

Either data, online from real sensors or from an emulation method, can be fed to the ego-state estimation (Chapter 5, pp 69) or landmark estimation (Chapter 6, pp 69) module. ESE received a continuous stream of data for uninterrupted localization of the robot itself. LE, triggered by events, estimates the localization of external landmarks and fuses this information with the ego-pose obtained in ESE.

5. Ego-State Estimation

Ego-state estimation (ESE) is the heart of the robot's localization system, providing robust and fault-tolerant estimation of the robot's movement. This process involves integrating data from various sensors, including visual and inertial, to accurately track the robot's trajectory and create detailed maps. Visual odometry, which uses image sequences to compute changes in position and orientation, plays an essential role in this system. Fault detection and correction mechanisms ensure the reliability of the estimates, allowing the robot to adapt to changes and maintain accurate localization over time. Additionally, ego-motion estimation supports mapping by providing accurate pose information that is essential for building and updating environmental maps. Overall, ego-state estimation (ESE) enables the robot to maintain a continuous representation of its pose history, ensuring effective navigation and interaction within its environment. An architectural drawing including all components is given in Figure 5.1.

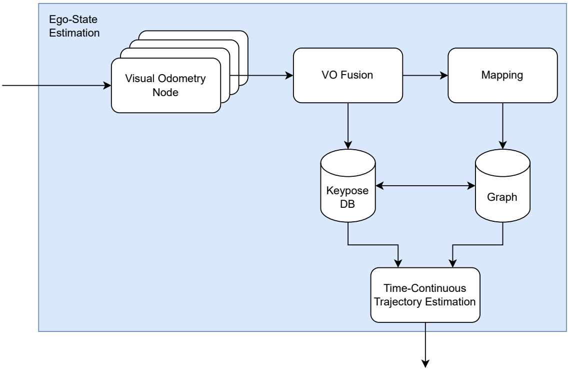


Figure 5.1.: Overview of the ego-state estimation sub-module.

5.1. Visual Odometry

The task of this node is to estimate the self-motion of the system.

Definitions

The **ego-motion** refers to the motion of a camera or other vision sensor relative to its environment. In robotics and computer vision, it specifically describes the process of estimating the sensor's own movement through the space it observes. It is defined as the trajectory and orientation of the sensor between two frames. This involves analyzing changes in position and perspective over time to determine how the sensor has moved.

Odometry is the accumulated motion since a defined reference frame, often the starting frame of the system. It involves continuous measurements of the ego-motion. Odometry is heavily affected by drift, which is the accumulation of errors that occur during the estimation of the motion.

To estimate its ego-motion, a robot compares consecutive camera images, extracting visual features to determine the displacement between them. Often this is supported by inertial measurements from an IMU to refine the estimation result. In this work, a vision-only odometry system is discussed.

Further, a definition for the terms Keyframe, Keypose as well as a virtual pose and clarify their distinction.

A **Keyframe** is a selected frame used by a VO module to estimate motion for consecutive frames. It represents a unique viewpoint for motion estimation.

The **Keypose** is the associated platform pose at the time the Keyframe is sampled. It reflects the pose at that particular point in time estimated using all available data. Since Keyframes are selected independently by each VO instance, a Keypose can have multiple Keyframes if they are sampled simultaneously by chance.

In the case, the pose has been determined based on a interpolation than using an actual Keyframe, we denote it as a **virtual pose**.

Feature Extraction

The first step is keypoint detection, which identifies distinctive points in an image that can be reliably tracked across multiple frames. These points are used for estimating the motion of the camera in visual odometry systems. Keypoint detection algorithms vary, but they generally aim to find points with strong local image gradients that are stable under transformations.

The Harris Corner Detector [46] identifies corners by analyzing the eigenvalues of the second-moment matrix (structure tensor) of image gradients. The algorithm computes image gradients using operators like Sobel. For each pixel, a second-moment matrix is formed from these gradients. The response for each pixel is calculated based on the determinant and trace of this matrix, highlighting areas with significant gradient variation. A threshold is applied to the response values to identify corners. This method is useful for detecting corners in structured environments.

The Shi-Tomasi Corner Detector [121] extends the Harris Corner Detector by using the minimum eigenvalue of the second-moment matrix for corner detection. Instead of a combined response function, it directly uses the smaller eigenvalue to determine the presence of a corner. This results in more stable and reliable corner detection, making it a preferred choice in many applications.

Features from Accelerated Segment Test (FAST) [108] is a highly efficient algorithm for real-time applications. It uses a circle of 16 pixels around a candidate pixel to determine if it is a corner. The intensity of the candidate pixel is compared with the intensities of the surrounding circle. If there are n contiguous pixels in the circle that are all either significantly brighter or darker than the candidate pixel, it is classified as a corner. This method is known for its speed and is well-suited for applications requiring fast keypoint detection.

The Adaptive and Generic Accelerated Segment Test (AGAST) [76]improves on FAST by introducing a decision tree that adapts to different image regions. This adaptation allows AGAST to maintain high detection speeds while increasing the robustness of the keypoint detection across various scales and rotations. The decision tree approach of AGAST ensures that the detection process can efficiently handle changes in image characteristics, providing more reliable keypoint detection under varying conditions.

Most state-of-the-art (SotA) approaches for VO use one of the aforementioned methods to find keypoints. However, these detectors often find features in local clusters, as shown in Figure 5.2. In a motion estimation system, this clustering can be problematic. Features may be too close together, and a single movement might remove the majority of them from the image. This reduces the estimation quality and can even prevent it. To overcome these limitations, a common approach is bucketing. This method splits the image into several smaller cells, often arranged as a grid. The keypoint detector is run for each cell, and the best scoring features are kept. This approach distributes the features across the image but may include low-scoring features in areas that do not offer reliable keypoints.

A different and globally-optimal distribution is achieved using range trees. The range tree adaptive non-maximal suppression (RT AMNS) [8] approach efficiently manages and distributes keypoints in an image by leveraging a range tree data structure. This data structure is a binary search tree where each node contains a nested data structure to facilitate multidimensional range queries. Initially, keypoints detected in the image are sorted based on their strength or cornerness score. These sorted keypoints are stored in the range tree, allowing for quick and efficient querying. During the suppression process, the algorithm iterates through the keypoints, starting with the strongest. For each keypoint, it uses the range tree to identify neighboring keypoints within a predefined search range. These neighboring keypoints, if found to be less significant, are suppressed to ensure that only the most relevant and well-distributed keypoints are retained.

The efficiency of RT AMNS comes from its ability to perform these range queries quickly, significantly reducing the computational complexity compared to brute-force methods. The search range is dynamically adjusted through a binary search process, optimizing the balance between the number of keypoints and their spatial distribution. This method ensures a more homogeneous spread of keypoints across the image. The number of keypoints varies from application and image size, in the case of indoor robotics using RealSense with 640 × 480 pixel, a good value is around 800 features after suppression.

The next step is feature description. This the process of computing a representation for each detected keypoint, which can be used to match keypoints across different images. The goal is to create descriptors that are invariant to

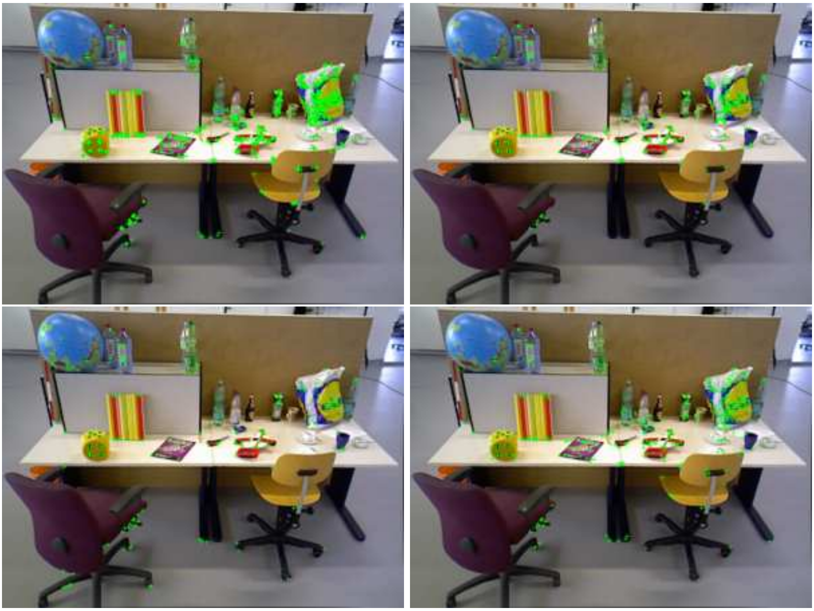


Figure 5.2.: Different distributions of features across a single image. Top left: The direct output of the ORB detector. Top right: Keeping the best *n* features. Bottom left: A 5 × 3-bucketing approach for detecting keypoints. Bottom right: Result of the global suppression method RT AMNS.

image transformations such as scale, rotation, and illumination changes. Common feature descriptors include SIFT, SURF, and ORB, each with different approaches to achieving robustness and efficiency.

The The Scale-Invariant Feature Transform (SIFT) [71] creates descriptors by extracting local image gradients around each keypoint and forming a histogram of gradient directions. These histograms are normalized to achieve invariance to illumination changes. The descriptors are then concatenated to form a high-dimensional vector, which can be used for matching. SIFT is known for its robustness to various transformations but is computationally intensive [135].

Speeded-Up Robust Features (SURF) [10] is designed to improve the speed of feature description while maintaining robustness. SURF uses a box filter approximation of the second-order Gaussian derivatives to compute image gradients efficiently. Similar to SIFT, it forms histograms of gradient directions but reduces the computational complexity by using integral images. This makes SURF faster than SIFT while still providing reliable descriptors [85].

Oriented FAST and Rotated BRIEF (ORB) [110] integrates the FAST [108] keypoint detector with the Binary Robust Independent Elementary Features (BRIEF) [20] descriptor, achieving both speed and rotation invariance. ORB determines the orientation of each keypoint using the intensity centroid method, and computes the BRIEF descriptor relative to this orientation. This method balances computational efficiency and descriptor robustness, making it suitable for real-time applications [86]. The binary descriptor used by ORB enhances matching speed. For BRIEF, the Hamming distance between the two 265 bit descriptors is calculated using a simple `AND` operation. In contrast, SIFT and SURF perform matching based on float comparisons.

Feature Tracking

For estimating motion based on images, the detected features must be tracked across a series of frames. A naive approach involves rerunning the detection pipeline and attempting to match the features with those from the previous frame. This method is resource-intensive and significantly impacts execution time. Without prior knowledge, a brute-force method must be employed, matching each feature from one image with all features from the other. If prior information is available, such as through a motion model based on previous estimations, the search radius for corresponding features can be reduced, improving execution time.

To further enhance performance, the Fast Library for Approximate Nearest Neighbors (FLANN) [89] can be employed. FLANN is an optimized algorithm for large datasets that accelerates the process of finding approximate nearest neighbors. By using a hierarchical clustering method, FLANN reduces the computational complexity of feature matching. This results in faster execution times and allows real-time processing even on standard hardware. Utilizing FLANN for feature matching can significantly increase the efficiency of motion estimation systems, providing a practical solution for applications requiring high frame rates and real-time processing. Typical average frame rates range from 10 Hz to 15 Hz on modern hardware, such as an Intel i7 10th generation processor.

Further improvements can be achieved by utilizing the fact that, with higher frame rates, the change in pixel positions of observed features is limited, and the similarity between the current frame and a previous one is significant. Instead of sampling, describing, and matching new features, already known features are searched for in the new image. The Lucas-Kanade method [73] assumes that the flow is essentially constant within a small window of pixels. This allows it to compute the flow vectors by solving a set of linear equations derived from the image intensity gradients. The method uses a least-squares approach to minimize the error in the image brightness constancy equation, resulting in a robust estimate of the optical flow.

Given the image intensity partial derivatives $I_x(\mathbf{q})$ and $I_y(\mathbf{q})$ and the temporal gradient $I_t(\mathbf{q})$ for a pixel $\mathbf{q}$, the optical flow $\mathbf{v} = (u, v)^T$ can be computed by minimizing the sum of squared errors in the image brightness constancy equation:

$$\min_{\mathbf{v}} \sum_i \big(I_x(\mathbf{q}_i)u + I_y(\mathbf{q}_i)v + I_t(\mathbf{q}_i)\big)^2 \tag{5.1}$$

for all pixels $\mathbf{q}_i$ that are part of the evaluation window. Applying the optical flow to the given features coordinates, we receive the estimated position for each in the new frame.

While this approach is faster than the naive method, it may introduce outliers and inaccuracies that negatively affect ego-motion estimation. Feature tracking based on optical flow is a directed operation between two frames. A straightforward method to check the correctness of the result is to apply the approach in the reverse direction. This process, known as backward tracking, reverses the order of the images and attempts to track the features back to the source image. Afterward, the distance between the original position of the feature and the position obtained from backward tracking is calculated. If the distance exceeds a threshold, such as 2 pixels, the feature is rejected.

Finally, due to occlusion as seen in Figure 5.3, features can merge into a single one. This occurs when two features are detected at different depths and the motion of the camera causes the object in front to hide the object behind it. To address this, the distance to the nearest features is calculated after each tracking step. If the distance is below a given threshold, the feature with

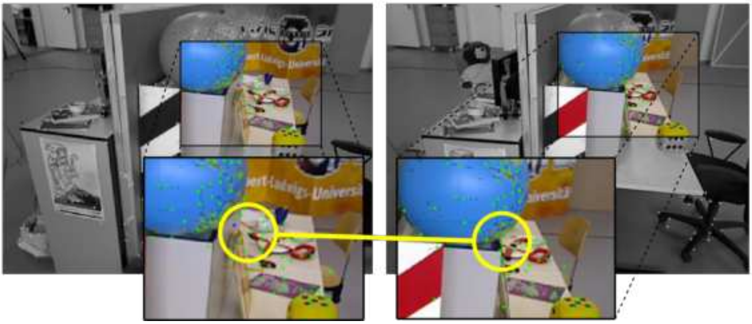


Figure 5.3.: Feature merging due to occlusion. An object in the front is hiding the feature in the back.

the higher distance is rejected as it either has the potential to merge or has already merged with another.

Feature Resampling

As features are rejected over time, resampling new features becomes necessary. This decision is based on two factors: the number of current features and their distribution. The feature count condition is determined by the original number of features from the initial sampling. If the ratio falls below 0.25, new features are sampled. The distribution condition is based on a statistical analysis of feature positions. The span of all features must cover at least 0.60 of the image dimensions. Additionally, the span of 50% of all features must be at least 0.20 of the image dimensions. This ensures that the features are not clustered, allowing for accurate tracking during fast movements, which prevents a loss of tracking (LOT) situation.

If resampling is necessary, features are detected as previously described. If new features are sampled close to existing ones, a matching step identifies if the same features have been sampled again. In this case, only the positions of the old features are updated. Otherwise, the old features are replaced by the new ones.

Keyframe Sampling

To reduce the amount of data inserted into the graph-structure for mapping, keyframes are used. Keyframes are selected based on their robustness and support for tracking the system's motion. During feature tracking, the number of stable features in each frame is recorded. A stable feature is defined as one that has been tracked continuously for at least $n_{stable} = 5$ frames. Every $n_{select} = 7$ frames, the frame with the most stable features is selected and added to a list of candidates for future keyframes. Finally, when 60% of features have been lost since the last keyframe, a new keyframe is picked from the set of candidates. For this selection, similar to Müller et al. [93], the pose for each candidate is calculated, and the one with the lowest normalized re-projection error is chosen. All candidates prior to the selected one are discarded. A maximum of $N_{candidates} = 5$ is kept for history.

Pose Estimation

To estimate the pose of the current frame, all feature correspondences with the last keyframe are selected. This is critical for maintaining consistency and accuracy in the pose estimation process. A point cloud is then constructed based on the 2D feature observations and the corresponding depth estimations. For each feature $(u, v)^T$, the 3D position $\mathbf{P}$ is calculated using the following formula:

$$\mathbf{P} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = d \cdot \mathbf{K}^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{d(u-c_x)}{f_x} \\ \frac{d(v-c_y)}{f_y} \\ d \end{pmatrix}, \tag{5.2}$$

Here, d represents the depth value at the feature point, as previous mentioned in Section 3.1 (p 29) $\mathbf{K}$ is the intrinsic camera matrix, and $(c_x, c_y)^T$ are the coordinates of the camera's principal point. The intrinsic matrix K is defined as:

$$\mathbf{K} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{5.3}$$

where f_x and f_y are the focal lengths in the x and y directions, respectively.

To determine the relative pose offset of the current frame with respect to the second image, the 3D points are projected into the second frame. This projection involves applying a transformation defined by a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$. The goal is to minimize the reprojection error, which is the difference between the observed 2D points $\mathbf{p}_i$ and the projected 3D points $\pi(\mathbf{R}\mathbf{P}_i + \mathbf{t})$. This is formulated as an optimization problem: [67]

$$\min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{n} \|\mathbf{p}_i - \pi(\mathbf{R}\mathbf{P}_i + \mathbf{t})\|^2 \tag{5.4}$$

In this equation:

- $\mathbf{p}_i$ represents the observed 2D points in the second image.
- $\mathbf{P}_i$ denotes the 3D points reconstructed from the first image.
- $\mathbf{R}$ is the rotation matrix that aligns the coordinate system of the first image with the second image.
- $\mathbf{t}$ is the translation vector that represents the position shift from the first image to the second image.
- $\pi(\cdot)$ is the projection function defined in Section 3.1 on page 29.

The optimization aims to find the optimal $\mathbf{R}$ and $\mathbf{t}$ that minimize the reprojection error across all feature correspondences. This involves iterating through possible values of $\mathbf{R}$ and $\mathbf{t}$ and evaluating the sum of squared differences between the observed and projected points. The method is implemented using the dense Cholesky factorization [28] from the ceres solver [3].

Finally, the resulting pose, which is relative to the keyframe, must be transformed into the global frame. This transformation is achieved by concatenating the computed relative pose with the keyframe's pose.

5.2. Visual Odometry Fusion

The approach described in Section 5.1 is initially for a single sensor. One key contribution of this work is extending this single-sensor estimation to a multi-sensor context. A graph fusion method is used to register and fuse multiple visual odometry (VO) modules. This multi-sensor approach aims to improve the reliability of motion estimation. Additionally, it enables a distributed architecture, supporting modern robotic systems with multiple computational nodes.[1]

Each VO module independently estimates a trajectory, represented initially as a series of discrete poses. By running independent VO modules, the local-optimal selection of keyframes is supported, which is important for robustness as discussed in Chapter 2 on page 13. These poses are then linearized, simplifying the trajectory. This linearized trajectory is then converted into a time-continuous representation using B-spline interpolation. This step maintains the temporal coherence of the estimated trajectory, ensuring accurate motion estimation.

After obtaining the time-continuous trajectory, it is registered and transformed into a common coordinate system. This step ensures that all trajectories, despite being from different sensors, are aligned and comparable. Accurate registration is essential for meaningful integration of data from multiple sensors.

Finally, the registered trajectories from the different VO modules are fused into a single, coherent motion model. This fused trajectory leverages the strengths and unique perspectives of each sensor, enhancing the system's overall accuracy and robustness, even in challenging environments. The fusion process helps to mitigate individual sensor weaknesses by combining their complementary data.

Trajectory Approximation

The ego-motion estimates provided by the VO system are structured as pose estimations connected by delta poses. These estimates are transformed into a time-continuous approximation, evaluable at any point between sensor

---

[1] Publication 8

measurements. This enables independent VO modules to operate optimally based on their observable field-of-view.[2]

A generally accepted approach in the automotive sector is using B-splines [148] for converting a discrete set of poses into a continuous representation. B-splines efficiently calculate the first and second-order derivatives at an evaluation point, making them useful for motion-model approximation. However, applying this to indoor service robotics, the system faces higher acceleration changes and spontaneous direction changes. While B-splines smooth the motion in automotive cases, this leads to missing trajectory coverage in corners for indoor cases.

To overcome this, a linear-motion constraint for the keyframe-sampling is required. The trajectory is linearized by reducing graph nodes that can be represented by linear motion. A set of consecutive poses $\{\mathbf{T}_0, \mathbf{T}_1, ..., \mathbf{T}_n\}$, where $\mathbf{T} \in \mathrm{SE}(3)$, is defined as linearizable if any $\mathbf{T} \in \{\mathbf{T}_1, ..., \mathbf{T}_{n-1}\}$ can be explained by linear motion from $\mathbf{T}_0$ to $\mathbf{T}_1$. An acceptable positional error $e_{\mathrm{lin,t}}$ and angular error $e_{\mathrm{lin,R}}$ enhance robustness against small estimation errors.

The result is a sparse set of discrete pose estimations whose density depends on the change of the first derivative of the initial trajectory. To achieve a continuous time representation, as proposed by Yang et al. [148], cumulative Spline-Fusion is applied. This ensures a twice continuously differentiable representation essential for motion models.

The initial set of discrete poses serves as control points for the B-spline, which fits a smooth curve through the data points. The B-spline method allows for efficient calculation of derivatives for modeling motion dynamics. By using cumulative Spline-Fusion, the resulting trajectory maintains a smooth and continuous profile, accommodating sudden changes in motion while ensuring accuracy in pose estimations.

The process is illustrated in Figure 5.4.

Combined Trajectory

To obtain a unified motion estimate, it is necessary to fuse the trajectories from all sensors into a common reference frame. Each sensor provides independent

---

[2] Publication 8

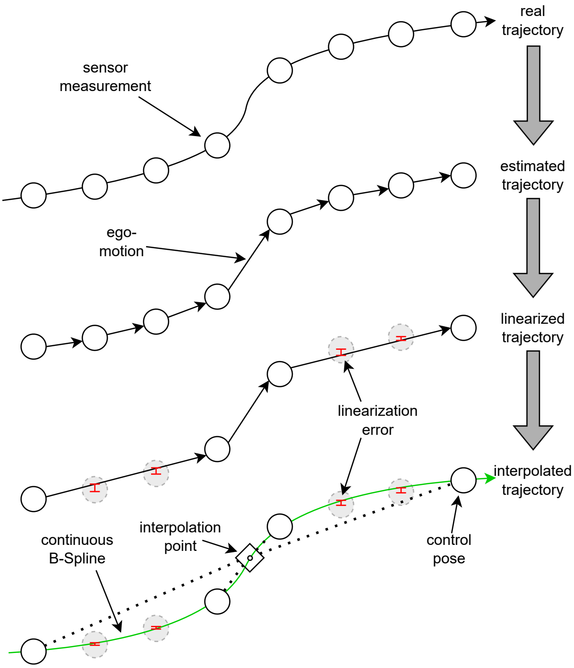


Figure 5.4.: Step-by-step illustration of acquiring a time-continuous interpolated motion-model from a sensor's trajectory.

trajectory estimates, which must be transformed into the robot's coordinate system due to their different positions on the platform.[3]

$$\mathbf{T}_O = \mathbf{T}_n^O \mathbf{T}_n, \tag{5.5}$$

where $\mathbf{T}_n^O$ denotes the transformation from sensor n to the robot's origin O, and $\mathbf{T}_n$ and $\mathbf{T}_O$ are the respective poses in their local frames.

For each new inserted Keyframe, the system queries the local motion estimates from all trajectories. These estimates are referred to as virtual poses, which

---

[3] Publication 8

are approximated based on earlier computations. The final fused pose is computed using a weighted combination of all virtual poses, with translation and rotation handled separately.

The translational component is defined as:

$$\mathbf{t}_{Platform}(t) = \sum^{n} w_n \mathbf{t}_n(t) = \sum^{n} \begin{bmatrix} w_n t_{1,n}(t) \\ w_n t_{2,n}(t) \\ w_n t_{3,n}(t) \end{bmatrix}, \tag{5.6}$$

where $\mathbf{t}_n(t) \in \mathbb{R}^3$ is the virtual position of sensor n at time t, and $w_n \in [0, 1]$ is its corresponding weight (see Equation (5.10)).

For the rotational component, the Spherical Linear Interpolation (Slerp) [122] algorithm is employed, which interpolates between two quaternions $\mathbf{q}_0$ and $\mathbf{q}_1$ along the shortest path on the unit quaternion sphere. This method is extended to multiple quaternions by concatenating individual Slerp operations:

$$\text{Slerp}(\mathbf{q}_0, \mathbf{q}_1, t) = \frac{\sin((1-t)\theta)}{\sin(\theta)} \mathbf{q}_0 + \frac{\sin(t\theta)}{\sin(\theta)} \mathbf{q}_1, \tag{5.7}$$

where θ is the angle between $\mathbf{q}_0$ and $\mathbf{q}_1$, and $t \in [0, 1]$ is the interpolation factor.

When applying this to multiple quaternions, the weights $w_0, w_1, \ldots, w_n \in [0, 1]$, where $\sum_{i=0}^{n} w_i = 1$, are corrected using:

$$u_j = \frac{\sum_{i=0}^{j} w_i}{\sum_{i=0}^{j+1} w_i}. \tag{5.8}$$

Using these corrected weights, the concatenated Slerp operations are as follows:

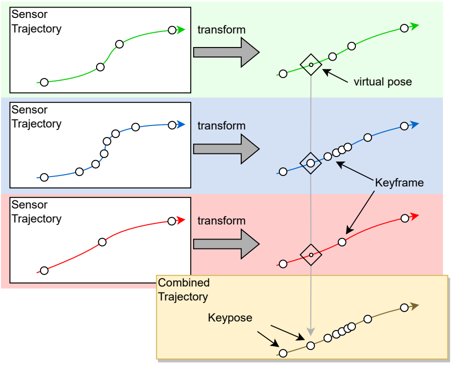


Figure 5.5.: Illustration showing the fusion of several independent estimation tracks into a single trajectory. The blue trajectory uses a directly computed pose based on a received Keyframe, the others use an interpolated virtual pose. The result is a combined estimated denoted as Keypose.

$$
\begin{aligned}
\widetilde{\mathbf{q}}_1 &= \mathrm{Slerp}(\mathbf{q}_0, \mathbf{q}_1, u_0)\,, \\
\widetilde{\mathbf{q}}_2 &= \mathrm{Slerp}(\widetilde{\mathbf{q}}_1, \mathbf{q}_2, u_1)\,, \\
&\dots \\
\widetilde{\mathbf{q}}_n &= \mathrm{Slerp}(\widetilde{\mathbf{q}}_{n-1}, \mathbf{q}_n, u_{n-1})\,, \qquad (5.9)
\end{aligned}
$$

resulting in the final quaternion.

The weighting of each track depends on the temporal distance from the nearest measurement. The weight for each track n at time t is defined as:

$$
w_{n,t} = \eta \max\left(1 - \frac{2\Delta t}{t_2 - t_1}, \hat{w}_{min}\right), \qquad (5.10)
$$

where $\hat{w}_{min}$ is a lower weight limit, and η normalizes the sum of weights to 1.

The final pose is applied for two purposes: registering new tracks to the platform and updating the fused pose for keyframe insertion into the global graph.

5.3. Mapping

In simultaneous localization and mapping (SLAM), representing a map using a factor graph involves key components that encapsulate the relationships between different variables. A factor graph is a bipartite graph consisting of variable nodes and factor nodes, where edges represent probabilistic dependencies among these nodes. The primary variable nodes in SLAM are pose variables, representing the robot's positions and orientations over time, and feature variables, denoting the positions of features in the environment. Factor nodes represent constraints or measurements that relate the variable nodes. In SLAM, these typically include odometry factors connecting consecutive robot poses, observation factors linking robot poses to landmarks, and loop closure factors connecting non-consecutive poses to correct drift in the map. The construction of a factor graph involves initializing with an initial guess of the robot's pose and landmarks' positions, incorporating odometry factors as the robot moves, adding observation factors when landmarks are observed, and handling loop closures to ensure global consistency. The mapping system in this work is based on the ORBSlam2 framework [92], which utilizes keyframes to build a map incrementally. ORBSlam2 allows for both tracking and mapping through the extraction and matching of keypoints across consecutive frames, ensuring robustness in various indoor environments.

Optimization in factor graphs is important for finding the most likely configuration of the robot's poses and landmark positions given the measurements. This is achieved through graph-based optimization techniques like Gauss-Newton [14] or Levenberg-Marquardt [88], which minimize the error represented by the factor nodes. The error function quantifies the difference between the predicted measurements, based on the current estimate of the variables, and the actual measurements. Incremental methods such as iSAM (incremental Smoothing and Mapping) [58] allow for real-time updates and re-optimization of the factor graph as new measurements are obtained.

Factor graphs offer several advantages in SLAM. They provide scalability, handling large-scale SLAM problems efficiently. The modularity of factor graphs, with a clear separation between variables and factors, makes it easy to integrate various types of sensors and motion models. Additionally, the flexibility of factor graphs allows for the incorporation of different types of measurements and constraints, providing a versatile framework for SLAM. This flexibility supports a modular approach, enabling the system to adapt to various scenarios and configurations. Overall, this structured and efficient representation facilitates robust and scalable solutions to the SLAM problem, and enables serialization of the underlying data structure.

In this work, the mapping capabilities of ORBSlam2 were extended to allow for the serialization of map data, enabling the system to save and load maps, improving reusability and adaptability in dynamic environments. Furthermore, a mapping system was introduced that allows for fast, reliable, and adaptable mapping of changing environments, making it suitable for real-world applications where environments are frequently altered.

Map Serialization

The map comprises several data classes:

1. Keyframe poses
2. Odometry estimations between Keyframes
3. Loop closures between Keyframes
4. Feature positions
5. Observations of features at different Keyframe poses

Figure 5.6 illustrates this relationship. The feature positions are explicitly included in the map, even though they can be derived from the keyframe poses and observations. This explicit inclusion has advantages. First, it accelerates the process of loading the map into memory by eliminating the need for re-triangulating feature positions. Second, once the positions have converged, parts of the map can be separated, creating independent maps. This is useful when maps become large or when only specific parts need to be serialized and saved.[4]

Furthermore, map serialization enables post-creation map editing. Using detected feature points, transformations between the map and the world or individual objects can be estimated. Additionally, features can be disabled for use in localization. This technique is used to remove objects from the scene that typically cause errors due to their appearance, such as screens,

---

[4] Publication 4

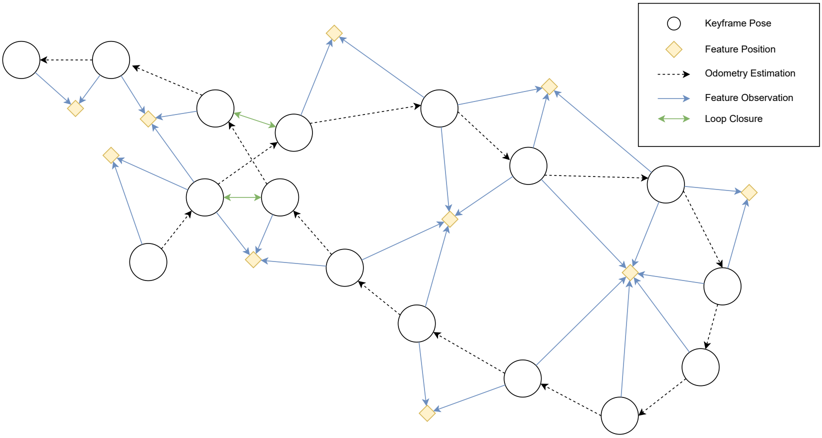


Figure 5.6.: A typical pose graph representing the data structure behind the mapping process. (Structure is simplified for better visualization)

plants, and curtains. By disabling these features, they are excluded from pose estimation.

Long-Term Map Creation

Creating a map for long-term use in indoor scenarios is a multi-step task performed by an expert. The process begins with the creation of a minimal map for re-localization. This map should be a small but distinguishable representation of the environment that remains constant over time. Examples include static furniture components like a kitchen counter, artistic objects like a sculpture, or electrical outlets in the wall. The key is that their location is fixed and their appearance is static.

Evolving from this minimal map, an extensive and detailed map is created. The purpose of this map is to cover the entire area of operation and achieve a converged representation. First, the system maps the circumference of the area. Second, it triggers loop-closures by linking diagonal points on this outline. The loop-closures between distant poses compensate for significant drift accumulated over time, ensuring fast map convergence. This detailed representation can be used to align the obtained map with other world representations, such as CAD representations of the environment or

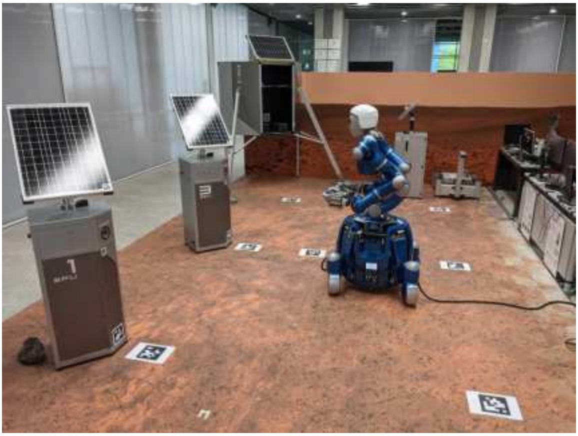


Figure 5.7.: Estimation process using artificial markers placed on the floor and captured by the robotic system.

predefined and known artificial markers (see Figure 5.7). Typically, several known landmarks are observed, and their positions are compared to the observations based on the detailed and converged map. An optimizer is then used to estimate a transformation that minimizes the spatial error between the observed and actual poses of the landmarks.

Finally, the minimal map and the obtained transformation can be used to create an application-specific map. This application map includes only the typical poses of the robot during its operation. The transformation from the detailed map is applied to the origin of the minimal map, ensuring accurate positioning since the origin is always the first Keyframe and has no drift. This makes the transformation applicable to any map derived from the minimal map. If the environment changes, a new map can be quickly created using this method, allowing the robot to adapt to different applications efficiently. Exemplary maps for DLR's Robotic and Mechatronic Center in Oberpfaffenhofen are shown in Figure 5.8.

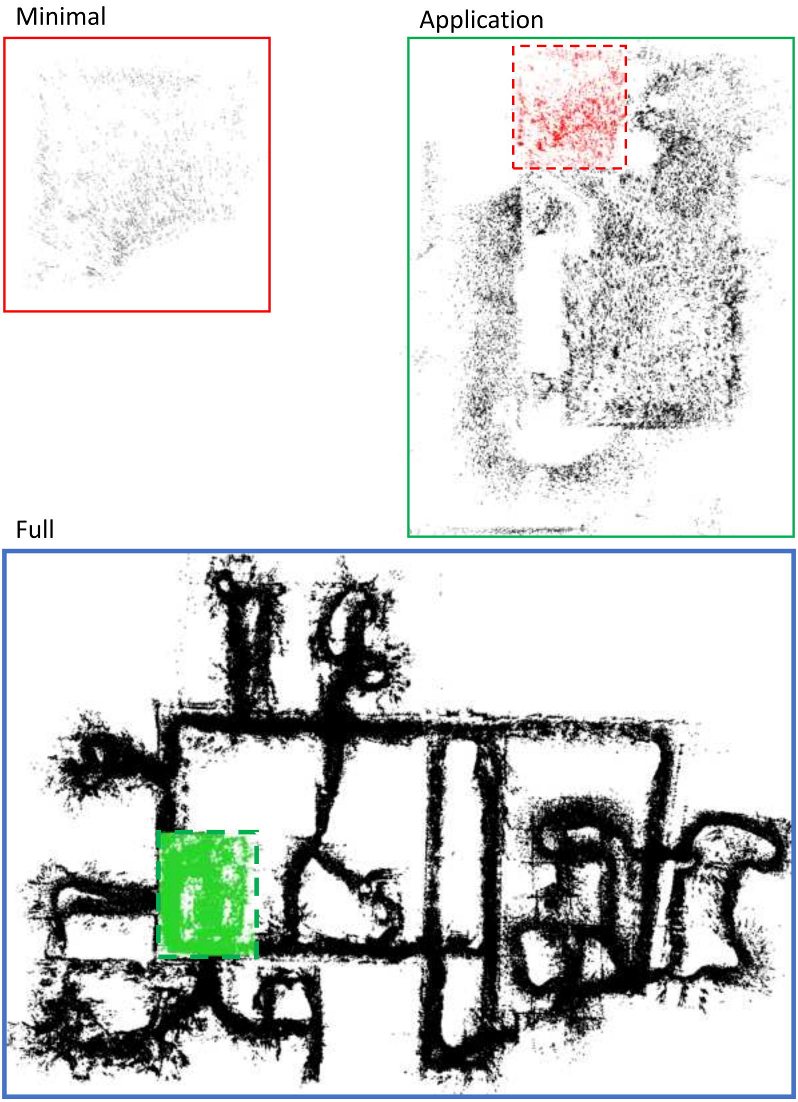


Figure 5.8.: Exemplary illustration of the different maps for the Robotic and Mechatronic Center at the German Aerospace Center in Oberpfaffenhofen. Top-left shows the minimal map for obtaining the world transformation information. Top-right is the application map used to start a mission. Bottom is the full map after a single mission.

6. Landmark Estimation

The sub-module described in Chapter 5 focuses on the localization of the robotic system itself in a quasi-static environment. This chapter explores the localization of external, possible dynamic landmarks. The primary focus will be on multi-modal detection for audio sources. However, the concepts can also be applied to the visual domain. An overview is given in Figure 6.1.

6.1. Landmarks

A key difference between detecting features for ego-state estimation and external landmarks is that the latter may not be continuously observable. Speakers emit sound only when actively speaking, and are undetectable during breaks or periods of silence. Objects produce different spectral profiles based on their internal state, and noise sources can disrupt the estimation. Smaller objects may be removed from the scene. Additionally, external landmarks may change location over time, a property not modeled in the proposed ego-state estimation approach. For example, a person may walk while speaking, causing the source's position to be dynamic during the estimation process.

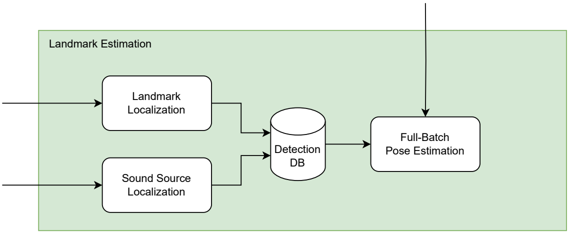


Figure 6.1.: Overview of the sub-module architecture

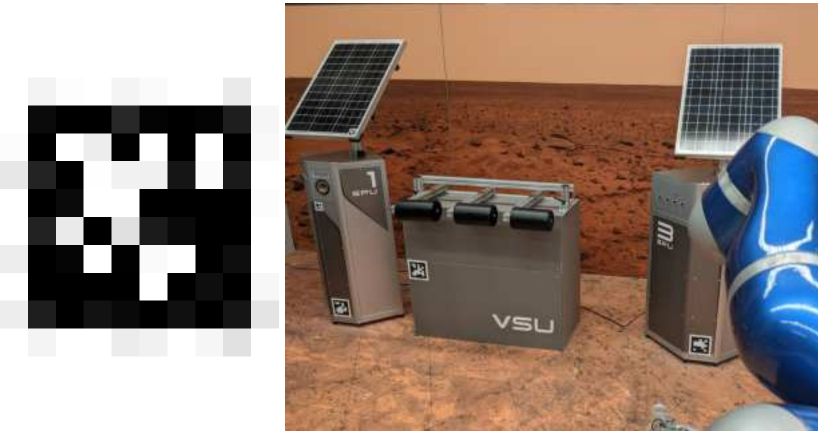


Figure 6.2.: AprilTag [145] fiducials used as visual aids for localization of objects in the environment.

6.1.1. Visual Landmarks

Detecting landmarks using visual sensor systems is a well-researched topic. There are different approaches to obtain pose information for an object. A common and simple approach is using artificial markers, called fiducials, mounted at defined positions. By detecting these markers in the image and estimating their 6D pose, the overall pose of the object can be determined. AprilTags [145], a fiducial family developed by the April Labs at the University of Michigan, are often used for this purpose. An example is given in Figure 6.2.

A more complex but less intrusive approach involves using neural networks to estimate the 6D pose directly from the object's visual appearance. This method does not require adding markers to the environment, making it suitable for indoor use, especially in environments shared with humans.

An in-depth discussion of methods for visual detection of objects and estimation of their pose is beyond the scope of this work.

6.1.2. Sound Sources

Before diving into the localization of sound sources, the signal space for a microphone array as described in Section 3.2 is defined. The sound source is modeled as a point emitting a sinusoidal wave with center frequency f_k and corresponding time-dependent amplitude $\lambda_k(t)$, where k is the index of one out of K frequency bands. Using the complex frequency notation, this is expressed as

$$s(t) = \lambda_k(t)e^{j2\pi f_k t} = \lambda_k(t)e^{j\omega_k t} \quad . \tag{6.1}$$

Consider a sensor array consisting of N microphones, leading to the system equation

$$\begin{bmatrix} 1 \\ e^{-jw_k\Delta_1} \\ \vdots \\ e^{-jw_k\Delta_{N-1}} \end{bmatrix} s(t) =: \mathbf{a}_k s(t) \quad , \tag{6.2}$$

where Δ_n is the relative propagation delay with respect to the n-th reference microphone. For a one-dimensional linear microphone array and under the assumption of planar waves, the delay is calculated as

$$\Delta_n = \frac{d_n \sin(\theta)}{c_0} \quad , \tag{6.3}$$

where d_n is the sensor's distance to the reference, θ the direction of arrival, and c_0 the speed of sound (see Equation (3.11)), approximately $334\,m/s$ at room temperature. The vector $\mathbf{a}_k \in \mathbb{C}^{\mathrm{N}}$ in Equation (6.2) is denoted as the steering vector for the frequency f_k. To obtain the complete signal vector, the system equation is extended to

$$\mathbf{x}(t) = \mathbf{a}_k s(t) + \mathbf{n}(t)\,, \tag{6.4}$$

where $\mathbf{n}(t)$ represents additional uncorrelated system noise.

Subspace Approach

When a new signal is received, it is split into smaller frames of fixed length and transformed into the frequency domain. The correlation matrix $\mathbf{R} \in \mathbb{C}^{N \times N}$ is computed using

$$\mathbf{R} = \overline{\mathbf{X}(k)\mathbf{X}^{\mathrm{H}}(k)} \ , \tag{6.5}$$

where $\mathbf{X}(k) \in \mathbb{C}^{N \times F}$ contains the transformed Fourier coefficients of band k for all F frames and N microphones. Here, $\mathbf{X}^{\mathrm{H}}$ denotes the Hermitian of $\mathbf{X}$. Applying singular value decomposition (SVD) on $\mathbf{R}$ to separate the contained subspaces results in

$$\begin{aligned} \mathrm{SVD}\left(\mathbf{R}\right) &= \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}} && (6.6) \\ \mathbf{U} &= [\mathbf{u}_0 \quad \mathbf{u}_1 \cdots \mathbf{u}_{N-1}] \\ &= [\mathbf{U}_{\mathrm{S}} \quad \mathbf{U}_{\Sigma}] \ , && (6.7) \end{aligned}$$

where $\mathbf{U}_{\mathrm{S}}$ represents the signal space and $\mathbf{U}_{\Sigma}$ the noise space. As the system noise is uncorrelated, it is present in all subspaces. The previously defined steering vector $\mathbf{a}_k$ is a property of a receiving signal and thus defined in the signal space. This implies

$$\begin{aligned} \boldsymbol{a}_k &\in \boldsymbol{U}_{\mathrm{S}} \ , && (6.8) \\ \Rightarrow \boldsymbol{a}_k &\perp \boldsymbol{U}_{\Sigma} \ . && (6.9) \end{aligned}$$

Hence, the inner product (denoted as $\langle \cdot, \cdot \rangle$) of the steering vector and the noise space is zero.

Natural sound events, especially human speech, are composed of several frequencies. To account for this, the complete frequency spectrum is considered and combined into a single representation. A common approach for this is the broadband pseudospectrum, which is defined over all frequency bands K as

$$P(\theta) = \sum_{k=1}^{K} \frac{1}{\langle \mathrm{a}_k(\theta), \mathrm{U}_{\Sigma} \rangle^2} \ . \tag{6.10}$$

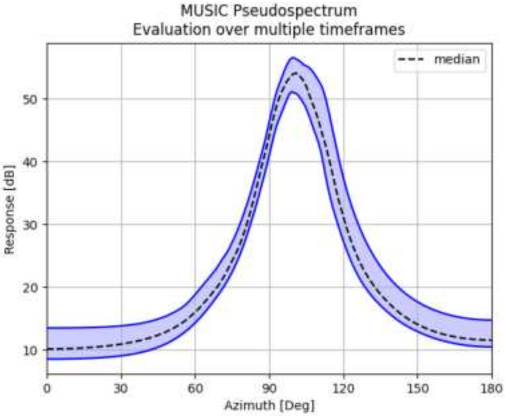


Figure 6.3.: Typical response for the MUSIC pseudospectrum. Shown is the median as well as the 25-th and the 75-th quantille for a series of estimations.

The DoA is found as the maximum of the estimator's response, i.e.

$$\tilde{\theta} = \operatorname{argmax} \quad P(\theta) \quad . \tag{6.11}$$

A typical response is exemplary depicted in Figure 6.3.

Adaptive Frequency Selection

Estimating the DoA is computationally intensive. Incorporating all frequency bands may prevent real-time applications. Most sound sources emit sound with base frequencies and their harmonics. Selecting these frequencies for estimating the DoA increases robustness against noise and reduces computation time. Naively selecting the strongest frequency and defining a bandpass filter around it does not exclude noise bands. A better approach compares the frequency bands against a known noise spectrum obtained in Section 4.1.2 on page 54. The bands with the highest divergence are then selected. The Long-Term Spectral Divergence (LTSD) approach by Ramirez et al. [104] can be applied to any sound source, even though it was initially designed for speech.[1].

The Long-Term Spectral Envelope (LTSE) is calculated to analyze the spectral characteristics of speech over a long period. Given a noisy speech signal $x(n)$, it is first decomposed into 25 ms frames with a 10 ms window shift. Let $X(k, l)$ denote the spectrum magnitude for the k-th frequency band at frame l. The LTSE is calculated using a $(2N + 1)$-frame window centered around the current frame l. It is defined as the maximum value of the spectral magnitude $Y(k, l + j)$ within this window:

$$\mathrm{LTSE}(k) = \max_{j=-N}^{N}\{Y(k, l + j)\} \tag{6.12}$$

Here, k represents the frequency band and l the current frame. This approach captures the spectral peaks over the window, providing a more stable representation of the spectral envelope in noisy conditions.

The LTSD is a metric used to determine the presence of speech. It measures the deviation of the LTSE from the estimated noise spectrum. The noise spectrum $N(k)$ is estimated and updated continuously during non-speech periods. The LTSD is defined as the logarithmic ratio of the LTSE to the noise spectrum averaged over all frequency bands:

$$\mathrm{LTSD} = 10\log_{10}\left(\frac{1}{N_{\mathrm{FFT}}}\sum_{k=0}^{N_{\mathrm{FFT}}-1}\frac{\mathrm{LTSE}^2(k)}{N^2(k)}\right) \tag{6.13}$$

Here, N_{FFT} is the number of frequency bands. The LTSD measures how much the current spectral envelope deviates from the noise estimate. However, only the information based on each band is needed for selection, simplifying eq. (6.13) to:

---

[1] Publication 3

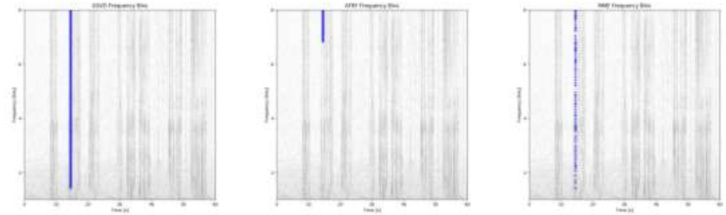


Figure 6.4.: Selected frequency bins for different approaches. Left: All estimated frequency bins used by GSVD. Center: Bandpass filter applied around the center frequency according Active Frequency Range Filtering [51]. Right: Adaptive approach based on Long-Term Spectral Divergence.

$$\mathrm{LTSD}'(k) = \frac{\mathrm{LTSE}^2(k)}{N^2(k)} \tag{6.14}$$

This is used to find the frequency bands with the highest divergence. The best $n_{Bands} = 50$ are then selected for estimating the DoA based on the pseudospectrum in Equation (6.10). An exemplary comparison for the selection is shown in Figure 6.4.

Motion-Model

The plausibility of the received angle is checked by evaluating it with a motion model. For the time span t_{mm}, it is assumed that the source moves with mean angular velocity $\bar{\omega}$, i.e.,

$$\bar{\omega}(t_{mm}) = \overline{\left(\frac{\Delta\theta}{\Delta t}\right)} \approx \frac{1}{M} \sum_{n \in \mathcal{N}(t_{mm})} \frac{\tilde{\theta}_n - \tilde{\theta}_{n-1}}{t_n - t_{n-1}}, \tag{6.15}$$

where $\mathcal{N}(t_{mm})$ is the index set of all M angular measurements $\tilde{\theta}_n$ within the time span t_{mm}. A subsequent measurement $\tilde{\theta}_{m+1}$ is considered valid if

$$\left|\tilde{\theta}_{m+1} - \bar{\omega}(t_{mm})\right| < \theta_{tol}, \tag{6.16}$$

with the constant motion tolerance $\theta_{tol} = 5^\circ$ which is the typical estimation accuracy of a microphone array.

When receiving a new DoA from the previous steps, all estimations within the time span t_{mm} are gathered. If at least two valid points are found, the motion model is used to verify the new one. Otherwise, all DoAs are used for the motion vector, requiring at least three estimations. The first estimations are used to calculate $\bar{\omega}(t_{mm})$, and the last one to verify the model. If the motion can be explained by the model, all DoAs are marked as valid estimations.

This motion model helps filter out echoes, as measurements stemming from echoes have a direction inconsistent with the source and occur shortly after the arrival of the original signal.

6.2. Landmark Pose Batch Optimization

The previous sections discuss the estimation of a landmark at a single point in time. However, this estimation is affected by measurement noise, leading to inaccuracies in the landmark's pose estimate. For example, as mentioned in Section 6.1.2, sound source localization is optimized for an accuracy of 5°. Integrating multiple measurements can reduce the uncertainty in these estimates.

Another source of uncertainty arises from the localization of the robot's ego pose, as previously described in Chapter 5 (pp. 69). Evaluations on the IndoorMCD dataset (Section 4.3.1, pp. 61) indicate a worst-case error of 10 cm and 4°. Additionally, as discussed in Section 4.2 (pp. 57), the robot's kinematics may be affected by configuration-dependent bending.

In scenarios involving multiple transformations with uncertainties, managing these uncertainties efficiently is essential for maintaining computational tractability. A common approach is to utilize Lie Algebra for representing and concatenating these transformations. The Lie group represents the transformations themselves, while the associated Lie algebra represents small perturbations around these transformations. The framework described in Section 4.2 (pp. 57) is particularly useful for systematically handling the nonlinearities and non-commutative properties inherent in robot kinematics.[2]

---

[2] Publications 6, 7 and 2

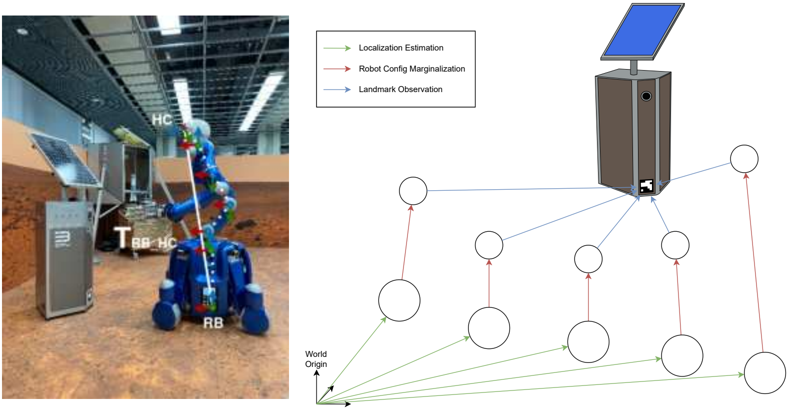


Figure 6.5.: Batch optimization of the pose estimation of an external landmark (here based on visual information). Left: The kinematic model of Rollin' Justin that is marginalized to a single transformation. Right: The optimization graph for the landmark pose estimation.

Marginalization is an important technique for reducing the complexity of the kinematic system. It involves integrating out certain variables to focus on a lower-dimensional subset, thereby simplifying the overall representation. For example, consider two transformations $\mathbf{T}_{AB}$ and $\mathbf{T}_{BC}$ with associated uncertainties represented by covariance matrices Σ_{AB} and Σ_{BC}. When combining these transformations, the overall uncertainty can be approximated by Equation (4.18) and Equation (4.19) on page 60

$$\begin{aligned}\mathbf{T}_{AC} &= \mathbf{T}_{AB} * \mathbf{T}_{BC},\\ \Sigma_{AC} &= \mathbf{Ad}_{\mathbf{T}_{BC}^{-1}} \Sigma_{AB} \mathbf{Ad}_{\mathbf{T}_{BC}^{-1}}^{T} + \Sigma_{BC}.\end{aligned}$$

Marginalization can then be applied to remove intermediate transformations or landmarks, updating the covariance of the remaining variables accordingly.

This entire process can be represented as a pose graph that incorporates these uncertain measurements. Figure 6.5 illustrates this graph for optimizing a landmark pose observed by the robotic system Rollin' Justin. Initially, the landmark is observed from multiple viewpoints. For each viewpoint, the observation in the sensor frame is recorded and included in the graph, alongside the ego-pose estimate and the estimated transformation from the

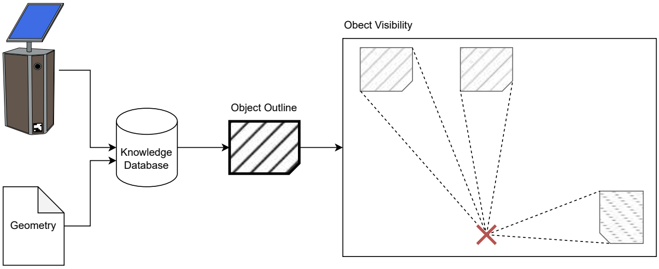


Figure 6.6.: An object's pose has been previously estimated using the batch optimization described in Section 6.2. Then, using the a-priori knwon geometry information in the knowledge database, the system is able project the visibility of the object in the environment with respect to the current robot pose.

robot's localization origin to the sensor origin. The entire robotic kinematic tree is marginalized to a single transformation.

The graph is constructed using all localization poses, connecting the corresponding sensor offsets and landmark observations. The final step in the optimization process involves the use of a smoothing and mapping (SAM) [58] approach. The optimization seeks to minimize the error between observed and predicted measurements, incorporating all localization, robot configuration, and landmark observations. This is achieved through non-linear least squares optimization leveraging Levenberg-Marquardt. Here the implementation of GTSAM [29] is used. The result is a globally consistent estimate of the landmark position with respect to all uncertainties in the system.

6.3. Audio-Visual Information Fusion

Audio and visual data from both modalities must be combined to achieve a fully multi-modal system[3]. Visual landmarks can directly estimate the pose of an object, whereas the SSL approach provides only a bearing. Since sound sources may emit audio signals for limited durations, a triangulation approach based on robot movement is not feasible. Instead, it is assumed that audio sources are contained within known objects.

---

[3] Publication 6.

First, objects are detected and mapped using a visual approach, and their poses are refined using the method outlined in Section 6.2 on page 96. The robot stores all information in a knowledge database [113], which includes not only world position and orientation but also geometric information such as 3D outlines, semantic properties, and other attributes. By querying this database, the robot can calculate the object's visible cross-section (VCS) relative to its current ego-pose, as shown in Figure 6.6. A received SSL bearing, including its estimation tolerance, is compared against the VCS of an object to determine whether the object is the source of the sound event. Since the plausibility is already checked and filtered by the motion model (Section 6.1.2, pp. 95), no further filtering is necessary.

Audio-visual data fusion occurs at a high level, after both modalities have been individually processed. The object's pose, refined using batch optimization, is the basis for comparison with the SSL-estimated audio bearing. The pose estimation from visual data is immediate, given the static nature of the visual observations. In contrast, audio source localization, a longer process, estimates the bearing based on the middle of the audio frame. It is assumed that during a single audio frame, the sound source remains static.

Uncertainty in the visual object's pose is derived from the batch optimization approach, which uses graph optimization implemented with the GTSAM library. This method provides a probabilistic measure of the pose accuracy. The uncertainty in the SSL bearing is obtained from empirical data, collected during experiments conducted in controlled environments such as anechoic chambers. These uncertainties from both modalities are then used to assess the plausibility of the object being the sound source.

Audio-Visual Information Fusion Challenges

One of the key challenges in audio-visual information fusion is ensuring that objects in the environment are distinguishable at any point in time during the fusion process. For successful fusion, a direct line of sight, or ray, must exist between the robot's sensors and the object in question. This ensures that both audio and visual data correspond to the same object, allowing accurate comparison of the SSL bearing and the object's pose.

However, the current system does not include a mechanism to handle cases where multiple objects are aligned along the same line of sight. Specifically, if one object is located directly behind another, the system may not differentiate between them. In such cases, the fusion process could mistakenly associate the SSL bearing with the wrong object. This limitation is particularly relevant in cluttered or dynamic environments where occlusions are common.

To address this, future work could involve incorporating methods to detect and handle occlusions, such as using depth data from cameras or introducing a more sophisticated environmental model that predicts object positions and handles visual occlusion. Additionally, filtering techniques could be applied to prioritize data from modalities that are less affected by occlusion in specific scenarios.

7. Evaluation

In the course of this work several publications (see Appendix A, pp. 137) have been prepared and presented. These publications evaluated the performance, accuracy, and applicability of the respective approaches. This chapter will highlight important results and put them into context for the whole system.

7.1. Multi-Sensor Approaches

As mentioned in Chapter 5 (pp. 69), adding more sensors to the perception system enhances two major aspects: robustness against LoT and an increase in the system's field-of-view (FoV).

While the increase in FoV is a direct consequence of using additional sensors with different viewpoints, the improvement in robustness stems from the system's ability to capture more features. With multiple sensors pointing in different directions, the likelihood of encountering views with insufficient or poor-quality visual features is significantly reduced. In a single-sensor system, certain scenes may lack enough distinctive features for accurate tracking, especially in scenarios with textureless surfaces or occlusions. However, with multiple cameras, the risk of this happening is minimized, as the system has access to complementary perspectives. This redundancy ensures that even if one camera's view lacks sufficient visual cues, others can continue to provide reliable feature data, preventing LoT events and ensuring continuous operation.

The positioning of the sensors plays a crucial role in determining the increase of the FoV. Parallel sensors with significant overlap in their views do not contribute to increasing the FoV, while sensors with completely different viewpoints can significantly expand it. However, depending on the specific application and scenario, overlapping views may be desirable, as they enhance redundancy and improve robustness. Therefore, a trade-off between expanding the FoV and maintaining overlapping views must be carefully considered.

Figure 7.1 provides a clear comparison of both configurations through the results of the same trajectory. The left image shows the map generated by a single-camera system, while the right image illustrates the map produced by a multi-camera system consisting of five cameras with four of them having non-overlapping views.

In the single-camera case, the system experienced a loss-of-tracking (LoT) partway through the trajectory. This is evidenced by the incomplete and fragmented map in the left image, where certain sections of the environment were missed entirely. The system was unable to recover from this tracking failure, leading to a substantial degradation in both map density and coverage. This highlights the vulnerability of single-camera systems, particularly in scenarios where the view becomes occluded or key feature points are lost due to sudden motion or environmental changes.

On the other hand, the multi-camera system (shown on the right) successfully tracked the entire trajectory without any LoT events. This is largely due to the increased FoV, where the additional cameras provided complementary viewpoints that compensated for occlusions or temporary loss of visual features. The result is a significantly denser map that covers a wider area, with more feature points accurately captured and aligned across the scene. The robustness against LoT is clearly visible in the right-side map, where the system's ability to consistently detect and track features throughout the entire path results in a more complete and reliable reconstruction of the environment.

The benefits of the multi-sensor configuration become particularly evident in complex or dynamic environments where a single camera might struggle with occlusions or fast movements. By integrating multiple cameras with varying viewpoints, the system can maintain continuous tracking and generate more reliable and detailed maps. Additionally, the redundancy offered by the overlapping views in certain configurations ensures that even if one sensor loses tracking temporarily, the other sensors can compensate, further enhancing the system's robustness.

Examining the individual improvements, we start with robustness. Typical reasons for losing tracking during ego-motion estimation in indoor environments include obstructed views due to proximity to obstacles, low-textured

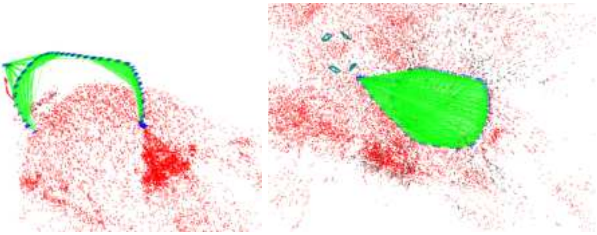


Figure 7.1.: Comparison of the same trajectory. Left side with a single camera, right side with five cameras. The single camera captures a limited view, while the multi-camera setup captures more feature points.

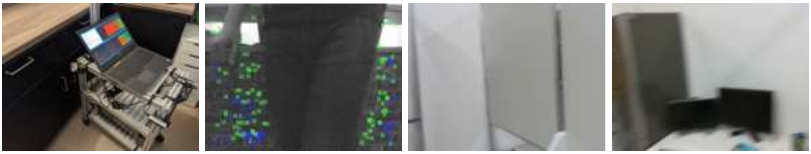


Figure 7.2.: Typical loss-of-tracking scenarios. These situation illustrate operation in close proximity to an obstacle, obstructed camera view, low-textured environment and motion-blur.

environments that lack sufficient features, and motion blur affecting feature detection and matching. These situations are displayed in Figure 7.2. The first two issues are examples where multi-sensor approaches can overcome the limitations of a single failing sensor. To address this, an extensive study using the IndoorMCD dataset was conducted. This study evaluated five different approaches for estimating a trajectory, along with two former single-sensor approaches using the fusion approach discussed in Section 5.2 (pp 79). The dataset is organized into six different scenarios, five of which offer high-accuracy ground truth information. In these scenarios including ground truth information, each estimation method is run on either a single sensor or all sensors, depending on the approach. A binary classification success-rate is defined to determine a successful and robust estimation. If the system reported valid tracking for at least 90 % of the time, it was considered successful. This threshold accounts for possible initialization issues at the beginning or short losses of tracking that could be immediately recovered.

The results are shown in Table 7.1. At first glance, it is clear that approaches using multi-sensors for estimation show superior performance in terms of robustness. Several single-sensor approaches, especially ORB-Slam2 and ORB-Slam3, struggle to estimate the trajectories reliably in challenging scenarios.

However, the approaches developed in the course of this work — namely MROSLAM, as well as the Multi- versions of ORB-Slam3 and VINS-Fusion — demonstrate significantly improved performance. The MROSLAM system uses fully independent, concurrently running VO and SLAM modules, with a final fusion of the pose estimate across the sensors. This design choice, while promoting robustness through independent operation, requires each individual module to regain tracking after a loss-of-tracking (LoT) event before continuing. In contrast, the Multi- variants of ORB-Slam3 and VINS-Fusion employ the graph fusion approach discussed in Section 5.2, where sensor data is shared between modules, allowing for online trajectory registration and enabling a more efficient recovery from LoT events.

Notably, the *Multi*-VINS-Fusion approach successfully estimates all trajectories in every scenario, highlighting the effectiveness of this multi-sensor extension. The fusion approach significantly impacts performance by allowing fast resets of individual modules after LoT, preventing prolonged loss of estimation and contributing to a more continuous ego-motion estimation. This is illustrated by the recovery behavior in Figure 7.3, where MROSLAM and the extended ORB-Slam3 are compared. The trajectory not only highlights the frequency of LoT events but also demonstrates the rapid recovery made possible by the fusion architecture in the extended ORB-Slam3, resulting in a more stable and reliable ego-motion estimation throughout the evaluation.

Overall, the evaluation confirms that the multi-sensor approaches, particularly the ones developed in this work, provide substantial robustness improvements in terms of handling LoT events and maintaining continuous motion tracking across a variety of challenging indoor scenarios.

Considering the system's FoV and the increased mapping area, a reliable indicator of system performance is the number of inserted observations into the final optimization graph. Figure 7.4 presents the number of features detected by single-, two-, three-, four-, and five-sensor systems. Each system was evaluated on three distinct trajectories: a linear path without any rotation, a spot rotation, and a loop that combines both translation and rotation.

As expected, the number of detected features increases with the number of sensors. This trend is a direct result of the expanded FoV provided by each ad-

Table 7.1.: Evaluation of the robustness of different approaches on the IndoorMCD dataset. Shown is the success-rate which indicates if the system was able to maintain valid tracking on all trajectories. MROSlam is a fully independent running Multi-VO system that fuses the final poses to a single estimate, *Multi*-ORB-Slam3 and *Multi*-VINS-Fusion use the fusion extension for handling multiple inputs. All three have been developed during the course of this work.

	S0	**S1**	**S2**	**S3**	**S4**
VINS-Mono [137]	0.82	0.93	0.88	0.92	0.93
VINS-Fusion [137]	0.81	0.86	0.96	**1.00**	0.93
ORB-SLAM2 [92]	0.00	0.24	0.00	0.63	0.73
ORB-SLAM3 [22]	0.05	0.00	0.05	0.75	0.53
MROSlam [119]	0.68	0.79	0.92	**1.00**	**1.00**
Multi ORB-Slam3	0.77	0.90	0.96	**1.00**	**1.00**
Multi VINS-Fusion	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

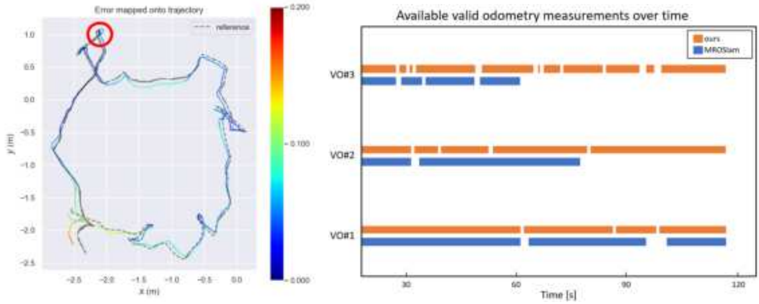


Figure 7.3.: Valid tracks for a three-sensor systems over time. Orange showing ORB-Slam3 with the Graph Fusion extension, blue the operation of MROSLAM on the same trajectory. The online registration of Graph Fusion enables the re-integration of lost estimation tracks and can faster recover a loss compared to completely independent running VO-modules.

ditional sensor. With each sensor pointing in a different direction or covering different parts of the environment, the system can capture significantly more visual features from multiple viewpoints. This broader coverage reduces the risk of missing landmarks, particularly in complex environments where parts of the scene might be occluded or out of view of a single sensor.

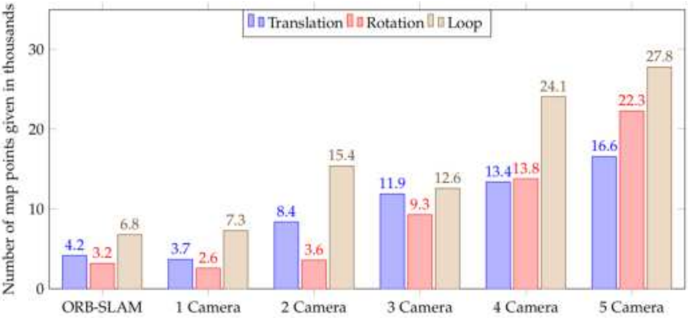


Figure 7.4.: Number of observed features for different amount of sensors. Each configuration is evaluated on three trajectories, a translation-only, a rotation-only and a combination of both.

For example, in the five-sensor configuration, the sensors collectively cover a much wider area than a single-sensor system. This not only allows the system to detect more features in total but also ensures that it can continue tracking even if some sensors temporarily lose sight of key landmarks due to occlusions or changes in the environment. The ability to capture more features contributes to the creation of a denser, more comprehensive map, which is crucial for accurate ego-motion estimation, especially in scenarios involving complex trajectories with both rotational and translational motion.

The increase in observed features, as shown in Figure 7.4, directly correlates with improvements in the accuracy and reliability of the overall mapping process. In the single-sensor system, feature detection is often limited by the constrained FoV, leading to less robust tracking and a sparser map. In contrast, the multi-sensor setups, especially with four and five sensors, significantly enhance the system's capacity to observe a greater number of points across the environment, ensuring more reliable localization and mapping even during fast rotations or in environments with varying levels of visual detail.

This improvement in FoV is particularly advantageous in real-world scenarios, where environments are often dynamic and contain objects that can occlude certain views. By distributing the sensors in such a way that they capture complementary views, the system becomes less vulnerable to localized visual deficiencies, ensuring a more consistent and reliable performance.

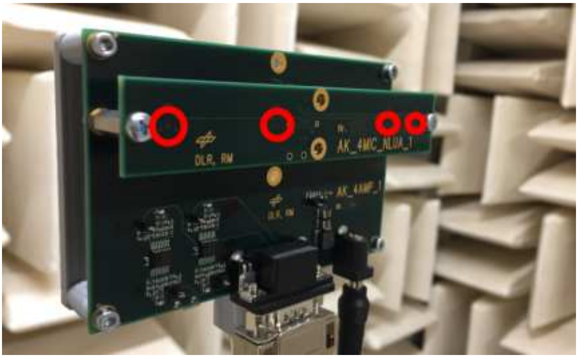


Figure 7.5.: Experimental audio setup. A linear microphone array with logarithmic spacing between the microphones (marked in red).

7.2. Indoor Robot Audition

As mentioned in Section 3.2 on page 39, the localization of audio sources in indoor environments is greatly affected by echo and reverberation. These environmental factors can distort the sound signals and challenge accurate source detection. To address these issues, the approach described in Section 6.1.2 (pp. 91) incorporates several countermeasures, including frequency pre-selection and the use of a motion model. These strategies help maintain reliable operation under reverberant conditions by focusing on dominant frequency bands and predicting source movement.

The system's performance has been evaluated in realistic indoor environments using a linear microphone array with four microphones, spaced logarithmically, as shown in Figure 7.5. This array configuration allows the system to capture sound waves effectively, even in complex acoustic environments.

The approach is compared with state-of-the-art implementations of the multiple signal classification (MUSIC) algorithm, including the Generalized Singular Value Decomposition (GSVD) method by Nakadai et al. [95] and the Active Frequency Range Filtering method by Hoshiba et al. [51]. A human speaker serves as the audio source, and the direction of arrival (DoA) of the sound is estimated by each approach. The methods are tested in six different locations with varying sizes and reverberation properties, as detailed in Ta-

Table 7.2.: Evaluation Data Set: Measured reverberation time T_{60} and room size for six different room types.

Room	T_{60} [s]	Area [m^2]
Lab (large)	1.158	291.3
Lab (small)	1.646	101.8
Entrance Hall	3.149	211.9
Common Room	1.971	80.3
Lecture Hall	1.077	142.0
Office	0.345	24.1

Table 7.3.: Experimental results. The first columns present the total number of estimated DoA for each room, the last ones the rate of successful estimations.

	n_{Total}			success rate		
Room	GSVD	AFRF	MME	GSVD	AFRF	MME
Lecture Hall	263	263	229	0.91	0.79	**0.95**
Common Room	77	77	69	0.82	0.78	**0.91**
Entrance	78	78	39	0.72	0.46	**0.95**
Office	98	98	57	0.55	0.46	**0.74**
Lab (large)	73	73	49	0.78	0.64	**0.82**
Lab (small)	52	52	24	0.58	0.48	**0.88**

ble 7.2. Room sizes range from 24 to 291 m^2, with reverberation times up to 3.15 s, providing diverse acoustic conditions for evaluation.

The experimental results, shown in Table 7.3, summarize the number of estimated DoA and the success rate for each method in these environments. A success is defined as an estimation with less than 5° deviation from the ground truth. While Motion-Model Enhanced MUSIC (MME) detects fewer sources overall, it achieves a higher success rate due to its active frequency selection and source tracking using the motion model. This enables it to filter out reverberation effects more effectively than GSVD and AFRF, which do not utilize a motion-based prediction model.

Two challenging scenarios are highlighted for a deeper evaluation. First, in the lecture hall, where reverberation is minimal given the size of the room, a speaker moves across three positions while talking. As shown in Figure 7.6, AFRF shows frequent miss-estimations, particularly when sound first arrives, as it tends to focus on reverberation-induced shadow sources. GSVD, which

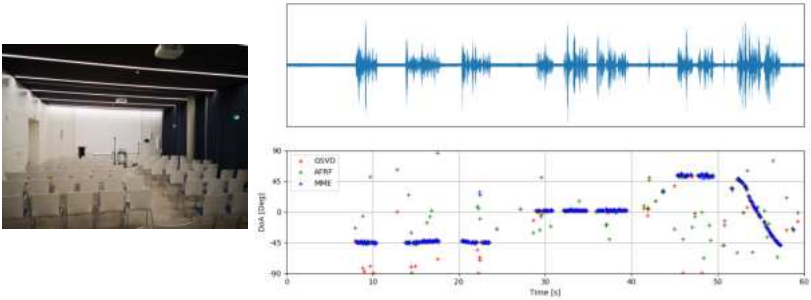


Figure 7.6.: Results of the direction of arrival estimations for the selected approaches in the lecture hall.

processes all frequency bins, performs better but still struggles with similar effects at certain points. In contrast, MME successfully mitigates this issue by filtering for dominant frequencies and applying a motion model to track the true sound source, thus preventing false detections from reflected sounds.

Second, in the entrance hall, the evaluation focuses on a scenario with high reverberation (3.149 s). Here, a speaker talks from two positions and moves across the room. As seen in Figure 7.7, GSVD and AFRF both struggle with this challenging acoustic environment, achieving success rates of only 72 % and 46 %, respectively. Many outliers in their estimations are caused by reverberation effects. However, MME filters out these outliers, yielding fewer total detections but a significantly higher success rate of 95 %. This demonstrates the effectiveness of combining frequency selection with a motion model in reducing the impact of reverberations.

A common challenge in audio-based localization is detecting a speaking person, especially when the speaker is not facing the vision system. While a vision-based system may fail under such conditions, an audio-based SSL system can detect sound sources without requiring line-of-sight. As demonstrated in Figure 7.8, the system successfully detects the speaker even when she is not facing the camera. This highlights the advantages of incorporating audio localization into the perception system for robust detection in dynamic environments.

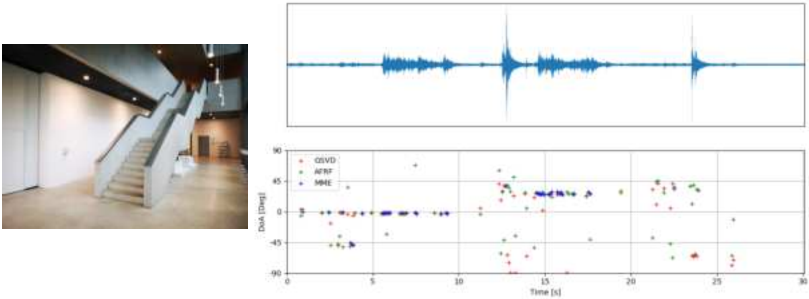


Figure 7.7.: Results of the direction of arrival estimations for the selected approaches in the entrance hall.

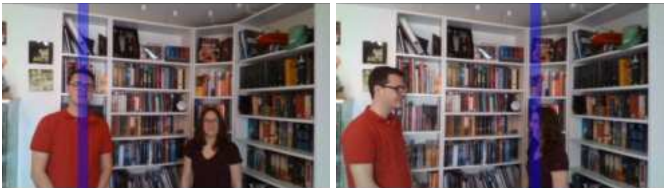


Figure 7.8.: Detection of speakers. Segmentation works even when the speakers do not face the system.

Only the real-time-capable methods (AFRF and MME) are included in this evaluation, as they need to detect speakers during speech. The results, shown in Figure 7.9, compare the true-positive (TP) and false-positive (FP) rates with a manually segmented ground truth. Visually, AFRF exhibits a frequent switching behavior between the left and right speakers, resulting in a TP rate of 79.5 % and an FP rate of 20.5 %. In contrast, MME demonstrates more stable performance, with sound events predominantly segmented to the correct speaker. This method achieves a higher TP rate of 93.1 % and a lower FP rate of 6.9 %.

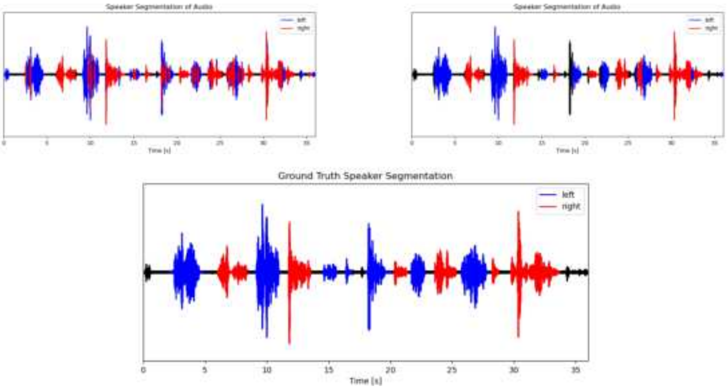


Figure 7.9.: Segmentation results for Active Frequency Range Filtering (left) and Motion-Model Enhanced MUSIC (right) for the first and second speaker. At the bottom, the manually labeled ground truth.

7.3. Audio-Visual Localization

As mentioned in Section 1.6 (pp. 9), combining audio and visual information enhances the system's understanding of the environment, providing multi-modal insight into the state and behavior of objects. This section demonstrates how integrating these two modalities allows the system to link sound sources to physical objects in its surroundings. The approach was previously described in Section 6.3 on page 98.

The experiment was conducted in the context of astronaut assistance, though the methodology is applicable to terrestrial scenarios as well. In space missions, an essential task is instrument maintenance as shown in Figure 7.10, which may be done proactively or after a failure occurs. Instrument failures often emit distinct sound patterns, which, if detected early, could prevent total system breakdown. This highlights the importance of audio perception as a complement to vision-based systems.

In this experimental setup, the vision system is enhanced with audio perception. The cameras detect and map objects in the environment (see Section 6.2 on page 96), while the robot's ego-state estimation system continuously tracks its position within the area (see Chapter 5 on page 69). Simultaneously, the audio system monitors the environment for sound events (see Section 6.1.2

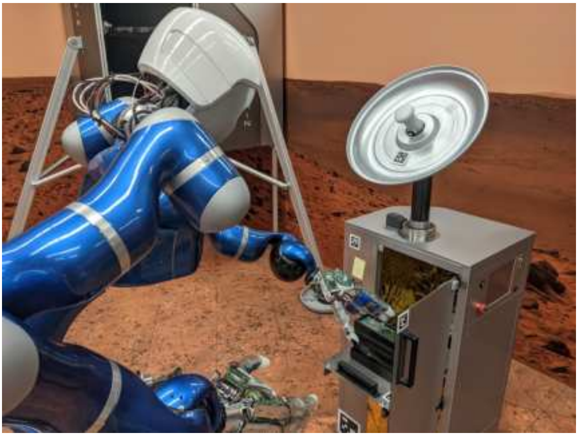


Figure 7.10.: Rollin' Justin maintaining a failed instrument box in a simulated Martian environment.

on page 91). When a sound is detected, the system estimates the DoA of the sound. Using the robot's current pose at the time of the sound event, ray-casting is applied to associate the sound source with one of the visually detected objects in the environment.

This process is illustrated in Figure 7.11, where the robot overlays the DoA with its own position and selects the object most likely responsible for the sound. The combination of audio and visual data enables the system to identify which object in the scene emitted the sound, providing more comprehensive situational awareness.

An additional experiment[1] was conducted to further utilize this audio-visual fusion for system diagnostics. Multiple distinct operational states of the system were pre-recorded and stored in the knowledge base, each associated with specific audio spectral profiles. By comparing the received sound spectrum with these known profiles, the system can estimate the current state of the instrument.

The received audio is compared in the frequency domain with pre-obtained spectral profiles. For each profile, an audio sample with a duration of at least 5s

---

[1] Publication 6

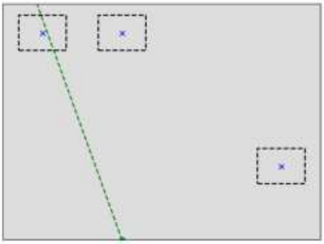


Figure 7.11.: Results of the fusion of visual and audio information. Object poses and robot ego-pose have been estimated using the vision system. DoA estimation of a received sound event is overlaid and used for source localization.

is recorded. These audio samples are transformed using a Short-Time Fourier Transform (SFTF) with small overlapping subframes and a hop-parameter of 32 samples. To reduce noise and capture the variability of the event, the median spectrum P_{50} is calculated across all received spectra, providing a representative frequency profile for each system state.

The highest value of the median spectrum is used to normalize the spectrum and constrain it to the range $[0, 1]$. To define an acceptance band for classification, the 20th percentile P_{20} and the 80th percentile P_{80} for each frequency bin are taken as the lower and upper bounds, respectively. When receiving a new, unclassified spectrum, the background noise components are subtracted from the input signal. The system then estimates and normalizes the median spectrum of the incoming signal.

The classification process involves calculating the sum of squared differences between the received spectrum and the stored profile spectra within the acceptance band of each frequency bin k:

$$
\begin{aligned}
s &= \sum_{k \in K} \frac{1}{s_k}, \\
s_k &= \begin{cases} 0 & X_k < P_{20,k} \\ \left(X_k - P_{50,k}\right)^2 & P_{20,k} \leq X_k \leq P_{80,k} \\ 0 & X_k > P_{80,k} \end{cases}
\end{aligned}
\tag{7.1}
$$

The score s quantifies the similarity of two frequency spectra within the acceptance band, with a higher score indicating a closer match to a known profile. If less than 10 % of the individual frequency scores exceeds the threshold, the system assumes the spectrum originates from an unknown or unrecorded state. This approach enables the robot to classify and identify the current operational state of the instrument based on its acoustic signature, preventing false classifications when unknown spectral profiles are encountered.

This approach is shown in Figure 7.12, depicting a known and an unknown sound event. In the evaluation, the system's ability to estimate the state of the instrument using both known and unknown audio profiles is demonstrated. Using the method based on the median profile and the scoring system described by Equation (7.1), the system successfully distinguishes between known profiles and identifies when a sound event does not match any of the previously recorded states. This allows for accurate classification of known operational states while flagging unrecognized events as potentially new or anomalous states.

In contrast, a naive approach based solely on the sum of squared differences (SSD) without incorporating the median-based acceptance band fails to identify unknown profiles. This method tends to force a match with the closest profile in the knowledge base, even when the sound event does not correspond to any known state. As a result, the naive SSD approach leads to false classifications in scenarios where the sound does not match any of the pre-recorded profiles, highlighting the limitations of such an approach. This comparison emphasizes the importance of using the median-based scoring system for distinguishing both known and unknown audio profiles.

Upon detecting a sound, the system retrieves the spectral profile from its database and matches it against the incoming sound signal. This comparison allows the robot to distinguish between normal operation and a potential

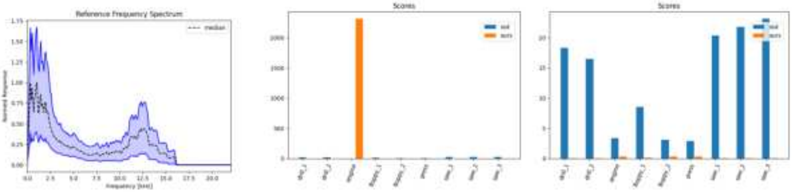


Figure 7.12.: Spectral analysis of a received audio event. On the left side, the reference audio profile measured a-priori for a given internal state. In the middle, a comparison of a sound event that corresponds to a known audio profile using the sum of squared differences (SSD) and an approach based on the median spectral profile. On the right, a comparison with a sound event that is not known.

failure condition. In the failure case, specific deviations in the audio spectrum trigger a warning, enabling the robot to flag the instrument for maintenance before a complete system breakdown occurs.

8. Conclusion

The conclusion of this work summarizes the key contributions made toward advancing multi-modal and multi-sensor fusion for robotic perception in indoor environments. The developed framework addresses critical aspects such as visual and auditory data fusion, landmark localization, and real-time navigation, showcasing robustness in handling the complexities of indoor spaces. The proposed methods have demonstrated their potential in improving the accuracy and reliability of robotic assistance systems operating in dynamic and cluttered environments.

Future work may focus on refining sensor models to enhance performance in confined spaces and exploring the adaptability of the system to broader application domains. Additionally, addressing the challenges of scaling and improving real-time processing remains a promising area for further research.

8.1. Summary

This work focused on developing a robust perception system for robots operating in indoor environments. These environments, such as homes or elderly care facilities, pose significant challenges due to confined spaces, clutter, and variable lighting conditions. To address these issues, the system integrated visual odometry from multiple sensor inputs, enabling accurate localization and ego-motion estimation even in visually ambiguous areas. The architecture allowed the robot to navigate and interact with its surroundings efficiently, reducing tracking failures commonly encountered in indoor spaces with textureless or repetitive surfaces. This was validated by testing in realistic apartment settings, where the robot maintained consistent performance without requiring environmental modifications.

A key part of the system is the audio-visual perception framework. While visual data is vital for understanding spatial structure and detecting objects, it can be unreliable in low-light conditions or when visual cues are obscured. To mitigate these limitations, audio data was incorporated to complement visual inputs. For example, in cases where a speaking person was occluded from the camera's view, sound localization provided additional context, improving the overall perception. The real-time integration of audio with visual data enabled robust tracking in dynamic environments, such as locating a speaker in cluttered indoor spaces, thereby enhancing situational awareness and interaction capabilities.

The system also implemented a multi-sensor framework to ensure robustness and reliability. Multiple cameras were used to provide redundancy, which helped in scenarios where one sensor might fail or encounter occlusions. For example, in environments where data for a single camera was temporarily unavailable due to an obstacle, the remaining sensors continued to supply data, allowing the system to recover from loss-of-tracking (LoT) events. This multi-sensor approach increased resilience, reducing the frequency of tracking failures and ensuring continuous operation in real-time. The approach was tested extensively using the IndoorMCD [118] dataset, where it showed superior performance compared to single-sensor setups, particularly in cluttered or low-texture environments.

Robust operation in proximity to obstacles was ensured by dynamically excluding sensors that were occluded or failed to provide accurate visual odometry (VO) estimates. In scenarios where the robot operated near obstacles, such as furniture or walls, sensors that lost sight of the environment were temporarily excluded from the pose estimation process. The multi-sensor framework took over, relying on the remaining sensors for accurate ego-motion estimation. When the occluded sensors regained visibility, their trajectory estimates were re-registered and seamlessly reintegrated into the overall pose estimation. This approach allowed the robot to maintain robust operation without sacrificing accuracy, even in highly cluttered or confined spaces.

Additionally, this work provides a simulation environment, URSim [120], which supports the development of multi-sensor systems by allowing for comprehensive testing in simulated indoor settings. URSim offers a realistic platform for evaluating system performance across various sensor configurations, without requiring physical deployments. Furthermore, the IndoorMCD [118] dataset, a first of its kind, enables rigorous evaluation of multi-sensor frameworks in complex indoor environments. This dataset was designed specifically to simulate real-world scenarios, such as cluttered apartments or offices, offering unique benchmarks for testing how multi-sensor systems perform under conditions involving dynamic obstacles, occlusions, and varying lighting. These tools significantly contribute to system development and validation, offering valuable resources for future research in this domain.

In summary, this work introduces several key technical contributions to the field of multi-sensor perception in robotics. First, the development of a robust multi-sensor visual odometry framework significantly improved the reliability of ego-motion estimation in cluttered and dynamic indoor environments. The integration of audio-visual perception added a novel layer of redundancy, enhancing performance in scenarios involving multiple modalities or hidden events. The real-time fusion of audio and visual data provided a more holistic understanding of the environment, particularly in human-centered spaces where interaction is crucial. Furthermore, the implementation of a dynamic sensor exclusion and reintegration mechanism ensured continuous operation near obstacles, an important advancement in maintaining pose estimation when sensors are occluded. Lastly, the creation of URSim and the IndoorMCD dataset provides valuable tools for simulating and evaluating multi-sensor systems, offering first-of-its-kind resources for testing in complex indoor scenarios. Together, these contributions advance the state-of-the-art in perception systems for indoor robotics, providing a robust foundation for future research and development.

8.2. Outlook

While the proposed system has shown significant improvements in multi-sensor perception and ego-motion estimation, there remain limitations that require further research. One key area for improvement is the ego-state estimation process, which currently relies solely on visual information for localization. While visual odometry has proven effective in many scenarios, it does not fully utilize other sensors that are commonly available in robotic systems. Future research should explore incorporating inertial data, typically available via inertial measurement units (IMUs) on most platforms.

The integration of IMU data could improve motion estimation, particularly during fast movements or when visual features are insufficient. Using other spectral information form the light's spectrum may further enhance the robot's perception. By incorporating data from a broader range of sensors, the system could achieve greater robustness and reliability in a wider range of operational environments.

Another key area for improvement is the mapping process, which currently assumes a static world during ego-state estimation. This assumption simplifies the process but becomes a limitation in dynamic indoor environments where objects can frequently change pose or appearance. The current system can handle dynamic objects for landmark estimation, but the fusion of dynamic object detection with the mapping framework has not yet been fully realized. Future research should focus on developing a dynamic mapping approach that not only updates the map when changes in the environment occur but also removes outdated parts of the map when objects move or change. This dynamic map updating would significantly improve system performance in real-world indoor environments, where such changes are common. A solution must be found that can effectively integrate dynamic objects into the ego-state estimation and mapping process, allowing the system to continuously update its understanding of the environment without requiring a full remapping. This would make the system more adaptable to environments like homes and offices, where both furniture and people are in constant motion.

Additionally, the current system requires manual intervention to configure minimal maps, ensuring that only the most essential features are included. This dependency on expert knowledge limits the usability of the system in broader applications. Future research should investigate how the system can autonomously generate these minimal maps online, without needing manual configuration. An automated approach would allow the robot to identify and select the most relevant features during mapping, enabling more scalable and accessible deployment across a wider range of environments. By reducing the need for expert input, this enhancement could significantly increase the system's versatility in real-world applications.

Another area that requires further research is the audio system, particularly in how it handles echoes during sound event localization. Currently, the system assumes that the received direction-of-arrival (DoA) of a sound directly points to its source. However, in environments where the sound source is occluded by an obstacle, the system may instead receive an echo that has reflected off a surface. This echo creates the illusion that the sound source is located at the point of reflection rather than at the true origin. To address this, future work should focus on incorporating echo detection and compensation mechanisms into the audio-visual fusion system. By understanding and modeling how sound reflects off surfaces, the system could more accurately attribute sound events to their correct sources, even when these sources are hidden. This would improve the reliability of the audio-visual perception system, particularly in complex indoor environments with many reflective surfaces.

Finally, the fusion of audio and visual data presents opportunities for further improvement. The system currently integrates these modalities for specific tasks, but future work could focus on tighter integration, particularly when handling dynamic environments. By further refining the fusion process, it may become possible to enhance object detection and tracking even when either modality alone would fail due to occlusion or environmental complexity. For example, developing algorithms capable of distinguishing between direct and reflected sound waves could enable more accurate association of sound events with objects in the environment. These enhancements would allow the system to better navigate environments with complex acoustics and ensure that sound source localization remains accurate even in cases of occlusion.

Bibliography

[1] Sharath Adavanne, Archontis Politis, and Tuomas Virtanen. “Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network”. In: *2018 26th European Signal Processing Conference (EUSIPCO).* IEEE. 2018, pp. 1462–1466.

[2] Triantafyllos Afouras et al. “Deep Audio-Visual Speech Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.12 (2022), pp. 8717–8727.

[3] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. *Ceres Solver.* Version 2.2. Oct. 2023. URL: `https://github.com/ceres-solver/ceres-solver`.

[4] Alin Albu-Schäffer et al. “The DLR lightweight robot: design and control concepts for robots in human environments”. In: *Industrial Robot: an international journal* 34.5 (2007), pp. 376–385.

[5] Sylvain Argentieri and Patrick Danes. “Broadband variations of the MUSIC high-resolution method for sound source localization in robotics”. In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE. 2007, pp. 2009–2014.

[6] F Asono, Hideki Asoh, and Toshihiro Matsui. “Sound source localization and signal separation for office robot" Jijo-2"”. In: *Proceedings. 1999 IEEE/SICE/RSJ. International Conference on Multisensor Fusion and Integration for Intelligent Systems.* IEEE. 1999, pp. 243–248.

[7] Christos Astaras et al. “Passive acoustic monitoring as a law enforcement tool for Afrotropical rainforests”. In: *Frontiers in Ecology and the Environment* 15.5 (2017).

[8] Oleksandr Bailo et al. “Efficient adaptive non-maximal suppression algorithms for homogeneous spatial keypoint distribution”. In: *Pattern Recognition Letters* 106 (2018), pp. 53–60.

[9] Barfoot et al. “Associating Uncertainty With Three-Dimensional Poses for Use in Estimation Problems”. In: (2014).

[10] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. “Speeded-Up Robust Features (SURF)”. In: *Computer Vision and Image Understanding* 110.3 (2008), pp. 346–359.

[11] Søren Bech and Nick Zacharov. *Perceptual Audio Evaluation—Theory, Method and Application.* John Wiley & Sons, 2006.

[12] Richard E. Berg and David G. Stork. *The Physics of Sound.* Englewood Cliffs, N.J.: Prentice Hall, 1982. ISBN: 9780136698328.

[13] Joydeep Biswas and Manuela Veloso. “Multi-sensor Mobile Robot Localization for Diverse Environments”. In: *IEEE Transactions on Robotics* 32.2 (2016), pp. 208–228.

[14] Åke Björck. *Numerical Methods for Least Squares Problems.* SIAM, 1996, pp. 105–114. ISBN: 9780898713602.

[15] Dennis A Bohn. “Environmental effects on the speed of sound”. In: *Audio Engineering Society Convention 83.* Audio Engineering Society. 1987.

[16] Christoph Borst et al. “Rollin’justin-mobile platform with variable base”. In: *2009 IEEE International Conference on Robotics and Automation.* IEEE. 2009, pp. 1597–1598.

[17] Browning et al. “Übersim: a multi-robot simulator for robot soccer”. In: *Proceedings of the second international joint conference on Autonomous agents and multiagent systems.* 2003, pp. 948–949.

[18] Jeffrey Bynum and David Lattanzi. “Combining convolutional neural networks with unsupervised learning for acoustic monitoring of robotic manufacturing facilities”. In: *Advances in Mechanical Engineering* 13.4 (2021), p. 16878140211009015.

[19] Cesar Cadena et al. “Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age”. In: *IEEE Transactions on Robotics* 32.6 (2016), pp. 1309–1332.

[20] Michael Calonder et al. “BRIEF: Binary Robust Independent Elementary Features”. In: *Proceedings of the European Conference on Computer Vision (ECCV).* 2010, pp. 778–792.

[21] Calvert. *Developing problem-solving skills in engineering.* 1953.

[22] Campos et al. “ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM”. In: *IEEE Transactions on Robotics* (2021).

[23] Carlsson et al. “DIVE—A platform for multi-user virtual environments”. In: *Computers & graphics* 17.6 (1993), pp. 663–669.

[24] Soo-Whan Chung et al. “FaceFilter: Audio-visual speech separation using still images”. In: *arXiv preprint arXiv:2005.07074*. 2020.

[25] Coelho et al. “OSGAR: A scene graph with uncertain transformations”. In: *IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE. 2004.

[26] James W. Cooley and John W. Tukey. “An Algorithm for the Machine Calculation of Complex Fourier Series”. In: *Mathematics of Computation* 19.90 (1965), pp. 297–301. DOI: `10.1090/S0025-5718-1965-0178586-1`.

[27] Toyota Motor Corporation. *Toyota Human Support Robot (HSR)*. 2019. URL: `https://www.toyota-global.com/innovation/robotics/hsr/`.

[28] Timothy A Davis. *Direct methods for sparse linear systems*. SIAM, 2006.

[29] Frank Dellaert. “Factor graphs and GTSAM: A hands-on introduction”. In: *Georgia Institute of Technology, Tech. Rep* 2 (2012), p. 4.

[30] Denavit and Hartenberg. “A kinematic notation for lower-pair mechanisms based on matrices”. In: (1955).

[31] George Devol. *Unimate Industrial Robot*. Danbury, CT: Unimation Inc., 1961.

[32] Andrew Digby et al. “A practical comparison of manual and autonomous methods for acoustic monitoring”. In: *Methods in Ecology and Evolution* 4.7 (2013), pp. 675–683.

[33] Drumwright et al. “Extending open dynamics engine for robotics simulation”. In: *Simulation, Modeling, and Programming for Autonomous Robots: Second International Conference*. Springer. 2010.

[34] Boston Dynamics. *Meet Spot, the Agile Mobile Robot*. 2020. URL: `https://www.bostondynamics.com/spot`.

[35] InGen Dynamics. *Aido: The Next Generation Social Family Robot*. 2018. URL: `https://www.aidorobot.com/`.

[36] Bengt Edlén. “The Refractive Index of Air”. In: *Metrologia* 2.2 (1966), p. 71. DOI: `10.1088/0026-1394/2/2/002`.

[37] Electrolux. *Trilobite: The First Robotic Vacuum Cleaner*. 1997. URL: `https://www.electroluxgroup.com/en/trilobite-the-first-robotic-vacuum-cleaner-3256/`.

[38] Ikenna et al. Enebuse. “Accuracy evaluation of hand-eye calibration techniques for vision-guided robots”. In: *PLOS ONE* (2022).

[39] Jakob Engel, Thomas Schöps, and Daniel Cremers. “LSD-SLAM: Large-scale direct monocular SLAM”. In: *European conference on computer vision*. Springer. 2014, pp. 834–849.

[40] Cliff Fitzgerald. “Developing baxter”. In: *2013 IEEE conference on technologies for practical robot applications (TePRA)*. IEEE. 2013, pp. 1–6.

[41] Foote. “tf: The transform library”. In: *IEEE Conference on Technologies for Practical Robot Applications (TePRA)*. IEEE. 2013.

[42] Christian Forster et al. “On-manifold preintegration for real-time visual–inertial odometry”. In: *IEEE Transactions on Robotics* 33.1 (2016), pp. 1–21.

[43] Jean Baptiste Joseph Fourier. *Théorie analytique de la chaleur.* Original work where the Fourier Transform was introduced. Paris: Chez Firmin Didot, père et fils, 1822. URL: `https://archive.org/details/bub_gb_ZBME0k9lK4oC`.

[44] Aviv Gabbay, Asaph Shamir, and Shmuel Peleg. “Visual Speech Enhancement”. In: *ICASSP*. IEEE. 2018.

[45] Annette Hagengruber and Lioba Suchenwirth. “SMiLE2gether: A prototype of a holistic ecosystem for robotic care assistants”. In: *Open-Access-Publikation im Sinne der CC-Lizenz BY* 4 (2022), p. 227.

[46] Chris Harris and Mike Stephens. “A Combined Corner and Edge Detector”. In: *Proceedings of the Alvey Vision Conference* (1988), pp. 147–151.

[47] Fredric J. Harris. “On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform”. In: *Proceedings of the IEEE* 66.1 (1978), pp. 51–83. DOI: `10.1109/PROC.1978.10837`.

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

[49] Nick Hawes et al. “The STRANDS Project: Long-Term Autonomy in Everyday Environments”. In: *IEEE Robotics and Automation Magazine* 24.3 (2017), pp. 146–156.

[50] Heiko Hirschmuller. “Stereo Processing by Semiglobal Matching and Mutual Information”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2 (2008), pp. 328–341. DOI: `10.1109/TPAMI.2007.1166`.

[51] Kotaro Hoshiba et al. “Assessment of MUSIC-based noise-robust sound source localization with active frequency range filtering”. In: *Journal of Robotics and Mechatronics* 30.3 (2018), pp. 426–435.

[52] Md Shahjahan Hossain and Hossein Taheri. “In situ process monitoring for additive manufacturing through acoustic techniques”. In: *Journal of Materials Engineering and Performance* 29.10 (2020), pp. 6249–6262.

[53] Jie Huang, Noboru Ohnishi, and Noboru Sugie. “Building ears for robots: sound localization and separation”. In: *Artificial Life and Robotics* 1.4 (1997), pp. 157–163.

[54] Jibo Inc. *Meet Jibo, The World’s First Social Robot for the Home.* 2017. URL: `https://www.jibo.com/`.

[55] iRobot. *PackBot: Proven Performance.* 2013. URL: `https://www.irobot.com/robots/packbot`.

[56] Carlos T Ishi et al. “Evaluation of a MUSIC-based real-time sound localization of multiple sound sources in real noisy environments”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE. 2009, pp. 2027–2032.

[57] Lloyd A Jeffress. “A place theory of sound localization.” In: *Journal of comparative and physiological psychology* 41.1 (1948), p. 35.

[58] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. “iSAM: Incremental smoothing and mapping”. In: *IEEE Transactions on Robotics* 24.6 (2008), pp. 1365–1378.

[59] Eric P Kasten et al. “The remote environmental assessment laboratory’s acoustic library: An archive for studying soundscape ecology”. In: *Ecological informatics* 12 (2012), pp. 50–67.

[60] Kennedy. *The Kinematics of Machinery.* New York: D. Van Nostrand, 1881.

[61] Christian Kerl, Jürgen Sturm, and Daniel Cremers. “Dense visual SLAM for RGB-D cameras”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE. 2013, pp. 2100–2106.

[62] Fakheredine Keyrouz, Youssef Naous, and Klaus Diepold. “A new method for binaural 3-D localization based on HRTFs”. In: *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. Vol. 5. IEEE. 2006, pp. V–V.

[63] Georg Klein and David Murray. “Parallel tracking and mapping for small AR workspaces”. In: *2007 6th IEEE and ACM international symposium on mixed and augmented reality*. IEEE. 2007, pp. 225–234.

[64] Lucas W Koester et al. “Acoustic monitoring of additive manufacturing for damage and process condition determination”. In: *AIP Conference Proceedings*. Vol. 2102. 1. AIP Publishing LLC. 2019, p. 020005.

[65] Ryosuke Kojima et al. “Semi-automatic bird song analysis by spatial-cue-based integration of sound source detection, localization, separation, and identification”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2016, pp. 1287–1292.

[66] Ingo Kossyk, Michael Neumann, and Zoltan-Csaba Marton. “Binaural bearing only tracking of stationary sound sources in reverberant environment”. In: *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE. 2015, pp. 53–60.

[67] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. “EP n P: An accurate O (n) solution to the P n P problem”. In: *International journal of computer vision* 81 (2009), pp. 155–166.

[68] Sheng Li et al. “Iterative spectral subtraction method for millimeter-wave conducted speech enhancement”. In: *Journal of Biomedical Science and Engineering* 3.02 (2010), p. 187.

[69] Ruth Y Litovsky et al. “The precedence effect”. In: *The Journal of the Acoustical Society of America* 106.4 (1999), pp. 1633–1654.

[70] Diego Llusia, Rafael Márquez, and Richard Bowker. “Terrestrial sound monitoring systems, a methodology for quantitative calibration”. In: *Bioacoustics* 20.3 (2011), pp. 277–286.

[71] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[72] Jiaoyang Lu, Ting Zou, and Xianta Jiang. “A Neural Network Based Approach to Inverse Kinematics Problem for General Six-Axis Robots”. In: *Sensors* 22 (2022).

[73] Bruce D Lucas and Takeo Kanade. “An iterative image registration technique with an application to stereo vision”. In: *IJCAI’81: 7th international joint conference on Artificial intelligence.* Vol. 2. 1981.

[74] Tobias Löw and Sylvain Calinon. “Geometric Algebra for Optimal Control With Applications in Manipulation Tasks”. In: *IEEE Transactions on Robotics* 39.5 (2023). DOI: `10.1109/TRO.2023.3277282`.

[75] Justin A MacDonald. “A localization algorithm based on head-related transfer functions”. In: *The Journal of the Acoustical Society of America* 123.6 (2008), pp. 4290–4296.

[76] Elmar Mair et al. “Adaptive and generic corner detection based on the accelerated segment test”. In: *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II 11.* Springer. 2010.

[77] Sagnik Majumder, Ziad Al-Halah, and Kristen Grauman. “Move2Hear: Active Audio-Visual Source Separation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* IEEE. 2022.

[78] Sagnik Majumder and Kristen Grauman. “Active Audio-Visual Separation of Dynamic Sound Sources”. In: *arXiv preprint arXiv:2202.00850.* 2022.

[79] Aryslan Malik et al. “A Deep Reinforcement-Learning Approach for Inverse Kinematics Solution of a High Degree of Freedom Robotic Manipulator”. In: *Robotics* 11 (2022).

[80] Larry et al. Matthies. “The Sojourner Rover: Autonomous Navigation for the Mars Pathfinder Mission”. In: *IEEE Expert* 12.2 (1997).

[81] Iain A McCowan. “Robust speech recognition using microphone arrays”. PhD thesis. Queensland University of Technology, 2001.

[82] Meng et al. “A tightly coupled monocular visual lidar odometry with loop closure”. In: *Intelligent Service Robotics* (2022).

[83] Meyer et al. “Robust Probabilistic Robot Arm Keypoint Detection Exploiting Kinematic Knowledge”. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop on Probabilistic Robotics in the Age of Deep Learning.* 2022.

[84] Meyer et al. “The Probabilistic Robot Kinematics Model and its Application to Sensor Fusion”. In: 2022.

[85] Ondrej Miksik and Krystian Mikolajczyk. “Evaluation of local detectors and descriptors for fast feature matching”. In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE. 2012, pp. 2681–2684.

[86] Breton Minnehan, Henry Spang, and Andreas E Savakis. “Robust and efficient tracker using dictionary of binary descriptors and locality constraints”. In: *International Symposium on Visual Computing*. Springer. 2014, pp. 589–598.

[87] Hans Moravec. “The Stanford Cart and the CMU Rover”. In: *Proceedings of the IEEE* 71.7 (1983), pp. 872–884.

[88] Jorge J. Moré. “The Levenberg-Marquardt algorithm: Implementation and theory”. In: *Numerical Analysis* (1978), pp. 105–116.

[89] Marius Muja and David G Lowe. “Fast approximate nearest neighbors with automatic algorithm configuration.” In: *VISAPP (1)* 2.331-340 (2009), p. 2.

[90] Enzo Mumolo, Massimiliano Nolich, and Gianni Vercelli. “Algorithms for acoustic localization based on microphone array in service robotics”. In: *Robotics and Autonomous systems* 42.2 (2003), pp. 69–88.

[91] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. “ORB-SLAM: a versatile and accurate monocular SLAM system”. In: *IEEE transactions on robotics* 31.5 (2015), pp. 1147–1163.

[92] Mur-Artal et al. “Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras”. In: *IEEE transactions on robotics* (2017).

[93] Müller et al. “Robust visual-inertial state estimation with multiple odometries and efficient mapping on an MAV with ultra-wide FOV stereo vision”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2018.

[94] Kazuhiro Nakadai et al. “Active Audition for Humanoid”. In: *Proceedings of the AAAI Conference*. American Association for Artificial Intelligence. 2000.

[95] Kazuhiro Nakadai et al. “Applying scattering theory to robot audition system: Robust sound source localization and extraction”. In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*. Vol. 2. IEEE. 2003, pp. 1147–1152.

[96] Kazuhiro Nakadai et al. “Real-time auditory and visual multiple-object tracking for humanoids”. In: *International Joint Conference on Artificial Intelligence*. Vol. 17. 1. LAWRENCE ERLBAUM ASSOCIATES LTD. 2001, pp. 1425–1436.

[97] Keisuke Nakamura et al. “Intelligent Sound Source Localization for Dynamic Environments”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2009, pp. 664–669.

[98] Thanh-Toan et al. Nguyen. “Covariance Analysis for Hand-Eye Calibration”. In: *Journal of Robotics* 35.4 (2018), pp. 451–459.

[99] Nils Nilsson. *Shakey the Robot*. Menlo Park, CA: SRI International, 1984.

[100] Ch. Ott et al. “A Humanoid Two-Arm System for Dexterous Manipulation”. In: *2006 6th IEEE-RAS International Conference on Humanoid Robots*. 2006, pp. 276–283. doi: `10.1109/ICHR.2006.321397`.

[101] Andrew Owens and Alexei A. Efros. “Audio-Visual Scene Analysis with Self-Supervised Multisensory Features”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2018, pp. 631–648.

[102] Panos Photinos. *The Physics of Sound Waves: Music, Instruments, and Sound Equipment*. Bristol: Institute of Physics Publishing, 2021. isbn: 9780750335379.

[103] Allan D Pierce and An Acoustics. “Introduction to its physical principles and applications”. In: *Acoustical Society of America and American Institute of Physics* (1981), p. 122.

[104] Javier Ramirez et al. “Voice activity detection with noise reduction and long-term spectral divergence estimation”. In: *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 2. IEEE. 2004, pp. ii–1093.

[105] Richard. *A Mathematical Introduction to Robotic Manipulation*. 1994.

[106] Rethink Robotics. *Meet Baxter, The Affordable Robot with Common Sense*. 2012. url: `https://www.rethinkrobotics.com/baxter`.

[107] Reinhild Roden et al. *On sound source localization of speech signals using deep neural networks*. 2015.

[108] Edward Rosten and Tom Drummond. “Machine Learning for High-Speed Corner Detection”. In: *European Conference on Computer Vision*. 2006, pp. 430–443.

[109] Richard Roy and Thomas Kailath. "ESPRIT-estimation of signal parameters via rotational invariance techniques". In: *IEEE Transactions on acoustics, speech, and signal processing* 37.7 (1989), pp. 984–995.

[110] Ethan Rublee et al. "ORB: An Efficient Alternative to SIFT or SURF". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2011, pp. 2564–2571.

[111] Thomas Ruehr. *uncertain tf*. `https://github.com/ruehr/uncertain_tf`. Last accessed 2023-11-30. 2013.

[112] Emilio et al. Ruiz. "Methods for Simultaneous Robot-World-Hand-Eye Calibration: A Comparative Study". In: *Sensors* 19.12 (2019), p. 2837.

[113] Ryo Sakagami et al. "Robotic world models—conceptualization, review, and engineering best practices". In: *Frontiers in Robotics and AI* 10 (2023), p. 1253049.

[114] Ralph Schmidt. "Multiple emitter location and signal parameter estimation". In: *IEEE transactions on antennas and propagation* 34.3 (1986), pp. 276–280.

[115] Rolf Dieter Schraft and Rainer Volz. "The Service Robot Concept Helpmate". In: *Advanced Robotics* 12.4 (1998), pp. 319–329.

[116] Marco Sewtz, Tim Bodenmüller, and Rudolph Triebel. "Design of a Microphone Array for Rollin Justin". In: *ICRA Workshop*. 2019.

[117] Marco Sewtz, Tim Bodenmüller, and Rudolph Triebel. "Robust MUSIC-Based Sound Source Localization in Reverberant and Echoic Environments". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 2474–2480.

[118] Marco Sewtz et al. "IndoorMCD: A Benchmark for Low-Cost Multi-Camera SLAM in Indoor Environments". In: *IEEE Robotics and Automation Letters* 8.3 (2023), pp. 1707–1714.

[119] Marco Sewtz et al. "Robust approaches for localization on multi-camera systems in dynamic environments". In: *2021 7th International Conference on Automation, Robotics and Applications (ICARA)*. IEEE. 2021, pp. 211–215.

[120] Marco Sewtz et al. "Ursim-a versatile robot simulator for extra-terrestrial exploration". In: *2022 IEEE Aerospace Conference (AERO)*. IEEE. 2022, pp. 1–14.

[121] Jianbo Shi and Carlo Tomasi. “Good Features to Track”. In: *1994 IEEE Conference on Computer Vision and Pattern Recognition.* 1994, pp. 593–600.

[122] Ken Shoemake. “Animating rotation with quaternion curves”. In: *Proceedings of the 12th annual conference on Computer graphics and interactive techniques.* 1985.

[123] Bruno Siciliano and Oussama Khatib. “Design of Robotic Joints”. In: *Springer Handbook of Robotics.* Springer, 2016, pp. 150–175.

[124] Bruno Siciliano and Oussama Khatib. *Springer Handbook of Robotics.* Springer, 2016.

[125] Sola et al. “A micro Lie theory for state estimation in robotics”. In: *CoRR* (2018).

[126] Frank Steinbrücker, Jürgen Sturm, and Daniel Cremers. “Real-time visual odometry from dense RGB-D images”. In: *2011 IEEE international conference on computer vision workshops (ICCV Workshops).* IEEE. 2011, pp. 719–722.

[127] Manuel Stoiber, Martin Sundermeyer, and Rudolph Triebel. “Iterative corresponding geometry: Fusing region and depth for highly efficient 3d tracking of textureless objects”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2022, pp. 6855–6865.

[128] Stoiber et al. “A sparse gaussian approach to region-based 6DoF object tracking”. In: *Proceedings of the Asian Conference on Computer Vision.* 2020.

[129] Klaus H. Strobl and Gerd Hirzinger. “Optimal Hand-Eye Calibration”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE. 2006, pp. 4647–4653.

[130] Su et al. “Manipulation and propagation of uncertainty and verification of applicability of actions in assembly tasks”. In: *IEEE Transactions on Systems, Man, and Cybernetics* (1992).

[131] Larissa Sayuri Moreira Sugai et al. “A roadmap for survey designs in terrestrial acoustic monitoring”. In: *Remote Sensing in Ecology and Conservation* 6.3 (2020), pp. 220–235.

[132] Larissa Sayuri Moreira Sugai et al. “Terrestrial passive acoustic monitoring: review and perspectives”. In: *BioScience* 69.1 (2019), pp. 15–25.

[133] Reiji Suzuki et al. “HARKBird: Exploring acoustic interactions in bird communities using a microphone array”. In: *Journal of Robotics and Mechatronics* 29.1 (2017), pp. 213–223.

[134] Ryu Takeda and Kazunori Komatani. “Discriminative multiple sound source localization based on deep neural networks using independent location model”. In: *2016 IEEE Spoken Language Technology Workshop (SLT).* IEEE. 2016, pp. 603–609.

[135] Yuehua Tao et al. “Research Progress of the Scale Invariant Feature Transform (SIFT) Descriptors.” In: *Journal of Convergence Information Technology* 5.1 (2010), pp. 116–121.

[136] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. “A probabilistic approach to concurrent mapping and localization for mobile robots”. In: *Autonomous Robots* 5 (1998), pp. 253–271.

[137] Tong et al. “Vins-mono: A robust and versatile monocular visual-inertial state estimator”. In: *IEEE Transactions on Robotics* (2018).

[138] Tramberend. “Avocado: A distributed virtual reality framework”. In: *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316).* IEEE. 1999, pp. 14–21.

[139] Juan Sebastian Ulloa et al. “Estimating animal acoustic diversity in tropical environments using unsupervised multiresolution analysis”. In: *Ecological Indicators* 90 (2018), pp. 346–355.

[140] J-M Valin et al. “Localization of simultaneous moving sound sources for mobile robot using a frequency-domain steered beamformer approach”. In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA’04. 2004.* Vol. 1. IEEE. 2004, pp. 1033–1038.

[141] J-M Valin et al. “Robust sound source localization using a microphone array on a mobile robot”. In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453).* Vol. 2. IEEE. 2003, pp. 1228–1233.

[142] Raquel Viciana-Abad et al. “Audio-Visual Perception System for a Humanoid Robotic Head”. In: *Sensors* 14.6 (2014), pp. 9522–9545.

[143] Kenneth J. Waldron and James P. Schmiedeler. “Kinematics”. In: *Springer Handbook of Robotics.* Springer, 2016, pp. 9–34.

[144] John Wang and Edwin Olson. “AprilTag 2: Efficient and robust fiducial detection”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE. 2016, pp. 4193–4198.

[145] John Wang and Edwin Olson. "AprilTag 2: Efficient and robust fiducial detection". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* 2016.

[146] Thomas Whelan et al. "ElasticFusion: Real-time dense SLAM and light source estimation". In: *The International Journal of Robotics Research* 35.14 (2016), pp. 1697–1716.

[147] Xiong Xiao et al. "A learning-based approach to direction of arrival estimation in noisy and reverberant environments". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE. 2015, pp. 2814–2818.

[148] Yang et al. "Asynchronous multi-view SLAM". In: *IEEE International Conference on Robotics and Automation.* IEEE. 2021.

[149] Hugh D. Young and Roger A. Freedman. *University Physics with Modern Physics.* 13th. San Francisco: Pearson, 2012. ISBN: 9780321696861.

[150] Yunfeng et al. "Error propagation on the Euclidean group with applications to manipulator kinematics". In: *IEEE Transactions on Robotics* (2006).

[151] Yunfeng et al. "Nonparametric Second-order Theory of Error Propagation on Motion Groups". In: *The International Journal of Robotics Research* (2008).

[152] Zhengyou Zhang. "A flexible new technique for camera calibration". In: *IEEE Transactions on pattern analysis and machine intelligence* 22.11 (2000), pp. 1330–1334.

[153] Zhao et al. "Super odometry: IMU-centric LiDAR-visual-inertial estimator for challenging environments". In: *International Conference on Intelligent Robots and Systems.* IEEE. 2021.

A. Publications

This chapter presents all publications prepared during this work. Each publication will be briefly introduced, and their contributions detailed. The full versions are attached to this thesis.

Journal Contributions:

- IndoorMCD: A Benchmark for Low-Cost Multi-Camera SLAM in Indoor Environments
- Representing Uncertain Spatial Transformations in Robotic Applications in a Structured Framework Leveraging Lie Algebra

Conference Contributions:

- Robust MUSIC-Based Sound Source Localization in Reverberant and Echoic Environments
- Robust Approaches for Localization on Multi-Camera Systems in Dynamic Environments
- URSim - A Versatile Robot Simulator for Extra-Terrestrial Exploration
- Audio Perception in Robotic Assistance for Human Space Exploration: A Feasibility Study
- A Structured Approach for Uncertain Transformations Trees
- Loosely-Coupled Multi-Sensor Visual Odometry: An Asynchronous Approach for Robust Household Robotics

Workshop Contributions:

- Design of a Microphone Array for Rollin' Justin
- Sound Source Localization for Robotic Applications

1. IndoorMCD: A Benchmark for Low-Cost Multi-Camera SLAM in Indoor Environments

Authors:

Marco Sewtz, Yunis Fanger, Xiaozhou Luo, Tim Bodenmüller and Rudolph Triebel

Journal:

Sewtz, Marco, et al. "IndoorMCD: A Benchmark for Low-Cost Multi-Camera SLAM in Indoor Environments." IEEE Robotics and Automation Letters 8.3 (2023): 1707-1714.

Abstract:

Navigating mobile robots within home environments is essential for future applications, e.g. in household or within the field of elderly care. Therefore, these systems, equipped with multiple sensors, have to deal with changing environments. This work presents the IndoorMCD dataset that allows for benchmarking SLAM algorithms within static and changing indoor environments of various difficulties. The dataset provides synchronized and calibrated RGB-D images from a low-cost multi-camera setup, as well as additional IMU data. Further, highly accurate ground truth movement data is provided. It is the first dataset that provides static and changing environments for a multi-camera setup. Evaluations with state-of-the-art SLAM algorithms show the dataset's applicability and also present limitations of current approaches. The dataset is made available in a structured format and a utility library with example scripts is provided.

Contributions:

The author of this dissertation designed the dataset scenarios, prepared the lab environments and was responsible for the correctness and integrity of the recorded data. Calibration was prepared and checked by the author. Dataset creation and evaluation was supported by Yunis Fanger and Xiaozhou Luo. Script was provided by the author and the publication was presented by the author.

Copyright:

©2023 IEEE

IndoorMCD: A Benchmark for Low-Cost Multi-Camera SLAM in Indoor Environments

Marco Sewtz[1], Yunis Fanger[1], Xiaozhou Luo[1], Tim Bodenmüller[1] and Rudolph Triebel[1,2]

***Abstract*—Navigating mobile robots within home environments is essential for future applications, e.g. in household or within the field of elderly care. Therefore, these systems, equipped with multiple sensors, have to deal with changing environments.**

This work presents the IndoorMCD dataset that allows for benchmarking SLAM algorithms within static and changing indoor environments of various difficulties. The dataset provides synchronized and calibrated RGB-D images from a low-cost multi-camera setup, as well as additional IMU data. Further, highly accurate ground truth movement data is provided. It is the first dataset that provides static and changing environments for a multi-camera setup. Evaluations with state-of-the-art SLAM algorithms show the dataset's applicability and also present limitations of current approaches. The dataset is made available in a structured format and a utility library with example scripts is provided.

***Index Terms*—Data Sets for SLAM, Visual-Inertial SLAM, Localization, Mapping, RGB-D, Multi-Camera**

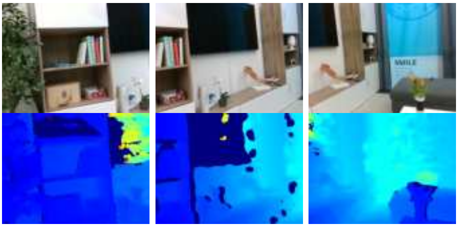


Fig. 1: Multi-camera view of a living room environment captured by commercial off-the-shelf RGB-D sensors.

I. INTRODUCTION

ROBOTIC assistance in home environments is an emerging field of research, opening up new opportunities and applications for autonomous systems. Symbiotic human-robot collaboration and interaction are essential for the success of those ambitions. Thus, robotic systems need to operate, especially navigate, in changing environments reliably. A central element for global navigation is Simultaneous Localization and Mapping (SLAM), as it continuously updates the environmental knowledge of the robot. Although the robustness of state-of-the-art applications is progressively enhanced with each subsequent generation, most of them still rely on a single sensor. A failure of the system likely results in the total loss of localization. However, modern commercial off-the-shelf (COTS) sensors, like RGB-D cameras, are cheap, small and only require little energy. Thus, adding multiple sensors becomes feasible and increases robustness by redundancy.

By using COTS hardware, redundancy can be added while limiting the increase in cost. However, this often results in degraded sensor measurements that integrated software has to consider. For the future development of frameworks to solve the problems mentioned above, a common dataset that includes representative scenarios is crucial. While several datasets have been released as benchmarks for SLAM or other navigation systems, most of them concentrate on the single-sensor case, the use of expensive sensors like LIDAR, or are meant for evaluation in autonomous outdoor-vehicle development.

In this work, we present a dataset that aims to enable research on SLAM systems that address both robustness and redundancy using COTS sensors. It contains five different scenarios, each consisting of several runs of increasing complexity. The recordings include highly accurate ground truth estimation measured by a high-speed motion capture system (MCS). Furthermore, we show the applicability of our data by evaluating the trajectories with state-of-the-art Visual-Inertial Navigation System (VINS) and SLAM systems, as well as an in-house development for multi-camera SLAM [1]. Finally, we also provide a utility library for easy access to the data.

https://rmc.dlr.de/rm/en/staff/marco.sewtz/benchmark

We summarize our contribution as following:

- The IndoorMCD dataset containing 105 individual sequences recorded in indoor environments using multiple COTS sensors, consisting of RGB-D and Inertial Measurement Unit (IMU) modules.
- An additional high accuracy ground truth reference.
- Various scenarios with increasing complexity in their trajectories including loops, motion blur and changes in the environment.
- An extensive evaluation of renowned approaches including performance benchmarks to demonstrate our data's applicability.

Manuscript received: August, 25, 2022; Revised November, 22, 2022; Accepted December, 28, 2022.

This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy (StMWi) by means of the project SMiLE2gether (LABAY102).

[1] Marco Sewtz, Yunis Fanger, Xiaozhou Luo, Tim Bodenmüller and Rudolph Triebel are with the Institute of Robotics and Mechatronics, German Aerospace Center, Wessling, Germany `{firstname.lastname}@dlr.de`

[2] Rudolph Triebel is with the Technical University of Munich (TUM), Munich, Germany

Digital Object Identifier (DOI): see top of this page.

The proposed dataset is, to our knowledge, the first dataset combining multiple sensors and high accurate ground truth for static and changing indoor environments.

II. RELATED WORK

Along with the rising potential of vision-based algorithms, datasets containing realistic environmental conditions have been proposed to provide a reference for new approaches and a baseline for performance benchmarks of existing developments. While the number of available datasets is growing continuously, we provide an overview of the most relevant datasets including visual and inertial data in Table I.

The TUM RGB-D dataset [2] provides a collection of synchronized color and depth data in an indoor scenario recorded in an office environment and an industrial hall. Supplemented by a ground truth reference recorded by a highly accurate MCS, it is one of the most extensively used and established benchmarks for RGB-D Visual Odometry (VO) and SLAM algorithms. Furthermore, the 7-Scenes dataset [3] focuses on realistic indoor-scenes captured by a RGB-D camera and generated ground-truth pose information. Around the same period, the KITTI benchmark suite [4] was proposed for research on vision-based navigation in autonomous driving. In addition to gray-scale mono and RGB stereo sequences, it also includes IMU information. However, the low-frequency inertial data is not synchronized with the visual information, which is mandatory for a well-designed visual-inertial (VI) benchmark. Nevertheless, KITTI has established itself well in the research community and serves as a foundation for further modifications and developments, e.g., object scene flow research [5].

Over time, the focus in the research community has shifted towards the fusion of information provided by different kinds of sensors. Most prominently, many recent datasets are designed to evaluate VO and SLAM applications by including time-synchronized high-frequency IMU measurements. The EuRoC MAV [6] and the more challenging UZH-FPV dataset [7] were recorded with a Micro Aerial Vehicle (MAV). In contrast, one can rely on benchmarks such as TUM VI [8] and OpenLORIS [9] in the case of ground-based carriers. These last four datasets are also equipped with sophisticated ground truth references, which are provided, at least partially, by MCS with an accuracy of approximately $1\mathrm{mm}$.

While the previously presented datasets only include one main viewing direction, the Field-of-View (FoV) size can be significantly expanded by deploying multiple sensing devices with differing orientations. However, most representatives of datasets that employ this approach, such as the NCLT [10] and PennCOSYVIO dataset [11], do neither include high-precision ground truth information nor a hardwired time-synchronization between IMU and the relevant sensors. Therefore, they cannot be considered as an evaluation reference for performance benchmarks between individual VO and SLAM approaches. Currently, the only dataset in the VI domain containing multiple viewing directions that fulfills the requirements for a benchmark is, besides our proposal, the M2DGR dataset [12].

Although the latter benchmark contains a sizable collection of information from different sensor types, data containing multiple viewing orientations are only available in RGB format. This is due to the original design purpose of those sensors, which has the target of achieving an omnidirectional coverage of the related sceneries. Lastly, we also want to mention RIO10 [13], an indoor visual dataset dedicated to changes in the environment – in specific different lightning conditions, object pose changes and appearance. To the best of our knowledge, there is no dataset available that contains multiple visual sensing modalities exceeding the information provided by RGB cameras and operating in dynamic indoor scenes. By supplementing multiple RGB sources with the respectively associated depth information on top of acceleration and angular data, our target is to foster research of multi-camera VO and SLAM approaches in the VI domain.

During the research process, we discovered a significant deficit of datasets for benchmarking the behavior of localization and mapping algorithms in the case of world-model alternation between static and dynamic changing objects within comparable environmental settings. While many conventional VINS and SLAM applications are based on the assumption of a static world, robust approaches must be able to deal with dynamic elements within this world. With the exception of OpenLORIS, all other datasets in Table I are recorded either in a static environment or a dynamic setting with moving objects. Although the benchmark includes static sequences and ones with dynamic moving objects by design, the world-model assumption does not change within individual scenes. Therefore, the performance differences between static and dynamic world assumptions cannot be evaluated in particular since no performance baseline can be provided for the world model within a specific scene.

Hence, we intended to establish our dataset as a benchmark for applications in home environments by providing realistic environmental conditions considering an urban housing scenario based on COTS hardware. In contrast to other established datasets, which are primarily recorded on industrial-grade and customer furnished hardware, the utilization of state-of-the-art COTS sensors allows for a rare peek into the ordinary application-related domain instead of the predominant, more or less idealized, scientific domain. Hence, algorithms have to demonstrate their practicability in real-world situations with imperfect data (e.g. motion blur) and changing environments (e.g. moved chair). However, we neglected temporary dynamic elements in our datasets, e.g. a human walking through the room, as they are more focused on permanent changes and not temporal disturbances.

In terms of emulating the kinematic behavior of typical applications, our dataset is recorded by two different carrier platforms representing either a ground-based robot or a handheld device. The latter assembly provides a total of six unlimited degrees of freedom (DoF) in contrast to the ground-based platforms utilized in benchmarks of similar quality, from which at least 3 DoF are fairly restricted in their magnitude of variability.

III. HARDWARE SETUP

Our hardware setup for recording the dataset consists of three RGB-D Intel RealSense D435i (denoted as *left*, *front*,

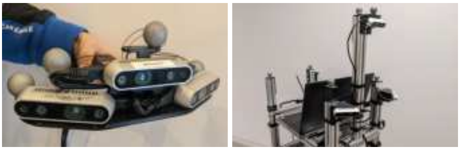
TABLE I: Overview of most common datasets for visual and inertial SLAM in changing indoor environments.

Dataset	**Environ.**	**Platform**	**Cameras**	**IMU**	**Scene mode**	**Ground truth**	**Accuracy**
NCLT [10]	In-/outdoors	Segway	6 RGB 1600×1200 @ 5Hz	1 3DM-GX3-45 3-axis acc./gyro @ 100Hz	Dynamic	Fused GNSS/IMU/Laser pose @ 150Hz	≤ 10cm
EuRoC MAV [6]	Indoors	MAV	1 stereo gray-scale 2 × 752×480 @ 20Hz	1 ADIS16488 3-axis acc./gyro @ 200Hz	Static	Laser tracker pose @ 20Hz, **MCS @ 100Hz**	**≤ 1mm (MCS)**
PennCOSYVIO [11]	In-/outdoors	Handheld	4 RGB (rolling shutter) 1920×1080 @ 30Hz, 1 stereo gray-scale 2 × 752×480 @ 20Hz, 1 fisheye gray-scale 640×480 @ 30Hz	1 ADIS16488 3-axis acc./gyro @ 200Hz, 2 Tango 3-axis acc. @ 128Hz 3-axis gyro @ 100Hz	Dynamic	Fiducial markers pose @ 30Hz	≤ 15cm
TUM VI [8]	In-/outdoors	Handheld	1 stereo gray-scale 2 × 1024×1024 @ 20Hz	1 BMI160 3-axis acc./gyro @ 200Hz	Static	Partial MCS pose @ 120Hz	**≤ 1mm**
UZH-FPV [7]	In/-outdoors	MAV	1 stereo gray-scale 2 × 640×480 @ 30Hz 1 event camera 346×260 @ 50Hz + events	1 MPU-9250 3-axis acc./gyro/magn. @ 500Hz, 1 3-axis acc./gyro @ 1000Hz	Static	Laser tracker pose @ 20Hz	**≤ 1mm**
OpenLORIS [9]	Indoors	Ground robot	1 RGB-D (rolling shutter) 848×480 @ 30Hz, 1 stereo fisheye RGB 2 × 848×480 @ 30Hz	2 BMI055 3-axis acc. @ 250Hz 3-axis gyro @ 400Hz	**Static or Dynamic**	Laser tracker pose @ 40Hz, **MCS pose @ 240Hz**	≤ 3cm (Laser), **≤ 1mm (MCS)**
M2DGR [12]	In-/outdoors	Ground robot	6 fish-eye RGB 1280×1024 @ 15Hz, 1 infrared camera 640×512 @ 25Hz, 1 event camera 640×480 @ 15Hz + events, 1 RGB-D (rolling shutter) 640×480 @ 15Hz	1 Handsfree A9 3-axis acc./gyro/magn. @ 150Hz, 1 BMI055 3-axis acc./gyro @ 200Hz	Dynamic	GNSS pose @ 100Hz, **Laser tracker pose @ 100Hz**, **MCS pose @ 50Hz**	≤ 2cm (GNSS), **≤ 1mm (Laser, MCS)**
7-Scenes [3]	Indoors	Handheld	1 RGB-D 640×480 @ 30Hz	None	Static	Visual Pose Tracking[2]	≤ 2cm
RIO10 [13]	Indoors	Handheld, synthetic	1 RGB 540×960[1], 1 synthetic depth 540×960[1]	None	**Dynamic**	Visual Pose Tracking[2]	≤ 10cm
IndoorMCD (Ours)	Indoors	**Handheld, ground robot**	3 RGB-D (rolling shutter) 640×480 @ 15Hz	3 BMI055 3-axis acc. @ 250Hz 3-axis gyro @ 400Hz	**Static and Dynamic**	**MCS pose @ 100Hz**	**≤ 1mm**

[1] Frame-rate unknown for this dataset. [2] Ground-truth accuracy is unknown and information is based on error metric.

right) in two different configurations.

The first one is a handheld camera device (HCD) which offers 6 DoF and can be easily moved around in the scene. The second one is a robotic platform mock-up called Marvin, which simulates the movement of wheel-based systems. Both are displayed in Figure 2.



(a) The handheld camera device used for capturing motion with six degrees of freedom. (b) The robotic mock-up platform Marvin used for simulating motion of wheel-based systems.

Fig. 2: The used hardware devices for this dataset.

A. Sensor Carriers

1) HCD: This device, as depicted in Figure 2a, integrates all sensors in a compact configuration. The small form-factor allows simple and uncomplicated use by the operator and enables mobile manipulation. While the center camera has an overlapping FoV with both outward-facing cameras, the sensors *left* and *right* do not share a common view. Hence the configuration can be used in algorithms that require visual overlap as well as systems that merely need a known rigid transform. Further, this platform offers a hardware synchronization of all camera modules.

2) Marvin: The used mock-up, as seen in Figure 2b, simulates the movement of a wheel-based robotic system. This reduces the motion to only 3 DoF, in particular x, y and θ. The design is intended to mimic the view of sensors equipped on real assistant systems like Rollin' Justin [14] or the motorized wheelchair EDAN [15]. Due to this fact, the sensors may be blocked by obstacles when closely approaching objects. Furthermore, the configuration of the outer cameras is comparable to the integration in the HCD. However, the center camera is tilted down and raised to offer an improved view of desktops or tables.

B. Sensors

The Intel RealSense D435i consists of a RGB camera, two infrared cameras for depth estimation and an Inertial Measurement Unit.

The image processing of the two infrared cameras is performed internally, and the resulting depth image is pixel-aligned to the color image. Furthermore, a pattern projector operating in the infrared range is integrated to enhance the depth estimation even in textureless environments. The cameras are operated 15Hz with a resolution of 640×480 pixels.

The Inertial Measurement Unit has a triaxial 12-bit linear acceleration and a triaxial 16-bit angular velocity module. The accelerometer is operated at 250Hz and the gyroscope at 400Hz. In our dataset, we provide the single data streams and a fused stream that interpolates the acceleration readings between the gyroscope measurements.

The carriers are equipped with a trigger synchronization circuit. The *front* camera is used as trigger commander and the *left* and *right* cameras are configured as receivers. Although this introduces a slight delay on the trigger for the receiving devices, our results with existing algorithms showed that this offset is negligible in practice.

C. Ground Truth

For all except the real indoor scenario we obtained a highly accurate ground truth estimation using a Vicon MX T40 motion capture tracking system. The recording devices are equipped with several reflective markers, which can be monitored by six infrared cameras hanging from the ceiling. The alignment configuration of the tracking system is individually adapted for each scene to obtain the best and at-all-time continuous estimation of the current pose. The system operates at 100Hz.

The Vicon cameras emit infrared light at the same wavelength as the RealSense pattern projector. However, as the pattern is projected statically and only small dots are visible, we did not measure any interference of the pattern with the tracking system.

IV. CALIBRATION

A. Cameras

The pinhole camera model is used to calibrate the intrinsic parameters of the sensors, which can be obtained using different views of a checkerboard target for each sensor [16]. These parameters consist of the focal-lengths f_x and f_y, the principal point (c_x, c_y) and the skew k_{skew}. The depth image is aligned to the color image on the hardware side of the RealSense devices result in a pixel-to-pixel correspondence in the images. In addition, the Brown-Conrady [17] model can be applied to remove distortion from the color image.

We provide the parameters of the pinhole as well as the Brown-Conrady model in our dataset.

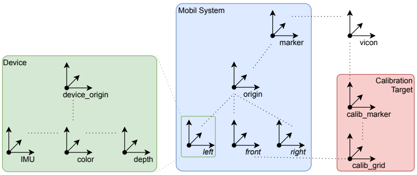


Fig. 3: Illustration of different frames in the dataset including markers and calibration utilities. All individual device origins are calibrated to the overall system's origin.

B. IMUs

The calibration procedure for IMU model estimation is defined by Intel for the RealSense devices [18]. Therefore, each device is orientated in six directions. Several thousand samples are acquired for each, and the parameters are finally optimized over the available set of data. The accelerometer parameters consist of the scale factor $\vec{s} = [s_x, s_y, s_z]^T$, the bias $\vec{b} = [b_x, b_y, b_z]^T$ and the axis alignment $c_{xy}, c_{yx}, c_{xz}, c_{zx}, c_{yz}, c_{zy}$. The intrinsics for the gyroscope include the bias values $\vec{\omega} = [\omega_x, \omega_y, \omega_z]^T$.

C. Extrinsics

The handling of extrinsic calibrations is organized on two levels. At first, all sensors of one RealSense device are handled on the device level, where the color sensor is set as the origin of each device. Therefore, the IMU is calibrated with respect to this sensor. As the depth stream provides a pixel-to-pixel alignment, the resulting displacement is zero.

On the system level, each device is also calibrated using the color sensor. Here, we make use of the fact that the *front* camera overlaps with both the *left* and the *right* camera. Multiple images of a checkerboard calibration target with distinctive origin are captured for estimating the relative pose transform from the *front* camera to the respective target camera. For each image, the correspondences between the checkerboard corners on the calibration target and the projected pixel coordinates are mapped and the transform is estimated by minimizing the reprojection error using Levenberg-Marquardt optimization [19].

For calibrating the Vicon system to the origin of the overall system, the same calibration target as before is used. In addition, several reflective markers are placed on the checkerboard and registered manually to its origin. Afterward, the transform of the *front* camera to the checkerboard and the transform of the markers in the tracking system is estimated and used for aligning the tracking markers to the system origin.

All frames and transforms are illustrated in Figure 3.

D. Time Domains

Within the dataset, different time domains are present as depicted in Figure 4. Each device has its own clock source, which is used for timestamps on the sensor measurements of each device. The timestamping of IMU readings is $\pm50\mu\text{s}$, which leads to tolerance of roughly 2% when operating the

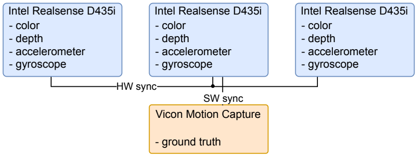


Fig. 4: Overview of the time domains in this dataset. Each RealSense has its own clock and the sensors are triggered device-central. In case the hardware synchronization is present, the trigger signals of the images are synced. The Vicon system is software synchronized.

gyroscope at 400Hz. Therefore, the temporal offset of IMU readings and image capturing on a specific RealSense device can be neglected. Image acquisition is hardware triggered, and the color and depth streams are temporal synced.

In scenarios where the hardware synchronization between the devices is available, the trigger of the image sensors is derived from the commanding camera. In all cases, this is the *front* camera. However, the clocks will not be synced, leading to different timestamps on the images. Exploiting the fact that the images are triggered simultaneously and that the offset between the trigger points is negligible, the clock offset can be estimated by the offset of the color images.

In the non-synced scenarios, the synchronization of the time domains between the devices is not possible without evaluating the trajectory.

The remaining time domain is the Vicon tracking system for ground truth estimation. Thereby, a calibration target is positioned in the view of the *front* camera and tracked by the Vicon system. The target is then slowly moved in the view of the camera. Afterward, the motion is estimated, and the temporal offset is determined by minimizing the absolute pose error (APE). This approach is based on the proposed calibration of Sturm et al. and we refer to their publication [2] for in-depth explanation.

E. Ground Truth

Accurate and continuous information of the actual pose is crucial for investigating the performance of navigation algorithms. Therefore close attention is paid to calibrating the Vicon system before every scenario recording.

The procedure is provided by the manufacturer. It involves operating a calibration stick which is moved in the area of operation and extensively observed by the cameras to create correspondences between individual views. Once enough samples are received, the system calibrates itself by performing optimization for low reprojection error.

V. DATASET

A. Calibration Sequences

These sequences contain the calibration runs used in Section IV. They contain the raw data without any further processing.

TABLE II: Overview of each scenario's (S) specific properties and number of runs (R), as well as if hardware sync has been enabled and if ground truth is available. Scenarios 0-4 have been captured in created environments in our lab, the last one is recorded in an actual apartment.

S	**#R**	**Environment**	**Device**	**Sync**	**GT**
0	19	kitchen, office, living-room	HCD	✓	✓
1	28	kitchen, office, living-room	HCD		✓
2	20	2 rooms: kitchen, living-room	HCD	✓	✓
3	15	2 office desktops	HCD	✓	✓
4	15	kitchen, office, living-room	Marvin		✓
5	10	actual apartment	HCD	✓	

B. Recorded Scenarios

Several scenarios have been recorded in varying setups. Three different environments are created in our labs, including a kitchen, an office area and a living room, which provide a broad set of visual inputs for algorithms. Temporary walls and a door are used to create different room layouts between the scenarios with a total available area of $6.50\text{m} \times 4.50\text{m}$. An exemplary subset of views is shown in Figure 5.

The kitchen consists of a counter including an oven, a fridge, several electronic appliances and commonplace items like apples, cucumbers, or a scale. Most of the structures are static and do not offer a lot of textures. The office area contains depending on the scenario either one or two desktops, including computer monitors, keyboards and a office chair. Further commodities like pens, scissors, or markers are added, which frequently change their position. The living room offers a sofa, including a coffee table, multiple plants and a television shelf. Furniture, as well as the appearance of objects, change over time to simulate human presence. Finally, we also provide a scenario captured in an actual apartment's living room. This room offers a sofa, a television, a fish tank, multiple bookshelves, plants and other common furniture objects. While this scenario does not offer a ground truth, we included it as a proof-of-concept whether proposed systems perform in real environments. An overview is provided in Table II. For measuring the impact of synchronization, scenario 0 and 1 are recorded with and without hardware synchronization in the same environment.

We took care that each run within a specific scenario increments the complexity of the trajectory. At first, they only contain a small number of rotations and single translations. The static-world assumption, meaning no dynamics in the perceived data, is held true. With progressing runs, the trajectories increase in length and amount of movement and ultimately include loops and revisits of previously explored areas. Final runs add changes in the environment that can be observed when places are viewed multiple times. The changes can be seen in Figure 6.

C. Utilities

Additionally to the datasets, we also provide a library for reading the data. It is able to parse the dataset and load the sensor measurements on-demand into the computer memory with a low footprint. Meta-information like extrinsic and

Fig. 5: Stitched panoramic images of views in the dataset. The image on the left-hand side shows the living room as seen in scenarios 0, 1 and 4, in the middle scenario 3 in the office, and on the right-hand side the actual apartment.

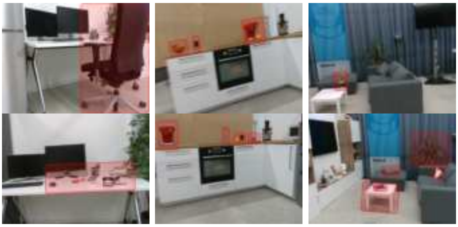


Fig. 6: The dataset captured several changes in the environment during each run. Objects like chairs, the table or the coffee machine are moved around in the scene, smaller objects like books are moved or completely removed, plants have a different appearance over the course of time.

TABLE III: Properties of SLAM systems used for evaluation.

SLAM-system	Type	RGB	IMU	Depth
VINS-Mono	feature-based	✓	✓	
ORBSLAM2	feature-based	✓		✓
ORBSLAM3	feature-based	✓	✓	✓
MROSLAM	feature-based	✓		✓
DSO	direct	✓		

intrinsic calibration and online interpolation of data points are also available. This shall ease access to our data. Furthermore, we provide sample scripts to generate *bag* files to be used within the Robot Operating System (ROS).

VI. EVALUATION

To assess the suitability of this dataset for benchmarking, we evaluate it with state-of-the-art SLAM systems. As examples for feature-based methods, we deploy VINS-Mono [20], ORBSLAM2 [21], ORBSLAM3 [22] and our in-house developed multi-camera approach MROSLAM [1]. Hereby, VINS-Mono processes both IMU and camera data while ORBSLAM2 and MROSLAM purely rely on RGB-D information. ORBSLAM3 incorporates color, depth and inertial data. As a representative of direct visual SLAM methods, we also deploy DSO [23] on the dataset. In this case, it requires only monocular RGB camera images as input. The general properties of the deployed SLAM algorithms are summarized in Table III.

All SLAM applications are configured using the calibration information provided within the dataset (see Section IV) but use the respective systems default parameters otherwise. For each run, three separate instances of these applications are deployed simultaneously to process the data provided by each of the devices. In order to evaluate the dataset's applicability, we assessed our selection of renowned algorithms both in a quantitative and qualitative scope. For the first one, we recorded how many of the devices reach the end of a run without losing tracking at any point or outright failing. The results are presented in Table IV, where scenarios 1-3 were recorded using the HCD and scenario 4 using Marvin. Since scenario 5 does not include ground truth trajectories, we do not consider it here. Therein, only devices where the respective SLAM instance ran for at least 90% of the ground truth trajectory's duration without losing tracking are declared as successful.

At a closer look, it is noteworthy that the multi-camera approach achieved the best results among the purely vision-based approaches. By utilizing information from all devices with different orientations at the same time, a robust construct with multiple redundancies is established, which results in the reduction of potential loss-of-tracking. Especially in comparison to ORBSLAM2, on which MROSLAM is primarily based, the rate of total failure is reduced by a factor of 2.5 in scenario 0 or 2.0 in total. Nevertheless, VINS-Mono already provides a very robust approach which only failed in situations where the sensor's view was blocked and the LoT was not resolvable. Lastly mentioning ORBSLAM3, the performance on tracking seems to be less stable compared to ORBSLAM2. We assume that the extensions focus primarily on the accuracy of the trajectory estimation (as shown later) accepting small deficits in the robustness.

Furthermore, a qualitative assessment is performed using the evo evaluation package [24]. It allows to align the pose estimates of the SLAM systems with ground truth information and the computation of performance measuring metrics from them.

To illustrate the usefulness of having multiple cameras on a single system, we determine the worst-performing device from each scenario. To compare the performance of each instance, we choose the relative pose error (RPE) as our metric. We assume that the utilization of multiple sensors has an measurable impact on local tracking as more data is available. In contrast, global estimation accuracy, in case of continuous tracking, is depending on the selected backend optimization strategy. Therefore we expect the improvements by the used sensor configuration to be observable in short-term domain and neglect APE evaluation. The respective mean and maximum RPE scores for each scenario are presented in Table V.

It is noteworthy that even though the SLAM algorithms occasionally have a high peak error, the mean errors are often reasonably small. This suggests that there were only temporary losses in tracking, which could be recognized and avoided by taking the output of other SLAM instances into account. The utilization of multiple devices has a beneficial

TABLE IV: Quantitative tracking evaluation for each algorithm. The table illustrates how many instances per run did not loose tracking. A value of 18% for 2 devices reads *In 18% of all runs in this scenario, two instances did not loose tracking*.

SLAM-system	**scenario 0**				**scenario 1**				**scenario 2**			
successful devices	**0**	**1**	**2**	**3**	**0**	**1**	**2**	**3**	**0**	**1**	**2**	**3**
VINS-Mono	0%	0%	18%	82%	0%	0%	7%	93%	0%	6%	6%	88%
ORBSLAM2	81%	5%	14%	0%	28%	14%	34%	24%	25%	60%	15%	0%
ORBSLAM3	77%	9%	9%	5%	90%	7%	3%	0%	95%	0%	0%	5%
MROSLAM(*)	32%			68%	21%			79%	8%			92%
DSO	58%	18%	18%	6%	79%	18%	0%	3%	75%	5%	10%	10%
SLAM-system	**scenario 3**				**scenario 4**				**Total**			
successful devices	**0**	**1**	**2**	**3**	**0**	**1**	**2**	**3**	**0**	**1**	**2**	**3**
VINS-Mono	0%	0%	8%	92%	0%	0%	7%	93%	0%	1%	12	87%
ORBSLAM2	0%	6%	31%	63%	0%	0%	27%	73%	30%	18%	25%	27%
ORBSLAM3	0%	0%	25%	75%	0%	0%	47%	53%	61%	4%	14%	21%
MROSLAM(*)	0%			100%	0%			100%	14%			86%
DSO	86%	7%	0%	7%	72%	14%	14%	0%	75%	12%	9%	4%

(*) MROSLAM is a multi-camera approach. There is not differentiation between single instances.

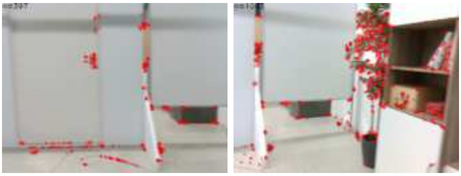


Fig. 7: ORBSLAM2 detected keypoints for two views in a low-texture environment at the same time. The left image shows significant less landmarks (n=397) which could lead to degraded estimation performance or loss of tracking compared to an adjacent camera view (n=1007). Utilizing both views at the same time would further increase the number of available landmarks for tracking and improve accuracy and robustness.

effect on the mean error since the results for MROSLAM rank as one of the lowest in our evaluation. Figure 7 shows the detected keypoints of ORBSLAM2 of two adjacent views. While relying only on a single input, the left image may not provide enough suitable landmarks and tracking will be lost. MROSLAM can use both and is more robust in low-texture cases. However, its maximum error measures are relatively high, indicating even more significant outliers produced in the fusion process which adds constant drift to the estimation as later seen in Figure 8. The more recent ORBSLAM3 occasionally outperforms the multi-camera approach, showing the progress since the introduction of ORBSLAM2 and the derived MROSLAM.

In addition, we also provide representative examples of the pose estimates for the employed SLAM systems compared to the ground truth trajectories in Figure 8. These results show that the visual-inertial system performs better than the purely visual systems in general. Especially during fast rotational movements, the additional information from the IMU leads to significantly better tracking result. Moreover, feature detecting systems perform better than the DSO algorithm, which uses a direct approach. However, the multi-camera MROSLAM suffers a constant drift as it does not implement loop-closure functionality on multiple sensors.

Finally, we evaluate the occurred loss of tracking. We manually examined the frame series in which tracking failure occurred. A significant amount of frames show motion blur or offer only few visual features which can be used for the estimation process. Figure 9 illustrates four individual selected events. They include motion blur and low-textured views offering only limited visual clues for the algorithms. Noteworthy, these defects are frequently observable at the same time. Regardless of the either using a direct approach or relying on features, all algorithms have reduced performance in these situations. However, due to it's multi-sensor nature, MROSLAM is able to recover tracking most of the times.

In summary, this evaluation demonstrates the validity of our dataset as a benchmark for evaluating SLAM systems but also shows the problems of state-of-the-art approaches with motion blur and low-texture environments. Particularly the feature-based visual-inertial system performed well. It also highlights the advantages which multi-camera SLAM approaches could provide. Even though a single device may have poor performance or lose tracking temporarily, others may be more accurate and therefore able to keep the entire system from losing localization.

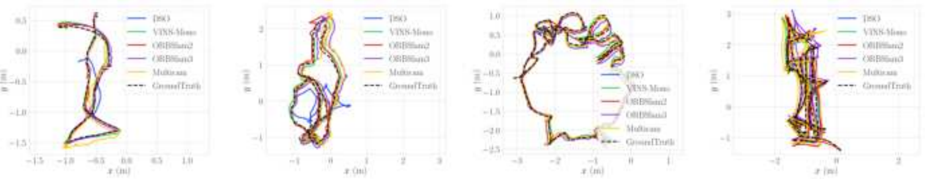
VII. CONCLUSION

This paper presents a novel dataset for the benchmark of SLAM systems in home environments. It mainly focuses on COTS hardware to decrease the costs for sensor setups while providing multiple similar devices to promote robustness. The environments shown represent common areas for service robotics as office, kitchen and living room settings, where static scenarios as well as ones with changes of objects can be observed. High accurate ground truth information obtained through a motion capture system accompanies the recorded data for evaluation of novel systems.

Finally, we analyzed the proposed data using diverse selections of state-of-the-art SLAM systems to prove its applicability. Furthermore, the outcome showed that these algorithms have difficulty tracking under the influence of motion blur, obstructed view, or in an environment of textureless surroundings. Multi-sensor approaches like MROSLAM however promise less loss of tracking and a better performance regarding local pose estimation. Nevertheless, it still has high outliers

TABLE V: Mean and maximum RPE of the worst performing SLAM instance in a scenario.

SLAM-system	**scenario 0**		**scenario 1**		**scenario 2**		**scenario 3**		**scenario 4**	
RPE	**mean**	**max**	**mean**	**max**	**mean**	**max**	**mean**	**max**	**mean**	**max**
VINS-Mono	0.112672	1.586197	0.126820	1.451341	0.0492078	**0.470913**	0.074746	0.619127	0.029876	0.617029
ORBSLAM2	0.245182	5.421701	0.153198	4.642067	0.181427	5.828487	0.159613	2.434654	0.120661	1.731525
ORBSLAM3	0.087262	5.432609	0.087736	3.990802	**0.035891**	4.908698	**0.023212**	3.492762	0.010758	4.931409
MROSLAM	**0.031168**	3.673559	**0.017651**	6.055320	0.042194	6.195327	0.058612	6.435569	**0.010456**	1.028161
DSO	0.118443	**0.573500**	0.072375	**0.359914**	0.064864	0.704010	0.069510	**0.354366**	0.081397	**0.277259**



(a) Scenario 0 run 5. (b) Scenario 1 run 22. (c) Scenario 3 run 11. (d) Scenario 4 run 10.

Fig. 8: Ground truth reference and estimated trajectories. The four runs have been manually selected out of the total 105 as they show the rare case of all methods not loosing tracking. The trajectories show the *front* instance for single-camera after final optimization or the fused pose for MROSLAM which does not have a final processing step or a loop-closure detection.

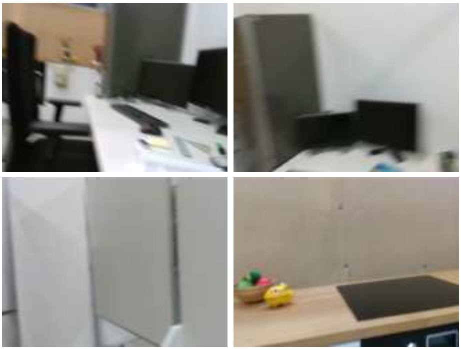


Fig. 9: Examples for views when a loss of tracking occurred. The majority of images is affected by motion blur (upper) or include few visual features (lower) for landmark detection.

which show the necessity of more research on adequate fusion strategies in the multi-sensor scenario.

We, therefore, hope that this dataset contributes to robust yet low-cost robots in home environments.

REFERENCES

[1] M. Sewtz et al., “Robust approaches for localization on multi-camera systems in dynamic environments,” in *2021 7th International Conference on Automation, Robotics and Applications*. IEEE, 2021.

[2] J. Sturm et al., “A benchmark for the evaluation of rgb-d slam systems,” in *2012 IEEE IROS*, 2012.

[3] B. Glocker et al., “Real-time rgb-d camera relocalization,” in *International Symposium on Mixed and Augmented Reality*. IEEE, October 2013.

[4] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *2012 IEEE CVPR*, 2012.

[5] M. Menze and A. Geiger, “Object scene flow for autonomous vehicles,” in *2015 IEEE CVPR*, 2015.

[6] M. Burri et al., “The euroc micro aerial vehicle datasets,” *The International Journal of Robotics Research*, vol. 35, no. 10, 2016.

[7] J. Delmerico et al., “Are we ready for autonomous drone racing? the uzh-fpv drone racing dataset,” in *2019 IEEE ICRA*, 2019.

[8] D.Schubert et al., “The tum vi benchmark for evaluating visual-inertial odometry,” in *2018 IEEE IROS*, 2018.

[9] Shi et al., “Are we ready for service robots? the openloris-scene datasets for lifelong slam,” in *2020 IEEE ICRA*, 2020.

[10] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, “University of michigan north campus long-term vision and lidar dataset,” *The International Journal of Robotics Research*, vol. 35, no. 9, 2016.

[11] B. Pfrommer et al., “Penncosyvio: A challenging visual inertial odometry benchmark,” in *2017 IEEE ICRA*, 2017.

[12] J. Yin et al., “M2dgr: A multi-sensor and multi-scenario slam dataset for ground robots,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, 2022.

[13] J. Wald et al., “Beyond controlled environments: 3d camera re-localization in changing indoor scenes,” in *European Conference on Computer Vision*, 2020.

[14] C. Borst et al., “Rollin’justin-mobile platform with variable base,” in *2009 IEEE ICRA*. IEEE, 2009.

[15] J. Vogel et al., “Edan: An emg-controlled daily assistant to help people with physical disabilities,” in *2020 IEEE/RSJ IROS*. IEEE, 2020.

[16] K. H. Strobl and G. Hirzinger, “More accurate pinhole camera calibration with imperfect planar target,” in *2011 IEEE ICCV*, 2011.

[17] A. E. Conrady, “Decentred Lens-Systems,” *Monthly Notices of the Royal Astronomical Society*, vol. 79, no. 5, 1919. [Online]. Available: https://doi.org/10.1093/mnras/79.5.384

[18] D. J. Mirota and J. J. Scaife, *Intel(R) RealSense(TM) Depth Camera D435i IMU Calibration*.

[19] K. Madsen, H. Nielsen, and O. Tingleff, *Methods for Non-Linear Least Squares Problems (2nd ed.)*, 2004.

[20] T. Qin, P. Li, and S. Shen, “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Transactions on Robotics*, vol. 34, no. 4, 2018.

[21] R. Mur-Artal et al., “Orb-slam: a versatile and accurate monocular slam system,” *IEEE transactions on robotics*, vol. 31, no. 5, 2015.

[22] C. Campos et al., “Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam.”

[23] J. Engel, V. Koltun, and D. Cremers, “Direct sparse odometry,” in *arXiv:1607.02565*, July 2016.

[24] M. Grupp, “evo: Python package for the evaluation of odometry and slam.” https://github.com/MichaelGrupp/evo, 2017.

2. Representing Uncertain Spatial Transformations in Robotic Applications in a Structured Framework Leveraging Lie Algebra

Authors:

Marco Sewtz, Lukas Burkhard, Xiaozhou Luo, Leon Dorscht and Rudolph Triebel

Journal:

Sewtz, Marco et al. "Representing Uncertain Spatial Transformations in Robotic Applications in a Structured Framework Leveraging Lie Algebra", International Journal of Mechanical Engineering and Robotics Research, 2025. (in print)

Abstract:

Accurately representing spatial transformations in robotics is crucial for reliable system performance. Traditional methods often fail to account for internal inaccuracies and environmental factors, leading to significant errors. This work introduces a framework that incorporates uncertainty into transformation trees using Lie Algebra, offering a consistent and realistic computation of spatial transformations. Our approach models inaccuracies from sensor decalibration, joint position errors, mechanical stress, and gravitational influences, as well as environmental uncertainties from perception limitations. By integrating probabilistic models into transformation calculations, we provide a robust and adaptable solution for various robotic applications. The framework is implemented using a C++ library with a Python wrapper, leveraging hierarchical transformation trees to simplify kinematic chains and apply uncertainty propagation. Real-world examples demonstrate the framework's effectiveness: compensating for gravitational bending in a robotic arm and handling uncertainties in a mapping task with an uncertain kinematic. These applications highlight the framework's ability to enhance the accuracy and reliability of tasks such as manipulation, navigation, and interaction with environments. This contribution aims to advance robotic systems' performance by providing a comprehensive method for managing spatial transformation uncertainties.

Contributions:

The author of this dissertation designed the software architecture, the tree structure and the experimental setup for evaluation. The mathematical framework was provided by Lukas Burkhard. Support for the software development was done by Leon Dorscht, support for the evaluation was done by Xiaozhou Luo. The script was provided by the author.

Copyright:

©2025 IJMERR

Representing Uncertain Spatial Transformations in Robotic Applications in a Structured Framework Leveraging Lie Algebra

Marco Sewtz, Lukas Burkhard, Xiaozhou Luo, Leon Dorscht, Rudolph Triebel
Institute of Robotics and Mechatronics
German Aerospace Center (DLR)
Wessling, Germany
{firstname.lastname}@dlr.de

***Abstract*—Accurately representing spatial transformations in robotics is crucial for reliable system performance. Traditional methods often fail to account for internal inaccuracies and environmental factors, leading to significant errors. This work introduces a framework that incorporates uncertainty into transformation trees using Lie Algebra, offering a consistent and realistic computation of spatial transformations. Our approach models inaccuracies from sensor decalibration, joint position errors, mechanical stress, and gravitational influences, as well as environmental uncertainties from perception limitations. By integrating probabilistic models into transformation calculations, we provide a robust and adaptable solution for various robotic applications. The framework is implemented using a C++ library with a Python wrapper, leveraging hierarchical transformation trees to simplify kinematic chains and apply uncertainty propagation. Real-world examples demonstrate the framework's effectiveness: compensating for gravitational bending in a robotic arm and handling uncertainties in a mapping task with an uncertain kinematic. These applications highlight the framework's ability to enhance the accuracy and reliability of tasks such as manipulation, navigation, and interaction with environments. This contribution aims to advance robotic systems' performance by providing a comprehensive method for managing spatial transformation uncertainties.**

***Index Terms*—robotics, transformation tree, uncertainty modeling, Lie Algebra**

I. INTRODUCTION

In the dynamic landscape of robotics, accurately representing spatial transformations is pivotal for reliable system performance. Conventional methods, which treat provided transformations as precise and deterministic, face difficulties with inherent inaccuracies within the system and environmental complexities. This work underscores the critical need for inaccuracies-aware spatial representations in robotics, often denoted as scene graphs. These representations allow modeling not only the spatial relationships in a robot-environment system but also the gaps in our knowledge about it.

An illustrative instance can be found in the distinction between a robotic arm's repetition accuracy, which signifies its capability to consistently reach the same point in a workspace, and the robot's absolute accuracy. For conventional robotic systems, the first can be assumed to be "exact". However, the error of the latter can be higher by several orders of magnitude, motivating the modeling of this error. Position measurements, constrained by both physical limitations and environmental influences, frequently fall short of the requisite precision. This constraint becomes especially critical in applications requiring high accuracy, such as surgical robotics [1].

An additional example is the process of registering a robot with respect to its environment, a task achieved through either an inaugural calibration procedure [2] or by means of the navigation implemented in mobile robotic systems [3].

Interestingly, various scholarly works [4], [5] have considered robot uncertainty within specific domains, such as the kinematic structure or autonomous navigation components. However, there is limited progress in combining these several domains into one single representation like a scene graph to achieve a unified consideration of inaccuracy-aware spatial relations. Conventional approaches that disregard uncertainty in scene graphs fall short in capturing the intricacies of real-world scenarios.

This paper advocates for a paradigm shift by introducing a framework that incorporates uncertainty into scene graphs, offering a more realistic and robust representation of transformations. By addressing challenges posed by both robot internal inaccuracies and the uncertainty of the robot's interaction with the environment, our approach aims to enhance the reliability and performance of robotic systems in practical applications.

We use the following terminology in this paper: Robotic systems can be subject to errors that cause *inaccurate* pose calculations, either within the system or with respect to its environment. A common simplification is to model such inaccuracies in a probabilistic way, thus subjecting nominal relative poses to an additional *uncertainty*. For a multitude of robotic applications, such uncertainty is modeled as a *zero-mean normal distribution*, thus an uncertain pose consists of a nominal pose and a covariance matrix. Generally, this simplification trades the exact representation of robotic errors for the availability of powerful mathematical tools and is thus well established in the robotic community. We adopt this error modeling as well, which allows us to immediately integrate the probabilistic pose information from other software components into our scene graph.

II. RELATED WORK

Accurately describing the spatial relationships of a robot and its environment is a key aspect of robotics specifically and mechanical mechanisms generally. This involves not only understanding the robot's position and orientation within its workspace but also how it interacts with various objects and obstacles around it. The ability to model and predict these interactions is crucial for tasks such as navigation, manipulation, and automated decision-making. Furthermore, a precise understanding of spatial relationships enhances the robot's efficiency, safety, and adaptability in complex and dynamic environments. Consequently, advancements in this area have significant implications for the development of more sophisticated and capable robotic systems.

Commencing with the early explorations in formulating a framework for kinematics in mechanical structures [6], [7], the field witnessed significant strides with one of the pivotal works by Denavit and Hartenberg [8]. In this groundbreaking contribution, the authors devised a structured yet elegant methodology to comprehensively describe the chain of transformations associated with robotic arms. Subsequent endeavors augmented the toolbox of robot kinematics representation, for example by considering the underlying Lie-Algebra of spatial transformations [9]. Advancements in the use of conformal geometric algebra have provided a unified approach to geometric reasoning, simplifying the computation of kinematics and dynamics of serial manipulators [10]. Moreover, neural network-based approaches and deep reinforcement learning have enhanced the precision and efficiency of solving inverse kinematics problems for high degrees of freedom manipulators [11], [12]. Our recent work [13][1] provides a kinematic robot description that allows considering inaccuracies from joint position measurements, mechanical stress-induced deformations, and gravitational influences in a probabilistic manner.

In the field of robotic navigation, numerous approaches account for the uncertainty of relative transformations, particularly in the domain of Simultaneous Localization and Mapping (SLAM). For instance, methods such as those proposed by Kaess et al. in iSAM2 [14] and Kümmerle et al. in g2o [15] utilize the covariance or information matrix to appropriately weigh different spatial transformations within a graph optimization framework. Recent advancements include the development of distributed pose graph optimization, which enhances collaborative SLAM by efficiently managing local and global uncertainties [16], and the integration of multi-level graph partitioning to improve scalability and accuracy [17]. These techniques enhance the accuracy and reliability of mapping and localization by effectively managing the inherent uncertainties in sensor measurements and environmental interactions.

The interaction of a robot with objects in its environment, specifically the uncertainties inherent in the workspace, has been investigated in [18]. Additionally, significant progress has been made in modeling the uncertainty in the perception process itself, including both classical [4] and deep-learning-based methods [5][1]. Recent research efforts have focused on sparse iterative approaches [19] to further enhance robustness in uncertain environments.

Finally, the hand-eye calibration of a robot is nothing else but an additional transformation between the real and the nominal robot geometry and can thus also be subject to inaccuracies, as discussed by [2]. Recent studies have further explored these uncertainties, proposing methods to enhance the accuracy and robustness of hand-eye calibration [20], [21]. These advancements highlight the ongoing need to address and mitigate calibration inaccuracies in vision-guided robotic systems.

In the end, all these sub-fields of robotics provide a multitude of different types of spatial transformations, where potentially all of them are subjected to errors which are being modeled as uncertainties.

Systematic approaches to order a multitude of interconnected transformations, particularly within the realm of virtual reality (VR) [22], [23], and robotic simulators [24], [25], considered the utilization of a scene graph to represent relative spatial relationships. This scene graph, akin to a tree structure, comprises multiple nodes arranged in a parent-child manner. This innovative approach enhanced the representation and simulation capabilities in both virtual reality and robotic simulation domains. The current state of the art is *tf* [26], the scene graph framework of ROS (robot operating system).

Interestingly, very little work has been published that considers the uncertainty of spatial information by interconnecting the different realms of robotics. Initial efforts have been directed towards acknowledging uncertainty within the scene graph, for example [27]. However, these early attempts typically fall short in correctly modeling the error propagation using Lie Algebra. Alternatively, some implementations resort to sampling-based approaches to represent the overall uncertainty within the system, such as [28], which however comes with computational costs.

The Lie-Algebra allows to acknowledge the manifold character of spatial relationships and is a powerful tool to compute and propagate uncertainty along chains of spatial transformations. An introduction to it together with the application to robotic navigation is provided by [29]. Similarly, Lie-Algebra-based concepts are provided for the error propagation within robotic manipulators, either for single errors [30] or as our comprehensive kinematic model [13].

Despite the widespread use of Lie Algebra in uncertainty estimation, to the best of our knowledge, no existing approach formulating a scene graph for robotics has integrated Lie Algebra-based uncertainty propagation. In our ongoing work, we aim to address this gap and demonstrate the efficacy of incorporating Lie Algebra into a scene graph framework for a more nuanced and accurate representation of uncertainty in kinematic systems.

[1] Now known as L. Burkhard *et al.*

III. ROBOTIC AND ENVIRONMENTAL CONFIGURATION STATE

Accurate assessment of the current configuration state in robotic systems holds significant importance across various applications. This is particularly pronounced in scenarios involving non-static components equipped with perception sensors, where precise positional data is crucial for effective operation. Registering cameras affixed to robotic manipulators to the robot's origin is imperative for seamlessly integrating spatial information within the correct coordinate framework.

Knowledge of the system's distance to the environment is indispensable for collision avoidance, especially when navigating confined spaces. To achieve this, it is crucial to carefully observe and organize the positions of joints into a transformation tree. This tree not only helps illustrate how the coordinate framework depends on a specified starting point known as the root frame, but also aids in obtaining an accurate estimate of the robot's spatial volume and movement range.

However, overlooking the inherent uncertainty in these measurements and the subtle non-static characteristics of certain links—attributable to mechanical stress and gravitational forces—can lead to erroneous state estimations. These factors can significantly impact the reliability of the robot's operation, particularly in dynamic or unpredictable environments.

In the ensuing discussion, we elaborate on representing the robotic and environmental configuration state (RECS) as a transformation tree. We discuss the methodology for constructing this tree, highlighting the importance of each node and its relationship to the overall framework. Subsequently, we introduce Lie Algebra as a robust solution for modeling uncertainty in this process. Lie Algebra provides a mathematical structure that allows for the representation and manipulation of spatial transformations, which is essential for accurately modeling the uncertainties and variances in the robot's configuration.

Finally, we detail our implementation of a managed and centralized approach for addressing the RECS problem within an inter-process communication (IPC) framework. This approach not only centralizes the data processing but also ensures that all components of the robotic system are synchronized and updated in real-time, enhancing the overall accuracy and efficiency of the system.

Throughout this work, we intend to conceptualize the inaccuracies within the system as a form of uncertainty. This approach is motivated by the computational convenience afforded through the utilization of a probabilistic model, as opposed to employing distinct models tailored to individual system errors. By treating all potential errors as probabilistic uncertainties, we can simplify the computational processes and improve the robustness of the system's state estimation.

We believe that this comprehensive approach to modeling and managing uncertainties will significantly enhance the performance and reliability of robotic systems, particularly in complex and dynamic operational environments.

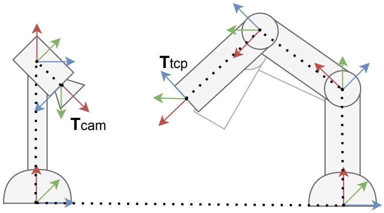


Fig. 1: An illustrated exampled of a robotic manipulator and an external camera.

A. Transformation Tree

Deriving the transformation between two coordinate frames is a pivotal task in robotics. A widely employed approach involves modeling the system as a hierarchical tree of frame transformations. Figure 1 illustrates a typical example involving a robotic manipulator and an external camera. To get the transformation between the coordinate frames $\mathbf{T}_{cam}$ and $\mathbf{T}_{tcp}$, the entire path involving multiple individual transformations must be calculated. In this example, the impact of uncaptured deviations in kinematics from the real world can be observed. The manipulator bends due to gravitational forces, causing the actual position of $\mathbf{T}_{tcp}$ to differ from the expected position derived from a naive approach based on exact measurements. This discrepancy highlights the importance of accounting for real-world factors such as mechanical flexibilities and external forces in kinematic modeling to ensure accurate predictions and reliable performance in practical applications.

A key optimization involves consolidating static displacements into a singular transformation, effectively pruning the tree for computational efficiency. This means that static transformations, which do not change over time, are combined into a single transformation matrix. Movable connections are represented as rotations or translations centered around joints, contributing to a chain of static links and dynamic joints. This approach not only streamlines computational complexity but also provides a comprehensive understanding of a robotic system's kinematic properties, enhancing both efficiency and reliability.

One significant advantage of using a hierarchical tree structure is that it can be directly derived from a CAD (computer-aided design) model, which inherently uses the same representation. computer-aided design (CAD) models are typically organized into a hierarchy of parts and subassemblies, mirroring the structure of the transformation tree. This direct correlation allows for seamless integration and accurate transfer of geometric data from design to implementation.

Following the comprehensive description of robot kinematics within the previously mentioned tree structure, the process of retrieving the direct transformation between any two arbitrary frames unfolds by traversing the path articulated within this structured tree. This systematic approach ensures a clear and methodical procedure for obtaining the specific transformation information required for precise spatial relationships between frames within the robotic system.

By organizing transformations into a hierarchical tree structure, we can simplify complex kinematic chains into more manageable sub-problems. This not only reduces the computational burden but also makes the system more scalable and adaptable to changes. Furthermore, the hierarchical model aids in debugging and enhances the modularity of the kinematic analysis, facilitating easier updates and maintenance. An illustration of this is provided in Figure 2.

B. Transformations and Uncertainty

Our treatment of uncertainties follows our previous work on probabilistic robot kinematics [13], which in turn builds upon the mathematical foundations provided by [29] and [31].

We briefly introduce the applied methods here, but refer the interested reader to the related works for more thorough insights. For a general introduction to Lie Algebra in the scope of robotics, we recommend the excellent [32], whose notation we mostly follow.

Lie Algebra provides a mathematical framework for describing the properties and behaviors of Lie groups, which are groups that also have the structure of a smooth manifold. This framework is particularly useful in robotics for representing rotations and rigid body transformations, as these operations form the basis of many kinematic and dynamic calculations.

A pose $\boldsymbol{T}_{AB} \in SE(3)$ describes the position and orientation of an object B with respect to a reference frame A. The Special Euclidean group $SE(3)$ includes both rotations and translations in three-dimensional space. While a pose quantity is generally an element of the manifold $SE(3)$, it can be described *locally* by its linear tangent space representation $\boldsymbol{\xi} = [\boldsymbol{\rho}\,\boldsymbol{\theta}]^T \in \mathbb{R}^6$, related by the exponential map [32]:

$$\boldsymbol{T} = \operatorname{Exp}(\boldsymbol{\xi}). \tag{1}$$

Here, $\boldsymbol{\rho}$ denotes the translational component and $\boldsymbol{\theta}$ the rotational component of the tangent space element. The exponential map allows for the conversion between the tangent space (Lie algebra) and the manifold (Lie group).

In Lie Algebra, the tangent space at the identity element of a Lie group forms a vector space called the Lie algebra of the group. For $SE(3)$, this tangent space can be represented as a six-dimensional vector comprising three translational and three rotational components. The adjoint representation provides a way to map local tangent space quantities between different coordinate frames.

Local tangent space quantities can be mapped between two different local spaces using the *adjoint matrix* $\mathbf{Ad}$ as:

$${}^{A}\boldsymbol{\xi} = \mathbf{Ad}(\boldsymbol{T}_{AB})\, {}^{B}\boldsymbol{\xi}, \tag{2}$$

with

$$\mathbf{Ad} = \begin{bmatrix} \boldsymbol{R} & [\boldsymbol{t}]_\times \boldsymbol{R} \\ \boldsymbol{0} & \boldsymbol{R} \end{bmatrix} \in \mathbb{R}^{6\times 6}, \tag{3}$$

where $\boldsymbol{R}$ is the rotation matrix of $\boldsymbol{T}$ and $[\boldsymbol{t}]_\times$ is the skew-symmetric matrix formed by the translation vector. The term $[\boldsymbol{t}]_\times \boldsymbol{R}$ illustrates how local rotation errors can create translation errors further down a chain of transformations, with the magnitude depending on the distance from the original error's location.

To understand this, consider that any rotation in three-dimensional space can be represented as an element of the $SO(3)$ group, the special orthogonal group, which deals with rotation matrices. Similarly, $SE(3)$ extends this concept to include translations. The Lie algebra of $SO(3)$ consists of skew-symmetric matrices that represent infinitesimal rotations, while the Lie algebra of $SE(3)$ includes both infinitesimal rotations and translations.

We describe the error of a pose as a local deviation $\boldsymbol{\xi}_{\text{B,err}}$ of a nominal pose $\boldsymbol{T}_{AB}$, i.e., in the tangent space of the pose's reference frame B. The corresponding covariance matrix $\boldsymbol{\Sigma}_{AB} = \mathbb{E}\big[\boldsymbol{\xi}_{\text{B,err}}\, \boldsymbol{\xi}_{\text{B,err}}^T\big] \in \mathbb{R}^{6\times 6}$ is therefore a locally defined tangent space quantity. This covariance matrix encapsulates the uncertainty in both the translational and rotational components of the pose.

The two essential mathematical operations on poses needed for the scene graph are concatenation and inversion. The *concatenation* operation combines two transformations, such as $\boldsymbol{T}_{AB}$ and $\boldsymbol{T}_{BC}$, to yield the transformation from A to C:

$$\boldsymbol{T}_{AC} = \boldsymbol{T}_{AB} * \boldsymbol{T}_{BC}, \tag{4}$$

$$\boldsymbol{\Sigma}_{AC} = \mathbf{Ad}_{\boldsymbol{T}_{BC}^{-1}} \boldsymbol{\Sigma}_{AB} \mathbf{Ad}_{\boldsymbol{T}_{BC}^{-1}}^T + \boldsymbol{\Sigma}_{BC}. \tag{5}$$

Here, the two covariance matrices are transformed into the common reference frame C using the adjoint matrix, where they can be added due to the linearity of the tangent space. The covariance composition in eq. (5) is a first-order approximation (referred to as second order in some publications) and is discussed in detail in [29].

Analogously, the *inverse* operation calculates the transformation from B to A given the transformation from A to B:

$$\boldsymbol{T}_{BA} = \boldsymbol{T}_{AB}^{-1}, \tag{6}$$

$$\boldsymbol{\Sigma}_{BA} = \mathbf{Ad}_{\boldsymbol{T}_{AB}} \boldsymbol{\Sigma}_{AB} \mathbf{Ad}_{\boldsymbol{T}_{AB}}^T, \tag{7}$$

This shifts the uncertainty from the tangent space of B to the tangent space of A. This representation can implicitly consider *exact* transformations, as zero-covariances simply vanish in eq. (5) and eq. (7).

For a more detailed introduction to Lie Algebra and its application in robotics, readers may refer to [32] and other comprehensive resources like [33] and [34].

C. Implementation

The presented methodology has been implemented within a C++ library, and the corresponding source code is accessible online[2]. Additionally, a wrapper for the scripting language Python is provided, facilitating ease of use and integration into various applications. Each coordinate frame is characterized by a node element. A frame is precisely defined by its pose matrix $\boldsymbol{T}$ and an accompanying covariance matrix $\boldsymbol{\Sigma}$, which

---

[2] https://github.com/DLR-RM/tf-dude

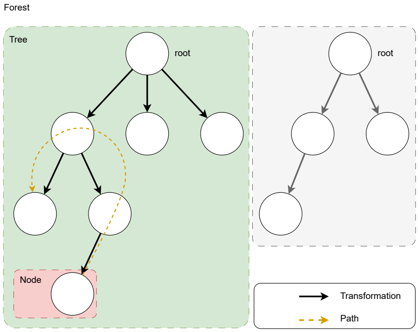


Fig. 2: A schematic overview of the forest and tree structure holding all transformation information.

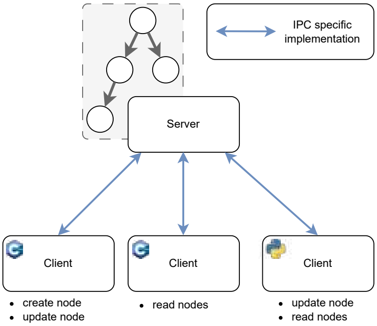


Fig. 3: Illustration of an exemplary server-client architecture with different API implementations.

may be set to zero for precisely known transformations. Distinctive identification of each frame is facilitated through the application of a unique character string.

Furthermore, the mathematical operations of *concatenation* and *inverse* for each frame are executed leveraging the computational capabilities provided by the *manif* library [32], which is augmented by the uncertainty propagation framework. This ensures that transformations account for any uncertainties in the positional data, thereby enhancing the robustness of the system.

The hierarchical structure is implemented using the Boost.Graph data structure [35]. Each vertex encapsulates a frame as its payload, and the edges define the direction of transformations. To determine a path between two nodes within the tree, a breadth-first search (BFS) routing algorithm is employed. The cumulative transformation along the identified path is computed based on the direction specified by the graph's edges, facilitating a comprehensive understanding of the transformations between the starting and ending points of the path.

The system allows for the addition of multiple root nodes, thereby declaring new trees that remain disconnected from preceding ones. It is imperative to underscore that the establishment of a path between nodes situated on distinct trees within the forest is not feasible. Each root node initiates an independent tree structure, and inter-tree connectivity is explicitly precluded within the system's framework.

The communication backend is implemented in an IPC-agnostic way, meaning that it can support various implementations of IPC such as ROS [36], ROS2 [37], native DDS [38], links_and_nodes [39] or other systems. This flexibility is achieved through the use of generic adapters that must be overloaded by the implementation using a plugin functionality. These adapters abstract the communication details, allowing the core library to remain independent of the specific IPC mechanism employed. This design ensures that the system can be easily integrated into different robotic frameworks without requiring significant modifications to the underlying codebase.

The default operational paradigm involves centralized control over all trees, nodes, and computations via a central server. A connected client has the capability to perform various operations such as creating, retrieving, updating, or deleting (CRUD) nodes. Additionally, the client can request the cumulative transformation of a specific path. Other clients can also access this information, but their requests must be routed through the server. This centralized architecture ensures efficient management and coordination of resources.

This implementation offers significant advantages in terms of flexibility and scalability. By leveraging well-established libraries and algorithms, the system ensures high performance and reliability. Furthermore, the clear separation of responsibilities between the server and clients facilitates efficient resource management and provides a robust framework for complex robotic applications. The IPC-agnostic design further enhances the system's adaptability, making it suitable for a wide range of robotic platforms and use cases. This architecture is illustrated in Figure 3.

IV. APPLICATION

To demonstrate the practical utility and broad applicability of the proposed framework, two application examples will be illustrated in the following sections. An in-depth analysis of applying Lie Algebra to the configuration modeling problem has been presented in [13]. Therefore, we will focus on the scene-graph implementation in this discussion.

The first application showcases the integration of the framework on a robotic arm, which is affected by bending induced by the gravitational pull of the Earth. This example highlights how the system compensates for real-world physical effects that deviate from ideal models. By applying the proposed methodology, we can accurately model and correct for these

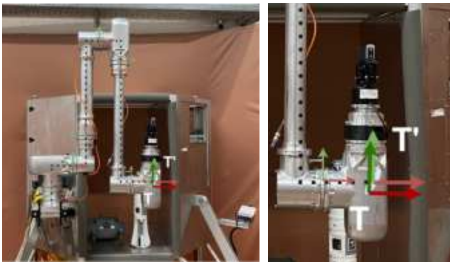


Fig. 4: TINA arm bending due to gravitation.

deviations, ensuring the robotic arm operates with high precision despite the bending.

The second application illustrates a mapping task on a system with an uncertain RECS, formulated as a graph optimization problem. This example demonstrates how the framework handles uncertainties in the robotic configuration space, ensuring accurate and reliable mapping. By using a robust scene-graph implementation, the system can dynamically adjust to changes and uncertainties in the environment, maintaining the integrity of the mapping process.

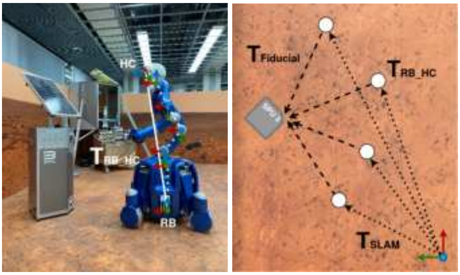
These examples are chosen to underscore the versatility and robustness of the proposed framework in handling various practical challenges in robotics. They provide concrete evidence of how the framework can be applied to real-world scenarios, demonstrating its effectiveness in improving the accuracy and reliability of robotic systems. Through these applications, we aim to showcase the framework's potential for widespread use in diverse robotic applications, highlighting its capability to address complex problems with innovative solutions.

A. Uncertain Robotic and Environmental Configuration State

As an integral component of the European Space Agency (ESA) project for a Sample Transfer Arm breadboard study, the German Aerospace Center (DLR) developed the TINA manipulator [40] as a compact, modular, and torque-controlled robotic system designed to meet the requirements of the Mars Sample Return mission. Figure 4 illustrates the robotic arm in its initial position mounted on a lander.

Upon closer inspection, it becomes evident that the manipulator, even in its initial configuration, experiences moderate deformations attributable to its own weight and joint play, particularly in the axial direction. These deformations introduce uncertainties in the pose of the end effector, which can be effectively modeled using the proposed framework.

By incorporating the expected variance parameters into the transformation tree, the state of the robot configuration can be predicted probabilistically. This allows the position of the end effector to be constrained within an anticipated uncertainty region. Consequently, considering these uncertainties provides a more realistic depiction of the arm's pose, acknowledging the impact of various factors, including gravitational forces, and enhances the accuracy of the positional assessment, enabling more precise manipulations.



(a) (b)

Fig. 5: Rollin' Justin mapping a SPU in a Martian environment (a) and the associated optimization graph is represented in (b).

The selection of appropriate probabilistic parameters heavily depends on the specific characteristics of the associated system and requires specialized technical knowledge. If necessary, experimental evaluations must be conducted to validate and fine-tune these parameters. This approach ensures that the manipulator's performance remains robust and reliable, even in the presence of inherent uncertainties.

B. Environmental Mapping

To enable more intricate manipulations and interactions between the robot and its environment, a significant challenge lies in achieving precise registration of the robot relative to its surroundings. This entails aligning various world representations generated for different types of tasks to ensure coherence and accuracy in the robot's perception of its environment.

As depicted in Figure 5a, Rollin' Justin [41] is mapping a Smart Payload Unit (SPU) in Martian surroundings. In addition to the unknown state of the environmental configuration, further challenges arise from within the robot itself. Although the upper body assembly is rigidly connected to the base platform, the wire rope construction in different parts of the torso is inherently less precise than the rigid joints of the arms, introducing uncertainties into the robot's configuration state.

Effectively managing and mitigating this uncertainty is crucial since information for navigation purposes is collected from sensors in the base, while other higher-level tasks, such as object recognition and manipulation, rely on information from the camera mounted in Justin's head. Therefore, modeling the spatial relations of the robot's configuration state, including uncertainties, is essential and can be addressed by the proposed framework. This framework simplifies the handling of transformations and their associated uncertainties by summarizing them into a single step.

In the context of environmental mapping, the transformation from the robot base to the head camera becomes particularly critical as it serves as the foundation for registering fiducials linked to the SPU. Combined with the spatial relationship to the registered fiducials and information regarding the global reference provided by MROSLAM [3], an optimization graph can be constructed, as illustrated in Figure 5b. The optimization problem can be effectively addressed using GTSAM [42] or comparable algorithms, leading to an optimized estimation of the SPU's pose.

This comprehensive approach significantly improves the reliability and quality of environmental mapping outcomes in the robot's operational context. By integrating precise registration techniques and robust uncertainty modeling, the framework enhances the robot's ability to interact accurately and efficiently with its environment, ensuring higher levels of performance in complex tasks.

V. CONCLUSION

This paper introduces a robust framework for representing uncertain spatial transformations in robotic systems, leveraging Lie Algebra for a structured and probabilistic approach. Traditional deterministic methods often fall short in accounting for the inherent inaccuracies and environmental factors that affect robotic operations. Our proposed framework addresses these limitations by incorporating uncertainty into transformation trees, providing a more realistic and reliable computation of spatial transformations.

The framework models inaccuracies arising from sensor decalibration, joint position errors, mechanical stress, and gravitational influences, as well as environmental uncertainties from perception limitations. By integrating probabilistic models into the transformation calculations, we offer a robust and adaptable solution for various robotic applications, enhancing the system's ability to handle real-world complexities.

We demonstrate the practical utility of the proposed framework through two application examples. The first example involves a robotic arm affected by gravitational bending, showcasing how the system compensates for real-world physical effects that deviate from ideal models. The second example illustrates a mapping task on a system with an uncertain robotic and environmental configuration state (RECS), formulated as a graph optimization problem. These applications highlight the framework's effectiveness in improving positional accuracy and enabling precise manipulations.

The hierarchical transformation tree structure not only simplifies complex kinematic chains but also provides a comprehensive understanding of the robot's spatial relationships. This approach reduces computational complexity and enhances the scalability and adaptability of the system. Additionally, the IPC-agnostic design allows for easy integration into different robotic frameworks, further enhancing the system's versatility.

Future work includes extending the framework to model temporal deviations, enabling configuration retrieval from previous time steps. We also aim to align the interface with ROS's tf implementation for seamless integration.

In summary, this contribution significantly advances the management of spatial transformation uncertainties in robotics, providing a versatile and robust tool that enhances the reliability and performance of robotic systems in diverse applications. The source code for this framework is accessible online https://github.com/DLR-RM/tf-dude.

ACKNOWLEDGMENT

This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy (StMWi) by means of the project SMiLE2gether (LABAY102).

REFERENCES

[1] U. Seibold, B. Kübler, T. Bahls, R. Haslinger, and F. Steidle, "The DLR MiroSurge surgical robotic demonstrator," in *The Encyclopedia of Medical Robotics: Volume 1 Minimally Invasive Surgical Robotics*. World Scientific, 2019, pp. 111–142.

[2] Nguyen et al., "On the covariance of X in AX = XB," *IEEE Transactions on Robotics*, 2018.

[3] Sewtz et al., "Robust approaches for localization on multi-camera systems in dynamic environments," in *2021 7th International Conference on Automation, Robotics and Applications (ICARA)*, 2021.

[4] Stoiber et al., "A sparse gaussian approach to region-based 6dof object tracking," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[5] Meyer et al., "Robust probabilistic robot arm keypoint detection exploiting kinematic knowledge," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop on Probabilistic Robotics in the Age of Deep Learning*, 2022.

[6] Kennedy, *The Kinematics of Machinery*. New York: D. Van Nostrand, 1881.

[7] Calvert, *Developing problem-solving skills in engineering*, 1953.

[8] Denavit and Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," 1955.

[9] Richard, *A Mathematical Introduction to Robotic Manipulation*, 1994.

[10] T. Löw and S. Calinon, "Geometric algebra for optimal control with applications in manipulation tasks," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3586–3600, 2023.

[11] J. Lu, T. Zou, and X. Jiang, "A neural network based approach to inverse kinematics problem for general six-axis robots," *Sensors*, vol. 22, 2022.

[12] A. Malik, Y. Lischuk, T. Henderson, and R. Prazenica, "A deep reinforcement-learning approach for inverse kinematics solution of a high degree of freedom robotic manipulator," *Robotics*, vol. 11, 2022.

[13] Meyer et al., "The Probabilistic Robot Kinematics Model and its Application to Sensor Fusion," 2022.

[14] Kaess et al., "isam2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, 2012.

[15] Kummerle et al., "g2o0: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*. IEEE.

[16] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, "Distributed trajectory estimation with privacy and communication constraints: A two-stage distributed gauss-seidel approach," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 5261–5268.

[17] C. Li, G. Guo, P. Yi, and Y. Hong, "Distributed pose-graph optimization with multi-level partitioning for multi-robot slam," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 4926–4933, 2024.

[18] Su et al., "Manipulation and propagation of uncertainty and verification of applicability of actions in assembly tasks," *IEEE Transactions on Systems, Man, and Cybernetics*, 1992.

[19] M. Stoiber, M. Sundermeyer, and R. Triebel, "Iterative corresponding geometry: Fusing region and depth for highly efficient 3d tracking of textureless objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6855–6865.

[20] E. e. a. Ruiz, "Methods for simultaneous robot-world-hand-eye calibration: A comparative study," *Sensors*, vol. 19, no. 12, p. 2837, 2019.

[21] I. e. a. Enebuse, "Accuracy evaluation of hand-eye calibration techniques for vision-guided robots," *PLOS ONE*, 2022.

[22] Carlsson et al., "Dive—a platform for multi-user virtual environments," *Computers & graphics*, vol. 17, no. 6, pp. 663–669, 1993.

[23] Tramberend, “Avocado: A distributed virtual reality framework,” in *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*. IEEE, 1999, pp. 14–21.

[24] Browning et al., “Übersim: a multi-robot simulator for robot soccer,” in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, 2003, pp. 948–949.

[25] Drumwright et al., “Extending open dynamics engine for robotics simulation,” in *Simulation, Modeling, and Programming for Autonomous Robots: Second International Conference*. Springer, 2010.

[26] Foote, “tf: The transform library,” in *IEEE Conference on Technologies for Practical Robot Applications (TePRA)*. IEEE, 2013.

[27] Coelho et al., “Osgar: A scene graph with uncertain transformations,” in *IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE, 2004.

[28] T. Ruehr, “uncertain tf,” https://github.com/ruehr/uncertain_tf, 2013, last accessed 2023-11-30.

[29] Barfoot et al., “Associating uncertainty with three-dimensional poses for use in estimation problems,” 2014.

[30] Yunfeng et al., “Error propagation on the euclidean group with applications to manipulator kinematics,” *IEEE Transactions on Robotics*, 2006.

[31] ——, “Nonparametric second-order theory of error propagation on motion groups,” *The International Journal of Robotics Research*, 2008.

[32] Sol et al., “A micro Lie theory for state estimation in robotics,” *CoRR*, 2018.

[33] A. R. for Manufacturing, “Introduction to lie theory and its application to robotics,” https://opentextbooks.clemson.edu/advancedroboticsmanufacturing/chapter/introduction-to-lie-theory-and-its-application-to-robotics/, accessed: 2024-05-18.

[34] P. Coelho and U. Nunes, “Lie algebra application to mobile robot control: a tutorial,” *Robotica*, vol. 22, no. 5, pp. 553–563, 2004, accessed: 2024-05-18. [Online]. Available: https://www.cambridge.org/core/journals/robotica/article/lie-algebra-application-to-mobile-robot-control-a-tutorial/

[35] Boost, “Boost C++ Libraries,” http://www.boost.org/, 2023, last accessed 2023-11-30.

[36] M. Quigley, B. Gerkey, and W. D. Smart, “Ros: an open-source robot operating system,” http://www.ros.org, 2009, accessed: 2024-05-18.

[37] R. Contributors, “Ros2: The robot operating system,” https://index.ros.org/doc/ros2/, 2018, accessed: 2024-05-18.

[38] O. M. G. (OMG), “Data distribution service (dds),” https://www.omg.org/spec/DDS/, 2004, accessed: 2024-05-18.

[39] F. Schmidt, “Links and nodes,” https://gitlab.com/links_and_nodes/links_and_nodes, 2024, accessed: 2024-05-18.

[40] Maier et al., “Tina: The modular torque controlled robotic arm - a study for mars sample return,” in *2021 IEEE Aerospace Conference (50100)*, 2021.

[41] Fuchs et al., “Rollin’ justin - design considerations and realization of a mobile platform for a humanoid upper body,” in *2009 IEEE International Conference on Robotics and Automation*, 2009.

[42] Dellaert et al., “borglab/gtsam,” May 2022. [Online]. Available: https://github.com/borglab/gtsam)

3. Robust MUSIC-Based Sound Source Localization in Reverberant and Echoic Environments

Authors:

Marco Sewtz, Tim Bodenmüller and Rudolph Triebel

Conference:

Sewtz, Marco, Tim Bodenmüller, and Rudolph Triebel. "Robust MUSIC-based sound source localization in reverberant and echoic environments." 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020.

Abstract:

Intuitive human robot interfaces like speech or gesture recognition are essential for gaining acceptance for robots in daily life. However, such interaction requires that the robot detects the human's intention to interact, tracks his position and keeps its sensor systems in an optimal configuration. Audio is a suitable modality for such task as it allows for detecting a speaker in arbitrary positions around the robot. In this paper, we present a novel approach for localization of sound sources by analyzing the frequency spectrum of the received signal and applying a motion model to the estimation process. We use an improved version of the Generalized Singular Value Decomposition (GSVD) based MUltiple SIgnal Classification (MUSIC) algorithm as a direction of arrival (DoA) estimator. Further, we introduce a motion model to enable robust localization in reverberant and echoic environments.

We evaluate the system under real conditions in an experimental setup. Our experiments show that our approach outperforms current state-of-the-art algorithm and demonstrate the robustness against the previously mentioned disruptive factors.

Contributions:

The author of this dissertation designed and implemented the robust and real-time approach for sound source localization in indoor environments. The author designed and executed the evaluation. The script was provided by the author and the publication was presented by the author.

Copyright:

©2020 IEEE

Robust MUSIC-Based Sound Source Localization in Reverberant and Echoic Environments

Marco Sewtz[1] Tim Bodenmüller[1] Rudolph Triebel[1,2]

***Abstract*— Intuitive human robot interfaces like speech or gesture recognition are essential for gaining acceptance for robots in daily life. However, such interaction requires that the robot detects the human's intention to interact, tracks his position and keeps its sensor systems in an optimal configuration. Audio is a suitable modality for such task as it allows for detecting a speaker in arbitrary positions around the robot. In this paper, we present a novel approach for localization of sound sources by analyzing the frequency spectrum of the received signal and applying a motion model to the estimation process. We use an improved version of the Generalized Singular Value Decomposition (GSVD) based MUltiple SIgnal Classification (MUSIC) algorithm as a direction of arrival (DoA) estimator. Further, we introduce a motion model to enable robust localization in reverberant and echoic environments.**

We evaluate the system under real conditions in an experimental setup. Our experiments show that our approach outperforms current state-of-the-art algorithm and demonstrate the robustness against the previously mentioned disruptive factors.

I. INTRODUCTION

The ability of mobile robots to interact with people in an intuitive and maybe anthropomorphic manner is a key to the acceptance of robots in human-dominated environments. Human-robot-interaction (HRI) can be visual (e.g. gestures), tactile (e.g. guiding) as well as auditive (e.g. instructing). However, all modalities require that the robot recognizes the intention of a human to interact. Visual systems can only recognize intention in the sensor's field of view, which is usually limited. Tactile systems require that the human is nearby. Robot audition, however, allows for detecting and tracking a speaker from arbitrary positions around the robot and also from distant places. Figure 1 illustrates a typical situation. The human on the sofa wants to interact with the robot, but the latter is currently performing another task, thus, positioning its visual sensor in the opposite direction. Moreover, audio also allows for gaining information about the environment or to separate between different speakers. The information about the speaker's position can also be used to enhance the audio input signal, e.g. to improve speech processing as well as getting more information about the position of humans in the scenario.

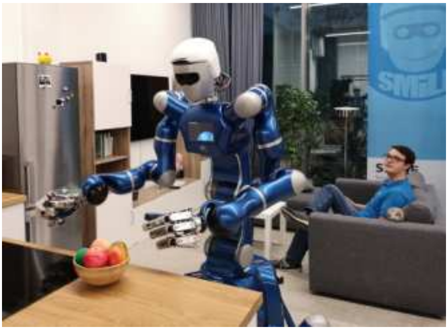


Fig. 1: Illustration of the interaction recognition problem: The robot is turned away from the operator. While the vision system might not recognize him, the audio input will do so.

In this work we present a novel approach for localization of speakers by use of a microphone array. First we detect speech phases in the audio stream using a voice activity detector. During the detection we calculate a score for analyzing the frequency spectrum. We introduce a frequency selection based on this score to enhance the MUSIC estimation and reduce the processing time. For estimation we use the established SEVD-MUSIC [1] algorithm. Further on we propose a motion model to check the calculated Direction of Arrival (DoA) of the received signal. We can show that this enhances robustness against reverberation and echo. Therefore, we present result of realistic experiments that verify our claims.

II. RELATED WORK

In recent years, research has been done to imitate the binaural audio localization of animals and humans [2]–[5]. Using both the interaural phase difference (IPD) and the interaural intensity difference (IID). These techniques take into account the head-related transfer function [6], [7] as well as the reverberant properties of the environment to achieve accurate results. Incorporation of a particle filter approach to be used on binaural measurements improves the estimation of sound sources as well [8]. Nonetheless these systems need a demanding hardware setup and calibration.

Other approaches use an array of microphones to overcome the hardware requirements and to estimate the direction of arrival (DoA) of a signal [9], [10]. It is possible to calculate the most probable DoA by estimating the time delay between the signals received by each microphone.

---

[1] Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Oberpfaffenhofen, Germany.

[2] Dep. of Computer Science, Technical Univ. of Munich, Germany

`marco.sewtz@dlr.de tim.bodenmueller@dlr.de rudolph.triebel@dlr.de`

`rmc.dlr.de/rm/de/staff/marco.sewtz/SSL`

Combining these methods with delay and sum beam forming (DSBF) as well as random sample consensus (RANSAC), more than one sound source can be localized [11]. However, these approaches have problems with low signal-to-noise-ratios (SNR) input signals, changing acoustic conditions and varying speakers. Different approaches using neural networks have been studied to tackle these problems. Nevertheless, they need training dedicated to the specific speaker or require very large amounts of data for generalizing [12]–[16]. Furthermore, Sasaki et al. present an approach incorporating a hypothesis tracking system which exploits the physical constraints of a dynamic moving object [17].

More recently, subspace approaches like Multiple Signal Classification (MUSIC) [18] and Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) [19] have received more interest. They overcome the resolution limit constrained by the sampling rate and are more robust to signal noise but they are computational costly [20]–[23].

There have been several extensions for MUSIC, e.g. using singular value decomposition [24] to reduce the computational complexity while enhancing robustness against noise. Incremental versions are introduced to reach real-time performance while enhancing robustness against noise [25], [26]. Enhancements to further reduce the computational costs in the representation space is done in [27], [28].

However, even recent sound source localization systems face problems when detecting humans in indoor scenarios under non-optimal acoustic conditions.

First, the estimation of speech is challenging. The receiving sound event consists of several words, each composed of vowels and consonants with different frequencies and durations. It is therefore hard to implement a filter a-priori. Active filter system which adapts to the current information in real-time as proposed by Hoshiba et al. [29] tackle this problem. However, human speech consists of frequencies distributed on a wide spectrum. Using only a bandpass which narrows the calculations to small portions of the complete spectrum neglects additional information encoded in the signal or may even led to falsely estimations when the filter adapts to a noise source.

Secondly, indoor scenes often face the problem of having a high reverberation time and shadow sources created by echo. The first phenomena is the superposition of several reflections of the same signal which results in a ”fading-out” effect and lower the SNR. The latter one is the reflection of the full signal at a surface and the system perceives an additional source at the location of the reflecting obstacle.

In this work we propose a novel framework based on the generalized singular value decomposition approach to reduce the complexity for estimating the DoA for localizing speakers. In addition we focus on raising the robustness in reverberant and echoic environments by exploiting intermediate steps of a noise evaluation process and validation based on a motion model. We aim to enable proven state-of-the-art methods for indoor scenarios under real-time constraints.

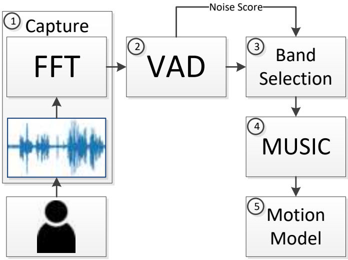


Fig. 2: System overview: ① Voice capture and transform into frequency domain. ② Classification of input as speech or noise phase. ③ Selection of appropriate frequencies. ④ DoA estiamtion with MUSIC. ⑤ Verification of DoA by a motion model.

III. SOUND SOURCE LOCALIZATION

In order to tackle the challenges of indoor environments, as discussed in the previous section, we propose a sound source localization framework called **Motion Model Enhanced MUSIC** (MME-MUSIC). Our system is based on the SEVD-MUSIC [1] approach. We enhance the estimation process by active selection of significant frequencies during Voice Activity Detection (VAD), as well as post-filtering the estimates by application of a motion model. A flow chart of our processing pipeline is given in Figure 2.

A. *Voice Activity Detection and Frequency Selection*

We split the incoming audio recordings into smaller and overlapping frames and transform them into the frequency domain using the fast fourier transform (FFT). Afterwards, the frames are classified into the categories “speech” or “noise”. We implement the Longterm Speech Divergence (LTSD) approach of Ramírez *et al.* [30], which assumes that the spectrum of noise differs significantly from frames containing speech. Yet, short time sound events like clapping or door closing are suppressed. For classification, the divergence of each frequency bin compared to a noise spectrum is computed, which we denote the **noise score** $\nu(k)$

$$\nu(k) = \frac{\overline{\mathrm{LTSE}}_{\tau}(k)^2}{\boldsymbol{X}_{\Sigma}(k)^2} \quad , \tag{1}$$

where $\overline{\mathrm{LTSE}}_{\tau}(k)$ is the average maximal amplitude of frequency band k in a frame neighborhood τ, and $\boldsymbol{X}_{\Sigma}$ a reference noise spectrum. The complete derivation can be found in [30]. Intuitively, a higher noise score means that the frequency bin differs more from the noise reference. If a frame is classified as “speech”, then the noise score is used to analyze the frequency spectrum.

As mentioned above, considering the complete signal spectrum is not practical. However, a simple bandpass filter approach as in Hoshiba *et al.* [29] omits a lot of useful information in the case of human speech. Therefore, we use the noise score ν to extract the m bands with the highest score. This removes frequencies from the computation that do not contribute to the source signal. We show the selected bins from each algorithm in Figure 4. These bins are then fed into the SEVD-MUSIC estimator.

B. SEVD-MUSIC

First, we derive the details to estimate the Direction of Arrival (DoA) for acoustic signals. We model our sound source as a point that emits a sinusoidal wave with center frequency f_k and corresponding time-dependent amplitude $\lambda_k(t)$, where k is the index of one out of K frequency bands. Using the complex frequency notation we have

$$s(t) = \lambda_k(t)e^{j2\pi f_k t} = \lambda_k(t)e^{j\omega_k t} \ . \tag{2}$$

We consider a sensor array that consists of N microphones, thus we obtain the system equation

$$\begin{bmatrix} 1 \\ e^{-jw_k\Delta_1} \\ \vdots \\ e^{-jw_k\Delta_{N-1}} \end{bmatrix} s(t) =: \boldsymbol{a}_k s(t) \ , \tag{3}$$

where Δ_n is the relative propagation delay with respect to the nth reference microphone. For a one-dimensional linear microphone array and under the assumption of planar waves, the delay is calculated as

$$\Delta_n = \frac{d_n \sin(\theta)}{c_0} \ , \tag{4}$$

where d_n is the sensor's distance to the reference, θ the direction of arrival and c_0 the speed of sound, i.e. approximately $334\,m/s$ at room temperature. The vector $\boldsymbol{a}_k \in \mathbb{C}^{\mathrm{N}}$ in Equation (3) is denoted the **steering vector** for the frequency f_k. To obtain the complete **signal vector** we extend the system equation to

$$\boldsymbol{x}(t) = \boldsymbol{a}_k s(t) + \boldsymbol{n}(t) \ , \tag{5}$$

where $\boldsymbol{n}(t)$ is additional uncorrelated system noise.

When a new signal is received, we split it into smaller frames of fixed length and transform them into the frequency domain. Then, we compute the correlation matrix $\boldsymbol{R} \in \mathbb{C}^{N\times N}$ using

$$\boldsymbol{R} = \overline{\boldsymbol{X}(k)\boldsymbol{X}^{\mathrm{H}}(k)} \ , \tag{6}$$

where $\boldsymbol{X}(k) \in \mathbb{C}^{N\times F}$ contains the transformed Fourier coefficients of band k for all F frames and N microphones. Here, $\boldsymbol{X}^{\mathrm{H}}$ is the Hermitian of $\boldsymbol{X}$. Using Singular Value Decomposition (SVD) on $\boldsymbol{R}$ to separate the contained subspaces, we get

$$\mathrm{SVD}\left(\boldsymbol{R}\right) = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\mathrm{T}} \tag{7}$$

$$\begin{aligned} \boldsymbol{U} &= [\boldsymbol{u}_0 \quad \boldsymbol{u}_1 \cdots \boldsymbol{u}_{N-1}] \\ &= [\boldsymbol{U}_{\mathrm{S}} \quad \boldsymbol{U}_{\Sigma}] \ , \end{aligned} \tag{8}$$

where $\boldsymbol{U}_{\mathrm{S}}$ is the **signal space** and $\boldsymbol{U}_{\Sigma}$ the **noise space**. As the system noise is uncorrelated it is present in all subspaces. The previously defined Steering Vector $\boldsymbol{a}_k$ is a property of a receiving signal and therefore defined in the signal space. This implies

$$\boldsymbol{a}_k \in \boldsymbol{U}_{\mathrm{S}} \ , \tag{9}$$

$$\Rightarrow \boldsymbol{a}_k \perp \boldsymbol{U}_{\Sigma} \ . \tag{10}$$

Hence, the inner product (denoted as $\langle\cdot,\cdot\rangle$) of the steering vector and the noise space is zero.

Natural sound events, especially the human speech, are composed of several frequencies. To take this into account we consider the complete frequency spectrum and combine it into a single representation. A common approach for that is the broadband pseudospectrum, which is defined over all frequency bands K as

$$P(\theta) = \sum_{k=1}^{K} \frac{1}{\langle \boldsymbol{a}_k(\theta), \boldsymbol{U}_{\Sigma}\rangle^2} \ . \tag{11}$$

The DoA is found as the maximum of the estimator's response, i.e.

$$\tilde{\theta} = \mathrm{argmax} \ \ P(\theta) \ . \tag{12}$$

C. Motion Model

We check the plausibility of the received angle by evaluating it with a motion model. To do this, we assume for time span t_m that the source moves with mean angular velocity $\bar{\omega}$, i.e.

$$\bar{\omega}(t_m) = \overline{\left(\frac{\Delta\theta}{\Delta t}\right)} \approx \frac{1}{M} \sum_{n\in\mathcal{N}(t_m)} \frac{\tilde{\theta}_n - \tilde{\theta}_{n-1}}{t_n - t_{n-1}} \ , \tag{13}$$

where $\mathcal{N}(t_m)$ is the index set of all M angular measurements $\tilde{\theta}_n$ within the time span t_m. A subsequent measurement $\tilde{\theta}_{m+1}$ is considered as valid, if

$$\left|\tilde{\theta}_{m+1} - \bar{\omega}(t_m)\right| < \theta_{tol}, \tag{14}$$

with the constant motion tolerance θ_{tol}.

When receiving a new DoA from the previous steps we gather all estimations within the time span t_m. If at least two valid points are found we use our motion model to verify the new one. Otherwise we use all DoAs for the motion vector, at least three estimations are necessary. The first estimations are used to calculate $\bar{\omega}(t_m)$ and the last one to verify the model. If the motion can be explained by our model we mark all DoAs as valid estimations.

This motion model allows for filter out echo, because measurements that stem from echoes have a direction that is not consistent with the source, and they are timed shortly after the arrival of the original signal.

IV. EXPERIMENTS

A. Evaluation Data Set

To evaluate the performance of our system in different and challenging conditions, we recorded static and moving speakers in an office building. We selected six representative rooms of different type and measured the reverberation time

TABLE I: Evaluation Data Set: Measured reverberation time T_{60} and room size for six different room types.

Room	T_{60} [s]	Area [m^2]
Lab (large)	1.158	291.3
Lab (small)	1.646	101.8
Entrance Hall	3.149	211.9
Common Room	1.971	80.28
Lecture Hall	1.077	141.97
Office	0.345	24.1

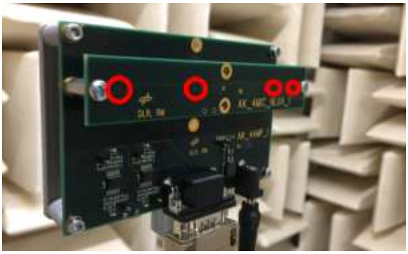


Fig. 3: The printed circuit board (PCB) with the 4 microphones (red circles). The microphones are spaced 1.5cm, 6cm and 9cm from the reference microphone on the right.

T_{60} for each. Table I lists the measured T_{60} time as well as the room sizes.

The data was recorded with a sensor array consisting of four microphones placed on a printed circuit board (PCB). A picture is shown in Figure 3. The microphones are arranged non-equally spaced over a distance of 9 cm, the positions of the microphones are marked by circles. The recording was done with a sampling rate of 16 kHz.

We created an evaluation data set with the aim to analyze different conditions where echo, reverberation and other effects degrade the localization performance. Hence, we placed the microphone array at different positions, to reflect a variety of scenarios for a robotic systems, and distances ranging from 3 m to 15 m. We took into account positions next to structures like walls or furniture, as well as placing the system in the center of the room. For example for the office room we placed the array into a corner next to two reflecting surfaces, centered in the room next to a desktop including screens and next to an open door.

B. Experiment Procedure

The recorded data sets were fed to the different sound source localization algorithms. For our experiments we compared our method (MME-MUSIC) with the well established Generalized Singular Value Decomposition based MUSIC (GSVD-MUSIC) [24], and the recently published MUSIC with Active Frequency Range Filtering (AFRF-MUSIC) [29]. We do not consider any cross-correlation-based algorithms as they use a different approach than the previous mentioned subspace-based algorithms. In addition, most methods need a significant larger amount of sensor input for enabling the same theoretical accuracy [31]. The used parameter set is given in Table II. We constrained all methods to a frequency band between 1 kHz and 8 kHz to remove low frequent system noise and focus on human speech. For each estimation a total frame of length n_{Total} was sliced into smaller frames of n_{FFT} points which are shifted by n_{Step}. For the number of bins n_{Bins} the improved MUSIC methods shall process we took 100 as it showed to be a good trade-off for accuracy, processing time and estimation miss-matches. For our motion model we used a motion time t_{motion} of 0.5 s and a tolerated motion deviations θ_{tol} of $4.5°$. Both have been determined empirically for static and dynamic sources.

TABLE II: Parameter constraints for experimental evaluation. If a parameter is applicable is indicated by a ✓.

Parameter	Value	GSVD	AFRF	MME
ω_L [Hz]	1000	✓	✓	✓
ω_H [Hz]	8000	✓	✓	✓
n_{FFT}	1024	✓	✓	✓
n_{Step}	64	✓	✓	✓
n_{Total}	$4 \cdot n_{\mathrm{FFT}}$	✓	✓	✓
n_{Bins}	100		✓	✓
t_{motion} [s]	0.5			✓
θ_{tol} [Deg]	4.5			✓

We examine only true speech phase of each recording. Miss-classification of the VAD are not considered. For all positions we define a tolerated corridor of $\pm 2.5°$ around the ground truth to classify an estimation as successful or miss. Ground truth was obtained by measuring the angles of placed markers and positioning the speakers on them. $2.5°$ correspond to a deviation of approx. 20cm at a distance of 5m. This is a sufficient accuracy to recognize a speaker within a group of people standing next to each other.

In addition we evaluated the performance of our frequency band selection based on the noise score. Our goal was to reduce the computational cost which are introduced by each estimation step of the MUSIC response for each frequency band. Furthermore we wanted to use the wide spectrum of the human voice to be represented in our selection. In Figure 4 we show the selection of each algorithm for a received frame, from left to right GSVD, AFRF and MME. As previously described the selection is limited to the range from 1 kHz to 8 kHz for all methods.

As GSVD does not use a filtering technique to reduce the amount of frequency bands, it uses every received bin and feeds it to the estimator. AFRF focuses on the bin with the highest fourier coefficient corresponding to the primary frequency contributing the signal. The bandpass tremendously reduces the amount of calculations in subsequent steps, however as seen in the figure it is only using a small and limited portion of the signal. In contrast, the selection of MME is as wide as in the GSVD approach, but the amount of used bins is the same as in AFRF. The figure illustrates how the selection is gathering bins around the main frequencies in the signal while omitting frequencies which do not contribute.

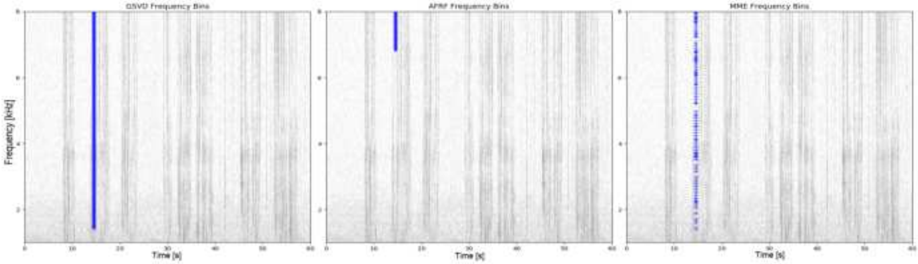


Fig. 4: Selected frequency bins for each algorithm. Each blue mark represents a selected frequency bin based on the algorithm's selection strategy. Left-hand side shows the GSVD approach, center the bandpass of AFRF and right-hand side the selected frequencies based on the noise score for MME.

TABLE III: Experimental results. The first columns present the total number of estimated DoA for each room, the last ones the rate of successful estimations.

	n_{Total}			success rate		
Room	GSVD	AFRF	MME	GSVD	AFRF	MME
Lecture Hall	263	263	229	0.91	0.79	**0.95**
Common Room	77	77	69	0.82	0.78	**0.91**
Entrance	78	78	39	0.72	0.46	**0.95**
Office	98	98	57	0.55	0.46	**0.74**
Lab (large)	73	73	49	0.78	0.64	**0.82**
Lab (small)	52	52	24	0.58	0.48	**0.88**

C. *Experimental Results*

The comparison of GSVD-MUSIC, AFRF-MUSIC and MME-MUSIC for all rooms is summarized in Table III. It shows that our MME-MUSIC approach outperforms GSVD-MUSIC and AFRF-MUSIC in all experiment.

We want to discuss the results of the experimental evaluation exemplary on the lecture and entrance hall. The first one represents an environment with average acoustics, the latter one illustrates the worst case scenario with huge reverberation time T_{60} and numerous reflecting surfaces. The DoA estimation over time of each algorithm is displayed in Figure 5 and 6. A corresponding image of the environment is shown on the left-hand side of each. In Figure 7 we show exemplary one estimation result with corresponding ground truth and tolerated corridor. For better readability of the figures we skipped them for the rest of the evaluation.

In the lecture hall the GSVD-MUSIC algorithm has a good performance with overall 91% successful estimations. However it has some outliers which are created by echo of dominate sounds which can be seen at $t = 10$s, $t = 15$s and $t = 22$s.

The AFRF-MUSIC algorithm is actively filtering for the main frequency in the current frame. This makes it faster than the standard GSVD approach and robust against other sources of noise, nevertheless it fails if the main frequency in the frame is not part of the source. Again at the endings of words, when the echo is dominant, AFRF-MUSIC solely focus on the frequencies which are created by the shadow source and neglects the frequencies of the original source. This yields to only 79% successful estimations.

MME-MUSIC introduces a motion model which checks if the estimated DoA is coherent to previous estimations. By that the algorithm removes outliers which were created from echo during the speech phases. The model not only considers static sources but also dynamic ones as the moving speaker at $t = 55$s. However the total amount of estimations is less compared to the other approaches. Despite that the rate of successful estimations is 95%.

The entrance hall is a more challenging environment for the algorithms as it has a high reverberation time and consists of a lot of reflecting surfaces. This is seen in the success rate of GSVD-MUSIC and AFRF-MUSIC with 72% and 46% respectively. In contrast MME-MUSIC has with 39 successful measurements a rate of 95%.

The results of these examples are consistent over the complete dataset.

Comparing execution times GSVD-MUSIC takes on average 1.049 s, AFRF-MUSIC 0.158 s and MME-MUSIC 0.208 s. This is a speed up of 5.1x of MME-MUSIC compared to GSVD-MUSIC. We think that the slower execution compared to AFRF is caused by cache-misses and we expect to remove this gap by optimizing the code for that.

V. CONCLUSION AND FUTURE WORK

In this paper we proposed our new MME-MUSIC approach for sound source localization in reverberant environments and under echoic conditions. We presented an intelligent way to select frequencies for the DoA estimation based on SEVD-MUSIC. We exploit the frequency evaluation of our VAD system and use the information to tighten the estimation process to the source bands. The results of the estimator are evaluated by our novel motion model. This takes into account the current motion of the speaker and is able to deal with static and dynamic sources.

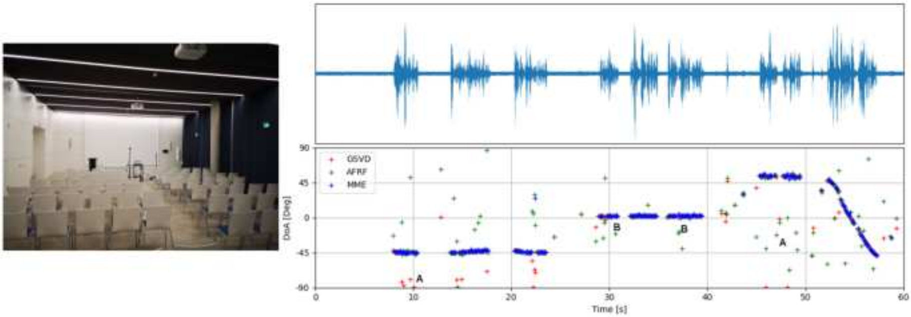


Fig. 5: Results of a recording in the lecture hall. We marked falsely estimations caused by echo with **A** and by reverberation with **B**. It can be clearly seen that MME is working better in this challenging scenario. Especially AFRF has miss-estimations at the silent endings of words.

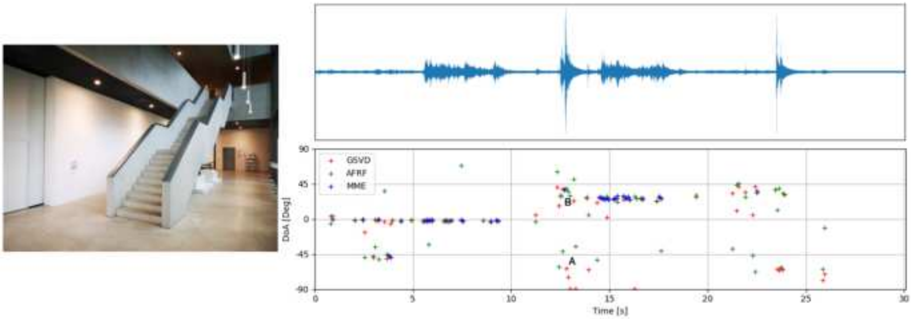


Fig. 6: Results of a recording in the entrance hall. We marked falsely estimations caused by echo with **A** and by reverberation with **B**. Because of the long reverberation time of more than 3.149s all methods have problems locating the source. At approx. $\mathrm{t} = 13s$ a loud sound event first creates inaccurate estimations caused by reverberation, afterwards the receiving echo introduces a shadow source which confuses GSVD and AFRF.

We evaluated our approach using a four channel microphone array. We showed that our system performs well in realistic scenarios with reverberations and echo. Our MME-MUSIC approach outperforms established and state-of-the-art algorithms in these scenarios while preserving real-time execution times.

In total we expect to enable robot audition as a usable and useful technology for robotic systems by our enhancements. We plan to investigate further aspects of our work. First the use of the motion model directly in the estimation process. We believe constraining the estimator towards valid positions enhances accuracy while further reducing processing time. Second we want to extend our system to handle multiple sources at the same time. This makes it possible to use robot audition for mapping tasks or in highly complex scenarios like crowds.

For future work, we will integrate the sound source localization on our humanoid robot system Rollin' Justin. We designed a microphone array which is integrated in the head of the system [32]. Here we will have to tackle further challenges, like compensating robot intrinsic noise and extend our system to a more complex array geometry due to design limitations of the robot system. We will use our technique to robustly detect speakers in a conversation.This can be used to enhance the acceptance of a robot as the system acknowledges the speaker by turning the head towards him or to annotate received speech to a specific speaker.

REFERENCES

[1] K. Nakamura, K. Nakadai, F. Asano, Y. Hasegawa, and H. Tsujino, "Intelligent sound source localization for dynamic environments," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2009, p. 664–669.

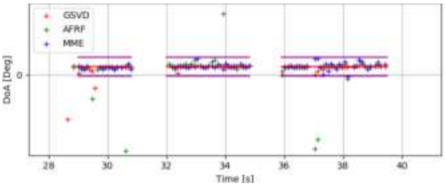


Fig. 7: Estimation results for the lecture hall dataset. The lines show the ground truth and the tolerated deviation (±2.5°). Miss-classifications of the VAD system without ground truth data are removed.

[2] K. Nakadai, K.-i. Hidai, H. Mizoguchi, H. Okuno, and H. Kitano, “Real-time auditory and visual multiple-object tracking for humanoids,” in *Artificial Intelligence. Proceedings. 17th International Joint Conference on*, 2001, pp. 1425–1432.

[3] K. Nakadai, D. Matsuura, H. G. Okuno, and H. Kitano, “Applying scattering theory to robot audition system: Robust sound source localization and extraction,” in *Intelligent Robots and Systems. Proceedings. IEEE/RSJ International Conference on*, vol. 2, 2003, pp. 1147–1152.

[4] J. Huang, N. Ohnishi, and N. Sugie, “Building ears for robots: sound localization and separation,” *Artificial Life and Robotics*, vol. 1, no. 4, pp. 157–163, 1997.

[5] L. A. Jeffress, “A place theory of sound localization.” *Journal of Comparative and Physiological Psychology*, vol. 41, no. 1, p. 35, 1948.

[6] J. A. MacDonald, “A localization algorithm based on head-related transfer functions,” *The Journal of the Acoustical Society of America*, vol. 123, no. 6, pp. 4290–4296, 2008.

[7] F. Keyrouz, Y. Naous, and K. Diepold, “A new method for binaural 3-D localization based on HRTFs,” in *Acoustics, Speech and Signal Processing (ICASSP). Proceedings. IEEE International Conference on*, vol. 5, 2006.

[8] I. Kossyk, M. Neumann, and Z.-C. Marton, “Binaural bearing only tracking of stationary sound sources in reverberant environment,” in *Humanoid Robots (Humanoids), IEEE-RAS 15th International Conference on*. IEEE, 2015, pp. 53–60.

[9] J.-M. Valin, F. Michaud, J. Rouat, and D. Létourneau, “Robust sound source localization using a microphone array on a mobile robot,” in *Intelligent Robots and Systems. Proceedings. IEEE/RSJ International Conference on*, vol. 2. IEEE, 2003, pp. 1228–1233.

[10] J.-M. Valin, F. Michaud, B. Hadjou, and J. Rouat, “Localization of simultaneous moving sound sources for mobile robot using a frequency-domain steered beamformer approach,” in *Robotics and Automation. Proceedings. IEEE International Conference on*, vol. 1. IEEE, 2004, pp. 1033–1038.

[11] Y. Sasaki, S. Kagami, and H. Mizoguchi, “Multiple sound source mapping for a mobile robot by self-motion triangulation,” in *Intelligent Robots and Systems. IEEE/RSJ International Conference on*. IEEE, 2006, pp. 380–385.

[12] E. Mumolo, M. Nolich, and G. Vercelli, “Algorithms for acoustic localization based on microphone array in service robotics,” *Robotics and Autonomous Systems*, vol. 42, no. 2, pp. 69–88, 2003.

[13] R. Roden, N. Moritz, S. Gerlach, S. Weinzierl, and S. Goetze, “On sound source localization of speech signals using deep neural networks,” in *Deutsche Jahrestagung für Akustik (DAGA)*, 2015, pp. 1510–1513.

[14] S. Adavanne, A. Politis, and T. Virtanen, “Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network,” in *26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 1462–1466.

[15] X. Xiao, S. Zhao, X. Zhong, D. L. Jones, E. S. Chng, and H. Li, “A learning-based approach to direction of arrival estimation in noisy and reverberant environments,” in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*. IEEE, 2015, pp. 2814–2818.

[16] R. Takeda and K. Komatani, “Discriminative multiple sound source localization based on deep neural networks using independent location model,” in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 603–609.

[17] Y. Sasaki, N. Hatao, K. Yoshii, and S. Kagami, “Nested igmm recognition and multiple hypothesis tracking of moving sound sources for mobile robot audition,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3930–3936.

[18] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[19] R. Roy and T. Kailath, “ESPRIT – estimation of signal parameters via rotational invariance techniques,” *Acoustics, Speech, and Signal Processing. IEEE Transactions on*, vol. 37, no. 7, pp. 984–995, 1989.

[20] S. Argentieri and P. Danes, “Broadband variations of the music high-resolution method for sound source localization in robotics,” in *Intelligent Robots and Systems. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 2009–2014.

[21] F. Asono, H. Asoh, and T. Matsui, “Sound source localization and signal separation for office robot” jijo-2”,” in *Multisensor Fusion and Integration for Intelligent Systems. Proceedings. IEEE/SICE/RSJ International Conference on*. IEEE, 1999, pp. 243–248.

[22] C. T. Ishi, O. Chatot, H. Ishiguro, and N. Hagita, “Evaluation of a MUSIC-based real-time sound localization of multiple sound sources in real noisy environments,” in *Intelligent Robots and Systems. IEEE/RSJ International Conference on*. Institute of Electrical and Electronics Engineers, 2009, pp. 2027–2032.

[23] K. Nakamura, K. Nakadai, and G. Ince, “Real-time super-resolution sound source localization for robots,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 694–699.

[24] K. Nakamura, K. Nakadai, and G. Ince, “Real-time super-resolution sound source localization for robots,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, p. 694–699.

[25] K. Okutani, T. Yoshida, K. Nakamura, and K. Nakadai, “Outdoor auditory scene analysis using a moving microphone array embedded in a quadrocopter,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, p. 3288–3293.

[26] T. Ohata, K. Nakamura, T. Mizumoto, T. Taiki, and K. Nakadai, “Improvement in outdoor sound source detection using a quadrotor-embedded microphone array,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep 2014, p. 1902–1907.

[27] G. Chardon, “A block-sparse music algorithm for the localization and the identification of directive sources,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, p. 3953–3957.

[28] R. Takeda and K. Komatani, “Noise-robust music-based sound source localization using steering vector transformation for small humanoids,” *Journal of Robotics and Mechatronics*, vol. 29, no. 1, p. 26–36, Feb 2017.

[29] K. Hoshiba, K. Nakadai, M. Kumon, and H. G. Okuno, “Assessment of music-based noise-robust sound source localization with active frequency range filtering,” *Journal of Robotics and Mechatronics*, vol. 30, no. 3, p. 426–435, 2018.

[30] J. Ramırez, J. C. Segura, C. Benıtez, A. De La Torre, and A. Rubio, “Efficient voice activity detection algorithms using long-term speech information,” *Speech communication*, vol. 42, no. 3-4, pp. 271–287, 2004.

[31] A. Pourmohammad and S. M. Ahadi, “N-dimensional n-microphone sound source localization,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, p. 27, 2013.

[32] M. Sewtz, T. Bodenmüller, and R. Triebel, “Design of a microphone array for rollin’ justin,” 2019.

4. Robust Approaches for Localization on Multi-Camera Systems in Dynamic Environments

Authors:

Marco Sewtz, Xiaozhou Luo, Johannes Landgraf, Tim Bodenmüller and Rudolph Triebel

Conference:

Sewtz, Marco, et al. "Robust approaches for localization on multi-camera systems in dynamic environments." 2021 7th International Conference on Automation, Robotics and Applications (ICARA). IEEE, 2021.

Award:

Best Oral Presentation Award

Abstract:

Localization of humanoid robots in real-life scenarios has to robustly tackle dynamic environments and provide coherent data and tight integration for follow-up tasks. However state-of-the-art solutions, like ORBSlam2, lack this ability.

In this work we present two adaptations of ORBSlam2 for a multi-camera setup on the DLR Rollin' Justin System, one distributed multi-slam and one combined single-process system. Further, we introduce the usage of pre-recorded maps with ORBSlam2 and the alignment with semantic maps for planning.

We compare performance of the adaptations against and the original approach in realistic experiments and discuss advantages and disadvantages of all methods.

Contributions:

The author of this dissertation designed the multi-sensor SLAM approaches and integrated the map saving and loading functionality. The author designed and executed the evaluation. The distributed implementation was supported by Xiaozhou Luo, the integrated implementation by Johannes Landgraf. The script was provided by the author and the publication was presented by the author.

Copyright:

©2021 IEEE

Robust Approaches for Localization on Multi-Camera Systems in Dynamic Environments

1st Marco Sewtz
Institute for Robotics and Mechatronics
German Aerospace Center (DLR)
Wessling, Germany
marco.sewtz@dlr.de

2nd Xiaozhou Luo
Institute for Robotics and Mechatronics
German Aerospace Center (DLR)
Wessling, Germany

3rd Johannes Landgraf
Institute for Robotics and Mechatronics
German Aerospace Center (DLR)
Wessling, Germany

4th Tim Bodenmüller
Institute for Robotics and Mechatronics
German Aerospace Center (DLR)
Wessling, Germany
tim.bodenmueller@dlr.de

5th Rudolph Triebel
Institute for Robotics and Mechatronics
German Aerospace Center (DLR)
Wessling, Germany
Technical University of Munich (TUM)
Munich, Germany
rudolph.triebel@dlr.de

***Abstract*—Localization of humanoid robots in real-life scenarios has to robustly tackle dynamic environments and provide coherent data and tight integration for follow-up tasks. However state-of-the-art solutions, like ORBSlam2 [1], lack this ability.**

In this work we present two adaptations of ORBSlam2 for a multi-camera setup on the DLR Rollin' Justin System, one distributed multi-slam and one combined single-process system. Further, we introduce the usage of pre-recorded maps with ORBSlam2 and the alignment with semantic maps for planning.

We compare performance of the adaptations against and the original approach in realistic experiments and discuss advantages and disadvantages of all methods.

***Index Terms*—SLAM, multi-camera SLAM, localization, mapping, ORBSlam, dynamic environments**

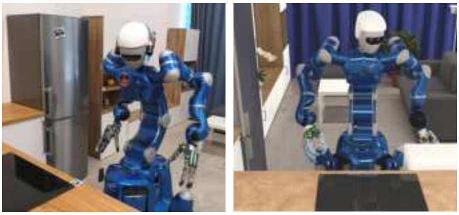


Fig. 1: DLR Rollin' Justin in kitchen scenario: real scene (left) and rendered planning map (right).

I. INTRODUCTION

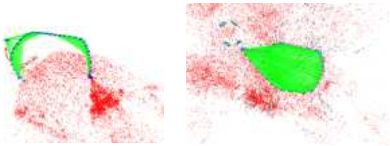
The use of humanoid robots as smart assistant in real-life scenarios like elderly care or housekeeping is still a challenging task. It not only requires robust methods for planning, perception, navigation and manipulation, but also the interaction between modules, hardware limitations and environmental issues have to be taken into account. Whole-robot and multi-task planning of robot actions requires a map with semantic object knowledge of larger environmental parts, e.g. the kitchen unit, as shown in Figure 1. However, localization and mapping (SLAM) systems usually create their own geometrical maps, not matching the planning system. Further, they usually do not re-use created maps but start over on a system restart. Complex scenarios also involve changing environments and dynamics, including humans, aggravating the localization.

Typical used perception sensors are RGB-D cameras witch usually have a finite working range. Manipulation tasks require proximity to objects or other structures that is often beyond the minimum range. Navigation tasks often move into open space, e.g. a floor, pointing the sensor into areas beyond maximum range. Both likely causes localization loss or wrong localization estimates. Multiple cameras, facing different directions, increase the visible environment, as shown in Figure 2, and thus are more robust to sensor limitations and scene dynamics.

In this work we present two different adaptations of the well-known ORBSlam2 [1] for a multi-camera setup on the DLR Rollin' Justin System, tackling sensor limitations and scene dynamics. The first is a multi-ORBSlam solution which estimates the pose per camera and finally fuse them into a single pose. The second one modifies the ORBSlam by combining the feature-maps of the cameras before the tracking step and estimating a combined pose. Additionally we modified ORBSlam to use a pre-acquired map. We present details on obtaining the static map and alignment with the planning map. Summarizing, our work has the following contributions:

- Two adaptations of ORBSlam2 for multiple cameras: Multi-ORBSlam fusion and multi-camera ORBSlam
- Initial map integration
- Comparison of adaptations and original ORBSlam2

This work is partly funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy project "SMiLE2gether" (LABAY102).



(a) Single camera (b) Multiple cameras

Fig. 2: Comparison of the same trajectory, left-hand side captured with a single camera, right-hand side with five cameras. The observed area is limited to one direction form the system. On the other side, the multi-camera setup is able to capture more feature points. In (a) the tracking is lost after half of the trajectory, which was prevented by the multi-camera setup in (b) as a previously visited area is visible in another camera view.

II. RELATED WORK

In recent times, graph-based simultaneous localization and mapping systems got a lot of interest in the robotics community. One major approach is mono-cam ORBSlam [2] proposed by *Mur-Artal et al.* It uses ORB features [3] for tracking and mapping and optimizes them in the g2o framework [4]. They extended their approach in ORBSlam2 [1] to also incorporate stereo and RGB-D images. And more recently ORBSlam3 [5] which features IMU integration.

Approaches for exploiting more than one visual sensor for localization were proposed by *Zou et al.* [6] and *Heng et al.* [7]. An arbitrary number of cameras and no necessary overlap is introduced by *Urban et al.* with MultiColSLAM [8].

To tackle the problem which is induced by the static-world assumption, typically outlier rejection is applied by statistically methods like RANSAC. In addition, *Tan et al.* propose a change detection comparing obtained maps to identify changes in the world and respond on it [9]. To avoid direct comparisons, approaches using detection of movement in the scene were heavily studied. These include 3D object tracker as proposed by *Wangsiripitak et al.* [10] or neural-network based semantics detection [11]. Movement detection has been extended already to the multi-camera setup by *Zou et al.*.

To use the a-priori information of the environment, *Wahl et al.* propose using static maps [12]. To extend this to ORBSlam, *Nobis et al.* implement a saving and loading mechanism to ORBSlam to reuse previously obtained maps [13]. However, they loose the ability to edit the map.

In this work, we propose a system capable of recording, editing and reusing static maps of the environment and utilizing them in typical applications. We extend ORBSlam2 to multi-camera setups and integrate modules to avoid typical problems of modern SLAM systems in indoor scenarios and human-robot-collaboration. With this, we are able to easily localize our systems in given environments and use the information of the estimation process for high-level tasks as navigation, task planing or object manipulation.

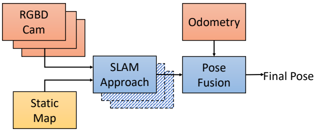


Fig. 3: Architecture of the multi-camera SLAM system. The image streams as well as a pre-recorded static map are fed to the SLAM approaches. Afterwards, they are fused with the wheel odometry and the final pose is obtained.

III. MULTI-CAMERA ORBSLAM

We use ORBSlam2 [1] as our core mapping and localization framework. It offers a robust solution for SLAM in static environments.

We propose two different approaches for estimating the pose of the robot in multi-camera setups which can be integrated into our localization framework. An overview of the architecture is given in Figure 3

A. *Multi Node Approach*

In this approach, we separately run a dedicated ORBSlam node on each camera. Each node uses the previously obtained map and estimates the robot's pose within the given static map. All estimates are fused together with the wheel odometry in single node by using a standard Discrete Kalman Filter on the state vector $\hat{\mathbf{x}} = \left(\mathbf{P}, \dot{\mathbf{P}}, \ddot{\mathbf{P}}\right)$ where $\mathbf{P}$ is the 6 DoF pose. As we lack the ability to calculate the covariance of the current SLAM estimate, we use a worst case assumption of $10\mathrm{cm}$ for the position and 2° for the orientation. Finally we obtain a pose based on all available SLAM nodes and odometry.

One major advantage of this approach is that we do not need synchronized sensors. The images may be captured at different points in time while still be able to be fused into the final pose. This reduced engineering overhead on the sensor setup as synchronization and jitter compensation normally induce a lot of effort.

Additionally, the system can deal with varying number of nodes. While running, nodes may be connected or disconnected on-demand, whether for performance reasons or because one sensor is blocked in his view.

Exploiting the fact that this system runs with different processes, we can outsource the estimation process to other processing nodes within the system to reduce CPU load on the main computer. This optimally utilizes the processing architecture and resources.

B. *Integrated Approach*

Furthermore, we developed an integrated solution fusing all camera information before the local mapping step of the SLAM system. Our main contributions are a new frame

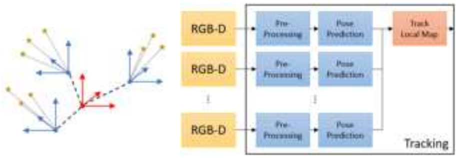


Fig. 4: Multi-frame tracking system of the integrated SLAM approach. We detect feature points in each camera frame (blue) in parallel, estimate the pose and register them to the local robot frame (red) for tracking of the local map.

system for holding the geometric information and multi-view adaptations of the pose estimation modules.

We extended the original ORBSlam tracking system to support a multi-camera setup. We estimate suitable feature points in each camera frame and track them separately. Detected features are used for pose estimation. Afterwards, we select keypoints for mapping and register them to the robot frame. We introduced a new frame system which consists of the robots ego-pose and the relative transformations of each camera to the robot's origin. The size of this structure is dynamical, it can may integrate one camera or multiple per frame. The process is depicted in Figure 4. However, this approach induces a new assumption on the recording: All images have to be captured synchronously. Especially in the case of non-overlapping camera views, a small time difference between the acquisition can led to deviation in the tracking and mapping of keyframes and reduce map quality.

For keyframe selection we additionally track if the seen area is already mapped. A previously visited scene which was mapped by another camera is not generating an additional keyframe. Nevertheless, it is still checked if the frame at hand has significant new keypoints which were not observed before. If this is the case, the above mentioned criteria is overruled.

Another extension had to be implemented in the optimization backend of ORBSlam. While for single-camera SLAM the robot's and the camera's origin are identical, in the case of arbitrary cameras the transformation from robot pose to camera pose has to be included in the graph as well. Otherwise a projection from 3D world coordinates to 2D camera coordinates would not be possible and global bundle adjustment will fail.

Relocalization and pose estimation have to be extended to a general multi-view reconstruction, also known as the generalized-camera exterior orientation problem. We adapt the minimal solution gP3P algorithm proposed by *Kneip et al.* [14]. In this approach we exploit the fact that the same feature might be seen from multiple cameras at once.

While this approach offers optimization of the mapping over all cameras and poses, it is no longer separable into smaller processes for a distributed processing architecture.

IV. APPLICATION INTEGRATION

Standard SLAM approaches described in the literature neglect registration of the map to application domain. However, manipulation in complex scenarios often require to tightly couple action planing and navigation, e.g. when opening a cabinet, the system might have to dodge the swinging door.

As a first step, we prepare the environment for recording of the static map. We remove all objects which often change their position in the world. This includes cups, pens, chairs and other objects. Afterwards we start an ORBSlam node for mapping. We use a handhold Intel Realsense D435 connected to a laptop for this task. By this we are able to easily handle the system and generate a dense map which highly covers the area of operation.

To have the ability to reuse previously recorded maps, we had to contribute saving and loading functionality to the core algorithm. This includes serialization of

- all keyframes
- all keypoints
- the essential graph
- the covisibility graph

Additionally, we save all keypoints to a pointcloud with a *valid/not-valid* flag. This way we are able to edit the generated map in an external software like Blender [15]. We are now able to remove objects from the map, which may change during the time of operation, but are hard to disassemble, e.g. computer screens or consumer electronics, or miss-interpreted map points which were caused by reflections, e.g. on mirrors. Disabled keypoints will stay in the map as they may give a hint for global relocalization, however they are not considered for local pose estimation anymore.

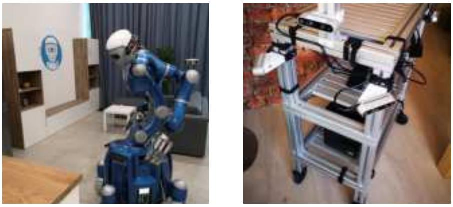
As by definition, the first pose of ORBSlam will be positioned at the origin of the map and has an orientation of zero. However, this normally does not coincide with the origin and orientation of the map used for mission planing. To transform the coordinates of the SLAM system into application coordinates, we have to find out the *Map2World* transformation.

We place several AprilTags [16] in the environment and measure their position in world coordinates. AprilTags offer a comfortable way of measuring highly-accurate position and orientation in 3D space. Nevertheless, we are only using the position as it is more easy to measure than the orientation of an object. Afterwards, we traverse multiple trajectories through the environment and estimate all detected tags in SLAM coordinates. Subsequently, we use the known world coordinates of the tags to optimize the *Map2World* transformation until the error between SLAM and world coordinates is minimized.

V. EXPERIMENTS

We conducted several experiments on different robotic platforms to evaluate the results of our approaches.

For our applications we use the mobile platform Rollin' Justin (Figure 5a). It is a wheel-based humanoid robot used in assisting tasks for human space exploration as well as elderly care. It is equipped with four Intel Realsense D435 RGB-D



(a) Rollin' Justin (b) Mock-up

Fig. 5: Systems used for evaluation. Left-hand side depicts Rollin' Justin, the system used for missions, right-hand side the mock-up.

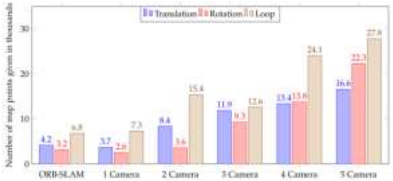
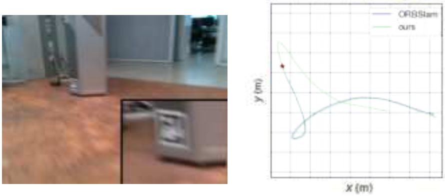


Fig. 6: Comparison of trackable map points over the number of included camera views. We included 3 different trajectories for comparison: One sole-translational, sole-rotational and one combined with loop-closure. As it can be seen, the number of used features for tracking increases with the number of views available for each scenario. The vanilla version is depicted on the left side.



(a) Motion Blur

(b) Tracking

Fig. 7: Loss of Tracking during fast movements. The original ORBSlam approach looses track ($\star$) and is not able to recover while our approach still keeps track of the robots motion. On the left-hand side, the image containing the motion blur. We highlighted one part of the image to illustrate the impact of motion blur.

cameras on its base and an additional auxiliary camera of the same type in its head. All sensors are connected to Nvidia Jetson TX2 boards for preprocessing and then send to the main computer.

Further we use a smaller mock-up (Figure 5b) of the platform for evaluation in environments which are inaccessible for the previously described system. The system is equipped with five rigid mounted Intel Realsense D435 RGB-D cameras and has to be moved manually.

A. *Mapping Capability*

One major advantage of using multiple cameras over a single camera on one system is that the number of observable map points in one frame drastically increases. To show the impact on the generated map by the SLAM system we executed three different trajectories on Rollin' Justin and compared the number of inserted map points. The trajectories consist of a translation along one axis, a single rotation and a more complex one combining both and also featuring a loop-closure. The results can be seen in Figure 6. The number of points in the original and the adapted approach for a single camera are comparable. Small deviations are due to different implementation of the integrated modules. Whereas the multi-camera approaches show significantly more tracked map points per trajectory.

This also reflects in the number of mapped area and is visualized in Figure 2. It can be seen that the approach with multiple cameras generates a more dense and wider feature map than the single camera approach on a normal trajectory. This is explained by the increased Field of View (FoV). For the Rollin' Justin system we can reach an observable area of 276° or 76% of the surroundings, for the mock-up 249° or 69%.

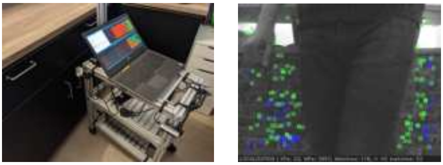
B. *Loss of Tracking*

A typical problem when dealing with SLAM systems is Loss of Tracking. This can happen during fast movements, particularly rotations, where the view changes significantly between two consecutive frames. Additionally, fast movements are accompanied with motion blur which hamper detection of features for tracking. With increasing velocity, the number of trackable points per frame reduces. Falling below the threshold of valid point matches between, the tracking is lost. However, having more camera views available for feature matching increases the total number of trackable points and prevents the Loss of Tracking event.

In Figure 7a we show an example of motion blur effecting the tracking. The robot is moving at $1.1\mathrm{m/s}$. While the original approach loses track of the movement and is not able to recover during the remaining trajectory, our approach is able to keep track.

Further, RGB-D cameras have the problem of requiring a minimum distance and enough texture for detecting depth. This interferes especially in scenarios where the robot has to move close to a rigid object, e.g. a kitchen counter (Figure 8a). We tested our approaches with a single-camera as well as full setup and were able to maintain tracking in the latter configuration.

Another common cause for Loss of Tracking is a person interacting with the system and obscuring the vision system.



(a) (b)

Fig. 8: Typical scenarios where single-camera systems fail. On the left-hand side a close approach to an object. Features can not be detected because of the short distance. On the right-hand side a person approaching the system and concealing the scene.

TABLE I: Comparison of the accuracy evaluation of each approach. We show the absolute pose error RMSE per trajectory.

	Positional Error			Angular Error		
	Trans	Rot	Loop	Trans	Rot	Loop
ORBSlam	**0.15**	9.11	13.18	**0.20**	33.41	3.68
Multi-Node	2.01	4.90	5.25	0.90	29.01	3.62
Combined	1.93	**4.47**	**5.15**	0.88	**24.52**	**3.48**

In Figure 8b we show an exemplary situation where a person is approaching the robot and stopping in front of it. The close proximity is usually around some tens of centimeters. If enough feature points are still detected, the system can localize on the static map points and is not confused by the dynamic parts of the image. However, if the number of valid map points drops below a certain threshold, the tracking will be lost. With more cameras, it is more unlikely to reach this threshold.

C. Trajectory Accuracy

As a last step, we want to show the accuracy of the pose estimation of each approach. We have equipped the system with apriltags [16] for obtaining the ground truth. For each trajectory we calculate the absolute pose error which illustrates the difference between estimated pose and ground truth. To compare it across the complete trajectory, we take the root-mean-square (RMSE) of all deviations. The results are shown in table I.

It can be seen that on a sole translational trajectory our approaches are outperformed by the original ORBSlam approach. However, in more complex scenarios the multi-camera approaches show better estimations.

VI. CONCLUSION

In this work, we presented two approaches for robustly localizing a robotic system equipped with multiple cameras in realistic scenarios. We were able keep track of the robot's motion in even complex scenarios with occlusion, motion-blur and human interaction. In addition, the overall accuracy of the pose estimation increased for complex scenarios. While the simplicity of the node-based approach simplifies the integration, the integrated approach promises an improved mapping and localization process.

In addition, we also shared our efforts on how to integrate visual SLAM systems into applications. In particular on how to obtain a map of the environment, register it to the application domain and enable use of the system on the robot.

VII. FUTURE WORK

Our goal is to integrate a navigation system which is able to robustly work in dynamic environments. We introduced multi-camera setups as they add more redundancy and therefore more robustness in many situations. However, we identified that a key-functionality is the estimation of map confidence. For our future work we plan to find a solution for multi-camera systems in human environments that are able to detect changes in map and incorporate updates for semantic planing.

REFERENCES

[1] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[4] G. Grisetti, R. Kümmerle, H. Strasdat, and K. Konolige, "g2o: A general framework for (hyper) graph optimization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China*, 2011, pp. 9–13.

[5] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual-inertial and multi-map slam," 2020.

[6] D. Zou and P. Tan, "Coslam: Collaborative visual slam in dynamic environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 354–366, 2013.

[7] L. Heng, G. H. Lee, and M. Pollefeys, "Self-calibration and visual slam with a multi-camera system on a micro aerial vehicle," *Autonomous robots*, vol. 39, no. 3, pp. 259–277, 2015.

[8] S. Urban and S. Hinz, "Multicol-slam - a modular real-time multi-camera slam system," 2016.

[9] Wei Tan, Haomin Liu, Z. Dong, G. Zhang, and H. Bao, "Robust monocular slam in dynamic environments," in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2013, pp. 209–218.

[10] S. Wangsiripitak and D. W. Murray, "Avoiding moving outliers in visual slam by tracking moving objects," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 375–380.

[11] L. Riazuelo, L. Montano, and J. M. M. Montiel, "Semantic visual slam in populated environments," in *2017 European Conference on Mobile Robots (ECMR)*, 2017, pp. 1–7.

[12] S. Wahl, P. Schlumberger, R. Rojas, and M. Stämpfle, "Localization inside a populated parking garage by using particle filters with a map of the static environment," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 95–100.

[13] J. B. Felix Nobis, Odysseas Papanikolaou and M. Lienkamp, "Persistent map saving for visual localization for autonomous vehicles: An orb-slam 2 extension," in *2020 Fifteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, 2020.

[14] L. Kneip, P. Furgale, and R. Siegwart, "Using multi-camera systems in robotics: Efficient solutions to the npnp problem," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 3770–3776.

[15] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

[16] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2016.

5. URSim - A Versatile Robot Simulator for Extra-Terrestrial Exploration

Authors:

Marco Sewtz, Hannah Lehner, Yunis Fanger, Jan Eberle, Martin Wudenka, Marcus G. Müller, Tim Bodenmüller and Martin J. Schuster

Conference:

Sewtz, Marco, et al. "Ursim-a versatile robot simulator for extra-terrestrial exploration." 2022 IEEE Aerospace Conference (AERO). IEEE, 2022.

Abstract:

We present URSim, a complete Software-in-the-Loop simulation of robotic systems, specially designed to meet the needs of testing platforms for planetary exploration. By simulating the sensors of a robotic system and providing similar interfaces to the real system, URSim enables developing and continuous testing of high-level software components in various scenarios. URSim is based on the Unreal Engine 4, which offers photo-realistic visual and physics support in real-time. A generic robot interface allows the integration of different robotic systems with various sensors that are simulated in real-time. Its modern and adaptable system architectures make it possible to customize URSim for different setups, frameworks, and modules. To demonstrate URSim and its advances, we simulate different robotic systems, including the LRU, the hexacopter **ardea!** (**ardea!**) and the humanoid robot Rollin' Justin. We integrate the complete navigation and mapping pipeline of the LRU in URSim and conduct an exploration mission in a simulated Martian and Lunar environment.

Contributions:

The corresponding authorship is shared between the author of this dissertation, Hannah Lehner and Yunis Fanger. The initial idea is by Hannah Lehner. The design of the simulation architecture, the software engineering and the communication infrastructure is proposed by the author. The integration and evaluation is by Yunis Fanger. Integration is supported by Jan Eberle and Martin Wudenka. The project was managed by the author, the script provided by the author, Hannah Lehner and Yunis Fanger. The publication was presented by the author.

Copyright:

©2022 IEEE

URSim – A Versatile Robot Simulator for Extra-Terrestrial Exploration

Marco Sewtz*, Hannah Lehner*, Yunis Fanger*,
Jan Eberle, Martin Wudenka, Marcus G. Müller, Tim Bodenmüller, Martin J. Schuster

German Aerospace Center (DLR)
Institut of Robotics and Mechatronics
Muenchener Str. 20, 82234 Weßling, Germany
{Marco.Sewtz, Hannah.Lehner, Yunis.Fanger, Jan.Eberle, Martin.Wudenka,
Marcus.Mueller, Tim.Bodenmueller, Martin.Schuster}@dlr.de
*The authors contributed equally to this work.

Abstract—**We present URSim, a complete Software-in-the-Loop simulation of robotic systems, specially designed to meet the needs of testing platforms for planetary exploration. By simulating the sensors of a robotic system and providing similar interfaces to the real system, URSim enables developing and continuous testing of high-level software components in various scenarios. URSim is based on the Unreal Engine 4, which offers photo-realistic visual and physics support in real-time. A generic robot interface allows the integration of different robotic systems with various sensors that are simulated in real-time. Its modern and adaptable system architectures make it possible to customize URSim for different setups, frameworks, and modules. To demonstrate URSim and its advances, we simulate different robotic systems, including the Lightweight Rover Unit (LRU), the hexacopter ARDEA and the humanoid robot Rollin' Justin. We integrate the complete navigation and mapping pipeline of the LRU in URSim and conduct an exploration mission in a simulated Martian and Lunar environment.**

TABLE OF CONTENTS

1. INTRODUCTION 1
2. RELATED WORK 2
3. URSIM 2
4. IMPLEMENTATIONS 6
5. EVALUATION 10
6. SUMMARY & CONCLUSION 10
7. FUTURE WORK 12
8. ACKNOWLEDGMENT 12
REFERENCES 12
BIOGRAPHY 14

1. INTRODUCTION

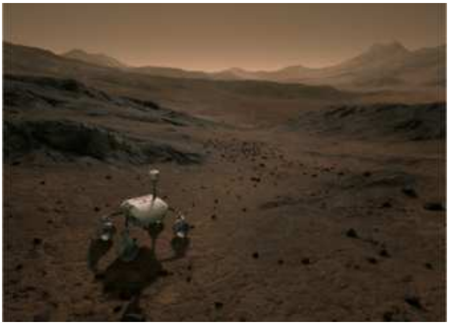


Figure 1: The Lightweight Rover Unit (LRU) setting out to explore a simulated Mars environment in URSim.

Robotic systems can withstand the prevailing harsh conditions on extra-terrestrial surfaces and accomplish challenging missions. As the communication between Earth and extra-terrestrial bodies is delayed, a robotic system has to autonomously navigate through unstructured environment, reliably localize itself and build an accurate map of the environment. To guarantee mission success it is essential to intensively test the single components, as well as the complete system during the development phase to implement reliable methods. However, testing high-level software components during the development of a robotic space exploration system is often limited. The prevailing conditions on the planetary surface cannot be reproduced on earth and field tests in analogue environments on earth are expensive and usually done at the end of the development phase. Often, limited access to the space rover hardware prevents a continuous test cycle during the development of the high-level software components, such as mapping and exploration.

Simulation tools offer a low-cost and easy possibility to extensively test a robotic system during the development phase. A simulator allows for a short integration time, a continuous test-cycle, and testing in various environments under conditions close to the real mission. We present a complete Software-in-the-Loop (SiL) simulator, called URSim, which is specially designed to meet the needs of testing robotic systems for planetary exploration. To test high-level software components for extra-terrestrial missions, it is important to simulate the robot in a photo-realistic environment with similar lightning, gravity and physics as expected on the target planetary body. URSim is based on the Unreal Engine 4 (UE4), which offers state-of-the-art photo-realistic rendering (see Figure 1) and physics support, which is required to create realistic missions scenarios. Our generic interface allows to integrate different robotic systems with various sensors. Robots and sensors are described with a descriptive language, which simplifies the integration and allows to create different setups within seconds. URSim is based on a modern and adaptable system architecture to customize the simulator

978-1-6654-3760-8/22/$31.00 ©2022 IEEE

for different setups, frameworks, and modules required for a specific exploration mission. We implemented several sensors, including visual sensors and physical sensors. For the Interprocess Communication (IPC) we provide a generic interface that allows to integrate different middle wares such as the Robot Operating System (ROS).

To demonstrate the simulation and its advances when developing software components for robotic space exploration systems, we use URSim to simulate different robotic systems in realistic mission scenarios. Figure 1 shows the Lightweight Rover Unit (LRU) [1] in a simulated Martian environment in URSim. By simulating the sensors of LRU and providing interfaces similar to the real robotic system, we can apply our complete navigation and mapping pipeline and evaluate its localization, mapping, and exploration performance with URSim. In addition, we show how to simulate our hexacopter ARDEA [2], a terrestrial prototype to develop algorithms for future space drones, as well as the humanoid robot Rollin' Justin [3], which assists astronauts in space missions.

This paper is structured in seven main sections. Section 2 provides a short overview of existing robotic simulators. Section 3 describes in detail the system architecture of URSim, as well as the generic robot generation, sensor, and IPC integration. In Section 4, we provide an overview on the currently implemented sensors and integrated robots in URSim. Finally, we show in Section 5 the results of the evaluation of our navigation and mapping pipeline with URSim. In Section 6, we give a summary of our work and in Section 7, we discuss the planned future work.

2. RELATED WORK

Access to robotic hardware operating in mission-relevant environments is typically limited and expensive. Thus, simulation as well as pre-recorded datasets play an important role for component and system evaluation and validation testing of robots for extra-terrestrial exploration. Datasets are valuable to test and evaluate specific software modules. For example, navigation sensor recordings from analogue test sites (such as the LRNT [4] Moon-analogue and the MADMAX [5] Mars-analogue datasets) can be used to evaluate robotic navigation pipelines, while annotated real images from other planets (such as the Deep Mars [6] dataset) are suitable for image classification tasks. However, in order to test and evaluate full robotic systems, simulations have become an invaluable tool as they allow perception-action control loops. Furthermore, they allow to test robotic systems in a resource-efficient manner in large varieties of different environments.

While there is a multitude of existing tools that allow Software-in-the-Loop simulations (e.g., [7], [8], [9], [8], [10], [11], [12], [13]), the focus of our extraterrestrial exploration use case are robots with vision-based navigation pipelines. Thus the quality and photo-realism of the rendering of the robots' environment with its simulated cameras is of critical importance. For this, modern game engines such as the Unreal Engine 4 [14] (used by our URSim and, e.g., by [10], [8]) or Unity Engine [15] (used, e.g., by [11]) are a suitable choice to base real-time robotics simulators on. Simulators such as OAISYS [16] or BlenderProc [17], which are based on 3D computer graphics software such as Blender [18], can deliver even more photo-realistic results. However, achieving that level of photo-realism typically is infeasible when running SiL simulations in realtime on standard hardware.

There are many simulators dealing with semi or fully structured human-made environments, such as urban scenes for autonomous driving [8], [10] or indoor scenes for factory automation and household-robotics [9], [17]. Most similar to URSim is AirSim [10], which is designed as a plugin for the UE4. However, AirSim is designed for simulating drones and for testing autonomous driving. Complex robotic systems with several movable joints cannot easily be integrated. In contrast, URSim offers an easy way to dynamically load complex robot systems specified via XML-based configuration files.

There are just a few simulators featuring fully unstructured environments, which are needed for the planetary exploration use case [19], [20], [16]. While [19], [20] focus on far-range spaceflight scenarios, [16] targets close-range surface exploration, but for passive data generation. To prepare and support recent and ongoing real surface exploration missions to, e.g., Mars, NASA/JPL employs their Multibody Modeling, Simulation and Analysis Software (DARTS) [13], which includes ROAMS, a physics-based simulator. In contrast to URSim, its primary focus is on simulating physics, e.g., wheel soil contact, and not on simulating visual sensors for testing visual navigation in photo-realistic environments.

With URSim, we can close the perception-action loop for a variety of full robot systems, targeting photo-realistic planetary environments with real-time rendering and game physics. The sensors are properly synchronized and provide the data in a sufficient frequency. This allows us to integrate and test all software components of our robots above the control level, including our entire vision-based navigation pipeline as well as higher-level modules for autonomous exploration and mission execution. The modularity of URSim enables the simulation of different types of complex robots, aerial and ground-based ones, with different sensor setups that can easily be specified via configuration files.

3. URSIM

In this section, we cover the general design and architectural philosophy of URSim. We will present the considerations for our simulation structure, project environment, and implementation decisions.

Simulation Architecture

The Unreal Robot Simulation (URSim) is a development to support the research and development on many different platforms at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR). Hence, it has to handle different robotic systems, environments and infrastructures. Differences between robots include specific controllers, individual communication and networking frameworks, design descriptions of the abstracted hardware, and much more. To support the large variety in system characteristics, we heavily work with abstraction layers within the simulation to convert between interfaces and the simulation backend. We visualize the overall architecture of the simulator in Figure 2.

URSim is split into a management, a world, and a robot module. While the management module covers start-up, scenario customization, and feature management, the world module is responsible for the environment, and the robot module for the simulation of the robotic target platform.

For each simulation run, the user is able to specify certain properties of the simulation. The user is able to specify

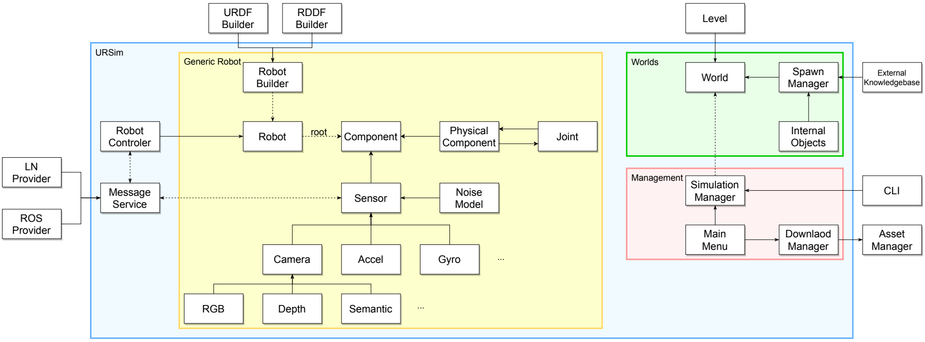


Figure 2: Architecture of the simulation framework.

the simulation environment. This includes the map and the robotic system. Additional visualization options like the screen resolution can also be selected.

Furthermore, the user is able to load additional assets like maps or robots from an external asset store on the fly. This offers a flexible way of distributing the application data across teams and organizations.

On startup of a simulation run, the simulation environment is loaded and the world and robot instances are generated.

Interprocess Communication—As mentioned before, the different robots implemented in URSim use different infrastructures for communication between process, commanding, and house-keeping. Therefore, URSim must support these same IPC frameworks in order to be used in a Software-in-the-Loop simulation. Our primarily used middlewares are the Robot Operating System (ROS) [21] and links_and_nodes (LN) [22]. However, due to its modular architecture, further IPC frameworks, such as for example the Data Distribution System (DDS) [23], can be easily integrated into URSim.

Most IPC frameworks heavily rely on specific environment setups or they may not be available on the target computer. Taking this into account, we decided to outsource the implementations of specific frameworks. To achieve this separation, we use dependency injection in combination with dynamic loading of the implementation libraries. The simulation framework offers a generic messaging service, which can be called and used by the simulation objects. This service is implemented by virtual interfaces. When a system requests access to a specific IPC framework, the messaging service will load during execution time the corresponding implementation and link the interfaces to it. Furthermore, using the injection technique, we can access multiple frameworks at the same time. We use this possibility, for example, to offer a logging and debugging interface, which outputs every message to the console. For more detailed information on the specific implementations of each IPC, we refer to Section 4.

The configuration of message streams is performed in two different ways, depending on their type. Incoming streams to URSim used for commanding the robots' movements are hardware-dependent and defined in the initial setup process of the simulation. Outgoing streams on the other hand, which transmit data gathered by virtual sensors in URSim, are able to be individually configured using a description file. This allows for fast development and testing with the simulation by changing the configuration between runs.

URSim Plugins—The philosophy of URSim is to create a generic simulation which can be easily adapted to new missions and scenarios. However at the same time, teams may want to only distribute robot models and data within a limited group. Therefore, URSim needs to be extendable and new maps or robots shall not be hardcoded directly into the software. Instead, we designed an asset manager, which is able to load and unload additional packages into the simulation.

By doing this, we are able to separate the maps and most of the robots specific implementations from the simulation's source code. Solely the controller implementation of each robot has to be compiled with the simulation code as it interacts directly with the engine's API.

We exploit the public asset manager API to support different asset stores. For our work at the institute, we have an implementation interfacing with a JFrog Artifactory instance, which is part of our internal CI/CD process. However, using the public interface, any other distribution system is possible, e.g., git, subversion or even file-based implementations. Using the asset manager, the management of URSim is able to list possible assets, download, delete, or update them. Depending on the external implementation, users and groups can restrict access and adapt the interface to their particular development and distribution processes.

User Interface—The User Interface (UI) has only a supportive role within the simulation. It is not part of the data streamed to external software The design is kept simple, however it offers configuration possibilities needed to run the simulation and alter the environment, like changing the speed of the simulation or spawning objects.

Robot Integration

One of the core components of URSim is the generic robot class, which abstracts the interfaces of the simulated hardware and generates the physical representation in the engine.

A robot within URSim is defined as a tree of components, each of which may have an arbitrary number of child components attached. There are different types of components that can be added to a robot. *Physical Components* are used to represent parts of the robot that have a physical body. They offer features like mechanical and visual models, the ability to detect collisions and physical properties such as mass and scale. *Sensors*, such as cameras or Inertial Measurement Units (IMU), are special components which offer access to the perception of the environment or the state of other components. In addition, invisible components can be created to add virtual attachment points for sensors.

Joints are connections between *Physical Components* that can add physical constraints on the attachments. By default, each joint allows translation and rotation around all six degrees of freedom (DoF). However, each DoF may be limited to a given range or disabled completely. Using these configuration options, a joint can represent all attachment methods of real robots such as hinges and couplings, but also complex connections like pulleys. Furthermore, we are able to add damping and friction components to mimic the physical behavior of real robots as closely as possible.

Once a robot has been spawned into a map, it is able to transmit received data from its sensors and react on external commands. Before this is able to happen, however, the robot needs to be created. Therefore again, we use an abstraction layer to support multiple ways of defining and creating robots in the simulation.

Robot Builder—The *Robot Builder* is an abstract module that is able to create an arbitrary robot system based on a given design description. One commonly known way of defining a robot is by means of an Unified Robot Description File (URDF). It is an XML formatted document which defines single components of the robot and their corresponding joint type to connect them. However, other types of design descriptions already exist and are being developed. These can be easily used to build robots by creating a specialized *Robot Builder* implementation within URSim, exploiting the abstraction layer.

Once requested, the *Robot Builder* will parse the given design description file and translate it into the URSim representation of the robot component structure. It will further bind sensors to their message service stream and setup noise models, if defined in the design description file.

Sensors—Sensors are one of the most complex and versatile components in URSim. They may represent access to the physical states of components like acceleration or orientation, or more complex renderings of the environment using cameras.

Sensors are a specialized form of components with no physical representation. They only provide external access to the simulated environment and can therefore be placed anywhere on the system. The *Robot Builder* is able to dynamically create sensor interfaces on demand and initialize them with the correct messaging stream. Furthermore, the sensor is able to generate noise according to a given noise model, e.g. Gaussian white noise, to artificially decrease the quality of the measurement. If enabled, the altered data will be sent over the stream instead of the original reading, which can be also be sent and used as additional ground truth information.

We focused on implementing a descriptive process for defining and configuring sensors that is easy to use and can be rapidly modified. This helps testing new sensor concepts, explore the characteristics of different setups, and offer a way of prototyping perception systems. To achieve this, the sensor infrastructure of the robot can be purely defined by the design description file.

Robot Design Description File—As mentioned before, the robots available in URSim are all described by a Robot Design Description File. In the current implementation, we use a customized version of the URDF [24], a well known format for the mechanical description of robotic system, which we extended to support sensor interfaces.

The file format describes the mechanical structure of the system using *links* and *joints*. Thereby, a link is the representation of a single, physical component of the mechanical structure and a joint is the representation of the kinematic interconnection of two components. An example of a simple link and joint structure is shown here

```
<robot name=”Justin”>

  <!-- mechanics -->

  <!-- link describing a physical component -->
  <link name=”Platform”>
     <!-- visual apperence of the link -->
    <visual>
      <origin rpy=”0 0 0” xyz=”0 0 0”/>
      <geometry>
        <mesh filename=”[...]/Meshes/platform”/>
      </geometry>
    </visual>
  </link>

  <link name=”Torso”>
    <visual>
      <origin rpy=”0 0 0” xyz=”0 0 0”/>
      <geometry>
        <mesh filename=”[...]/Meshes/torso”/>
      </geometry>
    </visual>
  </link>

  <!-- connections -->

  <!-- describing the connection property
       of two links -->
  <joint name=”torso_body_connection”
         type=”continuous”>
    <!-- links connected to each other -->
    <parent link=”Platform”/>
    <child link=”Torso”/>
    <!-- physical properties of the joint -->
    <origin rpy=”0 0 1.5708” xyz=”0 0.2 0.75”/>
    <axis xyz=”0 0 1”/>
    <dynamics damping=”0.7”/>
  </joint>

</robot>
```

Furthermore, the inertial properties of the links can be defined as well. It allows for more realistic modeling of the physical behavior of robotic systems. This additional information is encoded as follows

```
<link name=”arm”>
  <visual>
    ...
  </visual>
  <inertial>
    <mass value=3.4>
    <inertia ixx=”1” ixy=”0.0”  ixz=”0.0”
             iyy=”1” iyz=”0.0”  izz=”1”/>
  </inertial>
</link>
```

To allow the attachment of sensors to a link, we extended the original file format to support sensor interfaces. For our use case, we created an additional *sensor* tag. Within it, further attributes such as sampling frequency, resolution, sensor and noise model parameters, and the IPC to use for message streams can be defined. An exemplary listing for a RGB camera sensor is given here

```
<sensor type=”rgbcamera”>
  <!-- General sensor attributes-->

  <!-- Optional: frequency (default=1Hz)-->
  <attribute name=”frequency” value=”14”/>
  <!-- Optional: enabled (default=true)-->
  <attribute name=”enabled” value=”true”/>
  <!-- Optional: relative pose
  (default: zero relative transform
  from parent link)-->
  <pose rpy=”0 0 0” xyz=”0 -0.1 0”/>
  <!-- Optional: IPC definition
  (multiple possible)-->
  <ipc provider=”ROS”
       topic=”sensors/image”
       secondary_topic=”sensors/camera_info”/>

  <!-- Image sensor specific attributes-->

  <!-- Optional: resolution
  (default: 100 x 100 pixel)-->
  <resolution width=”640” height=”480”/>
  <!-- Optional: Field of view angle
  (default: 90 degrees)-->
  <attribute name=”fov_angle” value=”90”/>
  <!-- Optional: White noise strength
  (default: 0)-->
  <attribute name=”white_noise_strength”
             value=”0”/>
</sensor>
```

Controller—The task of the controller is to translate the input from a user device such as a keyboard into actions on the system. Typically this involves motion commands like *move-forward* or state changes like *toggle-movement-mode* if the systems supports multiple modes of motion. The controller is highly dependent on the robotic system and is therefore a custom module for each robot. However, the underlying engine UE4 needs the controller to be accessible at compile time. This means that every controller must be available in the source code and cannot be represented in the design description file.

The controller is able to listen to the input commands of the user devices and triggers the appropriate actions on the robot. It has access to all components and their corresponding states and is able to command actions like rotations and translations, continuous momentum drive or sequences of actions. It may trigger different actions depending on the current state or environment.

Out of the box, the engine supports input via mouse and keyboard and can be extended to use a game controller. Additionally, the controller can make use of incoming commands that are received via one or more of its supported IPCs. This offers a SiL behavior in already existing systems with autonomy or external remote control. A diagram illustrating the usage of URSim as a SiL simulator is shown in Figure 3.

Worlds

A world in URSim represents the physical environment in which the robot is working. The worlds, represented as levels in the engine, consist of both static and non-static components. The former group includes the terrain and several physical obstacles like walls, floors, or rocks within the environment. The latter includes objects, that are able able to move either kinematically along a trajectory or are completely dynamically by having their physics simulated using the game engine.

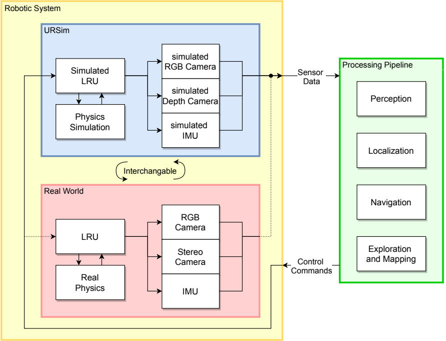


Figure 3: URSim and its usage as a Software-in-the-Loop (SiL) simulation by replacing real robots sensory measurements with virtual ones.

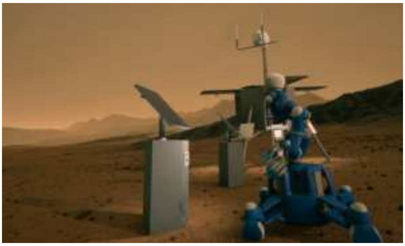


Figure 4: Simulated environment of the MARC-II and Meteron Supvis Justin mission. A solar farm on Mars is supervised and maintained by the robot Rollin' Justin.

Furthermore, we are able to spawn more objects on demand to populate the world. This offers the possibility to simulate dynamic worlds in which objects appear and disappear or change their position, similar environments with different objects.Moreover, the robot's knowledge can be used to dynamically create the estimated environment of the perception pipeline and display it in the virtual world which is useful for visualization and further exploration of the real world in simulation. An example of a mission scenario can be seen in Section 3, where the robot Rollin' Justin is illustrated performing a maintenance task on Mars.

One especially noteworthy aspect of the game engine used is its cutting edge graphical rendering pipeline. The Unreal Engine 4 is able to compute photo-realistic views of the environment in real-time. The output can be seen in Figure 5. Light is reflected on shiny surface like the floor. The irregularity of the tiles are easily identifiable. Reflections of the environment are distorted on the cabinet doors. Furthermore, lens-flares and other miss-readings in the capturing event can be modeled.

The environmental conditions of extra-terrestrial bodies like moons and planets are part of the world description. Gravity, being the main property of a environment that interacts with the locomotion and perception of robotic system, is a constant across the level. Due to the limited extent of most missions, we assume that this approximation is sufficient for most robotic developments. Atmospheric influence, especially on lighting, can be simulated using ambient lights. In the Mars environment, global lighting with no direction casting and no shadow is used in addition to the directional light of the sun. On the Lunar map, this feature is disabled to correctly simulate the missing atmosphere.

4. IMPLEMENTATIONS

In the following, we describe the robots that are implemented in URSim, the types of simulated sensors available, as well as the maps that we use for our experiments.

Robots

We simulate several of the robotic systems available in the labs at DLR in URSim. Table 1 presents a selection of our simulated robotic systems and their capabilities. In the following, we describe in detail the real robotic systems and their simulated counterparts.

LRU—The LRU is a rover prototype designed for planetary exploration. With a weight of approximately 40 kg, a length of 109 cm, and a width of 73 cm it is comparably light and compact, which is a big advantage for deployment in space. Its locomotion system consists of four wheels that are individually controlled. This allows the rover to maneuver autonomously through rough terrain in Moon- or Mars-like environments. The LRU is equipped with an IMU and a stereo camera system with a baseline of 9 cm for navigation. From the stereo camera system, dense depth images [25] are calculated and visual odometry is computed from the consecutive images. A key-frame based local reference filter [26] fuses the visual-odometry measurements with the IMU measurements for local pose estimation. Obstacles are detected in the depth images and a local costmap for path planning is generated [27]. Finally, a 6D global localization and mapping system based on supmaps [28], [29], [30] is used to generate a global 3D map. Thereby, the underlying Simultaneous Localization and Mapping (SLAM) graph is optimized by loop closures generated by matching pairs of submaps [31], [29]. The rover maps an area of interest autonomously or is directed to a point of interest. This is achieved by applying an autonomous exploration algorithm [32], [33] based on a Multi-Criteria Decision Making (MCDM) approach. The complete software stack was tested in an environment that resembles the moon on Mt. Etna, Sicily, in the ROBEX project [34]. Currently, we prepare for demonstrating a scientific exploration and sampling mission on Mt. Etna with a heterogeneous robotic team as part of the ARCHES project [35], [36].

We simulate the kinematic system of the LRU as well as its sensor setup. A local reference filter is used to fuse the simulated IMU measurements with the visual odometry, derived from the simulated depth sensor. Furthermore, the local mapping, obstacle detection, path planning, global mapping and exploration modules are the same as the ones deployed on the real robot.

ARDEA—ARDEA is a Micro Aerial Vehicle (MAV) that has been developed from the ground up at DLR [37]. The MAV was mainly developed for the autonomous exploration of unknown environments and part of our heterogeneous robotic team. ARDEA has four ultra-wide angle lens cameras which enable a 240° view in the vertical direction. Thus the system has simultaneous coverage of the ground and directly above, facilitating navigation and mapping in narrow spaces such as caves.

This also allows the MAV to have a robust Visual-Inertial Navigation system [2], fusing multiple visual odometry estimates. It uses the same navigation filter and 6D mapping components as LRU, allowing both robots to operate and exchange data as part of a heterogeneous team – so far with the real systems [36] and planned as future work with the simulated ones as well. Besides the robust navigation system, ARDEA has many more autonomy skills [38], which are important for future planetary exploration missions.

Rollin' Justin— Rollin' Justin is a wheel-based humanoid robot. The base platform is equipped with four wheels, each can be controlled and rotated individually, enabling three degrees-of-freedom in motion. Four RGB-D camera devices are mounted on the platform to perceive the environment with nearly 270° coverage of the surroundings. The visual input is accompanied by an IMU for inertial measurements and wheel encoders for odometry readings. The upper part of Rollin' Justin consists of a torso with two robotic arms attached and a head. The torso can extend and contract as well as rotate by $\pm 90^\circ$ to alter the manipulation space area of the arms. Each arm has five DoF and has a four-fingered human-like hand attached for manipulation. The head is mounted on a pan-tilt device and has a stereo-camera for perception of the manipulation space. The complete system is capable of teleoperation or might be commanded via shared-control where the operator only issues high-level commands. Furthermore, the system is able to operate fully autonomously. This has been demonstrated in several experiments with astronauts on board the International Space Station (ISS) in the frame of the Meteron Supvis Justin [39], [40], [41] mission, and in the terrestrial scenario of SMiLE [42] in the domain of elderly care. To support this, the robot makes use of a multi-camera SLAM system [43]. Based on the streams of the base cameras, a trajectory is estimated for localization of the robot. Furthermore, the system offers the possibility to load persistent maps of the environment to navigate in already explored areas and localize the robot in already given world coordinates. Semantic annotations provide the possibility to extract static elements in the area to find stable landmarks. Additionally, the camera streams are used to obtain an obstacle map to avoid collisions with the environment and humans in the vicinity.

Sensors

URSim provides the user with a suite of virtual sensors. These senors are able to measure current states of the robot and the environment at a given frequency. While every sensor has its own type of information that it measures, they all include timestamps, which reference the simulation time of triggering. Each sensor can be enabled or disabled during runtime. Sensor may trigger other sensors, creating a cascading sequence of measurements. All sensors can be attached to a robot component with additional offsets in rotation and translation. The configuration of sensors and the addition of new sensors to a robot can be easily accomplished using the RDDF introduced in Section 3. All sensor readings are relative to the component's current position. Most sensor

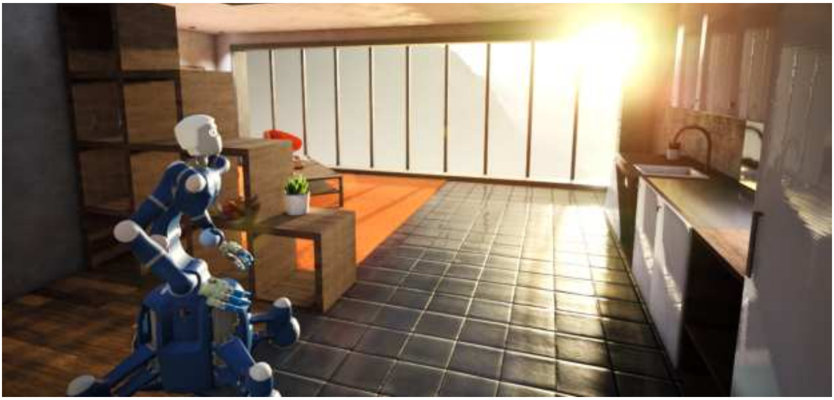
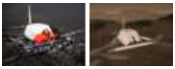


Figure 5: Indoor scene showing the graphical capabilities of the Unreal Engine 4 (UE4). Light is reflected on shiny surfaces, reflections are scattered according to surface roughness, and lights can be blinding.

Table 1: Integrated robots with their capabilities and sensor setup.

System	**Features and Capabilities**
LRU [1] 	• Simulated sensors: IMU, depth sensor, RGB sensor. • Capabilities: Autonomous exploration, mapping and navigation in rough terrain. • Features: four individually controlled wheels, pan/tilt camera head with stereo camera system.
ARDEA [2] 	• Simulated sensors: IMU, depth sensor, RGB sensor. • Capabilities: Fast autonomous exploration and mapping. • Features: High maneuverability, ultra wide-angle stereo camera setup (240° vertical/80° horizontal FoV).
Rollin' Justin [3] 	• Simulated sensors: IMU, depth sensors, RGB sensors • Capabilities: Robotic assistance and collaboration in human exploration • Features: Multi-camera perception, motion maneuvers in 3 DoF, humanoid interaction and tele-operation

types allow for configuration of a suitable noise model that is then used to generate simulated noise on the respective sensor measurements. In the following, we describe each of the currently implemented sensors in more detail.

RGB-Camera—As the first of our image sensors, we introduce the RGB-camera sensor. When triggered, it renders a colored image of the scene that is currently visible from the point of view of the robot component that it is attached to. The image resolution, defined by a certain pixel width and height in the robot's URDF file, as well as the field of view of the camera are customizable. As with all image sensors, we are able to apply Gaussian white noise to the image in a post processing step to generate more realistic data.

Depth-Camera—The second image sensor in URSim is the depth-camera sensor. It is used to generate depth information images from the scene of the environment. While in non-simulated systems, a depth image is often generated from the disparity of a stereo camera pair, we skip this step in favor of directly reading the depth information provided by the Unreal Engine. This is done to increase performance, since less processing effort is required to generate the depth image. However, a stereo camera can still be simulated by attaching two RGB-cameras with a slight offset to each other. The noise model customization options are the same as for the RGB-camera.

Semantic-Camera—The last image sensor we provide is the semantic-camera. This is a special type of camera that is able to capture semantically segmented images. An internal stencil value is provided for every object in the environment to enable the segmentation of objects in the scene. Contrary to the RGB-camera, the semantic-camera does not render the texture of each object but uses the respective stencil values instead. The value is encoded in the *red* channel of the image. Thus, we receive a labeled image with the semantic ground truth information gathered from the environment. The semantic segmentation is performed on a terrain class level. Therefore, separate object instances of the same terrain type are not distinguished between. An example use case for this sensor type is the semantic mapping presented in Section 5. In Figure 6, we show images of the same scene captured with all image sensor types side by side.

*Gyroscope—*Our gyroscope sensor is able to gather the rotational velocities of the robot component it is attached to. These velocities are directly supplied by the Unreal Engine 4's physics simulation. A noise model that considers both white noise and bias of the sensor is available.

Accelerometer— To measure the linear accelerations of a robot component, the accelerometer sensor is used. Since the Unreal Engine 4 does not provide acceleration data of simulated objects natively, we must fall back on approximated data. We use the linear velocities of the respective component at the current and previous simulation frame, as well as the corresponding time difference between the frames, to approximate the experienced acceleration. As with the gyroscope sensor, a configurable noise model with white noise and bias is available.

*IMU—*With our IMU sensor, it is possible to measure the linear accelerations, the rotational velocities, and the orientation of a robot component at the same time. Thus, it combines the functionality of our gyroscope and our accelerometer with the additional capabilities of a magnetometer. The noise model configuration options are therefore a combination of the ones of the previous two sensors.

IMU with trigger— This sub-type of the IMU sensor has the additional capability to trigger measurements of other sensors. This is particularly useful if processing components, e.g., the local reference filter of LRU, require camera data and IMU data with matching timestamps. In that case, the IMU with trigger sensor is able to trigger measurements of image sensors, thus generating synchronized data. Since a cameras frame rate is usually lower than the IMU frequency, the cameras are able to be triggered on only a subset of IMU measurements. Whether or not a camera image was triggered is indicated by a flag that is included with the IMU measurement.

Maps

In this section, we briefly describe all of the maps currently available in URSim. We point out their properties, give examples for use cases, and show snapshots in Figure 7 of each of them to give an impression.

*Lunar—*With its craters of varying sizes and rocky terrain shown in Figure 7a the Lunar map simulates an environment similar to Earth's moon in a visually quite detailed fashion. A special property of this environment is the reduced gravitational pull to its surface. Furthermore, the lightning is solely based on the sun and no ambient lights are added.

*Mars—*The Mars map, as shown in Figure 7b, resembles the surface of Mars. It is a large-scale environment containing different types of rocks, pebbles, and the signature red-colored dirt floor. Some parts of the map are flat with large plateaus, while others are rough mountainous terrain.

*DLR-OP—*The DLR-OP map we present is of the DLR site in Oberpfaffenhofen, Germany. It was created from real flight data taken in multiple passes by a drone. The separate sections of the map were then stitched together to create the top-down view seen in Figure 7c.

*Modern House—*Furthermore, we use the Modern House map an indoor environment for evaluation of service robots in typical house-hold scenarios. The house consists of several rooms on different levels as can be seen in Figure 7d. Most of the objects have reflecting and repetitive textures.

Test and Calibration Map— Lastly, we show our Test and Calibration map in Figure 7e, which we use to calibrate and test our robots and their sensory measurements. The map features a visualized coordinate system which helps with calibration and debugging processes. It also has concentric circles drawn on the ground with in regular intervals of increasing radii. For image sensor calibration, a checkerboard can be spawned into the map as seen in Figure 7f. While the camera sensors in URSim use the precise camera intrinsics they are configured with, camera calibration can still be useful to simulate calibration errors.

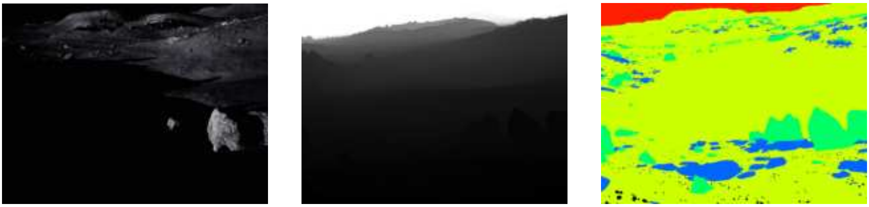
IPC

As described in Section 3 we designed URSim with the different inter-process communication interfaces used at the German Aerospace Center (DLR) in mind. We provide a convenient way to choose which IPC implementations are available on each robot using the RDDF presented in Section 3. Furthermore, each sensor can be individually configured to use one or more of the enabled IPCs. This configuration is performed within the robot description file.

*ROS—*The first IPC interface implemented in URSim is the well known Robot Operating System. When publishing sensor measurements with ROS, they are converted from URSim internal data structures to the ones defined by the ROS framework. Furthermore, we also provide and interface for subscribers to ROS topics in URSim. These are able to receive command messages that can be used to control the robots movements and actions. The connection between URSim and the externally running roscore is facilitated by a rosbridge [44]. ROS is the IPC used on the robotic systems LRU and ARDEA.

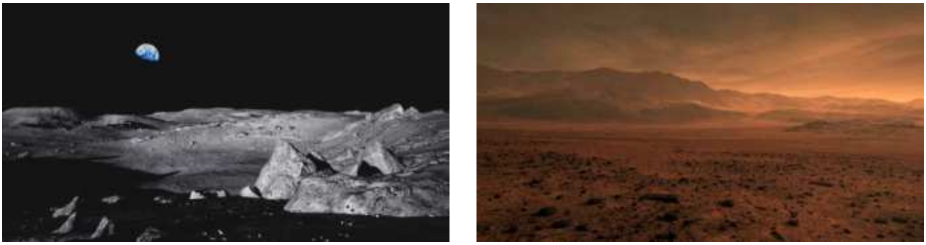
LN— The second interface available in URSim is to the middleware links_and_nodes, that has been developed at the German Aerospace Center (DLR). LN provides real-time messaging streams, process management, and low-overhead marshaling. Similarly to the ROS framework, LN supports the subscription and publishing to message streams. It is the IPC deployed to communicate with the robotic platform Rollin' Justin.

*Console—*Lastly, we provide the Console IPC that is useful for debugging and testing purposes. It simply outputs all published sensor messages in text format to the console using the Unreal logging system. Thus, it is independent from specialized hardware and software and therefore usable on every machine URSim runs on. The logged messages all contain the name of the emitting sensor, a time stamp, the reference frame name, and the measured data. Messages containing large amounts of data, such as image measurements, are condensed

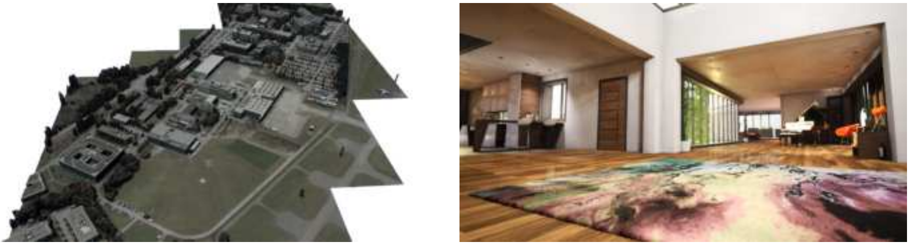


(a) Image captured with RGB-camera. (b) Image captured with depth-camera. (c) Image captured with semantic-camera.

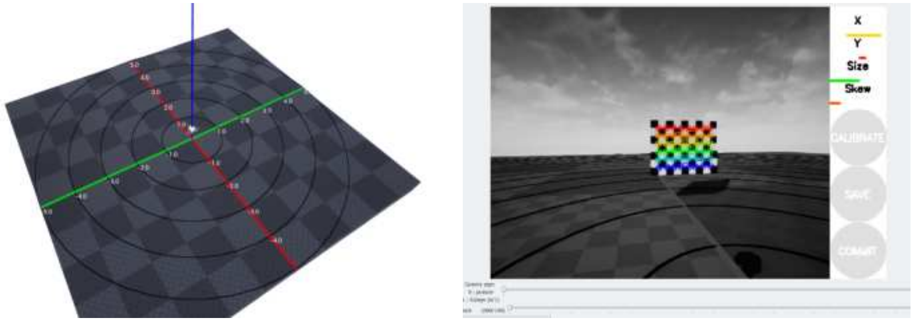
Figure 6: Side by side comparison of scene in moon environment captured by URSim's three image sensor types.



(a) Lunar landscape. (b) Martian landscape.



(c) Map of the DLR site in Oberpfaffenhofen. (d) Modern house map.



(e) Test and calibrations map for robots and sensors. (f) Checkerboard for camera calibration in URSim's calibration map.

Figure 7: Renderings of the currently available maps in URSim.

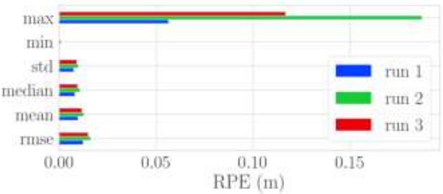


Figure 8: Translation error of navigation filter pose estimation relative to a change in position of 1 m for multiple runs in the Mars map.

to maintain readability.

5. EVALUATION

To demonstrate the capabilities of URSim, we conducted an exploration mission with the simulated LRU in the virtual Mars and Moon landscapes. For this evaluation, we examine three different aspects of that mission. Firstly we analyzed the accuracy of the pose estimation provided by the local reference filter [26]. Secondly we assessed the results of the local and global mapping. Finally we performed semantic mapping with the LRU.

Pose estimate evaluation— In this section, we show that data generated with URSim can be processed with a visual-inertial navigation system to accurately estimate the pose of the LRUs. To this end, we manually steered the rover to followed three different trajectories within the Mars map. Afterwards, we evaluated the accuracy of the LRUs pose estimated with the local reference filter by comparing the estimated trajectory with the ground truth trajectory. The ground truth is directly provided by the simulation's internal physics state. The following trajectory evaluations and plots where generated utilizing the evo [45] package. Figure 8 displays the relative translational errors and Figure 9 shows the relative rotational errors for the three followed trajectories.

The mean relative translational error per meter of displacement was between 0.0097 m and 0.0127 m for the three trajectories. The mean relative rotational error per meter of displacement was below one degree for all trajectories. When only integrating the output of a visual-inertial system, the absolute translational error increases as the relative errors accumulate over time. This effect can be seen in Figure 10, which shows a top-down view of the ground truth and estimated trajectory for the second run. The accuracy of the pose estimation of the simulated LRU is similar to the one that can be achieved with the real robot.

Local and global mapping— In this section, we present the maps built while following trajectories through the Mars(Figure 7b) and Moon world (Figure 7a). These maps are created using the geometric information of the measurements made by depth camera sensor that is attached to the simulated LRU. In conjunction with the current pose estimate, these measurements are incrementally added to a gridmap. Figure 11a shows the resulting gridmap when navigating through the Mars world. Similarly to the real system, obstacles are detected directly on the virtual depth images. A costmap is derived from the detected obstacles and steep slopes. Figure 11b shows an example of a costmap used to plan a path (green line) through the rough Mars environment. Finally, we show the probabilistic voxel-grid representation of the 3D map in Figure 11c generated by applying our 6D global localization and mapping system [29].

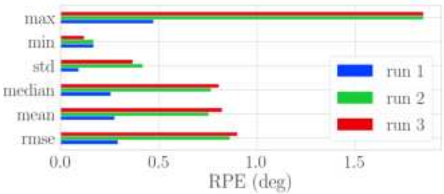


Figure 9: Rotational error of navigation filter pose estimation relative to a change in position of 1 m for multiple runs in the Mars map.

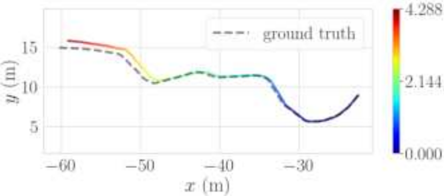
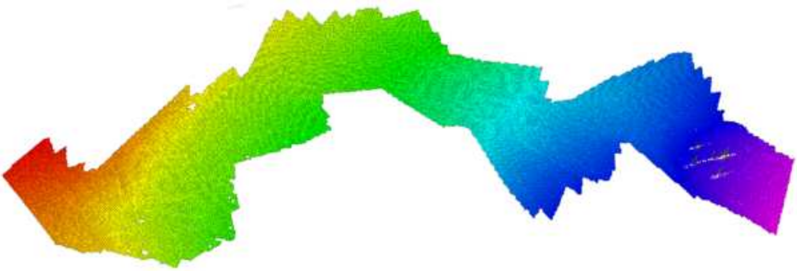


Figure 10: Top-down view of ground truth and pose estimate trajectories of run #2 in the Mars map.

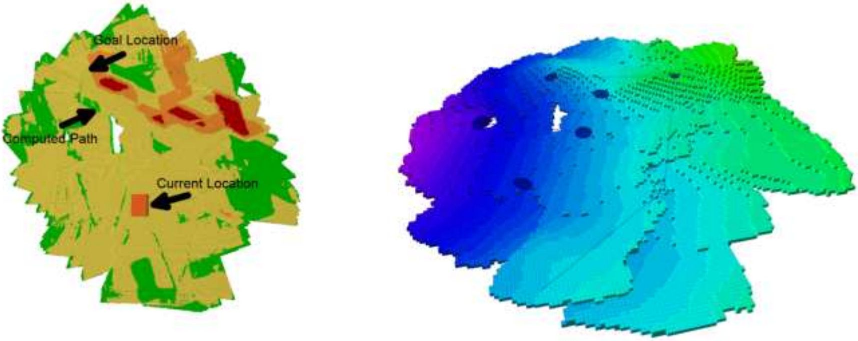
Semantic mapping— In this section, we present the results of our third experiment, where we applied our semantic mapping pipeline in the Lunar world. The goal is to annotate the geometric map with additional semantic information about respective terrain types. To this end, the meshes used to represent different objects in the simulated Moon environment, e.g., various rock types, were annotated with ground truth labels in the simulation map. This enables the virtual semantic camera sensor (see section 4) to render these semantic annotations into a semantically segmented image. Next, an additional gridmap layer was created for each of the semantic terrain types. Each of the semantic layers includes the probability distribution of its respective terrain type. During the semantic mapping process, these layers were then incrementally filled with information based on the current sensor input. In Figure 12a, we show the geometric local gridmap layer and in Figure 12b several semantic mapping representation, including the probability distributions for the individual terrain classes for each grid cell, retrieved from the semantic mapping layers.

6. SUMMARY & CONCLUSION

With URSim, we solve the common issues of developing software for complex robotic systems. URSim is a versatile Software-in-the-Loop (SiL) simulator to develop and test high-level software components, which allows for a continuous test cycle without requiring access to the real system. A generic robot class is used to interface the simulated hardware



(a) Height-colored elevation layer of the rolling gridmap. It was built while driving down a slope in the Mars environment.



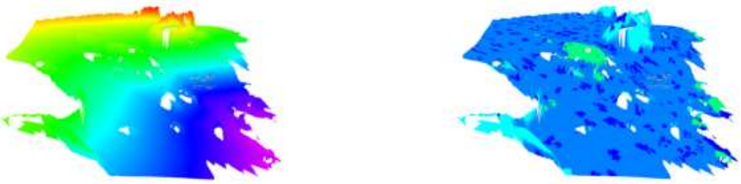
(b) Local costmap with terrain classification results used for path planning (green → red: traversible → obstacle). The narrow green line indicates the planned path between the rover's current and its goal location.

(c) Height-colored global 3D probabilistic voxel-grid map overlaid with the SLAM graph (blue nodes and edges).

Figure 11: Different local and global map representations generated while navigation through the unstructured Mars environment.

and generates the physical representation of the engine. It can be customized by adding different physical and virtual components to represent the real system as accurately as possible. Each robot is described by a Robot Design Description File and the various visual and physical sensors provided by URSim can be easily attached to the robot by adding them to the description file, which allows for a fast change between different sensor setups. The modern and adaptable system architecture allows to customize the simulator for different setups, frameworks and modules required for a specific mission. A layer of abstraction for the IPC offers the possibility to integrate different communication infrastructures, such as ROS and LN.

Testing high-level software components is also often limited as the challenging conditions in space cannot be reproduced on Earth. URSim is based on the Unreal Engine 4, which is capable of rendering photo-realistic scenes. The UE4 allows us to reproduce many of the challenging light conditions relevant to space missions and thus to test robotic software components in realistic scenarios. The graphical capabilities of the underlying UE4 are highlighted by our integrated maps, e.g., the Martian Landscape, Lunar Landscape, and Modern House map. We demonstrate the capabilities and advantages of URSim when developing and testing software components by simulating a planetary exploration mission. We have integrated our flying system ARDEA, our rover prototype LRU and the humanoid robot Rollin' Justin in URSim. For the LRU, we have simulated the kinematic system, as well as the sensor setup and provide similar interfaces as on the real robotic system. Furthermore, we were able to apply and test the complete navigation and mapping stack of the LRU, without changing parameters compared to the real system and could show that the provided sensor data can be used as input for an visual-inertial navigation system. The evaluation of navigation and mapping tests in multiple virtual environments, using only simulated measurements, yields compelling results. We have achieved similar translational and rotational errors for the pose estimation as on the real system and the mapping of the virtual environment is accurate and consistent. Thus, URSim can be applied to test complete software stacks of robotic systems and to evaluate visual-inertial navigation systems.



(a) Local elevation map of a region in the Lunar map.

(b) Semantic map color coded by most likely terrain class.

(c) Probability distribution of first rock terrain type.

(d) Probability distribution of second rock terrain type.

Figure 12: Visualized semantic terrain types of the local gridmap for the Moon environment.

7. FUTURE WORK

Currently URSim can be used to simulate and test a single robot at the same time. However, as multi-robot scenarios, in which a heterogeneous team of robots is working together to maximize the scientific return of space missions is currently in the focus of the research community, we want to extend URSim to be able to simulate space exploration missions with several robots. Furthermore, we want to provide a better interface to set the lightning conditions expected at a certain day or night time on the target planetary surface. This is important to easily adapt tests of robotic vision algorithms in different realistic conditions.

8. ACKNOWLEDGMENT

This work was supported by the Helmholtz Association, project ARCHES (contract number ZT-0033)

REFERENCES

[1] M. J. Schuster *et al.*, "Towards Autonomous Planetary Exploration: The Lightweight Rover Unit (LRU), its Success in the SpaceBotCamp Challenge, and Beyond," *Journal of Intelligent & Robotic Systems (JINT)*, Nov. 2017.

[2] M. G. Müller, F. Steidle, M. J. Schuster, P. Lutz, M. Maier, S. Stoneman, T. Tomić, and W. Stürzl, "Robust Visual-Inertial State Estimation with Multiple Odometries and Efficient Mapping on an MAV with Ultra-Wide FOV Stereo Vision," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[3] M. Fuchs, C. Borst, P. R. Giordano, A. Baumann, E. Kraemer, J. Langwald, R. Gruber, N. Seitz, G. Plank, K. Kunze *et al.*, "Rollin'justin-design considerations and realization of a mobile platform for a humanoid upper body," in *2009 IEEE International Conference on Robotics and Automation*, 2009.

[4] M. Vayugundla, F. Steidle, M. Smisek, M. J. Schuster, K. Bussmann, and A. Wedler, "Datasets of Long Range Navigation Experiments in a Moon Analogue Environment on Mount Etna," in *International Symposium on Robotics (ISR)*, 2018.

[5] L. Meyer, M. Smíšek, A. Fontan Villacampa, L. Oliva Maza, D. Medina, M. J. Schuster, F. Steidle, M. Vayugundla, M. G. Müller, B. Rebele, A. Wedler, and R. Triebel, "The madmax data set for visual-inertial rover navigation on mars," *Journal of Field Robotics*, vol. 38, no. 6, 2021.

[6] K. L. Wagstaff, Y. Lu, A. Stanboli, K. Grimes, T. Gowda, and J. Padams, "Deep mars: Cnn classification of mars imagery for the pds imaging atlas," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[7] N. Koenig and A. Howard, "Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, Sep 2004.

[8] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *1st Annual Conference on Robot Learning*, 2017.

[9] NVIDIA, "Isaac Sim Software," https://developer.nvidia.com/isaac-sim.

[10] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2018.

[11] Y. Song, S. Naji, E. Kaufmann, A. Loquercio, and D. Scaramuzza, "Flightmare: A flexible quadrotor simulator," in *Conf. on Robot Learning*, 2020.

[12] M. Hellerer, M. J. Schuster, and R. Lichtenheldt, "Software-in-the-loop Simulation of a Planetary Rover," in *International Symposium on Artificial Intel-*

ligence, Robotics and Automation in Space (i-SAIRAS), 2016.

[13] A. Jain, “Darts-multibody modeling, simulation and analysis software,” in *European Congress on Computational Methods in Applied Sciences and Engineering*, 2019.

[14] Epic Games, “Unreal Engine,” https://www.unrealengine.com.

[15] Unity, “Unity Engine,” https://unity.com.

[16] M. G. Müller, M. Durner, A. Gawel, W. Stürzl, R. Triebel, and R. Siegwart, “A Photorealistic Terrain Simulation Pipeline for Unstructured Outdoor Environments,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021.

[17] M. Denninger, M. Sundermeyer, D. Winkelbauer, Y. Zidan, D. Olefir, M. Elbadrawy, A. Lodhi, and H. Katam, “Blenderproc,” 2019.

[18] B. O. Community, *Blender - a 3D modelling and rendering package*, 2018.

[19] S. Parkes, I. Martin, M. Dunstan, and D. Matthews, “Planet surface simulation with PANGU,” in *Int. Conf. on Space Operations*, 2004.

[20] R. Brochard, J. Lebreton, C. Robin, K. Kanani, G. Jonniaux, A. Masson, N. Despré, and A. Berjaoui, “Scientific image rendering for space scenes with the SurRender software,” *arXiv:1810.01423*, 2018.

[21] Stanford Artificial Intelligence Laboratory et al., “Robotic operating system.” [Online]. Available: https://www.ros.org

[22] F. Schmidt, “links and nodes.” [Online]. Available: https://gitlab.com/links_and_nodes/links_and_nodes

[23] S. D. Organization, “Data distribution service.” [Online]. Available: https://www.omg.org/spec/DDS/

[24] I. Sucan and J. Kay, “Unified robot description format.” [Online]. Available: http://wiki.ros.org/urdf

[25] H. Hirschmüller, “Stereo Processing by Semiglobal Matching and Mutual Information,” *TPAMI*, vol. 30, no. 2, 2008.

[26] K. Schmid, F. Ruess, and D. Burschka, “Local Reference Filter for Life-Long Vision Aided Inertial Navigation,” in *FUSION*, 2014.

[27] C. Brand, M. J. Schuster, H. Hirschmüller, and M. Suppa, “Stereo-Vision Based Obstacle Mapping for Indoor/Outdoor SLAM,” in *IROS*, 2014.

[28] M. J. Schuster, C. Brand, H. Hirschmüller, M. Suppa, and M. Beetz, “Multi-Robot 6D Graph SLAM Connecting Decoupled Local Reference Filters,” in *IROS*, 2015.

[29] M. Schuster, K. Schmid, C. Brand, and M. Beetz, “Distributed stereo vision-based 6d localization and mapping for multi-robot teams,” *Journal of Field Robotics (JFR)*, October 2018.

[30] M. J. Schuster, “Collaborative localization and mapping for autonomous planetary exploration: Distributed stereo vision-based 6d slam in gnss-denied environments,” Ph.D. dissertation, University of Bremen, Bremen, Germany, 2019.

[31] C. Brand, M. J. Schuster, H. Hirschmüller, and M. Suppa, “Submap Matching for Stereo-Vision Based Indoor/Outdoor SLAM,” in *IROS*, 2015.

[32] H. Lehner, M. J. Schuster, T. Bodenmüller, and R. Triebel, “Exploration of Large Outdoor Environments Using Multi-Criteria Decision Making,” in *ICRA*, 2021.

[33] H. Lehner, M. J. Schuster, T. Bodenmüller, and S. Kriegel, “Exploration with active loop closing: A trade-off between exploration efficiency and map quality,” in *IROS*, 2017.

[34] A. Wedler *et al.*, “From the ROBEX Experiment Toward the Robotic Deployment and Maintenance of Scientific Infrastructure for Future Planetary Exploration Missions,” in *42nd COSPAR Scientific Assembly*, Pasadena, California, USA, Jul. 2018.

[35] A. Wedler *et al.*, “From single autonomous robots toward cooperative robotic interactions for future planetary exploration missions,” in *69th International Astronautical Congress (IAC)*, Bremen, Germany, Oct. 2018.

[36] M. J. Schuster *et al.*, “The ARCHES Space-Analogue Demonstration Mission: Towards Heterogeneous Teams of Autonomous Robots for Collaborative Scientific Sampling in Planetary Exploration,” *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, no. 4, pp. 5315–5322, Oct 2020.

[37] P. Lutz, M. G. Müller, M. Maier, S. Stoneman, T. Tomić, I. von Bargen, M. J. Schuster, F. Steidle, A. Wedler, W. Stürzl, and R. Triebel, “ARDEA - an MAV with skills for future planetary missions,” *Journal of Field Robotics*, 2020.

[38] M. G. Müller, S. Stoneman, I. von Bargen, F. Steidle, and W. Stürzl, “Efficient terrain following for a micro aerial vehicle with ultra-wide stereo cameras,” in *2020 IEEE Aerospace Conference*, 2020.

[39] N. Y. Lii, D. Leidner, P. Birkenkampf, B. Pleintinger, R. Bayer, and T. Krueger, “Toward scalable intuitive telecommand of robots for space deployment with meteron supvis justin,” 2017.

[40] P. Schmaus, D. Leidner, T. Krüger, A. Schiele, B. Pleintinger, R. Bayer, and N. Y. Lii, “Preliminary insights from the meteron supvis justin space-robotics experiment,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, 2018.

[41] P. Schmaus, D. Leidner, R. Bayer, B. Pleintinger, T. Krüger, and N. Y. Lii, “Continued advances in supervised autonomy user interface design for meteron supvis justin,” in *2019 IEEE Aerospace Conference*. IEEE, 2019.

[42] J. Vogel *et al.*, “An ecosystem for heterogeneous robotic assistants in caregiving: Core functionalities and use cases,” *IEEE Robotics Automation Magazine*, vol. 28, no. 3, 2021.

[43] M. Sewtz, X. Luo, J. Landgraf, T. Bodenmüller, and R. Triebel, “Robust approaches for localization on multi-camera systems in dynamic environments,” in *2021 7th International Conference on Automation, Robotics and Applications (ICARA)*, 2021.

[44] P. Mania and M. Beetz, “A framework for self-training perceptual agents in simulated photorealistic environments,” in *International Conference on Robotics and Automation (ICRA)*, Montreal, Canada, 2019.

[45] M. Grupp, “evo: Python package for the evaluation of odometry and slam.” https://github.com/MichaelGrupp/evo, 2017.

BIOGRAPHY

Marco Sewtz *received his B.Eng. degree in electrical engineering at the University of Applied Sciences of Munich and his M.Sc. degree at the Technical University of Munich. He works at the Institute for Robotics and Mechatronics at the German Aerospace Center (DLR) as a researcher since 2018. His interests focuses on SLAM and multi-model perception of the environment. Before his current role, he worked as an electrical designer for high-performance processing modules for space hardware at Airbus Defence and Space.*

Hannah Lehner *is a researcher at the Department of Perception and Cognition at the Robotics and Mechatronics Center, German Aerospace Center (DLR). She received her master's degree in Geodesy and Geoinformation Science from the Technical University of Berlin in 2014. Her current research activities at the Robotics and Mechatronics Center at DLR are methods for autonomous planetary exploration with mobile robots.*

Yunis Fanger *works at the Institute for Robotics and Mechatronics at the German Aerospace Center (DLR) as a scientific researcher since 2020. He received his M.Sc. degree in electrical engineering from the Technical University of Munich in 2019 with the specialization on robotics and automation. His research focuses on the topic of semantic mapping in distributed robotic systems.*

Jan Eberle *is a master student at the Technical University of Munich (TUM). He received his Bachelor's degree in Computer Science: Games Engineering in 2021.*
His current studies at TUM focus on robotics and artificial intelligence.

Martin Wudenka *Martin Wudenka received his B.Sc. Informatics from TU Dresden in 2018 and his M.Sc Robotics, Cognition and Intelligence from TU Munich in 2021. Between 2018 and 2021 he conducted research as a working student at the Robotics and Mechatronics Institute of the German Aerospace Center. His work focused on visual odometry and simulation engines for robotic research.*

Marcus G. Müller *is a researcher in the department of "Perception and Cognition" at the German Aerospace Center (DLR) since 2016 and Ph.D. student at ETH Zurich. He is the leader of the MAV Exploration Team at the Institute of Robotics and Mechatronics (DLR-RM), where he is working on autonomous navigation algorithms for MAVs. Before joining DLR he conducted research at the Jet Propulsion Laboratory (JPL) of NASA in Pasadena, USA, where he worked on visual inertial navigation for MAVs and on radar signal processing. Marcus received his Master's and Bachelor's degree in Electrical Engineering from the University of Siegen, Germany.*

Tim Bodenmüller *is senior researcher at the Institute for Robotics and Mechatronics at the German Aerospace Center (DLR). He joined DLR as a full-time researcher in 2001. and is working on 3D-sensing, robotic middleware and software design. He further is member of the institutes software engineering group. He received his Dipl.-Ing. degree in electrical and information engineering in 2001 from the Technical University of Darmstadt and his PhD in electrical engineering in 2009 from the Technical University of Munich.*

Martin J. Schuster *is senior researcher and leader of the Planetary Exploration Operations team at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR). He received his Ph.D. (Dr.-Ing.) degree in Computer Science (CS) from the University of Bremen in 2019, his first Master's degree in CS from the Georgia Institute of Technology in Atlanta, USA in 2010, his Bachelor's and second Master's degree in CS from the Technische Universität München (TUM) in 2008 and 2011, and his Bachelor's degree in Philosophy from the Ludwig-Maximilians-Universität München (LMU) in 2014. His research focus is on multi-robot SLAM in GNSS-denied areas.*

6. Audio Perception in Robotic Assistance for Human Space Exploration: A Feasibility Study

Authors:

Marco Sewtz, Werner Freidl, Adrian Bauer, Anne Köpken, Florian Lay, Nicolai Bechtel, Peter Schmaus, Rudolph Triebel and Neal Y. Lii

Conference:

Sewtz, Marco, et al. "Audio Perception in Robotic Assistance for Human Space Exploration: A Feasibility Study." 2023 IEEE Aerospace Conference. IEEE, 2023.

Abstract:

Future crewed missions beyond low earth orbit will greatly rely on the support of robotic assistance platforms to perform inspection and manipulation of critical assets. This includes crew habitats, landing sites or assets for life support and operation.

Maintenance and manipulation of a crewed site in extra-terrestrial environments is a complex task and the system will have to face different challenges during operation. While most may be solved autonomously, in certain occasions human intervention will be required. The telerobotic demonstration mission, Surface Avatar, led by the German Aerospace Center (DLR), with partner European Space Agency (ESA), investigates different approaches offering astronauts on board the International Space Station (ISS) control of ground robots in representative scenarios, e.g. a Martian landing and exploration site.

In this work we present a feasibility study on how to integrate auditory information into the mentioned application. We will discuss methods for obtaining audio information and localizing audio sources in the environment, as well as fusing auditory and visual information to perform state estimation based on the gathered data. We demonstrate our work in different experiments to show the effectiveness of utilizing audio information, the results of spectral analysis of our mission assets, and how this information could help future astronauts to argue about the current mission situation.

Contributions:

The author of this dissertation designed and integrated the audio-visual perception system. He designed and conducted the experimental evaluation. The script was provided by the author and the publication was presented by the author.

Copyright:

©2023 IEEE

Audio Perception in Robotic Assistance for Human Space Exploration: A Feasibility Study

**Marco Sewtz, Werner Friedl, Adrian Bauer, Anne Köpken, Florian Lay, Nicolai Bechtel,
Peter Schmaus, Rudolph Triebel, and Neal Y. Lii**

**German Aerospace Center (DLR)
Institut of Robotics and Mechatronics
Muenchener Str. 20, 82234 Weßling, Germany
{*Firstname.Lastname*}@dlr.de**

Abstract—**Future crewed missions beyond low earth orbit will greatly rely on the support of robotic assistance platforms to perform inspection and manipulation of critical assets. This includes crew habitats, landing sites or assets for life support and operation.**

Maintenance and manipulation of a crewed site in extra-terrestrial environments is a complex task and the system will have to face different challenges during operation. While most may be solved autonomously, in certain occasions human intervention will be required. The telerobotic demonstration mission, Surface Avatar, led by the German Aerospace Center (DLR), with partner European Space Agency (ESA), investigates different approaches offering astronauts on board the International Space Station (ISS) control of ground robots in representative scenarios, e.g. a Martian landing and exploration site.

In this work we present a feasibility study on how to integrate auditory information into the mentioned application. We will discuss methods for obtaining audio information and localizing audio sources in the environment, as well as fusing auditory and visual information to perform state estimation based on the gathered data. We demonstrate our work in different experiments to show the effectiveness of utilizing audio information, the results of spectral analysis of our mission assets, and how this information could help future astronauts to argue about the current mission situation.

TABLE OF CONTENTS

1. INTRODUCTION 1
2. RELATED WORK 2
3. SYSTEM OVERVIEW 2
4. METHODOLOGY 4
5. EVALUATION 5
6. CONCLUSION AND OUTLOOK 8
REFERENCES 8
BIOGRAPHY 10

1. INTRODUCTION

Accomplishing the goals of bringing humankind to the Moon and Mars is some of the greatest challenges ahead for the space community. To help meet these challenges, robotic assistance will be key, particularly for the construction and support of habitat infrastructure, as well as for carrying out scientific tasks. However, due to the long distances, communication round trip will cause delays of 20min to several hours between Earth and Mars.

Surface Avatar, a telerobotic technology validation mission led by German Aerospace Center (DLR) with partner European Space Agency (ESA), gives astronauts on board the International Space Station (ISS) control over robotic assets [1]. It investigates a combined approach offering scalable autonomy through multi-modal teleoperation to perform tasks in different scenarios. These can range from simple surveillance to complex maintenance tasks which often include a search for failure in which the astronaut has to detect an anomaly in the environment. The astronaut has to investigate multiple objects to observe their state, often accompanied by detailed inspection and manipulation of inner components.

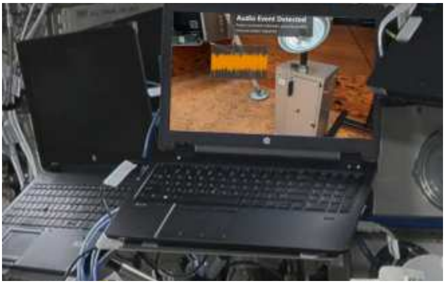


Figure 1: Integrated audio perception into the telerobotic system of Surface Avatar. The robot in the experimental area detected a sound event with an unknown spectral profile and requests manual action from an astronaut on board the International Space Station (ISS).

Audio perception provides an additional modality that may decrease crew time to find the anomaly in extra-terrestrial environments with an atmosphere like Mars. The direction of arrival of a sound event received by the system can be estimated and displayed to the astronaut. Furthermore, the robot's knowledge of the world can be used to infer the current state of a known object remotely and detect failures. All of these can be displayed to the crew as illustrated in the simulated view in Figure 1.

In this feasibility study, we aim to show our preliminary results on using audio perception, in the context of a telerobotic mission, to help understand the world around the robot and propose an approach to:

- detect sound events
- localize sound sources
- fuse sound input with vision sensors and prior knowledge
- obtain spectral knowledge and infer objects' state based on the received data

978-1-6654-9032-0/23/$31.00 ©2023 IEEE

2. RELATED WORK

Early research in the field of sound source localization has focused on the imitation of binaural audio perception of humans and animals [2][3][4][5]. They are based on the interaural phase difference (IPD) and interaural intensity difference (IID) of received signals. The inclusion of the head-related transfer function [6] and the modeling of the reverberation of the environment [7][8] increases the robustness further. However, these approaches require an accurate calibration process, where deviations and unexpected components in the environmental modeling greatly influence the outcome.

Successive work has been carried out on the estimation of Direction of Arrival (DoA) of a signal [9][10]. Incorporating a delay and sum beamformer (DSBF) these approaches estimate the direction using the time delay between the input signal of individual sensors. But low signal to noise ratio (SNR) environments or varying spectral profiles of the sound sources prevent usable results. Approaches based on deep learning [11][12][13][14][15] promise to overcome the mentioned problems, but require dedicated data sets for specific sources for training or immense data for generalization.

More recently, research attention has shifted toward subspace-based approaches like multiple signal classification (MUSIC) [16] or Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) [17]. To overcome the limitations and constraints of the chosen sampling frequency, they offer increased robustness and angular resolution [18][19][20]. The initial high computational demand could be decreased with recent advances in offering real-time estimations for outdoor [21] and indoor [22] environments.

The field of acoustic monitoring is well established in the area of ecological research, especially for ornithology [23][24]. Semi-automated analysis [25][26][27] are utilized for temporal and spatial estimation of bird behavior, which has been developed to detect and monitor audio events. However, expert knowledge is necessary to label received audio fragments. Full-automation methods [28][29][30][31] offering an unsupervised approach, which requires intense training. These methods have been applied toward factory and technical applications for process monitoring for additive manufacturing [32][33]. Furthermore, convolutional neural networks have been added for detecting the degradation state of robotic system [34]. However, the unknown spectral profiles or signals with high variances are still problematic.

In this work, we aim to show that acoustic perception can be effectively used as an additional modality in telepresence applications by implementing it in a ISS-to-Earth demonstration missions, Surface Avatar [1]. It depends on knowledge gained in previous space-to-ground missions, Analog-1 [35][36][37] and Meteron Supvis Justin [38]. We focus on a system that extends the immersion of the robot operator to obtain more knowledge about the environment and which keeps the astronaut in the loop.

3. SYSTEM OVERVIEW

This work is intended to be integrated into DLR's Rollin' Justin [39]. It is a dexterous humanoid robot with a mobile wheel-base, which has served in a wide array of research toward space exploration and terrestrial applications [38] [40]. Equipped with an Intel Realsense D435i RGBD camera, it is able to visually perceive its environment. The sensor is mounted on the head to mimic human-like anatomy and follows the head movement to stay aligned with the visual processing pipeline.

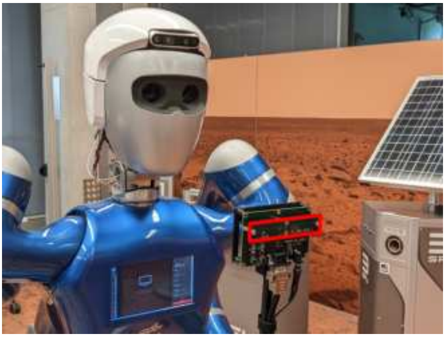


Figure 2: DLR's dexterous humanoid robot, Rollin' Justin, and the microphone array (red) used in this work.

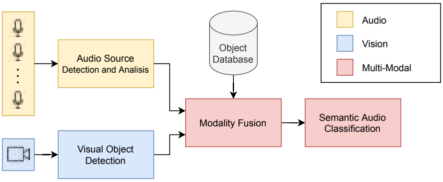


Figure 3: Overview of the system architecture. The approach is divided into auditoral, visual and prior knowledge.

We utilize a four-sensor microphone array as depicted in Figure 2 to receive audio information on the environment. The sensors are arranged linearly with located $d = [0.00, 0.015, 0.06, 0.09]$ cm along the x-axis and enables broadband estimation of signals in the audible range. We investigated a future integration of the array into a novel head design [41] consisting of eight microphones heterogeneously placed on the forehead of the robot. The estimated directivity patterns (-3dB at $\pm 40^\circ$) are integrated into this feasibility study to assure the applicability.

Furthermore, the robot operates in an environment at the DLR simulating a Martian exploration and science site [42]. The environment includes a mechanical mock-up of a lander, several Smart Payload Units (SPUs) for scientific experiments and monitoring and a visual representation of Martian setting. All objects a marked with Apriltags [43] for easy identification and localization.

All data are recorded and pre-processed before fusing them together. Afterwards, using prior knowledge on the environment, the semantic information on the current perception of the world will be jointly inferred. An overview is given in Figure 3.

Audition

Audio is captured using the microphone array. For processing it is essential to have synchronous data acquisition.

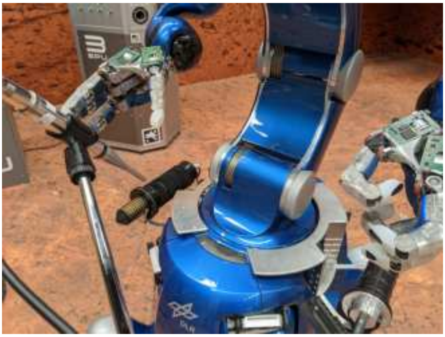


Figure 4: Estimation of the background noise profile for the robot environment. An audio probe is used to capture a highly accurate frequency spectrum that can be used for spectral subtraction in noise filtering.

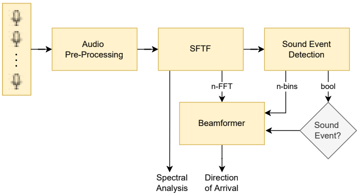


Figure 5: Audio processing branch. After pre-processing the received signals are transformed into the frequency domain and the presence of a sound source is estimated. Afterwards, a possible source is localized.

Therefore, the analog-to-digital conversion is triggered on hardware side. The sampling rate is set to 44100Hz to capture the full spectrum of most signals available in our environment.

Background noise such as wind or system noise created by mechanical components, e.g. cooling fans, induce an omnipresent spectral component that is always accumulated to the received signal. A prior statistical profile is estimated using a sound probe as shown in Figure 4 to obtain an accurate recording of the actual noise. Then, a Fourier analysis is performed to obtain the gains of the spectral components. These can be applied later for noise reduction by spectral subtraction. To prohibit unnecessary detection and estimation efforts that may lead to false positive results in subsequent modules, the presence of a suitable input signal is detected. An evaluation of the power equivalent of the sound signal is performed, comparing the active input to the previously acquired noise spectrum. The received response is used to classify the audio as *noise* or *sound event*. Afterwards, the DoA of the signal is estimated to obtain the spatial information of the sound source used later in the fusion process. The chain of modules is shown in Figure 5.

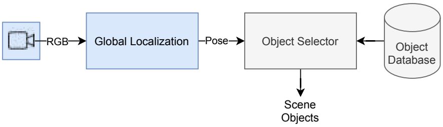


Figure 6: Components of the vision processing. The camera data is used for a global position strategy based on a Simultaneous Localization and Mapping (SLAM) approach and refinement using AprilTags. This information is used to receive current scene objects from a knowledge base.

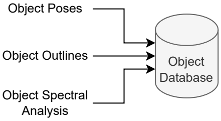


Figure 7: The object database is storage of the robots perception and knowledge about the world. It included the poses of known objects, their geometric outlines and previously obtained spectral profiles of observed states.

Visual Perception

The vision system is primarily used to obtain the localization information of the system. The coarse ego-pose estimation is retrieved by a SLAM system based on a multi-camera approach [44] in the base. Further refinement of the pose is obtained by using visible AprilTags in the environment. Finally, based on the current localization, all scene objects are loaded from a central object database. It is noteworthy that the query returns more objects than visible to the camera as the auditory system is capable of perceiving more of the world than the field-of-view of the camera. As seen in Figure 6 the system returns the list of scene objects needed for the fusion process.

Multi-Modal Fusion and Processing

In this step the information of the audio and visual branch are fused together to obtain a multi-modal description of the world. The DoA estimation retrieved from the audio beamforming module is used to cast a ray from the current position of the robot and infer the 3D position of the sound source using the known geometric outline of scene objects obtained from the vision branch. If a sound source can be located within an object, the relevant spectral information of the given entity is loaded from the database. Finally, this is compared to the received spectrum and the state is inferred.

Object Database

The aforementioned object database is a storage of prior knowledge obtained before the operation of the system (Figure 7). It contains for each object in the environment its exact position, orientation and geometric outline. Furthermore, it also contains a list of spectral information of different states. Each consists of the median and an acceptance band of normalized frequency spectra, e.g. Figure 8 displays the characteristic spectrum of a running drill. This is used to estimate the state of an object or infer if the observed situation is unknown.

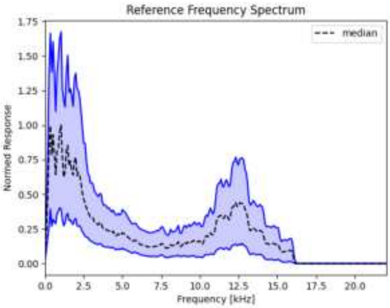


Figure 8: Normalized frequency spectrum of a running drill. The median is depicted as a dashed line. The acceptance band of the sound spectrum is shown by the 20th and 80th percentile.

4. METHODOLOGY

The following sections describes the key aspects of the selected approaches and applied customizations in depth.

LTSD Power Evalutation

The module for detecting sound events is based on the voice activity detection (VAD) approach by Ramırez et al. [45]. The received input signal is analyzed on smaller chunks. Each is further divided into overlapping subframes, which are transformed into the frequency domain using a short-term Fourier transform. We estimate the spectral envelope for the chunk for the frequency bin l on N subframes as

$$\mathrm{LTSE_N(k)} = \max\left(\mathrm{X(k,0), X(k,1), ..., X(k,N)}\right) \tag{1}$$

with $X(k,n)$ representing the k-th bin of the n-th subframe.

Each long-term spectral envelope (LTSE) value represents the current maximal gain for each frequency bin in the envelope. To receive information on the overall spectrum differs from the noise reference ξ, we calculate the long-term spectral divergence (LTSD) as given by

$$\mathrm{LTSD_N} = 10\log_{10}\left(\frac{1}{\mathrm{n_{FFT}}}\sum\frac{\mathrm{LTSE^2(k)}}{\xi^2\mathrm{(k)}}\right) \tag{2}$$

with $\mathrm{n_{FFT}}$ as the amount of frequency bins in each subframe analysis. Subsequently inserting the audio chunks, we receive a temporal trend of the LTSD responses. A typical result can be seen in Figure 9. Furthermore, we exploit Equation (2) and retrieve the m-most deviating frequency bins compared to the reference ξ and propagate this information to the beamformer module.

MUSIC DoA Estimation

We integrate a modified implementation [22] of the MUSIC algorithm [16] [21] to locate sources using the directed subspaces of the frequency domain. Considering the complex short-term input signal $s_k(t)$ for the k-th frequency band, we get

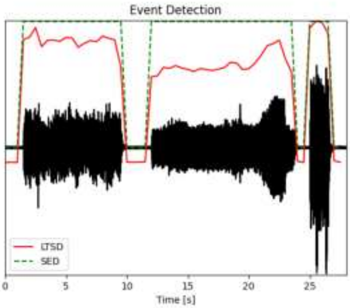


Figure 9: LTSD response for three different sound sources. The input audio is separated into *sound event* (SED=high) and *noise* (SED=low).

$$\begin{aligned} s_k(t) &= \lambda_k(t)e^{i2\pi f_k t} \\ &= \lambda_k(t)e^{i\omega_k t} \end{aligned} \tag{3}$$

For a linear microphone array of N sensors where each signal is delayed by

$$\Delta_n = \frac{d_n \sin(\theta)}{c_0} \tag{4}$$

with the DoA θ and the speed of sound c_0, we can construct the system equation as

$$\begin{bmatrix} 1 \\ e^{i\omega_k\Delta_1} \\ e^{i\omega_k\Delta_2} \\ \vdots \\ e^{i\omega_k\Delta_N} \end{bmatrix} s_k(t) =: \boldsymbol{a_k} s(t) \tag{5}$$

We denote $\boldsymbol{a_k}$ as the *steering vector* of the sound source, describing the angular dependency of the received signal to the direction of arrival. As described in the referenced work, the source subspace $\boldsymbol{U}_S$ of the received signal is extracted. The aforementioned *steering vector* is an element of the signal subspace, therefore

$$\boldsymbol{a_k} \in \boldsymbol{U}_S, \tag{6}$$

$$\Rightarrow \boldsymbol{a_k} \perp \boldsymbol{U}_\Sigma \tag{7}$$

of the noise subspace $\boldsymbol{U}_\Sigma$. We can formulate the response equation as

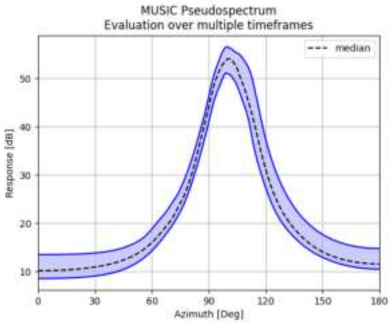


Figure 10: Pseudospectrum as returned from the custom MUSIC implementation. The frequency evaluation is adapted to the current received spectrum and the DoA can be reconstructed from the signal maximum.

$$P(\theta) = 10\log_{10}\sum_{k=1}^{K_N}\frac{1}{\langle \boldsymbol{a_k}, \boldsymbol{U}_\Sigma\rangle^2} \tag{8}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product. We further only examine the N-most deviating frequency bins as calculated in the LTSD power evaluation to integrate into the final response. This reduces the amount of resources needed to process the data while increasing the robustness in low SNR scenarios. An exemplary pseudospectrum is displayed in Figure 10 showing a detected sound source at $\approx 100°$.

Modality Fusion

Processing the separate modalities independently, the modality fusion combines both branches and estimates the joint state. Based on the global position of the system, a set of scene objects is loaded from the object database. The received geometry is projected on the 2D ground plane as the microphone array is only capable to distinguish between azimuth but not elevation angles. As ray is casted starting at the microphones reference position and with the estimated orientation. The ray is tested with each outline of the scene objects for an intersection. The point is reprojected to the microphone array and checked against the sensor accuracy to take measurement tolerances into account. Finally, after testing all lines, the intersection with the shortest ray length is taken as the source position.

Spectral Classification

As a last step, the spectral information of the object is examined. The received audio is compared in the frequency domain with already obtained spectral profiles. For each profile, a audio sample is recorded with a duration of at least 5s. These audio samples are transformed with a short-term Fourier transform (SFTF) using small overlapping subframes with a hop-parameter of 32 samples. The median spectrum P_{50} is calculated over all received spectra. The highest value of the median is used to normalize the spectrum and constraint it to $[0, 1]$. Afterwards, the 20th percentile P_{20} and the 80th percentile P_{80} for each frequency bin are taken as the lower and upper bound of the acceptance band. When receiving a new and unclassified spectrum, first the spectral components of the background noise is subtracted from the input signal. Afterwards, the median spectrum is estimated and normalized. We calculate the sum of the squared differences of frequencies that are within the acceptance band range of each bin.

$$s = \sum_{k\in K}\frac{1}{s_k} \tag{9}$$

$$s_k = \begin{cases} 0 & X_k < P_{20,k} \\ (X_k - P_{50,k})^2 & P_{20,k} \le X_k \le P_{80,k} \\ 0 & X_k > P_{80,k} \end{cases} \tag{10}$$

The received score describes the similarity of two frequency spectra within the acceptance band. Further we can set a threshold τ for recognizing known profiles. A analyzed spectrum is only considered if the $s \ge \tau$, ultimately leading to the assumption, if no score passes the threshold, the spectrum originates from an unknown source.

5. EVALUATION

For the evaluation, we consider the scenario of a dexterous mobile robot operating in a Martian environment. During the final ISS-Earth experiment session of METERON SUPVIS Justin [38] [46], ESA astronaut Alexander Gerst was tasked with finding, and replacing a failed component in a SPUs in the simulated Martian environment on ground. To recover to nominal operation, the operator first had to search for the problem with visual inspection of all components in the environment. This failure investigation and maintenance (shown in Figure 11), was, as expected, time-consuming. This inspired us to consider other modes of surveying the environment to achieve faster failure detection and localization. This desire turned us to audio perception, to remotely infer the state of an object.

We start with an evaluation in a simulated environment showing the applicability of our method for audio perception and finally show experiments conducted in our laboratory to show the transferability to actual applications.

Simulation

We use a simulated environment of a room with a rectangular floor shape of $W = 8$m, $L = 8$m and a constant height of $H = 4$m. Further, we define the absorption properties of the walls, the floor and the ceiling based on the data in [47] to mimic the acoustic behavior of our lab. The floor is constructed of rigid plywood with a linoleum surface. The northern and eastern wall are of hard surfaces. The ceiling and southern, as well as the western wall are with high absorption to reflect open space. All parameters are shown in Table 1. We design a reverberation time of $t_{60} = 0.5$s for our evaluation.

We further placed three sound sources (an *engine*, a *press*, and an unknown *air valve*) in the room, each emitting a different pre-recorded sound. An eight-sensor microphone array with the same directivity pattern as the future integrated sensor array of the system is placed at the south wall of the room. The resulting room is shown in Figure 12 and the source-specific room impulse response (RIR) in Figure 13.

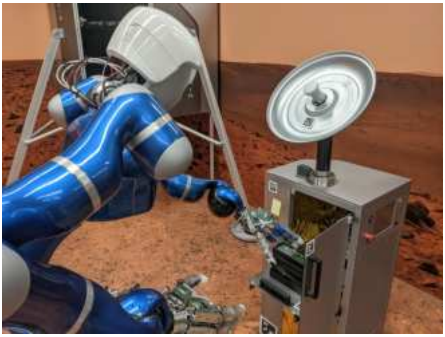


Figure 11: Detecting and replacing a failed component in a simulated Martian habitat. Prior missions required visual inspection of the enclosed modules for failure detection. Robot audition can enable remote detect the components' state, which can speed up anomaly detection.

Table 1: Material absorption properties at different frequencies were used for the simulation.

Element	250Hz	500Hz	1kHz	2kHz	4kHz	8kHz
Floor	0.21	0.10	0.08	0.06	0.06	0.06
Ceiling	0.45	0.55	0.60	0.90	0.86	0.75
Wall N	0.02	0.03	0.03	0.04	0.05	0.05
Wall E	0.02	0.03	0.03	0.04	0.05	0.05
Wall S	0.93	1.00	1.00	1.00	1.00	1.00
Wall W	0.93	1.00	1.00	1.00	1.00	1.00

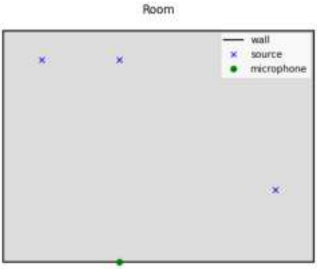


Figure 12: Simulated room environment. Displayed are the three sound sources, the position of the first microphone of the sensor array and the dimensions of the room. Absorption properties of all elements can be extracted from Table 1.

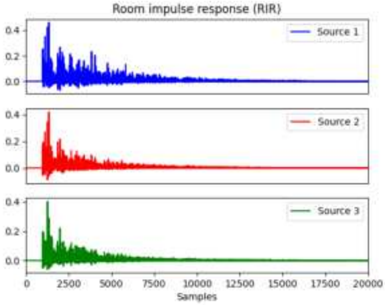


Figure 13: Estimated RIR of the simulated environment in Figure 12. The graphs show the propagation delay each signal needs to reach the first microphone. Further, echo can be identified as the following peaks in the graph. The slow drop after the impulse is due to the reverberation of $t_{60} = 0.5\text{s}$.

The simulated audio data is loaded into the proposed processing pipeline. Positions of the system and scene objects are altered by an uncertainty of 10cm. An exemplary result of the data fusion is shown in Figure 14 and shows the localization of a simulated source. For classification, we evaluate the naïve approach of comparing the sum of squared differences (SSD) and our proposed method of calculating the difference in the acceptance band.

The resulting score distribution is shown in Figure 15. The SSD approach for classification yields to individual class scores that are mostly in the range of $[10, 20]$. In general, narrow spectral profiles like *drill* or *saw* result in similar scoring results. Since the complete spectrum is compared, and in the case of a narrow-band signal, most of the spectral components are the background noise which scores a high similarity in this approach. Contrary, our approach takes the variance of the pre-recorded profile into account. While still a fairly simple approach, it results in high deviating class scores and is more robust to narrow-band profiles.

Further, we expect unforeseen sound events to occur and the spectral information of those is unknown. Since our classification approach is explicitly designed to handle this case, it estimates the score only on the acceptance band, thus yielding a significantly lower score compared to known sound profiles. An example can be seen in Figure 16. SSD scores in a comparable range as in the case of a known source. In the given example, it results in the selecting the *saw* class as it is a highly narrow-band profile and therefore more frequency bins with only the background noise. Our approach scores higher values for wide-band profiles like *engine* or *press* due to the higher probability of components of the unknown source laying coincidentally within the acceptance band. However, the overall scoring range is below 1 and by deploying a threshold of $\tau = 5$ including a safety margin, we can safely classify the input signal as *unknown*.

We further investigate the impact of the SNR and the number of simultaneously emitting sources on the successful inference in the modality fusion outcome. We place one, two and three sources in the room and artificially change the SNR

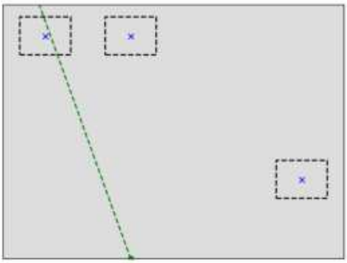


Figure 14: Illustration of the simulated room including three objects and the microphone array. The estimated DoA is shown as a dashed line. By using ray tracing, the source can be located within the object on the upper left-hand side.

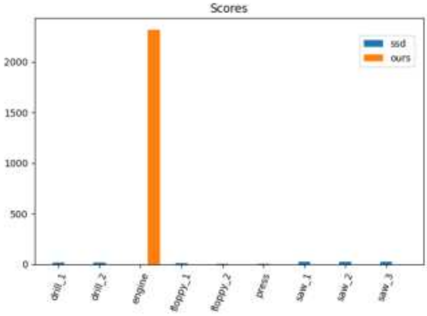


Figure 15: Results of the classification process for a known spectral profile of an engine. While the naïve approach SSD performs poorly and only small deviations between different sound profiles are recognizable, our approach correctly classifies the profile.

of the target in the simulation. The noise sources are set to be at $\mathrm{SNR} = 10\mathrm{dB}$ compared to the background noise. We sample 50 different scenarios where the sources a placed at random positions in the $3\mathrm{m}$ cone as defined in [41] at distances in the range of $[0.5\mathrm{m}, 5.0\mathrm{m}]$. We define a threshold of 0.90 for the desired hit rate as this is a good trade-off on correctly detected sound events and misses in our scenario. The results of the evaluation are shown in Figure 17. While in the single source case the threshold is already reached at $\mathrm{SNR}_{\mathrm{target},1} \approx 5\mathrm{dB}$, additional sources decrease the performance. For two active sources the minimum ration is increased to $\mathrm{SNR}_{\mathrm{target},2} \approx 20\mathrm{dB}$, for three sources the threshold is reached at $\mathrm{SNR}_{\mathrm{target},3} \approx 30\mathrm{dB}$.

Concluding with respect to a future use-case within the Surface Avatar mission, the results show that the perception of audio events and the fusion of the different modalities is feasible. The simulated signals could be identified for their origin and a simple yet effective approach based on an acceptance band in the spectral profile led to the successful estimation of different states. However, the presence of additional sources in the environment affects the performance of the processing pipeline and the rate of successful identification of emitting objects. Assuming that the robotic system itself will be acting as an emitting source in the world, a SNR of at least $20\mathrm{dB}$ must be assured for operation.

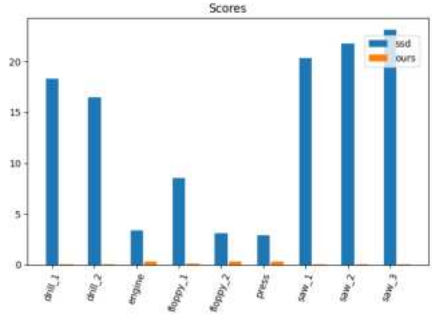


Figure 16: Results of the classification process for an unknown sound event. Compared to the results in Figure 15 the score of our approach is significantly lower and it can be easily stated the system received an unknown spectral profile.

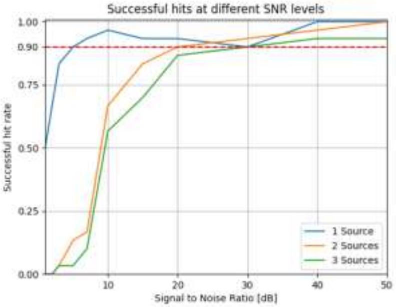


Figure 17: Evaluation of the relation between SNR, the number of active emitting sources and the success rate of detecting the target object. The three curves (blue, orange, green) show the rates for one, two and three sources respectively. We added a threshold of 0.90 as the minimum success rate for use in our scenario. It can be seen that additional sources increase the minimum SNR for successful operation.

Lab Evaluation

Further evaluation is conducted on recordings taken in a laboratory environment. In this experiment, we aim to show the transferability of our approach into a realistic scenario.

A speaker is placed inside of one of the scene objects and

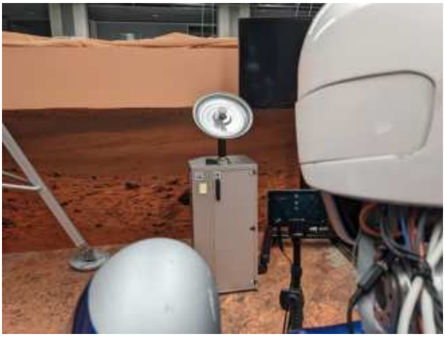


Figure 18: Audio perception evaluation in the METERON environment. A speaker is placed inside the SPU, shown with an antenna mounted on top, and is emitting a recording of running drill at low and high speed. The system shall differentiate between both states.

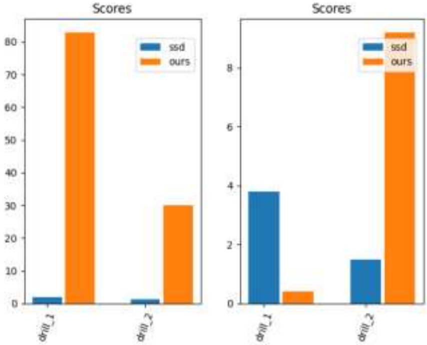


Figure 19: Scoring results for the state estimation of the drill object. The approach correctly estimated the correct states of the object on *low* and *high* speed. As the system identified the object according to its prior knowledge, only the associated sound profiles are loaded for estimation.

is set to alternately emit the sound of the pre-recorded drill at low and high speed. As discussed before, the robot is emitting noise and is a sound source itself in the environment resulting in a minimum of at least two sources at the same time. The speaker is set to transmit at an average of 50dB taking into account the transmission from inside the object and over the distance to the sensor array to meet the requirement of $\text{SNR}_{\text{target},2} \approx 20\text{dB}$. The sensor array is positioned in front of the robot facing the same direction as the camera interface. The setup is shown in Figure 18 including the robot, the sensor array and the target object. All data is fed into the proposed processing pipeline, including the prior knowledge of the positions of objects, possible spectral profiles and the mapping of object's emitting frequencies. The source object is detected in the localization module, the robot's knowledge is updated according to the database content and the state is determined based on the received audio signal.

An exemplary result is shown in Figure 19. Based on the prior knowledge, the system reduced the total amount of possible spectral profiles to two, *drill_1* and *drill_2*. The classification resulted in correct associations with the emitting profiles. However, the yielded scores a significantly lower than compared to the simulated ones. This is due to further sources in the environment emitting sounds that are overlaying the audio signal and induce further spectral noise. The transmission through the scene objects and the frequency depending sampling accuracy of the microphones were not simulated. Nevertheless, the preliminary results already show that the desired estimation can be achieved under lab conditions.

6. CONCLUSION AND OUTLOOK

In this work we presented a first study on the integration of audio perception into the context of the Surface Avatar mission led by DLR with partner ESA. We aimed to show the usability of audio input as an additional perception modality to improve situational awareness of the surface environment where the robot is operating in. This can offer further information on the world and the state of objects in the robot's surrounding.

Our approach is divided into an audio and vision branch, which eventually are fused into a single state estimation of located sources withing known objects. We further introduces a method to compare the received spectral information with prior learned profiles. Moreover, the system is able to detect unknown profiles which are not part of the set of known data.

We showed the performance of our proposed method in simulation as well as the transferability in a real scenario. The sound source localization yields high accuracy in combination with the visual perception and results in robust fusion of the two modalities for spectral profile and state estimation. Deployed to our laboratory, the system was able to detect and estimate the current state of the target object.

The presented system shall be integrated into the perceptional system of the robotic assistance platforms and provide the audio modality in the upcoming Surface Avatar ISS-Earth telerobotic experiment sessions in 2023-2024.

REFERENCES

[1] Lii, N. Y. et al., "Introduction to Surface Avatar: the First Heterogeneous Robotic Team to be Commanded with Scalable Autonomy from the ISS," in *Proceedings of the 73rd International Astronautical Congress (IAC)*. International Astronautical Federation, September 2022.

[2] L. A. Jeffress, "A place theory of sound localization." *Journal of comparative and physiological psychology*, vol. 41, no. 1, p. 35, 1948.

[3] J. Huang, N. Ohnishi, and N. Sugie, "Building ears for robots: sound localization and separation," *Artificial Life and Robotics*, vol. 1, no. 4, pp. 157–163, 1997.

[4] K. Nakadai, K.-i. Hidai, H. Mizoguchi, H. G. Okuno, and H. Kitano, "Real-time auditory and visual multiple-

object tracking for humanoids," in *International Joint Conference on Artificial Intelligence*, vol. 17, no. 1. LAWRENCE ERLBAUM ASSOCIATES LTD, 2001, pp. 1425–1436.

[5] K. Nakadai, D. Matsuura, H. G. Okuno, and H. Kitano, "Applying scattering theory to robot audition system: Robust sound source localization and extraction," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 2. IEEE, 2003, pp. 1147–1152.

[6] J. A. MacDonald, "A localization algorithm based on head-related transfer functions," *The Journal of the Acoustical Society of America*, vol. 123, no. 6, pp. 4290–4296, 2008.

[7] F. Keyrouz, Y. Naous, and K. Diepold, "A new method for binaural 3-d localization based on hrtfs," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 5. IEEE, 2006, pp. V–V.

[8] I. Kossyk, M. Neumann, and Z.-C. Marton, "Binaural bearing only tracking of stationary sound sources in reverberant environment," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 53–60.

[9] J.-M. Valin, F. Michaud, J. Rouat, and D. Létourneau, "Robust sound source localization using a microphone array on a mobile robot," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 2. IEEE, 2003, pp. 1228–1233.

[10] J.-M. Valin, F. Michaud, B. Hadjou, and J. Rouat, "Localization of simultaneous moving sound sources for mobile robot using a frequency-domain steered beamformer approach," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 1. IEEE, 2004, pp. 1033–1038.

[11] E. Mumolo, M. Nolich, and G. Vercelli, "Algorithms for acoustic localization based on microphone array in service robotics," *Robotics and Autonomous systems*, vol. 42, no. 2, pp. 69–88, 2003.

[12] R. Roden, N. Moritz, S. Gerlach, S. Weinzierl, and S. Goetze, *On sound source localization of speech signals using deep neural networks*, 2015.

[13] S. Adavanne, A. Politis, and T. Virtanen, "Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 1462–1466.

[14] X. Xiao, S. Zhao, X. Zhong, D. L. Jones, E. S. Chng, and H. Li, "A learning-based approach to direction of arrival estimation in noisy and reverberant environments," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 2814–2818.

[15] R. Takeda and K. Komatani, "Discriminative multiple sound source localization based on deep neural networks using independent location model," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 603–609.

[16] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[17] R. Roy and T. Kailath, "Esprit-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 37, no. 7, pp. 984–995, 1989.

[18] S. Argentieri and P. Danes, "Broadband variations of the music high-resolution method for sound source localization in robotics," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 2009–2014.

[19] F. Asono, H. Asoh, and T. Matsui, "Sound source localization and signal separation for office robot" jijo-2"," in *Proceedings. 1999 IEEE/SICE/RSJ. International Conference on Multisensor Fusion and Integration for Intelligent Systems. MFI'99 (Cat. No. 99TH8480)*. IEEE, 1999, pp. 243–248.

[20] C. T. Ishi, O. Chatot, H. Ishiguro, and N. Hagita, "Evaluation of a music-based real-time sound localization of multiple sound sources in real noisy environments," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 2027–2032.

[21] K. Nakamura, K. Nakadai, and G. Ince, "Real-time super-resolution sound source localization for robots," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 694–699.

[22] M. Sewtz, T. Bodenmüller, and R. Triebel, "Robust music-based sound source localization in reverberant and echoic environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2474–2480.

[23] L. S. M. Sugai, C. Desjonqueres, T. S. F. Silva, and D. Llusia, "A roadmap for survey designs in terrestrial acoustic monitoring," *Remote Sensing in Ecology and Conservation*, vol. 6, no. 3, pp. 220–235, 2020.

[24] L. S. M. Sugai, T. S. F. Silva, J. W. Ribeiro Jr, and D. Llusia, "Terrestrial passive acoustic monitoring: review and perspectives," *BioScience*, vol. 69, no. 1, pp. 15–25, 2019.

[25] D. Llusia, R. Márquez, and R. Bowker, "Terrestrial sound monitoring systems, a methodology for quantitative calibration," *Bioacoustics*, vol. 20, no. 3, pp. 277–286, 2011.

[26] E. P. Kasten, S. H. Gage, J. Fox, and W. Joo, "The remote environmental assessment laboratory's acoustic library: An archive for studying soundscape ecology," *Ecological informatics*, vol. 12, pp. 50–67, 2012.

[27] R. Kojima, O. Sugiyama, R. Suzuki, K. Nakadai, and C. E. Taylor, "Semi-automatic bird song analysis by spatial-cue-based integration of sound source detection, localization, separation, and identification," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1287–1292.

[28] A. Digby, M. Towsey, B. D. Bell, and P. D. Teal, "A practical comparison of manual and autonomous methods for acoustic monitoring," *Methods in Ecology and Evolution*, vol. 4, no. 7, pp. 675–683, 2013.

[29] R. Suzuki, S. Matsubayashi, R. W. Hedley, K. Nakadai, and H. G. Okuno, "Harkbird: Exploring acoustic interactions in bird communities using a microphone array," *Journal of Robotics and Mechatronics*, vol. 29, no. 1, pp. 213–223, 2017.

[30] C. Astaras, J. M. Linder, P. Wrege, R. D. Orume, and D. W. Macdonald, "Passive acoustic monitoring as a law

enforcement tool for afrotropical rainforests," *Frontiers in Ecology and the Environment*, vol. 15, no. 5, 2017.

[31] J. S. Ulloa, T. Aubin, D. Llusia, C. Bouveyron, and J. Sueur, "Estimating animal acoustic diversity in tropical environments using unsupervised multiresolution analysis," *Ecological Indicators*, vol. 90, pp. 346–355, 2018.

[32] L. W. Koester, H. Taheri, L. J. Bond, and E. J. Faierson, "Acoustic monitoring of additive manufacturing for damage and process condition determination," in *AIP Conference Proceedings*, vol. 2102, no. 1. AIP Publishing LLC, 2019, p. 020005.

[33] M. S. Hossain and H. Taheri, "In situ process monitoring for additive manufacturing through acoustic techniques," *Journal of Materials Engineering and Performance*, vol. 29, no. 10, pp. 6249–6262, 2020.

[34] J. Bynum and D. Lattanzi, "Combining convolutional neural networks with unsupervised learning for acoustic monitoring of robotic manufacturing facilities," *Advances in Mechanical Engineering*, vol. 13, no. 4, p. 16878140211009015, 2021.

[35] Carey, W. et al., "Analog-1: A touch remote," in *Proceedings of the 73rd International Astronautical Congress (IAC)*. International Astronautical Federation, September 2022.

[36] Wedler, A. et al., "Finally! Insights into the ARCHES lunar planetary exploration analogue campaign on etna in summer 2022," in *Proceedings of the 73rd International Astronautical Congress (IAC)*. International Astronautical Federation, September 2022.

[37] T. Krueger, E. Ferreira, A. Gherghescu, L. Hann, E. den Exter, F. P. van der Hulst, L. Gerdes, A. Pereira, H. Singh, M. Panzirsch *et al.*, "Designing and testing a robotic avatar for space-to-ground teleoperation: the developers' insights," in *71st International Astronautical Congress, IAC 2020*. International Astronautical Federation, 2020.

[38] N. Y. Lii, D. Leidner, P. Birkenkampf, B. Pleintinger, R. Bayer, and T. Krueger, "Toward scalable intuitive telecommand of robots for space deployment with meteron supvis justin," 2017.

[39] C. Borst, T. Wimbock, F. Schmidt, M. Fuchs, B. Brunner, F. Zacharias, P. R. Giordano, R. Konietschke, W. Sepp, S. Fuchs *et al.*, "Rollin'justin-mobile platform with variable base," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 1597–1598.

[40] V. et al., "An ecosystem for heterogeneous robotic assistants in caregiving: Core functionalities and use cases," *IEEE Robotics & Automation Magazine*, vol. 28, no. 3, pp. 12–28, 2021.

[41] M. Sewtz, T. Bodenmüller, and R. Triebel, "Design of a microphone array for rollin justin," in *ICRA Workshop*, 2019.

[42] R. Bayer, P. Schmaus, M. Pfau, B. Pleintinger, D. Leidner, F. Wappler, A. Maier, T. Krueger, and N. Y. Lii, "Deployment of the solex environment for analog space telerobotics validation," in *Proceedings of the International Astronautical Congress, IAC*, 2019.

[43] J. Wang and E. Olson, "Apriltag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4193–4198.

[44] M. Sewtz, X. Luo, J. Landgraf, T. Bodenmüller, and R. Triebel, "Robust approaches for localization on multi-camera systems in dynamic environments," in *2021 7th International Conference on Automation, Robotics and Applications (ICARA)*. IEEE, 2021, pp. 211–215.

[45] J. Ramırez, J. C. Segura, C. Benıtez, A. De La Torre, and A. Rubio, "Efficient voice activity detection algorithms using long-term speech information," *Speech communication*, vol. 42, no. 3-4, pp. 271–287, 2004.

[46] P. Schmaus, D. Leidner, T. Krüger, R. Bayer, B. Pleintinger, A. Schiele, and N. Y. Lii, "Knowledge driven orbit-to-ground teleoperation of a robot coworker," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 143–150, 2019.

[47] M. Vorländer, *Auralization*. Springer, 2020.

BIOGRAPHY

***Marco Sewtz** received his B.Eng. degree in electrical engineering at the University of Applied Sciences of Munich and his M.Sc. degree at the Technical University of Munich. He works at the Institute for Robotics and Mechatronics at the German Aerospace Center (DLR) as a researcher since 2018. His interests focuses on SLAM and multi-model perception of the environment. Before his current role, he worked as an electrical designer for high-performance processing modules for space hardware at Airbus Defence and Space.*

***Werner Friedl** received his Dipl.-Ing.(FH) in Mechatronic at the University of applied science In Munich and starts at DLR in 2004. 2006 he develop the torso of DLR Humanoid Justin. In the DLR Hand-Arm- project he developed the forearm of the AWIWI hand and AWIWI II. Since 2015 he is responsible for the mechanical hand development at DLR. His main research focus includes variable stiffness actuation, tendon driven hands and grasping.*

***Adrian Bauer** received a Bachelor in Mechanical Engineering in 2012, a Bachelor in Cognitive Sciences in 2015, and a master in Robotics, Cognition, Intelligence from TU Munich in 2018. Currently he is pursuing a PhD in robotics at the German Aerospace Center. His interest is in enabling robotics to generate meaningful symbolic plans in presence of epistemic uncertainty.*

***Anne Köpken** received a Bachelor in Electrical Engineering from TU Munich in 2019, and a Master in Robotics, Cognition, Intelligence from TU Munich in 2021. She spent one semester at the JSK Laboratory at the University of Tokyo in 2019/20. Currently she is pursuing a PhD in robotics at the German Aerospace Center in Oberpfaffenhofen near Munich. She is interested in enabling robots to cope with unexpected situations and finding ways to prevent and recover from failures.*

***Florian S. Lay** received his B.Sc. degree in Engineering Science in 2018, and his M.Sc. in "Robotics, Cognition, Intelligence" in 2020 both from the Technical University of Munich. Since 2020 he is pursuing a PhD in robotics at the German Aerospace Center (DLR). His interests range from symbol grounding and emergence for task and motion planning to multi-robot world representations.*

***Nicolai Bechtel** received his Master of Science in Computational Engineering from the University of Applied Sciences Munich in 2018. Since then, he has been conducting research in the field of haptics and virtual reality as a research assistant at the Center for Robotics and Mechatronics of the German Aerospace Center (DLR) in Oberpfaffenhofen. His research focuses on haptics, multi-body-dynamic simulations, and development of virtual reality environments. He is currently working on topics such as Model-Based Teleoperation and GUI development involving Augmented Reality.*

***Peter Schmaus** received his M.Sc. Degree in "Robotics, Cognition, Intelligence" from Technical University of Munich, Germany, in 2013. He joined the German Aerospace Center (DLR) Institute of Robotics and Mechatronics in 2011 where he was involved in the ISS-to-ground telerobotics projects Kontur-2, METERON SUPVIS Justin, and became Co-Investigator of the Surface Avatar experiment suite. His main interests lie in Shared Autonomy and effective Human-Robot Interaction.*

***Rudolph Triebel** received his PhD in 2007 from the University of Freiburg in Germany. From 2007 to 2011, he was a postdoctoral researcher at ETH Zurich, where he worked on machine learning algorithms for robot perception. From 2011 to 2013 he worked in the Mobile Robotics Group at the University of Oxford. In 2015, he was appointed as leader of the Department of Perception and Cognition at the Robotics Institute of DLR.*

***Neal Y. Lii** is the domain head of Space Robotic Assistance, and the co-founding head of the Modular Dexterous (Modex) Robotics Laboratory at the German Aerospace Center (DLR). Neal received his BS, MS, and PhD degrees from Purdue University, Stanford University, and University of Cambridge, respectively. He has served as the principal investigator of the ISS-to-Earth telerobotic experiments, Surface Avatar, and METERON SUPVIS Justin. Neal is primarily interested in the use of telerobotics in both space and terrestrial applications.*

7. A Structured Approach for Uncertain Transformations Trees

Authors:

Marco Sewtz, Lukas Burkhard, Xiaozhou Luo, Leon Dorscht and Rudolph Triebel

Conference:

Sewtz, Marco, et al. "A Structured Approach for Uncertain Transformations Trees." 2024 International Conference on Automation, Robotics and Applications, ICARA 2024. IEEE, 2024.

Award:

This publication has been awarded to be submitted to a journal

Abstract:

In the field of robotics, ensuring precise representation of spatial transformations is imperative for maintaining reliable system performance. However, conventional approaches often prove inadequate due to their failure to consider internal inaccuracies in the robot and environmental factors. In the context of robotic systems, deviations from nominal transformations arise from various sources such as sensor decalibration, inaccuracies in joint positions, deformations induced by mechanical stress, and gravitational influences, among other contributing factors. The same applies to environmental uncertainties, where the registered poses of objects and landmarks suffer from limitations in the perception methods. This paper advocates for a paradigm shift by introducing a framework that incorporates uncertainty into transformation trees, utilizing Lie Algebra for a consistent computation. Our approach addresses the aforementioned challenges, providing a realistic and robust representation of transformations. We demonstrate the applicability and efficacy of our framework through real-world examples.

Contributions:

The author of this dissertation designed the software architecture, the tree structure and the experimental setup for evaluation. The mathematical framework was provided by Lukas Burkhard. Support for the software development was done by Leon Dorscht, support for the evaluation was done by Xiaozhou Luo. The script was provided by the author and the publication was presented by the author.

Copyright:

©2024 IEEE

A Structured Approach for Uncertain Transformations Trees

Marco Sewtz, Lukas Burkhard, Xiaozhou Luo, Leon Dorscht, Rudolph Triebel
Institute of Robotics and Mechatronics
German Aerospace Center (DLR)
Wessling, Germany
{firstname.lastname}@dlr.de

***Abstract*—In the field of robotics, ensuring precise representation of spatial transformations is imperative for maintaining reliable system performance. However, conventional approaches often prove inadequate due to their failure to consider internal inaccuracies in the robot and environmental factors. In the context of robotic systems, deviations from nominal transformations arise from various sources such as sensor decalibration, inaccuracies in joint positions, deformations induced by mechanical stress, and gravitational influences, among other contributing factors. The same applies to environmental uncertainties, where the registered poses of objects and landmarks suffer from limitations in the perception methods. This paper advocates for a paradigm shift by introducing a framework that incorporates uncertainty into transformation trees, utilizing Lie Algebra for a consistent computation. Our approach addresses the aforementioned challenges, providing a realistic and robust representation of transformations. We demonstrate the applicability and efficacy of our framework through real-world examples.**

***Index Terms*—robotics, transformation tree, uncertainty modeling, Lie Algebra**

I. INTRODUCTION

In the dynamic landscape of robotics, accurately representing spatial transformations is pivotal for reliable system performance. Conventional methods, which treat provided transformations as precise and deterministic, face difficulties in coping with inherent inaccuracies within the system and environmental complexities. This paper underscores the critical need for inaccuracies-aware spatial representations in robotics, often denoted as scene graphs. These representations allow modeling not only the spatial relationships in a robot-environment system but also our missing knowledge about it.

An illustrative instance can be found in the distinction between a robotic arm's repetition accuracy, which signifies its capability to consistently reach the same point in a workspace, and the robot's absolute accuracy. There, the first can be assumed to be "exact" for conventional robotic systems. However, the error of the latter can be higher by several orders of magnitude, motivating the modeling of the error. Position measurements constrained by both physical limitations and environmental influences, frequently fall short of the requisite precision. This constraint becomes especially critical in applications requiring high accuracy, such as surgical robotics.

An additional example is the process of registering a robot with respect to its environment, a task achieved through either an inaugural calibration procedure or by means of the navigation implemented in mobile robotic systems.

Interestingly, various scholarly works have considered robot uncertainty within specific domains, such as the kinematic structure or autonomous navigation components.However, there is limited progress in combining these several domains into one single representation like a scene graph to have a unified consideration of inaccuracy-aware spatial relations. Conventional approaches that disregard uncertainty in scene graphs fall short in capturing the intricacies of real-world scenarios.

This paper advocates for a paradigm shift by introducing a framework that incorporates uncertainty into scene graphs, offering a more realistic and robust representation of transformations. By addressing challenges posed by both robot internal inaccuracies and the uncertainty of the robot's interaction with the environment, our approach aims to enhance the reliability and performance of robotic systems in practical applications.

We use the following terminology in this paper: Robotic systems can be subject to errors that cause *inaccurate* pose calculations, either within the system or with respect to its environment. A common simplification is to model such inaccuracies in a probabilistic way, thus subjecting nominal relative poses to an additional *uncertainty*. For a multitude of robotic applications, such uncertainty is modeled as a *zero-mean normal distribution*, thus an uncertain pose consists of a nominal pose and a covariance matrix. Generally, this simplification trades the exact representation of robotic errors for the availability of powerful mathematical tools and is thus well established in the robotic community. We adopt this error modeling as well, which allows us to immediately integrate the probabilistic pose information from other software components into our scene graph.

II. RELATED WORK

Accurately describing the spatial relationships of a robot and its environment is a key aspect of robotics specifically

This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy (StMWi) by means of the project SMiLE2gether (LABAY102).

©2024 IEEE Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

and mechanical mechanisms generally.

Commencing with the early explorations in formulating a framework for kinematics in mechanical structures [1], [2], the field witnessed significant strides with one of the pivotal works by Denavit and Hartenberg [3]. In this groundbreaking contribution, the authors devised a structured yet elegant methodology to comprehensively describe the chain of transformations associated with robotic arms. Subsequent endeavors augmented the toolbox of robot kinematics representation, for example by considering the underlying Lie-Algebra of spacial transformations [4]. Our recent work [5][1] provides a kinematic robot description that allows to consider the inaccuracies from joint position measurements, mechanical stress-induced deformations, and gravitational influences in a probabilistic manner.

In the field of robotic navigation, many approaches already consider the uncertainty of relative transformations, especially in the area of SLAM where e.g. [6] or [7] use the covariance or information matrix, respectively, to weigh different spatial transformations in a graph optimization.

The interaction of a robot with objects in its environment, specifically the uncertainties inherent in the workspace, has been investigated in [8]. Additionally, notable strides have been made in recent research towards modeling the uncertainty embedded within the perception process of classical [9] and deep-learning-based [10][1] methods.

Finally, the hand-eye-calibration of a robot is nothing else but an additional transformation between the real and the nominal robot geometry, and can thus also be subject to inaccuracies, as discussed by [11].

In the end, all these sub-fields of robotics provide a multitude of different types of spatial transformations, where potentially all of them are subjected to errors which are being modeled as uncertainties.

Systematic approaches to order a multitude of interconnected transformations, particularly within the realm of virtual reality (VR) [12], [13], and robotic simulators [14], [15], considered the utilization of a scene graph to represent relative spatial relationships. This scene graph, akin to a tree structure, comprises multiple nodes arranged in a parent-child manner. This innovative approach enhanced the representation and simulation capabilities in both virtual reality and robotic simulation domains. The current state of the art is *tf* [16], the scene graph framework of ROS (robot operating system).

Interestingly, very little work has been published that considers the uncertainty of spatial information by interconnecting the different realms of robotics. Initial efforts have been directed towards acknowledging uncertainty within the scene graph, for example [17]. However, these early attempts typically fall short in correctly modeling the error propagation using Lie Algebra. Alternatively, some implementations resort to sampling-based approaches to represent the overall uncertainty within the system, such as [18], which however comes with computational costs.

The Lie-Algebra allows to acknowledge the manifold character of spatial relationships and is a powerful tool to compute and propagate uncertainty along chains of spatial transformations. An introduction to it together with the application to robotic navigation is provided by [19]. Similarly, Lie-Algebra-based concepts are provided for the error propagation within robotic manipulators, either for single errors [20] or as our comprehensive kinematic model [5].

Despite the widespread use of Lie Algebra in uncertainty estimation, to the best of our knowledge, no existing approach formulating a scene graph for robotics has integrated Lie Algebra-based uncertainty propagation. In our ongoing work, we aim to address this gap and demonstrate the efficacy of incorporating Lie Algebra into a scene graph framework for a more nuanced and accurate representation of uncertainty in kinematic systems.

III. ROBOTIC AND ENVIRONMENTAL CONFIGURATION STATE

Accurate assessment of the current configuration state in robotic systems holds significant importance across various applications. This is particularly pronounced in scenarios involving non-static components equipped with perception sensors. Registering cameras affixed to robotic manipulators to the robot's origin is imperative for seamlessly integrating spatial information within the correct coordinate framework. Knowledge of the system's distance to the environment is indispensable for collision avoidance, especially when navigating confined spaces. To achieve this, it's crucial to carefully observe and organize the positions of joints into a transformation tree. This tree helps illustrate how the coordinate framework depends on a specified starting point known as the root frame and obtaining an estimate of the robot's spatial volume. However, overlooking the inherent uncertainty in these measurements and the subtle non-static characteristics of certain links—attributable to mechanical stress and gravitational forces—can lead to erroneous state estimations. In the ensuing discussion, we elaborate on representing the robotic and environmental configuration state (RECS) as a transformation tree. Subsequently, we introduce Lie Algebra as a robust solution for modeling uncertainty in this process. Finally, we detail our implementation of a managed and centralized approach for addressing the RECS problem within an inter-process communication (IPC) framework.

A. Transformation Tree

Deriving the transformation between two coordinate frames is a pivotal task in robotics. A widely employed approach involves modeling the system as a hierarchical tree of frame transformations, as seen in the example Figure 1. This facilitates information extraction from the CAD model, allowing for the calculation of spatial offsets between structural points.

A key optimization involves consolidating static displacements into a singular transformation, pruning the tree for computational efficiency. Movable connections are represented as rotations or translations centered around joints, contributing

---

[1] Now known as L. Burkhard, *et al.*

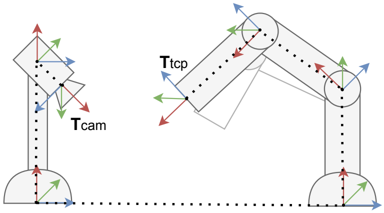


Fig. 1: An illustrated exampled of a robotic manipulator and an external camera. The transformation from the camera to the tool center point of the robot can be calculated concatenating all individual frame transformations.

to a chain of static links and dynamic joints. This approach not only streamlines computational complexity but also provides a comprehensive understanding of a robotic system's kinematic properties, enhancing efficiency and reliability.

Following the comprehensive description of robot kinematics within the previously mentioned tree structure, the process of retrieving the direct transformation between any two arbitrary frames unfolds by traversing the path articulated within this structured tree. This systematic approach ensures a clear and methodical procedure for obtaining the specific transformation information required for precise spatial relationships between frames within the robotic system.

B. *Transformations and Uncertainty*

Our treatment of uncertainties follows our previous work on probabilistic robot kinematics [5], which in turn builds upon the mathematical foundations provided by [19] and [21].

We briefly introduce the applied methods here, but refer the interested reader to the related works for more thorough insights. For a general introduction to Lie Algebra in the scope of robotics, we recommend the excellent [22], who's notation we mostly follow.

A pose $\boldsymbol{T}_{AB} \in SE(3)$ describes the position and orientation of an object B with respect to a reference frame A. While a pose quantity is generally an element of the manifold $SE(3)$, it can be described *locally* by its linear tangent space representation $\boldsymbol{\xi} = [\boldsymbol{\rho}\,\boldsymbol{\theta}]^T \in \mathbb{R}^6$, related by the exponential map [22]

$$\boldsymbol{T} = \mathrm{Exp}(\boldsymbol{\xi}). \tag{1}$$

There, $\boldsymbol{\rho}$ denotes the translational and $\boldsymbol{\theta}$ the rotational component of the tangent space element. Local tangent space quantities can be mapped between two different local spaces using the *adjoint matrix* $\mathbf{Ad}$ as

$${}^{A}\boldsymbol{\xi} = \mathbf{Ad}(\boldsymbol{T}_{AB})\,{}^{B}\boldsymbol{\xi}, \tag{2}$$

with

$$\mathbf{Ad} = \begin{bmatrix} \boldsymbol{R} & [\boldsymbol{t}]_\times \boldsymbol{R} \\ \mathbf{0} & \boldsymbol{R} \end{bmatrix} \in \mathbb{R}^{6\times 6}, \tag{3}$$

where $\boldsymbol{R}$ being the rotation matrix of $\boldsymbol{T}$ an $[\boldsymbol{t}]_\times$ the skew-symmetric matrix formed by the translation. The term $[\boldsymbol{t}]_\times \boldsymbol{R}$ illustrates, how local rotation errors create translation errors further down a chain of transformations, with the magnitude depending on the distance from the original error's location.

Recall that we describe the error of a pose as local deviation $\boldsymbol{\xi}_{\text{B,err}}$ of a nominal pose $\boldsymbol{T}_{AB}$, i.e., in the tangent space of the pose's reference frame B. The corresponding covariance matrix $\mathbf{\Sigma}_{AB} = \mathbb{E}\big[\boldsymbol{\xi}_{\text{B,err}}\,\boldsymbol{\xi}_{\text{B,err}}^T\big] \in \mathbb{R}^{6x6}$ is therefore a locally defined tangent space quantity.

The two mathematical operations on poses, which are needed for the scene graph, are thus defined in these terms. The *concatenation* is computed as

$$\boldsymbol{T}_{AC} = \boldsymbol{T}_{AB} * \boldsymbol{T}_{BC} \tag{4}$$

$$\mathbf{\Sigma}_{AC} = \mathbf{Ad}_{\boldsymbol{T}_{BC}^{-1}} \mathbf{\Sigma}_{AB} \mathbf{Ad}_{\boldsymbol{T}_{BC}^{-1}}^T + \mathbf{\Sigma}_{BC}. \tag{5}$$

Note that the two covariance matrices are transported into the common reference frame C using the adjoint matrix, where they can be added due to the linearity of the tangent space. The covariance composition eq. (5) is a first order approximation (called *second* order in some publications) and is discussed in detail in [19].

Analogously, the *inverse* is computed as

$$\boldsymbol{T}_{BA} = \boldsymbol{T}_{AB}^{-1} \tag{6}$$

$$\mathbf{\Sigma}_{BA} = \mathbf{Ad}_{\boldsymbol{T}_{AB}} \mathbf{\Sigma}_{AB} \mathbf{Ad}_{\boldsymbol{T}_{AB}}^T, \tag{7}$$

shifting the uncertainty from the tangent space of B in the tangent space A. We omit the discussion on the specific modeling of probabilistic rover kinematics here and refer the reader to our previous publication [5]. Note that this representation can implicitly also consider *exact* transformations, as zero-covariances simply vanish in eq. (5) and eq. (7).

C. *Implementation*

The presented methodology has been implemented within a C++ library, and the corresponding source code is accessible online[2]. Further, a wrapper for the scripting language Python is provided. Each coordinate frame is characterized by a node-element. A frame is precisely defined by its pose matrix $\boldsymbol{T}$ and an accompanying covariance matrix $\mathbf{\Sigma}$ which may be set to zero for precisely known transformations. Distinctive identification of each frame is facilitated through the application of a unique character string. Furthermore, the mathematical operations of *concatenation* and *inverse* for each frame are executed leveraging the computational capabilities provided by the manif library [22] augmented by the uncertainty propagation.

The hierarchical structure is implemented using the Boost.Graph data structure.Each vertex encapsulates a frame as its payload, and the edges define the direction of transformations. To determine a path between two nodes within the tree, a breadth-first search (BFS) routing algorithm is employed. The cumulative transformation along the identified path is computed based on the direction specified by the graph's edges, facilitating a comprehensive understanding of

---

[2] https://rmc.dlr.de/rm/en/staff/marco.sewtz/software

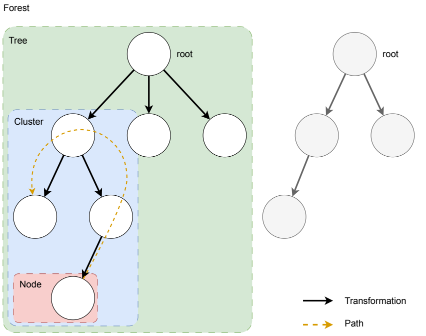


Fig. 2: A schematic overview of the tree structure holding all transformation information. The whole system is consisting of separate trees that do not share any connection. Each tree is constructed by child nodes that are added by directed transformations to their parent node. Further, neighboring nodes can be grouped to a cluster. A transformation between non-neighboring nodes is described by a path.

the transformations between the starting and ending points of the path.

The system allows for the addition of additional root nodes, thereby declaring new trees that remain disconnected from preceding ones. It is imperative to underscore that the establishment of a path between nodes situated on distinct trees within the forest is not feasible. Each root node initiates an independent tree structure, and inter-tree connectivity is explicitly precluded within the system's framework.

The default operational paradigm involves centralized control over all trees, nodes, and computations via a central server. A connected client possesses the capability to perform operations such as creation, retrieval, updating, or deletion of nodes. Additionally, the client can request the cumulative transformation of a specific path. An added feature allows the definition of a local cluster within a tree, enabling the transfer of ownership from the server to a designated client. Consequently, the client gains the ability to locally compute a path within this cluster without necessitating network calls for information retrieval, thereby enhancing computational speed for that particular client. Other clients will be still able to access this information however it must be routed through the server. An illustration of this architecture is given in Figure 2.

IV. APPLICATION

To demonstrate the practical utility of the proposed framework, two examples of application will be illustrated in the following. An in-depth analysis of the applying Lie Algebra to the configuration modeling problem has been presented in [5], therefore we want to focus on the scene-graph implementation. At first, the initial application showcases the integration on a robotic arm affected by bending introduced by gravitational pull of the Earth. The second instance will illustrate a mapping application on a system featuring an uncertain RECS, formulated as a graph optimization problem.

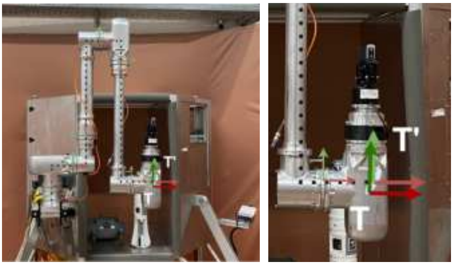


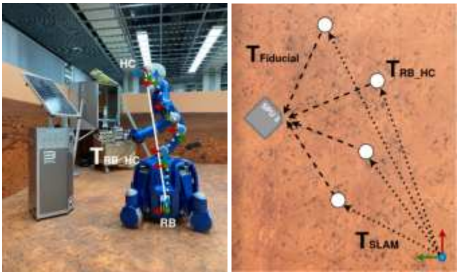
Fig. 3: TINA arm bending due to gravitation. The computed position, designated as $\mathbf{T}'$, represents the theoretical location without accounting for uncertainties.

A. Uncertain robotic and environmental configuration state

As an integral component of the European Space Agency (ESA) project for a Sample Transfer Arm breadboard study, the German Aerospace Center (DLR) developed the TINA manipulator [23] as a compact, modular, and torque-controlled robotic system designed to adhere to the requirements of the Mars Sample Return mission. Figure 3 illustrates the robotic arm in its initial position mounted on a lander. Upon closer inspection, it becomes evident that the manipulator, even in its initial configuration, experiences moderate deformations attributable to its own weight and joint play, particularly in the axial direction. As a result, the pose of the end effector is subjected to several uncertainties, which can be modeled with the proposed framework. By incorporating the expected variance parameters into the transformation tree, the state of the robot configuration can be predicted probabilistically, and the position of the end effector is constrained to an anticipated uncertainty region. Consequently, the consideration of uncertainties provides a more realistic depiction of the arm's pose, acknowledging the impact of various factors, including gravitational forces, and enhances the accuracy of the positional assessment, enabling more precise manipulations. The selection of adequate probabilistic parameters heavily depends on the associated system's specific characteristics and requires specialized technical knowledge. If necessary, an experimental evaluation has to be conducted to validate and fine-tune these parameters.

B. Environmental Mapping

To enable more intricate manipulations and interactions between the robot and its environment, a significant challenge lies in achieving precise registration of the robot relative to its surroundings. This entails aligning various world representations generated for different types of tasks to ensure coherence and accuracy in the robot's perception of its environment.



(a) (b)

Fig. 4: Rollin' Justin mapping a SPU in a Martian environment (a) and the associated optimization graph is represented in (b). The uncertainty-ridden transformation is summarized as T_{RB_HC} from robot base (RB) to the head camera (HC), from which fiducials associated to the SPU are registered.

As depicted in Figure 4a, Rollin' Justin [24] is mapping a Smart Payload Unit (SPU) in a Martian surroundings. In addition to the unknown state of the environmental configuration, a further challenge arises from within the robot. Although the upper body assembly is rigidly connected to the base platform, the wire rope construction in different parts of the torso is inherently less precise than the rigid joints of the arms, introducing uncertainties into the robot's configuration state. Effectively managing and mitigating this uncertainty is crucial since information for navigation purposes is collected from sensors in the base, while other higher-level tasks, e.g., object recognition and manipulation, rely on information from the camera mounted in Justin's head. Therefore, modeling the spatial relations of the robot configuration state, including uncertainties, is essential and can be addressed by the proposed framework. It is further capable of simplifying the handling of transformations and their associated uncertainties by summarizing them into one single step.

In the context of environmental mapping, the transformation from the robot base to the head camera becomes particularly critical as it serves as the foundation for registering fiducials linked to the SPU. Combined with the spatial relationship to the registered fiducials and information regarding the global reference provided by MROSLAM [25], an optimization graph can be constructed, as illustrated in Figure 4b. The optimization problem can be effectively addressed using GTSAM [26] or comparable algorithms, leading to an optimized estimation of the SPU's pose. This comprehensive approach significantly improves the reliability and quality of environmental mapping outcomes in the robot's operational context.

V. CONCLUSION

We present a Lie Algebra-based framework for uncertainty estimation, realized as a transformation tree. Our work develops a scene-graph-like structure and details the library implementation. Real-world examples demonstrate practical applicability, and comparative analysis highlights method superiority. This contribution enhances robotic transformations, offering a versatile tool for improved reliability and performance.

Future work includes temporal deviation modeling for enhanced capabilities, enabling configuration retrieval from previous timesteps. We aim to align the interface with ROS's *tf* implementation for seamless integration.

REFERENCES

[1] Kennedy, *The Kinematics of Machinery*. New York: D. Van Nostrand, 1881.
[2] Calvert, *Developing problem-solving skills in engineering*, 1953.
[3] Denavit and Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," 1955.
[4] Richard, *A Mathematical Introduction to Robotic Manipulation*, 1994.
[5] Meyer et al., "The Probabilistic Robot Kinematics Model and its Application to Sensor Fusion," 2022.
[6] Kaess et al., "isam2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, 2012.
[7] Kummerle et al., "g2o0: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*. IEEE.
[8] Su et al., "Manipulation and propagation of uncertainty and verification of applicability of actions in assembly tasks," *IEEE Transactions on Systems, Man, and Cybernetics*, 1992.
[9] Stoiber et al., "A sparse gaussian approach to region-based 6dof object tracking," in *Proceedings of the Asian Conference on Computer Vision*, 2020.
[10] Meyer et al., "Robust probabilistic robot arm keypoint detection exploiting kinematic knowledge," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop on Probabilistic Robotics in the Age of Deep Learning*, 2022.
[11] Nguyen et al., "On the covariance of X in AX = XB," *IEEE Transactions on Robotics*, 2018.
[12] Carlsson et al., "Dive—a platform for multi-user virtual environments," *Computers & graphics*, vol. 17, no. 6, pp. 663–669, 1993.
[13] Tramberend, "Avocado: A distributed virtual reality framework," in *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*. IEEE, 1999, pp. 14–21.
[14] Browning et al., "Übersim: a multi-robot simulator for robot soccer," in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, 2003, pp. 948–949.
[15] Drumwright et al., "Extending open dynamics engine for robotics simulation," in *Simulation, Modeling, and Programming for Autonomous Robots: Second International Conference*. Springer, 2010.
[16] Foote, "tf: The transform library," in *IEEE Conference on Technologies for Practical Robot Applications (TePRA)*. IEEE, 2013.
[17] Coelho et al., "Osgar: A scene graph with uncertain transformations," in *IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE, 2004.
[18] T. Ruehr, "uncertain tf," https://github.com/ruehr/uncertain_tf, 2013, last accessed 2023-11-30.
[19] Barfoot et al., "Associating uncertainty with three-dimensional poses for use in estimation problems," 2014.
[20] Yunfeng et al., "Error propagation on the euclidean group with applications to manipulator kinematics," *IEEE Transactions on Robotics*, 2006.
[21] ——, "Nonparametric second-order theory of error propagation on motion groups," *The International Journal of Robotics Research*, 2008.
[22] Sol et al., "A micro Lie theory for state estimation in robotics," *CoRR*, 2018.
[23] Maier et al., "Tina: The modular torque controlled robotic arm - a study for mars sample return," in *2021 IEEE Aerospace Conference (50100)*, 2021.
[24] Fuchs et al., "Rollin' justin - design considerations and realization of a mobile platform for a humanoid upper body," in *2009 IEEE International Conference on Robotics and Automation*, 2009.
[25] Sewtz et al., "Robust approaches for localization on multi-camera systems in dynamic environments," in *2021 7th International Conference on Automation, Robotics and Applications (ICARA)*, 2021.
[26] Dellaert et al., "borglab/gtsam," May 2022. [Online]. Available: https://github.com/borglab/gtsam)

8. Loosely-Coupled Multi-Sensor Visual Odometry: An Asynchronous Approach for Robust Household Robotics

Authors:

Marco Sewtz, Xiaozhou Luo, Tim Bodenmüller, Martin J. Schuster and Rudolph Triebel

Conference:

Sewtz, Marco et al. "Loosely-Coupled Multi-Sensor Visual Odometry: An Asynchronous Approach for Robust Household Robotics." Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2025. (Under Review)

Abstract:

Assistant robotics is an evolving field of research and promises support for humans in their everyday life. Household robots are designed to handle cleaning, maintenance, transportation, and monitoring tasks. However, typical indoor environments are challenging for the localization and navigation system as they lack suitable features, or mobile platforms have to operate close to obstacles. Furthermore, the hardware requirements must make robotic platforms affordable for household use. In this work, we propose an extension of single-sensor approaches to the multi-sensor case. It fuses various loosely-coupled and independent-running odometries, including vision-only and visual-inertial approaches, taking into account robust countermeasures for frequent loss-of-tracking (LoT) scenarios while preserving high accuracy. We minimized hardware requirements to only rigidly-connected visual sensors, eliminating the need for high-priced LIDAR or additional synchronization circuitry. We demonstrate the effectiveness of our approach in realistic experiments in a representative indoor environment and simulation.

Contributions:

The author of this dissertation designed and implemented the methodology of a graph-based approach. He provided the mathematical framework for weighted N-Slerp. The author designed and executed the evaluation. The script was provided by the author and the publication was presented by the author.

Copyright:

©2025 IEEE

A Robust Graph-Based Extension to Multi-Sensor Applications for Visual Odometry Systems in the Indoor Domain

Marco Sewtz[1], Xiaozhou Luo[1], Tim Bodenmüller[1], Martin J. Schuster[1] and Rudolph Triebel[1,2]

***Abstract*— Assistant robotics is an evolving field of research and promises support for humans in their everyday life. Household robots are designed to handle cleaning, maintenance, transportation, and monitoring tasks. However, typical indoor environments are challenging for the localization and navigation system as they lack suitable features, or mobile platforms have to operate close to obstacles. Furthermore, the hardware requirements must make robotic platforms affordable for household use. In this work, we propose an extension of single-sensor approaches to the multi-sensor case. It fuses various loosely-coupled and independent-running odometries, including vision-only and visual-inertial approaches, taking into account robust countermeasures for frequent loss-of-tracking (LoT) scenarios while preserving high accuracy. We minimized hardware requirements to only rigidly-connected visual sensors, eliminating the need for high-priced LIDAR or additional synchronization circuitry. We demonstrate the effectiveness of our approach in realistic experiments in a representative indoor environment and simulation.**

I. INTRODUCTION

Mobile robots designed for home assistance tasks demand robust ego-motion estimation and localization for successful deployment. Challenges persist in the form of limited landmarks, confined operational spaces, and close interaction with obstacles or humans, posing difficulties for localization methods.

Incorporating these robots seamlessly into everyday life on a large scale demands resource-saving and sustainable hardware design. Advancements in commercial-of-the-shelf (COTS) hardware have rendered the integration of multiple sensors into home assistants increasingly viable. Most commercial systems in indoor environments, such as server robots, rely on simple CMOS cameras for perception [1].

This work introduces a novel approach for robust odometry fusion on a robotic platform. This approach is based on ego-motion estimation using multiple cameras, as illustrated in Figure 1. The aim is to avoid adding any hardware requirements to the system design, thereby supporting cost-effective use of COTS components found in current systems. Notably, this methodology eliminates the need for trigger and timing circuitry, substituting them with software equivalents. Furthermore, it allows for flexibility in sensor positioning and orientation.

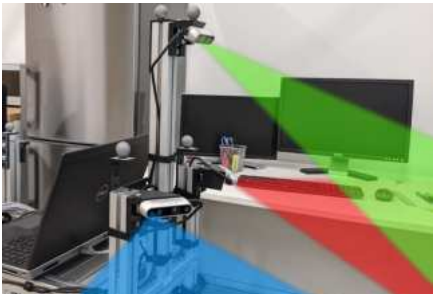


Fig. 1: Illustration of a multi-camera localization and navigation system used in an indoor environment. Each camera is individually mounted, with no requirements on position, orientation, trigger, or frequency.

The proposed two-step approach entails running a visual odometry (VO) module independently for each sensor, aiming to derive the locally-optimal estimate for each unique view. These individual estimates are then approximated using a time-continuous representation, acknowledging the distinctiveness of each sensor's perspective and addressing challenges like loss-of-tracking (LoT) in close proximity to obstacles.

In the second step, a fusion process is used to construct a joint motion model, leveraging the time-continuous approximation from each individual module. This results in a trajectory estimation that integrates distinctive information from each sensor. By independently processing and fusing sensor data, the approach enhances robustness and simplifies hardware requirements. We believe that this promotes resource-saving and sustainable architectures for mobile robots in home environments.

Our contributions are summarized as follows:

- A graph-based extension of state-of-the-art approaches using a unified fusion of multiple independent-running and loosely-coupled VOs that enables locally-optimal operation for each module while aiming for minimal system complexity.
- A generalization of the Slerp algorithm to the weighted n-elements case.
- Evaluation of the approach on a realistic dataset and in simulation.

This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy (StMWi) by means of the project SMiLE2gether (LABAY102).

[1] Marco Sewtz, Xiaozhou Luo, Tim Bodenmüller, Martin J. Schuster and Rudolph Triebel are with the Institute of Robotics and Mechatronics, German Aerospace Center, Wessling, Germany `{firstname.lastname}@dlr.de`

[2] Rudolph Triebel is with the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

II. RELATED WORK

Starting from early pioneering work on filter-based methods for ego-motion estimation [2], which introduced the concept of VO and stereo feature tracking [3], research has increasingly shifted towards graph-based formulations. Tong et al. [4] proposed a Gaussian process for modeling the robot's movement. The work of Lim et al. [5] proposes to use the FAST detector and BRIEF descriptor to decrease processing power to estimate the current state. This eventually resulted in ORB-SLAM3 [6]. In addition to its real-time capabilities, a robust approach for selecting *Keyframes* in the tracking thread contributes to the results of this work.

In particular, robust *Keyframe* selection has been a critical component for most systems, and a great number of fundamental research has been dedicated to it. With PTAM [7], the utilization of *Keyframes* was introduced to reduce the processing amount to only a subset of all available frames. The work of Lim et al. [5] further introduces the process of a double-window selection for local batch optimization. Lastly, a stable approach for *Keyframe* culling was introduced with VINS-Mono [8] and its extension VINS-Fusion was focused on finding the trade-off between sparse graphs and optimal representation.

In the field of multi-camera localization and mapping, Kaess and Dellaert [9] proposed filter-based methods to integrate eight camera views into the mapping process. Further, MultiCol-SLAM [10] introduces the formulation of *Multi-Keyframes* to construct a central graph of time-synchronized camera views. In contrast, Müller et al. [11] introduced a formulation for independent running VO systems, which yield *Keyframes* being sampled at different points in time. However, most multi-sensor or multi-modal systems, e.g., by Zhao et al. [12], Meng et al. [13], or Xu and Zhang [14], use a tightly coupled approach but require a central sensor that all other sensors depend on. Eckenhoff et al. [15] in addition approach this using multiple visual and inertial sensors in a tightly-coupled fashion to integrate online extrinsic calibration based on the IMU data. Lastly, in our latest work, we presented MROSlam [16] as a multi-sensor fusion approach running several instances of ORB-SLAM2 [17] in parallel, but not including a joint motion-model. However, all instances are run independently, and information is not exchanged between nodes. Therefore loop-closures can not be utilized as they lead to large offsets in unknown environments.

In summary, most previous work requires synchronous and centralized behavior, including simultaneous sampling of frames and respective *Keyframes*. Our objective is to propose a unified solution that utilizes multiple VO modules to estimate the robot's ego-motion without imposing the aforementioned requirements or expensive hardware. We enable asynchronous and independent operation of each sensor module to obtain an optimal motion estimate for each individual observable view. Our primary focus is to enhance the system's robustness without significantly increasing hardware requirements or system complexity.

III. METHODOLOGY

In our multi-sensor approach, we fuse the information of all available cameras and thus can compensate for LoT situations. For each device, a separate VO instance is executed. An exemplary architecture for a system with three modules is illustrated in Figure 2.

First, we define the terms *Keypose* and *Keyframe* and clarify their distinction.

A *Keyframe* is a selected frame used by a VO module to estimate motion for consecutive frames. It represents a unique viewpoint for motion estimation. The *Keypose* is the associated platform pose at the time the *Keyframe* is sampled. It reflects the pose at that specific time, estimated using all available data. Since *Keyframes* are selected independently by each VO instance, a *Keypose* can have multiple *Keyframes* if they are sampled simultaneously.

Additionally, we clarify the terms *ego-motion* and *odometry*. Motion refers to the change in position and orientation between two time frames. *Ego-motion* specifies the motion estimation of the system performing the calculation. In contrast, *odometry* measures the pose difference between the current pose and a reference frame, requiring continuous *ego-motion* measurements.

In the multi-sensor scenario, the trajectory estimated by *odometry* for each sensor is called a *track*.

A. Time-Continuous Local Trajectory Approximation

A design goal of this work is tracking-agnostic implementation, independent of the sensor, feature type, or filter design. We anticipate any tracking approach to produce a continuous stream of discrete estimations of the *ego-motion*, structured as pose estimations connected by metric delta poses. We aim to transform these estimates into a time-continuous approximation that can be evaluated at any point between measurements received from the sensors. This offers the opportunity to have independent VO modules that can operate optimally based on their currently observable field-of-view.

A generally accepted approach in the automotive sector, as shown in [18], is using B-spline for converting a discrete set of poses into a continuous representation. B-splines can be used to calculate the first and second order derivative efficiently at an evaluation point, making them useful for a motion-model approximation [19]. However, transferring the mentioned approach into the domain of indoor service robotics, the system has to deal with higher acceleration-changes and spontaneous changes in direction. While the smoothing behavior of B-splines adds a more linear motion in the automotive case which better reflects the actual vehicle's motion, in the aforementioned cases this leads in combination with *Keyframe*-discretization to missing trajectory coverage in corners. Evaluating this on the KITTI benchmark [20] (an automotive dataset) compared to TUM-RGBD [21] and IndoorMCD [22] (two indoor, hand-held/robot datasets) using the *Keyframe*-sampling approach of ORBSlam3 [6], we can identify a worst-case increase of only 2.6% on the median trajectory error on all KITTI

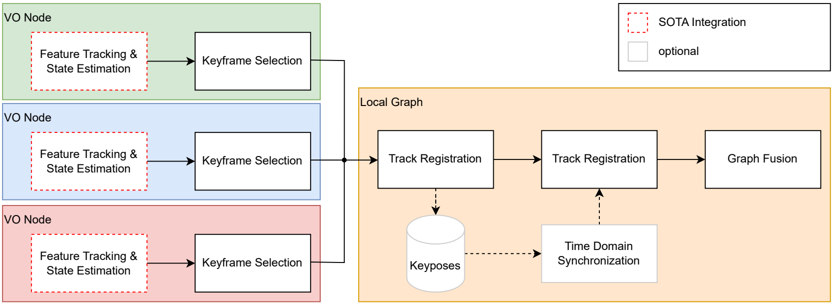


Fig. 2: Overview of an exemplary architecture for three visual odometry modules and the integration of SOTA approaches.

data. However, on TUM-RGBD the median trajectory error is increased by worst-case 12.8% and on IndoorMCD by 20.1%.

To overcome this domain-issue, we require a linear-motion constraint for the *Keyframe*-sampling described in the following. We linearize the trajectory by reducing graph nodes that can be represented by a linear motion. To achieve this, we define a set of consecutive poses $\{\mathbf{P}_0, \mathbf{P}_1, ..., \mathbf{P}_n\}$, where $\mathbf{P} \in \mathrm{SE}(3)$, as linearizable if any $\mathbf{P} \in \{\mathbf{P}_1, ..., \mathbf{P}_{n-1}\}$ can be explained by a linear motion from $\mathbf{P}_0$ to $\mathbf{P}_1$. We further introduce an acceptable positional error $e_{\mathrm{lin,t}}$ and angular error $e_{\mathrm{lin,R}}$ to enhance the process's robustness against small errors in the estimation of the *ego-motion*.

The result is a sparse set of discrete pose estimations whose density is depending on the change of the first derivative of the initial trajectory. To achieve a continuous time representation, the method proposed by Yang et al. [18] is followed, utilizing cumulative Spline-Fusion [23]. This assures a twice continuously differentiable representation which is crucial for motion-models.

The whole approximation process is illustrated in a step-by-step example in Figure 3.

B. Local Graph

To obtain a combined motion estimation of the system as seen in Figure 4, it is necessary to fuse the trajectory from each sensor into a single one. Since each sensor is positioned at a different location on the robotic system, their trajectory estimations result in different outputs. Therefore, all trajectory poses must be transformed into a common coordinate system, referred to here as the robot's origin

$$\mathbf{P}_O = \mathbf{F}_n^O \mathbf{P}_n, \tag{1}$$

where $\mathbf{F}_n^O$ is a transform from sensor n to the robot's origin O and $\mathbf{P}_n$ and $\mathbf{P}_O$ poses in the respective frames.

Subsequently, each estimated pose can be translated into a *Keypose*. At each timestep when a control pose is inserted into the trajectory, all other trajectories are queried for their local motion estimate. This is denoted as a *virtual pose*, which is based on the approximation described in

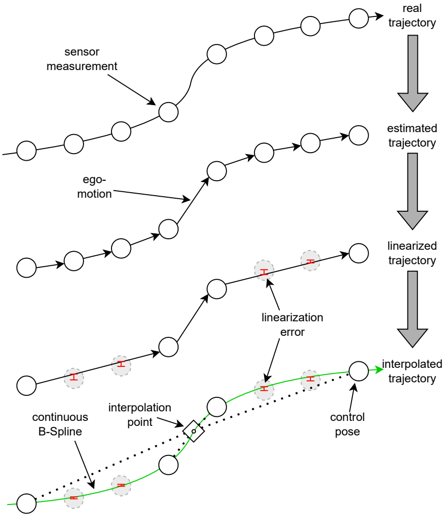


Fig. 3: Step-by-step illustration of the process for acquiring a time-continuous interpolated motion-model from a sensor's trajectory.

Section III-A. The combined pose is determined through a weighted average that includes all *virtual poses*. Translation and rotation are considered separately in this process.

The position is

$$\mathbf{T}_{Platform}(t) = \sum^{n} w_n \mathbf{T}_n(t) = \sum^{n} \begin{bmatrix} w_n T_{1,n}(t) \\ w_n T_{2,n}(t) \\ w_n T_{3,n}(t) \end{bmatrix}, \tag{2}$$

with the virtual position $\mathbf{T}_n(t)$ at time t, $\mathbf{T} \in \mathbb{R}^3$, and the corresponding weight $w_n \in [0; 1]$ (see Equation (7)) for the valid *track* of sensor n.

For the rotational component, we employ the Spherical

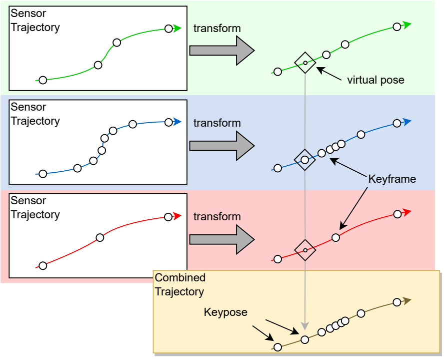


Fig. 4: Exemplary fusion of three independent running visual odometry modules.

Linear Interpolation (Slerp) [24] algorithm, which utilizes Lie Algebra to interpolate two rotations with a constant angular velocity, similar to the approach in Equation (2). We extend this method to the *n-element* and *weighted average* case by employing a concatenation of individual Slerp operations. However, it is important to note that these operations themselves are not commutative, so the weights applied are not accounted for as intended. For instance, when interpolating two unit quaternions $\mathbf{q}_a$ and $\mathbf{q}_b$ with equal weights, the resulting quaternion $\mathbf{q}^*$ is composed of equal contributions (1:1) of $\mathbf{q}_a$ and $\mathbf{q}_b$. Adding a third quaternion $\mathbf{q}_c$ with equal weight results in a shifted contribution towards the latter one (1:1:2).

To overcome this miss-alignment and obtain the correct weights after all operations have been applied, the weights have to be adapted. With the set of all weights

$$\begin{gathered} w_0, w_1, ..., w_n \in [0;1] \\ \sum^{n} w_n \overset{!}{=} 1, \end{gathered} \tag{3}$$

the corrected weights u_n to apply can be calculated as a series starting with the last element

$$\begin{aligned} u_n &\overset{!}{=} w_n \\ u_{n-1} &= \frac{w_{n-1}}{1-\widetilde{u}_{n-1}} \\ \widetilde{u}_n &= 1 - u_n . \end{aligned} \tag{4}$$

Afterward, the final quaternion can be calculated by concatenating the individual operations

$$\begin{aligned} \widetilde{\mathbf{q}}_2 &= \mathrm{Slerp}\left(\mathbf{q}_0, \mathbf{q}_1, u_1\right) \\ \widetilde{\mathbf{q}}_3 &= \mathrm{Slerp}\left(\mathbf{q}_2, \widetilde{\mathbf{q}}_2, u_2\right) \\ &\ldots \end{aligned} \tag{5}$$

The weights of each *track* are based on the temporal distance between the nearest measurement and the requested *virtual pose*. Let $t_{max,e}$ be the point in time in which the error of the velocity model is at its maximum. Since we constrain our motion-model at the control points, the error is increasing with the minimal difference $\Delta t = \min\left(t - t_1, t_2 - t\right)$ and the maximum being at

$$t_{max,e} = t_1 + \frac{t_2 - t_1}{2} = \frac{t_1 + t_2}{2}. \tag{6}$$

The weight for each track n at time t is defined as follows

$$w_{n,t} = \eta \ \max\left(1 - \frac{2\Delta t}{t_2 - t_1}, \hat{w}_{min}\right), \tag{7}$$

with a minimal weight limit of $\hat{w}_{min}$, and a factor η to normalize them to a sum of 1.

The final pose is used for two important registrations. Firstly, when a new *track* is added to the system and its relative pose estimates have to be registered to the platform. Secondly, when a new *Keyframe* is issued and the corresponding *Keypose* has to be estimated for insertion into the global graph. Therefore, the final pose is used to register the new *track* in terms of spatial offset with respect to the other *tracks*. Similarly, if a *Keypose* is registered, its pose will be estimated in the same manner to incorporate the information of all sensors and will be inserted into a global graph.

C. *Time Domain Synchronization*

Our approach allows transferring individual processes to different nodes within the robot's network. This flexibility comes with the downside that two arbitrary nodes' clock domains can have a time offset which has to be estimated.However, accurate synchronization often involves recurring complex calibrations or additional interfaces on the sensor. By exploiting the dynamic behavior and rigid transformation of the sensor configuration, this offset can be estimated as long as an initial guess of the offset is known.

For any two *tracks* that have a valid *ego-motion* estimation within the same time-span $[t_n, t_m]$, we determine the interval $\mathbf{I} = \{\mathbf{T}_n, \mathbf{T}_{n+1}, ..., \mathbf{T}_m\}$ in which the positional change delta-poses exceeds a threshold $\lambda_{\Delta\mathbf{T}}$. While in the general case this describes a change in acceleration, experimental evaluation has shown that our requirements for the sampling of *Keyframes* lead to similar results when determining the intervals with the highest *Keyframe* sampling frequency. We select the *track* with the higher density of *Keyframes* as the source and the other as the target. The process of finding the offset can be formulated as a constrained minimization problem

$$\min_{\Delta t \in \mathbf{I}} \sum \left(\mathbf{F}_S^T \mathbf{T}_S\left(t\right) - \mathbf{T'}_T\left(t - \Delta t\right)\right)^2, \tag{8}$$

where the estimated positions $\mathbf{T}_S\left(t\right)$ of the source track are transformed onto the target using $\mathbf{F}_S^T$. The timestamps of the target track are altered by the offset parameter Δt and the interpolation $\mathbf{T'}_T$ at the given time is used. The parameter Δt minimizes the sum of squared differences.

IV. EVALUATION

In the following, we want to demonstrate the effectiveness of our approach. We first investigate the tracking accuracy on a multi-sensor indoor benchmark [22]. This involves comparing the performance of single-instance systems, the impact of our fusion method on two selected methods, and one multi-tracking by-design approach on 97 trajectories in 5 different scenarios and a varying distance from $3\mathrm{m}$ to $37\mathrm{m}$. Selected images are presented in Figure 5. To our knowledge, this is the only multi-sensor dataset in the indoor domain with high-resolution ground truth information. Afterward, we demonstrate the individual sampling of *Keyposes* and the reduction of factor insertions into the pose graph. We conduct an experiment showing the robustness by avoiding loss-of-tracking due to single module failure. Lastly, the possibility to estimate an offset in the clock domain is demonstrated.

A. Accurate and robust ego-motion estimation

For this evaluation, we utilized the IndoorMCD dataset [22]. A small wheeled system and a hand-held camera device recorded sensor data from three Intel RealSense D435i devices, each providing color, depth, and Inertial Measurement Unit (IMU) streams. The evaluated systems include VINS-Mono [8] and its extension VINS-Fusion, ORB-SLAM2 [17] and its successor ORB-SLAM3 [6], as well as the multi-camera MROSlam [16].

For assessing tracking accuracy, we computed the relative pose error (RPE) with a delta-pose of $0.1\mathrm{m}$ for each approach. Additionally, we determined the success rate (SR), defined as the percentage of trajectories with valid tracking (according to the method) for at least 90% of the time to compensate for initial setup phase of each approach. This metric has been selected to specifically address the robustness of the system to reach the end of the trajectory. In the case of single-sensor approaches, we iterated through each sensor instance, selecting the worst-performing one since the advantage of multiple views cannot be utilized. If the SR fell below 10% for a single instance, we evaluated the error on the next better-performing instance to ensure comparable results. We employed the *evo* package [25] for error evaluation. The compiled results of the investigation are presented in Table I.

Analyzing the results reveals that extending the motion estimation process can enhance accuracy or, at the very least, maintain the same error level. MROSlam, a fully decoupled multi-sensor extension of ORB-SLAM2, outperforms the single-sensor variant by a factor of 4.5 in scenario 3 and a factor of 12.0 in scenario 4. Our extension similarly yields significant performance improvements when considering mean error, proving our assumptions that multiple sensors enhance the estimation process. Examining the maximum error in each scenario, our proposed fusion approach significantly enhances estimation accuracy. In all five scenarios, both the *Multi* variants of ORB-SLAM3 and VINS-Fusion outperform their counterparts, demonstrating our robust fusion capability. This is confirmed by the success rate, unequivocally validating the initial claim of a LoT-robust method. Particularly noteworthy is the exceptional performance of the *Multi* VINS-Fusion, accurately estimating every trajectory in the IndoorMCD benchmark.

To visually demonstrate our approach, we selected a single trajectory and plotted the estimated motion using our extension on ORB-SLAM3. The trajectory of run 15 from scenario 0, as depicted in Figure 6, utilizes the absolute pose error (APE) to color-code the segments for better visual inspection of the close-to-real estimation. However, we want to underline that this metric is not suitable for evaluating the performance of VO systems as shown in [26]. The chosen approach incorporates an initialization step at the beginning of each run, affecting the origin and orientation at the start. Therefore, we optimized the final trajectory over the ground truth to minimize errors for each system. Consequently, the trajectories of the estimation and ground truth may not align at the starting point. Additionally, we marked the position when the multi-sensor approach MROSlam lost all three VO instances and reported a loss-of-tracking. Due to its requirement for an already mapped area to recover from LoT, MROSlam was unable to estimate the entire trajectory. In contrast, our approach can dynamically handle a reset of any VO, ensuring full-trajectory coverage. A plot indicating the currently valid VO track estimates is presented in Figure 7 showing the frequent LoT events for individual modules.

The results show that our fusion approach does not negatively impact the estimation accuracy. Instead, it slightly improves the maximal error experienced by all systems due to the averaging effect of the local map. Our claim of higher robustness is clearly demonstrated by the significant improvements of the success-rate.

B. Keyframe and Keypose selection

Keypose selection is a crucial factor impacting the robust operation of a localization system, as discussed in Section II. In scenarios where sensors are mounted with different orientations and have no overlap, the decision to sample new *Keyposes* should be left to the individual VO modules, as views may vary significantly between sensors. While the first sensor may observe a feature-rich area, the second one might enter a low-textured region and delay sampling (or sample earlier if suitable). Sampling too many *Keyframes* can lead to a complex pose graph that may not be optimized online and in real-time. Therefore, individual sampling may result in the optimal set of *Keyframes* and minimal *Keyposes* for mapping.

We illustrate that our system independently samples *Keyframes* and the corresponding *Keyposes* for each sensor in Figure 10. The figure shows three plots of the same trajectory, each depicting the sampling process for a different sensor. Starting with the upper plot, it displays the selected poses for the front-facing camera. As the system rotates, more poses are registered, but in forward motion, only a few are required. In contrast, consider the left- and right-facing cameras. Both are oriented to the side, resulting in a higher sampling rate during forward motion compared to the

Fig. 5: IndoorMCD [22] benchmark. On the left, panoramic views of three out the five scenarios are shown. On the right, the images depict typical low-texture environments where loss-of-tracking frequently occurs due to homogeneous appearance, missing edges, or low-quality features.

TABLE I: Evaluation of the relative trajectory accuracy estimated on the scenarios with ground truth information of the IndoorMCD dataset [22] in meters. Shown are the mean and maximal RPE for the different approaches in meters. The success-rate (SR) indicates if the system was able to maintain valid tracking on all trajectories. The rows starting with *Multi* at the bottom correspond to extensions using our approach.

	Scenario 0			**Scenario 1**			**Scenario 2**			**Scenario 3**			**Scenario 4**		
	mean	max	SR	mean	max	SR	mean	max	SR	mean	max	SR	mean	max	SR
VINS-Mono [8]	0.11	1.59	0.82	0.13	1.45	0.93	0.05	0.47	0.88	0.07	0.62	0.92	0.03	0.62	0.93
VINS-Fusion [8]	0.13	1.57	0.81	0.10	1.40	0.86	0.07	0.32	0.96	0.05	0.50	**1.00**	0.04	0.59	0.93
ORB-SLAM2 [17]	0.25	5.42	0.00*	0.15	4.64	0.24	0.18	5.82	0.00*	0.16	2.43	0.63	0.12	1.73	0.73
ORB-SLAM3 [6]	0.09	5.43	0.05*	0.09	3.99	0.00*	0.04	4.91	0.05*	0.02	3.49	0.75	**0.01**	4.93	0.53
MROSlam [16]	**0.03**	3.67	0.68	**0.02**	6.06	0.79	**0.04**	6.20	0.92	0.06	6.44	**1.00**	**0.01**	1.03	**1.00**
Multi ORB-SLAM3	0.04	1.91	0.77	**0.02**	1.80	0.90	**0.04**	2.05	0.96	0.18	1.73	**1.00**	**0.01**	0.99	**1.00**
Multi VINS-Fusion	0.07	**1.22**	**1.00**	0.10	**1.31**	**1.00**	0.07	**0.27**	**1.00**	**0.03**	**0.41**	**1.00**	0.03	**0.40**	**1.00**

*) the trajectory error has been evaluated on the second worst instance due to bad performance

Full trajectory estimation on S0R15

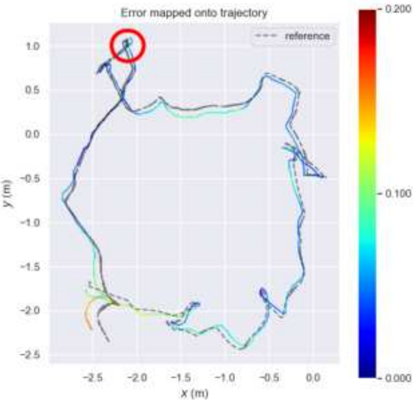


Fig. 6: Estimation result using our approach based on ORB-SLAM3 [6] that was able to recover the full trajectory. The red circle marks the area, where the multi-camera approach MROSlam [16] (trajectory not displayed to enhance readability; please refer to original publication) lost all motion estimations and was not able to recover until an already mapped position was revisited.

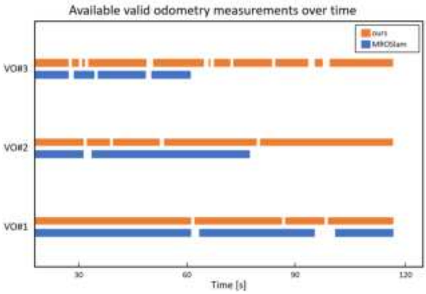


Fig. 7: Number of valid tracking estimations received from the VO modules during traversing the trajectory of Figure 6.

previously discussed front sensor. It is noteworthy that the density of sampled *Keyposes* is higher in curves compared to the front sensor. Both side sensors are located outside the rotational axis of the platform and, therefore, move faster relative to the others.

C. Reduction of Pose Graph Factors

Besides optimal and independent selection of *Keyframes*, one of the major advantages of using a decoupled approach for the *odometry* estimation, mapping, and fusion of the simultaneous localization and mapping (SLAM) system is the effective reduction of factors in the final pose graph for global optimization. Our *Keypose* formulation allows single VO modules to insert observations solely based on the state to maintain efficient, robust, and optimal tracking, as discussed in Section II.

To further investigate the impact of utilizing decentralized *Keyposes* over centralized and synchronized Multi-*Keyframes* (e.g. [27]), we evaluate the trajectories obtained from the TUM SLAM Benchmarks [21], [28], the KITTI dataset [20], and IndoorMCD [22] in a simulation that can be seen in Figure 8. We sample 3D landmarks in the proximity of the trajectory and assign unique IDs to simulate the sampling of features in a VO system. In addition, an intended error is induced consisting of a local normal for the pose error and a constant directed delta to mimic realistic *odometry* behavior. In the simulation, eight different sensor configurations are selected with two (C1 front/back, C2 front/down) three (C3 3 in line, C4 2 in line/down, C5 $-45°$/front/$+45°$, C6 $-90°$/front/$+90°$) and four (C7 front/left/right/back, C8 $-135°$/$-45°$/$+45°$/$+135°$) sensors to reflect typical hardware configuration across household robots, exploration rovers, drones and automotive. Features are tracked across the frames, and *Keyframes* are sampled according to our requirements using $e_{\mathrm{lin,t}} = 0.02\mathrm{m}$ and $e_{\mathrm{lin,R}} = 2°$. We then obtain the number of observations and compare the result with the number of observations that would have been inserted in the centralized and synchronized system, in which every module samples simultaneously.

Since numerous approaches offer the possibility to limit the number of features per *Keyframe*, we introduce a similar behavior by selecting $n_{\mathrm{best}} = 20, 50, 100, 300, \infty$ random features. Each pair of configurations and parameters is evaluated 20 times to overcome statistical outliers. Figure 9 illustrates the reduction of pose graph factors of the evaluated sensor configurations. It can be seen that the number of cameras has an impact on the ratio. In the case of two cameras, the median number of inserted factors reaches 0.46, decreasing to 0.32 for a three-camera setup and 0.26 for a four-camera configuration.

D. Computational Complexity

It is crucial to localization systems of any kind to run under real-time constraints. For this approach, we define the criteria for real-time capability as the time period between two received *Keyframes*. Therefore, we implemented the approach using Python 3.10 and evaluated the runtime on a computer with an Intel i7-8650H CPU. We measured the execution time for estimating a combined pose for both approaches described in Section IV-A on all trajectories. For our ORB-SLAM3 implementation, we observed a mean computation time of $0.13\mathrm{ms}$ (variance $< 0.01\mathrm{ms}$) and for VINS-Fusion $0.15\mathrm{ms}$ (variance $< 0.01\mathrm{ms}$). In a worst-case scenario, we assume that *Keyframes* are sampled for every frame at a frame-rate of $60Hz$. As a result, the time between consecutive frames would be $16\mathrm{ms}$, hence the margin is a factor of 100. For this reason, we expect no complication for running this approach under real-time constrains.

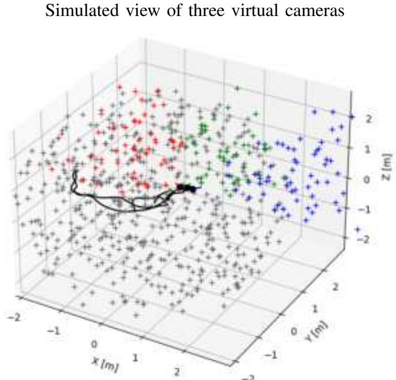


Fig. 8: Simulated view of a sensor setup with three cameras. The scene displays a single frame and the current visible features for each sensor coded by color.

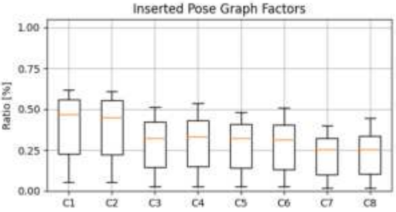


Fig. 9: Estimated reduction of inserted observation factors into the final pose graph for global optimization.

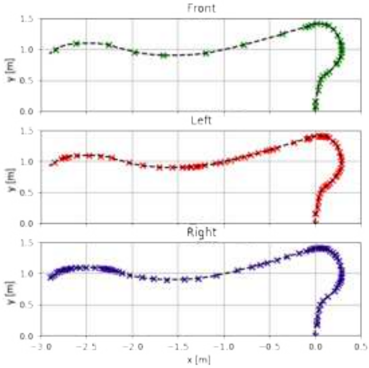


Fig. 10: Locally-optimal selection of *Keyframes* for each VO module depending on their internal state along the trajectory.

E. Time Offset Estimation

We reuse the simulation from Section IV-C for this evaluation, in which the timestamps of received frames are altered by an offset of $10\mathrm{ms}$. That corresponds to the worst-case scenario of an Network Time Protocol (NTP)-based clock synchronization and unknown offset due to interface stacks like USB. We limit the evaluation to indoor datasets only, as described in Section III-C. For the minimal change in pose-delta we select $\lambda_{\Delta\mathbf{T}} = 20\mathrm{cms}^{-2}$ and a minimum of 30 *Keyframes* per *track*. Some trajectories contain very few pose-delta changes in which the condition above is not fulfilled. For all others, we receive a median offset recovery of $0.9\mathrm{ms}$. For indoor systems with a theoretical maximum velocity of $2\mathrm{ms}^{-1}$ this corresponds to a miss-match of $1.8\mathrm{mm}$, which is an acceptable error.

V. CONCLUSION AND FUTURE WORK

This work presents an extension to state-of-the-art approaches by fusing multiple, loosely-coupled, and independently running visual odometries. We carefully ensured that our proposed work does not impose any additional hardware requirements for the placement and connection of sensors. Our decoupled approach provides a locally-optimal estimation based on the current view of each camera while minimizing the overall graph complexity for optimization. The primary design goal was to enhance the robustness of current *ego-motion*-estimation approaches to overcome challenging indoor environments, where loss-of-tracking events occur frequently due to low texture and close distances to obstacles. Additionally, we offer a time-continuous representation of the full motion based on all input sensors that can be evaluated at any point in time.

We demonstrate the applicability of our system in a representative indoor environment using multiple D435i sensors. Our approach decrease the maximum relative pose error while maintaining the mean error at the same level as a comparable multi-sensor by-design method. In addition, our approach significantly improves the robustness by successfully recovering nearly all trajectories. Furthermore, our evaluation in a simulation shows that our formulation reduces the number of inserted nodes in the final graph for optimization. Finally, we present a first demonstration of online time synchronization, acknowledging that further research is necessary to apply the method outside the indoor domain.

Hereby, we also want to underscore the limitations of our system. At first, fast movements, particularly rotations, introduce a significant level of motion blur across all sensors. Therefore, the disturbances degrade the quality of sensor information, and they cannot be resolved through the fusion approach. Additionally, this method does not address highly dynamic environments where scene appearance spontaneously changes. In the end, our goal is to integrate this approach into a large-scale SLAM system with active loop-closure detection and place recognition. We anticipate having a robust localization system in operation, which is crucial for our research on robotic assistance in elderly care.

REFERENCES

[1] Liu et al., "A review of sensing technologies for indoor autonomous mobile robots," *Sensors*, 2024.

[2] Nistér et al., "Visual odometry," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2004.

[3] Johnson et al., "Robust and efficient stereo feature tracking for visual odometry," in *2008 IEEE international conference on robotics and automation*. IEEE, 2008.

[4] Tong et al., "Gaussian process gauss–newton for non-parametric simultaneous localization and mapping," *The International Journal of Robotics Research*, 2013.

[5] Lim et al., "Real-time 6-dof monocular visual slam in a large-scale environment," in *IEEE international conference on robotics and automation*. IEEE, 2014.

[6] Campos et al., "ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM," *IEEE Transactions on Robotics*, 2021.

[7] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*. IEEE, 2007.

[8] Tong et al., "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, 2018.

[9] Kaess et al., "Visual slam with a multi-camera rig," Georgia Institute of Technology, Tech. Rep., 2006.

[10] S. Urban and S. Hinz, "Multicol-slam-a modular real-time multi-camera slam system," *arXiv preprint arXiv:1610.07336*, 2016.

[11] Müller et al., "Robust visual-inertial state estimation with multiple odometries and efficient mapping on an mav with ultra-wide fov stereo vision," in *International Conference on Intelligent Robots and Systems*. IEEE, 2018.

[12] Zhao et al., "Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments," in *International Conference on Intelligent Robots and Systems*. IEEE, 2021.

[13] Meng et al., "A tightly coupled monocular visual lidar odometry with loop closure," *Intelligent Service Robotics*, 2022.

[14] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.

[15] Eckenhoff et al., "Multi-camera visual-inertial navigation with online intrinsic and extrinsic calibration," in *International Conference on Robotics and Automation*. IEEE, 2019.

[16] Sewtz et al., "Robust approaches for localization on multi-camera systems in dynamic environments," in *International Conference on Automation, Robotics and Applications*. IEEE, 2021.

[17] Mur-Artal et al., "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, 2017.

[18] Yang et al., "Asynchronous multi-view slam," in *IEEE International Conference on Robotics and Automation*. IEEE, 2021.

[19] Sommer et al., "Efficient derivative computation for cumulative b-splines on lie groups," in *Conference on Computer Vision and Pattern Recognition*, 2020.

[20] Geiger et al., "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition*, 2012.

[21] Sturm et al., "A benchmark for the evaluation of rgb-d slam systems," in *International Conference on Intelligent Robot Systems*, Oct. 2012.

[22] Sewtz et al., "Indoormcd: A benchmark for low-cost multi-camera slam in indoor environments," *Robotics and Automation Letters*, 2023.

[23] Lovegrove et al., "Spline fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras." in *BMVC*, 2013.

[24] K. Shoemake, "Animating rotation with quaternion curves," in *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, 1985.

[25] M. Grupp, "evo: Python package for the evaluation of odometry and slam." https://github.com/MichaelGrupp/evo, 2017.

[26] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *International Conference on Intelligent Robots and Systems*. IEEE, 2018.

[27] D. Zou and P. Tan, "Coslam: Collaborative visual slam in dynamic environments," *IEEE transactions on pattern analysis and machine intelligence*, 2012.

[28] Schubert et al., "The tum vi benchmark for evaluating visual-inertial odometry," in *International Conference on Intelligent Robots and Systems*, 2018.

9. Design of a Microphone Array for Rollin' Justin

Authors:

Marco Sewtz, Tim Bodenmüller and Rudolph Triebel

Workshop:

Sewtz, Marco, et al. "Design of a Microphone Array for Rollin' Justin." Sound Source Localization and Its Applications for Robots. ICRA Workshop. 2019.

Abstract:

The paper presents the design and implementation of a microphone array for the humanoid robot Rollin' Justin, aiming to enhance human-robot interaction by enabling robust sound source localization and speech processing. Recognizing the importance of natural and intuitive interaction in populated environments, the study addresses the need for robots to detect and track speakers from various positions. The proposed microphone array, integrated into the robot's head, facilitates this by localizing and tracking sound sources within a specific range. The design considers an indoor environment with multiple noise sources and utilizes a sub-array approach to handle different frequency bands. The system processes include ambient noise removal, ego-noise suppression, and sound source localization using a modified MUSIC algorithm. This setup enhances interaction capabilities by reorienting the robot's head sensors towards the speaker. Preliminary evaluations indicate the system's potential, with further experiments planned to validate its performance in realistic scenarios. This work contributes to the development of human-centered interfaces in service robotics, leveraging advanced audio processing techniques to improve robot audition capabilities.

Contributions:

The author of this dissertation designed the simulation and numerical analysis for the proposed microphone array. The mechanical design was provided by Werner Friedl. The script was provided by the author and the publication was presented by the author.

Copyright:

None

Design of a Microphone Array for Rollin' Justin

Marco Sewtz Tim Bodenmüller Rudolph Triebel

I. INTRODUCTION

For a humanoid robot operating in populated environments it is a key competence to interact with humans naturally and intuitively. Therefore, research in human-robot interaction explores the interpretation of visual, auditive and even tactile sensing modalities. One important aspect here is to recognize robustly the intention of the human interacting with the robot. To achieve this, robot audition is a suitable modality, as it allows for detecting and tracking speakers from arbitrary positions around the robot and also from distant places. In this paper we present the design of a microphone array for the head of our humanoid robot Rollin' Justin that allows us to localize and track sound sources within a certain distance to the robot. Until now, high-level interfacing to the robot was only possible using a tablet or by verbal communication via a headset using speech recognition. However, neither method allows to localize the operator. With the microphone array in the head we can do sound source localization and then re-position the head sensors towards the speaker, enabling advanced interaction possibilities.

In recent years, the research in the field of service robotics focuses on human-centered interfaces. This includes easy-to-use as well as easy-to-understand systems like robots with audio input [1]. Several systems have been developed using microphone arrays to extract speech [2], [3]. More complex systems use multiple techniques for processing including sound source localization, feature extraction and speech-to-text engines [4].

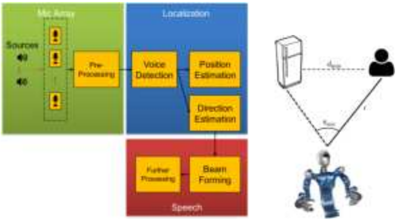
In the following we present the design of a microphone array for Rollin' Justin for sound source localization and speech recognition. First we describe the considered household scenario and present an overview of the proposed processing system. We than discuss our design considerations regarding the microphone array as well as possible sound source localization and speech processing. We conclude with a description of system integration.

II. DESIGN

For the design of our microphone array we consider the following scenario: our robot is located in a typical indoor environment, for instance an apartment (Figure 1). This implies that we have multiple sources of noise, e.g. a fridge, and reverberations. In addition, we suppose that there is only one person at a time speaking to the robot, called the operator. The expected distance r between the robot and the operator is between $1\,\mathrm{m}$ and $4\,\mathrm{m}$. We assume that, from the robot's point of view the operator and any other sound source have a minimum tangential distance of at least $d_{\min} = 1\,\mathrm{m}$. This results in a minimum angular distance $\theta_{\min} = \arctan(d_{\min}/r_{\max}) \approx \pm 14^\circ$ separating the speech from any other sound source. Figure 2b illustrates this scenario.

A. System Overview

Figure 1: Lab environment imitating a typical small apartment



(a) (b)

Figure 2: (left) System overview and (right) illustration of assumed audio scenario.

We plan our system as illustrated in Figure 2a. The sound from the sound sources is received and sampled by a microphone array. In a preprocessing step we remove ambient noise and the robot's ego-noise. Then, we localize the sound source by calculating the direction and distance of the source. This is used by subsequent beam forming to improve the signal for speech recognition.

B. Microphone Array

For the design of the microphone array, we consider the expected sound signals, the processing requirements, as well as geometrical limitations. In general, we follow the approach of a broadband microphone sub-array [5]. Hence, we divide the frequency spectrum into three sub-bands ($< 1\,\mathrm{kHz}$, 1-2 kHz, $> 2\,\mathrm{kHz}$), each handled by a specific sub-array. For lower frequencies, we need large distances between microphones to capture longer delays. For higher frequencies, we have to use a smaller spacing. However, the range of feasible distances between the microphones is small due to limited space within the robot's face mask. Also, they have to be placed on a single plane because of limited space and appearance constraints. In total we use eight microphones that are arranged as shown in Figure 4b and grouped into sub-arrays as shown in Figure 4a. The outer microphones have a distance of $146\,\mathrm{mm}$. To improve the speech quality [6], we sample with at least $16\,\mathrm{kHz}$ instead of the usual $8\,\mathrm{kHz}$. The signals are bandpass filtered, amplified and summed up to the complete signal. We optimized the microphone positions by a free-air simulation. The resulting combined directivity pattern is shown in Figure 3. The main lobe of the array is focused around our defined 14° corridor at -3 dB, and sources at larger angles are suppressed. In the lower frequency range, the lobe is somewhat less focused due to a limited maximum inter-signal delay.

All authors are with: Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany. Email: `marco.sewtz@dlr.de`, `tim.bodenmueller@dlr.de`, `rudolph.triebel@dlr.de`

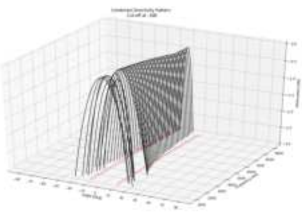
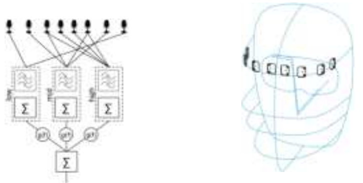


Figure 3: Directivity pattern for the combined array approach. The figure shows the pattern truncated at -3 dB. The dashed red line illustrates a ±14° corridor.



(a) Sub-Array concept

(b) CAD drawing of the proposed microphone positions

Figure 4: (left) Illustration of the sub-array approach, (right) CAD drawing of the microphones on the robot's face.

C. Sound Source Localization

For localization, we only use signal snippets that contain speech, which we identify by means of the Long-Term Speech Divergence [7]. For an accurate and robust localization from these snippets we use a modified version of the MUSIC algorithm [8].

The main principle of the estimation process relies on the delay between the received signals. In case of a linear array, the delay is proportional to the spacing between the microphones. For our 2D array design, the delay Δt_i between microphone i and a reference point assuming a signal from direction θ is given by the projection onto the direction vector of the arriving sound wave, i.e.

$$\Delta t_i = (\mathbf{p}_i^\top \mathbf{e}_\theta)/c_0 \ ,$$

where $\mathbf{p}_i$ is the position of the i-th microphone with respect to the reference point, $\mathbf{e}_\theta$ the direction unit vector of the sound wave and $c_0 \approx 343\,\mathrm{m/s}$ the speed of sound. Figure 5 illustrates this relation.

D. Further Processing

With the direction of the sound source, we steer our system towards the source using a delay-and-sum beamformer in combination with our sub-array approach. We expect a high increase in signal-to-noise ratio as well as noise suppression from this technique. The retrieved speech signal will be used to extract commands from the operator. We will use already available offline speech processing engines such as CMU Sphinx [9], which has already been used with our headset. We also plan to adapt voice assistant paradigms as in the NAOMI Project [10].

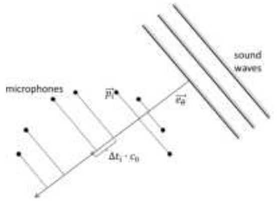


Figure 5: Illustration of the delay calculation for an arriving sound wave. The intersections of the dashed lines show the projection onto the direction vector $\vec{e_\theta}$. The red dot represents the reference point.

III. INTEGRATION

For the array we chose SPH0645LM4H-8 MEMS microphones with $\mathrm{I^2S}$ support[1]. They will be sampled simultaneously by the native $\mathrm{I^2S}$ ports of an Nvidia Jetson TX2 board mounted onto an Auvidea J140 carrier. The whole system will be integrated physically into the head of the robot, and the software will be part of our "Links and Nodes" framework.

IV. CONCLUSION

We presented the design of a microphone array for sound source localization and speech processing on the humanoid robot Rollin' Justin. We introduced our overall system architecture, gave an overview of the subsequent sound processing, and described design considerations, implementation details and preliminary evaluations. In a next step, we will finalize the implementation on the robot and execute several experiments in realistic scenarios.

V. ACKNOWLEDGMENT

We want to thank Werner Friedl for his contributions to the mechanical design and implementation.

REFERENCES

[1] A. Di Nuovo, F. Broz, N. Wang, T. Belpaeme, A. Cangelosi, R. Jones, R. Esposito, F. Cavallo, and P. Dario, "The multi-modal interface of Robot-Era multi-robot services tailored for the elderly," *Intelligent Service Robotics*, 2018.

[2] I. Hara, F. Asano, H. Asoh, J. Ogata, N. Ichimura, Y. Kawai, F. Kanehiro, H. Hirukawa, and K. Yamamoto, "Robust speech interface based on audio and video information fusion for humanoid HRP-2," in *IROS*, 2004.

[3] J.-M. Valin, J. Rouat, and F. Michaud, "Enhanced robot audition based on microphone array source separation with post-filter," in *IROS*, 2016.

[4] K. Nakadai, T. Takahashi, H. G. Okuno, H. Nakajima, Y. Hasegawa, and H. Tsujino, "Design and implementation of robot audition system "HARK" — open source software for listening to three simultaneous speakers," *Advanced Robotics*, 2010.

[5] I. A. McCowan, "Robust speech recognition using microphone arrays," Ph.D. dissertation, Queensland University of Technology, 2001.

[6] B. B. Monson, E. J. Hunter, A. J. Lotto, and B. H. Story, "The perceptual significance of high-frequency energy in the human voice," *Frontiers in Psychology*, 2014.

[7] J. Ramırez, J. C. Segura, C. Benıtez, A. De La Torre, and A. Rubio, "Efficient voice activity detection algorithms using long-term speech information," *Speech Communication*, 2004.

[8] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, 1986.

[9] P. Lamere, P. Kwok, E. Gouvea, B. Raj, R. Singh, W. Walker, M. Warmuth, and P. Wolf, "The CMU Sphinx-4 speech recognition system," in *ICASSP*, 2003.

[10] The Naomi Community and Project Naomi. [Online]. Available: https://projectnaomi.com/

---

[1] https://www.knowles.com/

10. Sound Source Localization for Robotic Applications

Authors:

Marco Sewtz, Tim Bodenmüller, and Rudolph Triebel.

Workshop:

Sewtz, Marco, et al. "Sound Source Localization for Robotic Application." Unconventional Sensors in Robotics: Perception for Online Learning, Adaptive Behavior, and Cognition. ICRA Workshop. 2020.

Abstract:

Intuitive human robot interfaces like speech or gesture recognition are essential for gaining acceptance for robots in daily life. However, such interaction requires that the robot detects the human's intention to interact, tracks his position and keeps its sensor systems in an optimal configuration. Audio is a suitable modality for such task as it allows for detecting a speaker in arbitrary positions around the robot. In this paper, we present an extension of our proposed sound source localization approach Motion Model Enhanced MUltiple SIgnal Classification (MME-MUSIC) for segmenting speech input.

We evaluate the system with speech captured under real conditions in an experimental setup and show the use of our estimation in real applications.

Contributions:

The author of this dissertation designed and implemented the presented sound source localization framework. He designed and conducted the experimental evaluation. The script was provided by the author and the publication was presented by the author.

Copyright:

None

Sound Source Localization for Robotic Applications

Marco Sewtz[1] Tim Bodenmüller[1] Rudolph Triebel[1,2]

Abstract— **Intuitive human robot interfaces like speech or gesture recognition are essential for gaining acceptance for robots in daily life. However, such interaction requires that the robot detects the human's intention to interact, tracks his position and keeps its sensor systems in an optimal configuration. Audio is a suitable modality for such task as it allows for detecting a speaker in arbitrary positions around the robot. In this paper, we present an extension of our proposed sound source localization approach Motion Model Enhanced MUltiple SIgnal Classification (MME-MUSIC) for segmenting speech input.**

We evaluate the system with speech captured under real conditions in an experimental setup and show the use of our estimation in real applications.

I. INTRODUCTION

The ability of mobile robots to interact with people in an intuitive and maybe anthropomorphic manner is a key to the acceptance of robots in human-dominated environments. Human-robot-interaction (HRI) can be visual (e.g. gestures), tactile (e.g. guiding) as well as auditive (e.g. instructing). However, all modalities require that the robot recognizes the intention of a human to interact. Visual systems can only recognize intention in the sensor's field of view, which is usually limited and may also be occluded by obstacles. Tactile systems require that the human is nearby. Robot audition, however, allows for detecting and tracking a speaker from arbitrary positions around the robot and also from distant places. Figure 1 illustrates a typical situation. The human on the sofa wants to interact with the robot, but the latter is currently performing another task, thus, positioning its visual sensor in the opposite direction. Moreover, audio also allows for gaining information about the environment or to separate between different speakers. The information about the speaker's position can also be used to enhance the audio input signal, e.g. to improve speech processing as well as getting more information about the position of humans in the scenario.

We presented a novel approach for localization of speakers in reverberant and echoic environments by use of a microphone array in [1]. We classify received audio streams as speech or non-speech using a voice activity detector (VAD). We transform the signal into the frequency domain and analyze the fourier coefficients to calculate a score. Afterwards we select the most significant bins and fed them into our direction of arrival (DoA) estimator. Further on we propose a motion model to check the calculated direction spectrum to improve the robustness.

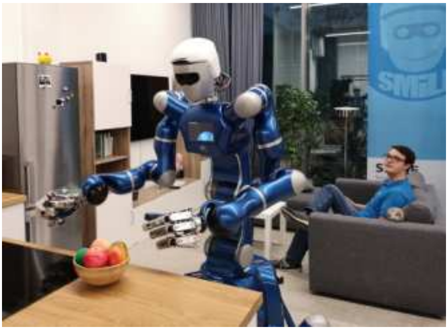


Fig. 1: Illustration of the interaction recognition problem: The robot is turned away from the operator. While the vision system might not recognize him, the audio input will do so.

In this work we want to show the application and the use of the motion model to segment received speech and assign them to different speakers. We deliberately avoid using other techniques like mel cepstrum analysis [2]–[5] or vision-based aid [6]–[9] to illustrate the performance of a single DoA estimator.

II. RELATED WORK

At first, research focused on imitating the binaural audio localization of animals and humans [10]–[13]. Using both the interaural phase difference (IPD) and the interaural intensity difference (IID). Further, some techniques take into account the head-related transfer function [14], [15] as well as the prior information on reverberant properties of the environment to achieve accurate results. Incorporation of a particle filter approach to be used on binaural measurements improves the estimation of sound sources as well [16]. Nonetheless these systems need a demanding hardware setup and calibration.

Other approaches use an array of microphones to overcome the challenging requirements on the hardware and to estimate the direction of arrival (DoA) of a received signal [17], [18]. It is possible to calculate the most probable DoA by estimating the time delay between the signals received by each microphone. Combining these methods with delay and sum beam forming (DSBF) as well as random sample consensus (RANSAC), more than one sound source can be

[1] Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Oberpfaffenhofen, Germany.
[2] Dep. of Computer Science, Technical Univ. of Munich, Germany
`marco.sewtz@dlr.de` `tim.bodenmueller@dlr.de`
`rudolph.triebel@dlr.de`

localized simultaneously [19]. However, these approaches have problems with low signal-to-noise-ratios (SNR) input signals, changing acoustic conditions and varying speakers. Different approaches using neural networks have been studied to tackle these problems. Nevertheless, they need training dedicated to the specific speaker or require very large amounts of data for generalizing [20]–[24].

Recently, exploiting the properties of the subspace as in Multiple Signal Classification (MUSIC) [25] and Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) [26] have received more interest. They overcome the resolution limit constrained by the sampling rate and are more robust to signal noise but they are computational costly [27]–[30].

Several extensions have been proposed for enhancing the performance of MUSIC, e.g. using singular value decomposition [31] to reduce the computational complexity while enhancing robustness against noise. Incremental versions are introduced to reach real-time performance while enhancing robustness against noise [32], [33]. Additional research to further reduce the computational costs in the representation space is done in [34], [35].

However, even recent sound source localization systems face problems when detecting humans in indoor scenarios under non-optimal acoustic conditions. We identified significant effects that degrade the performance, namely reverberation and echo. The first one is the reflection of numerous acoustic wavelets at every surface which results in a "fading-out" effect and lower SNR. The latter one is the complete reflection and delayed reception of the original source. This leads to miss-classification.

III. SYSTEM

Our system Motion Model Enhanced Multiple Signal Classification (MME-MUSIC) is based on the SEVD-MUSIC [36] approach. We enhance the process by limiting the estimation only to speech phases classified by the voice activity detector. Furthermore we reduce the number of frequency bins by selecting the most significant ones based on a score calculated in the previous step. Additionally we post-filter our results using a motion model. Lastly we exploit the decision of the model to segment the speech and assign it to the speakers. For capturing the audio we use a microphone array consisting of four acoustic sensors. An overview on the system is illustrated in Figure 2.

A. Voice Activity Detector and Band Selection

We use the VAD proposed by Ramírez *et al.* to classify the incoming signal [37]. First, we transform the audio into the frequency domain. Afterwards, we use the Longterm Speech Divergence (LTSD) approach which assumes that the spectrum of noise differs significantly from frames containing speech. Yet, short time sound events like clapping or door closing are suppressed.

Subsequently we use the gained information on the difference of individual frequency bins compared to noise to find the significant components. This enables the reduction of calculation costs while preserving estimation accuracy.

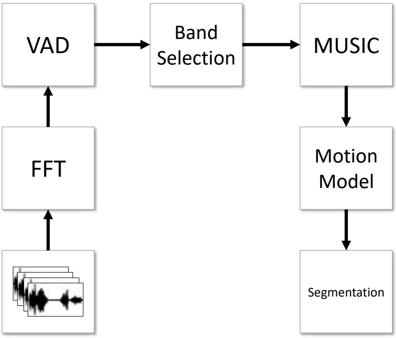


Fig. 2: System overview.

B. DoA Estimation

We assume that our received signal consists only of the direction-depending source signal and independent system noise. The approach of MUSIC exploits this dependency and decomposes the transformed audio into noise and source subspace. Ultimately it tries to find the corresponding direction vector which fulfills the constraints given by the system and the subspaces. We repeat this estimation for all selected frequency bins and accumulate a total pseudospectrum to reflect the direction dependencies.

C. Motion Model and Segmentation

We check the plausibility of the estimated angle by evaluating it with a motion model. To do this, we assume that for a given time span the source moves with mean angular velocity. We take into account a constant motion tolerance to cope with dynamic changes and measurement noise.

When receiving a new DoA from the previous steps we gather all estimation within the time span. If we can explain the measurement given our motion model, we flag them as valid. We need at least 3 valid estimations, the first ones to calculate the motion vector, the last one to verify the model.

Furthermore we exploit the verification for our segmentation. We consider a scenario with two persons speaking. If we receive new measurements which are marked as valid but based on a different motion vector than previous measurements, we assign them to a different speaker. This is a fairly naïve approach, however the performance shown in the next section is notable.

IV. EXPERIMENTS

We show the application of our sound source localization in a segmentation process where two persons are having a conversation. Our system uses the estimated position to assign the speech to the corresponding speaker. The scenario is shown in Figure 3. We illustrate both cases, the speakers facing the system and each other. We assume the last one as

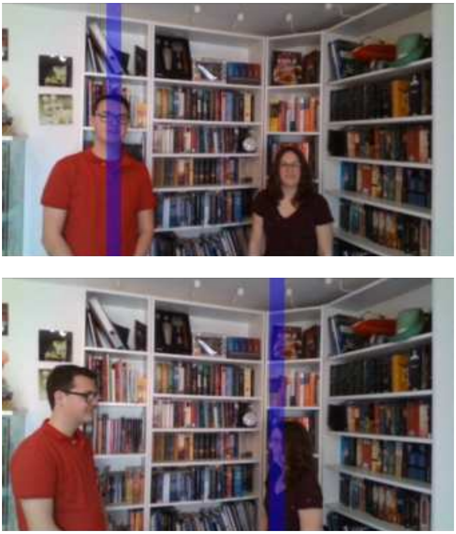


Fig. 3: Conversation between two person. Left side shows the case where both of them are speaking towards the camera. In this scenario a vision-based system may lead comparable performance. Right side shows the case where both speakers are facing each others. This is a hard task for vision classifier. As indicated by the blue bar, the auditory system succeeded in identifying the current speaker.

a hard task for camera-based systems, as the visual clues for identifying the speaker are reduced to a minimum.

We compare our approach with AFRF-MUSIC [38], which is an optimized version of SEVD-MUSIC [31] according to execution time. In contrast, our approach is also optimized for use in indoor scenarios.

For AFRF-MUSIC we add the information, that the left speaker can be localized by positive angles, the right speaker by negative, as the system has no indicator for changing sources.

We manually labeled the data for left and right speaker and compare it with the outcome of the algorithms. The results are shown in Figure 4.

For AFRF-MUSIC we get correct assignment in 79.5% of all estimated cases, for MME-MUSIC in 93.1%. In total comparing all cases where the approaches did not assign a speaker, AFRF-MUSIC performs with 61.0% and MME-MUSIC with 73.0% successfull assignments (see Table I).

V. CONCLUSION

In this work we showed the application of our recently developed sound source localization system Motion Model Enhanced MUSIC (MME-MUSIC). We shortly introduced the pitfalls of indoor scenarios and the resulting effects on auditory systems. We developed a simple segmentation algorithm based on our approach to assign speech phases of a received signal to specific speakers. Furthermore we showed that this naïve approach is reliable enough in situations where classical approaches using vision-based systems may fail to locate the correct speaker.

TABLE I: Segmentation results.

	Total		Segmented	
Method	TP	FP	TP	FP
AFRF	61.0%	39.0%	79.5%	20.5%
MME	73.9%	26.1%	93.1%	6.9%

With this work we want to propagate the benefit of using robot audition as an additional modality for robust robotic systems. We expect enhanced perception systems which operate robustly in complex environments.

REFERENCES

[1] M. Sewtz, T. Bodenmüller, and R. Triebel, "Robust music-based sound source localization in reverberant and echoic environments," in *Intelligent Robots and Systems (IROS). IEEE/RSJ International Conference on*, 2020, submitted.

[2] M. R. Hasan, M. Jamil, M. Rahman *et al.*, "Speaker identification using mel frequency cepstral coefficients," *variations*, vol. 1, no. 4, 2004.

[3] S. Agrawal and D. Mishra, "Speaker verification using mel-frequency cepstrum coefficient and linear prediction coding," in *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*. IEEE, 2017, pp. 543–548.

[4] A. Charisma, M. R. Hidayat, and Y. B. Zainal, "Speaker recognition using mel-frequency cepstrum coefficients and sum square error," in *2017 3rd International Conference on Wireless and Telematics (ICWT)*. IEEE, 2017, pp. 160–163.

[5] A. Awais, S. Kun, Y. Yu, S. Hayat, A. Ahmed, and T. Tu, "Speaker recognition using mel frequency cepstral coefficient and locality sensitive hashing," in *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*. IEEE, 2018, pp. 271–276.

[6] J. M. Rehg, K. P. Murphy, and P. W. Fieguth, "Vision-based speaker detection using bayesian networks," in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, vol. 2. IEEE, 1999, pp. 110–116.

[7] R. Cutler and L. Davis, "Look who's talking: Speaker detection using video and audio correlation," in *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No. 00TH8532)*, vol. 3. IEEE, 2000, pp. 1589–1592.

[8] P. Chakravarty and T. Tuytelaars, "Cross-modal supervision for learning active speaker detection in video," in *European Conference on Computer Vision*. Springer, 2016, pp. 285–301.

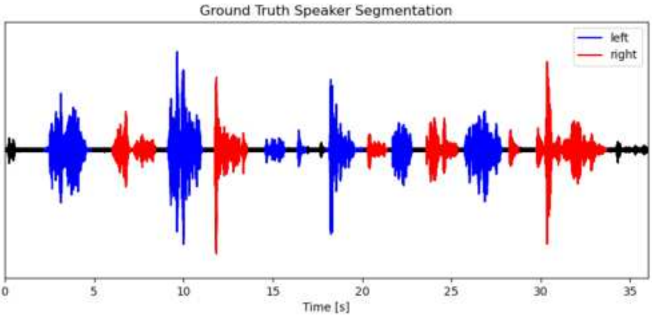
[9] K. Stefanov, J. Beskow, and G. Salvi, "Vision-based active speaker detection in multiparty interaction," in *Grounding Language Understanding GLU2017 August 25, 2017, KTH Royal Institute of Technology, Stockholm, Sweden*, 2017.

[10] K. Nakadai, K.-i. Hidai, H. Mizoguchi, H. Okuno, and H. Kitano, "Real-time auditory and visual multiple-object tracking for humanoids," in *Artificial Intelligence. Proceedings. 17th International Joint Conference on*, 2001, pp. 1425–1432.

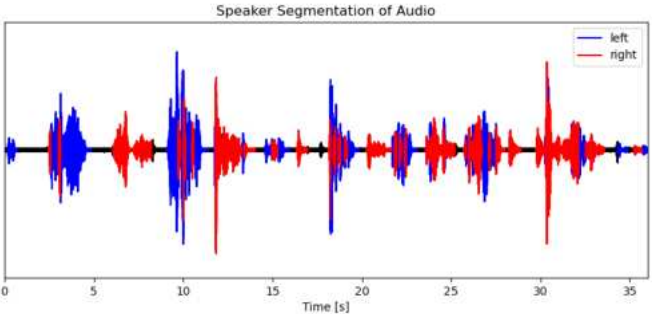
[11] K. Nakadai, D. Matsuura, H. G. Okuno, and H. Kitano, "Applying scattering theory to robot audition system: Robust sound source localization and extraction," in *Intelligent Robots and Systems. Proceedings. IEEE/RSJ International Conference on*, vol. 2, 2003, pp. 1147–1152.

[12] J. Huang, N. Ohnishi, and N. Sugie, "Building ears for robots: sound localization and separation," *Artificial Life and Robotics*, vol. 1, no. 4, pp. 157–163, 1997.

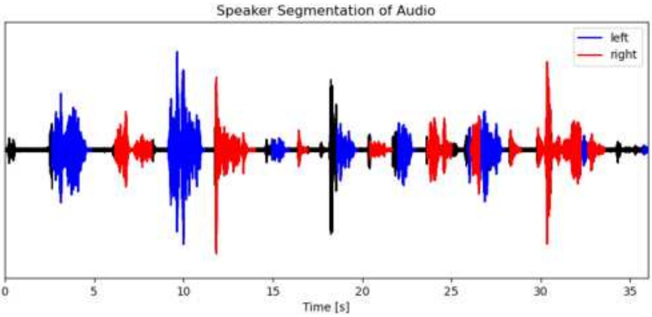
[13] L. A. Jeffress, "A place theory of sound localization." *Journal of Comparative and Physiological Psychology*, vol. 41, no. 1, p. 35, 1948.


Ground Truth Speaker Segmentation

(a) Ground Truth


Speaker Segmentation of Audio

(b) AFRF-MUSIC method


Speaker Segmentation of Audio

(c) MME-MUSIC method

Fig. 4: Experimental results of our segmentation. Top shows the manually labeled ground truth. Center shows the result using AFRF-MUSIC, a state-of-the-art and real-time capable approach. Bottom shows our approach using MME-MUSIC. It can be seen, that AFRF-MUSIC has a lot of miss-classifications. MME-MUSIC has less segmented points while having better assignments.

[14] J. A. MacDonald, “A localization algorithm based on head-related transfer functions,” *The Journal of the Acoustical Society of America*, vol. 123, no. 6, pp. 4290–4296, 2008.

[15] F. Keyrouz, Y. Naous, and K. Diepold, “A new method for binaural 3-D localization based on HRTFs,” in *Acoustics, Speech and Signal Processing (ICASSP). Proceedings. IEEE International Conference on*, vol. 5, 2006.

[16] I. Kossyk, M. Neumann, and Z.-C. Marton, “Binaural bearing only tracking of stationary sound sources in reverberant environment,” in *Humanoid Robots (Humanoids), IEEE-RAS 15th International Conference on*. IEEE, 2015, pp. 53–60.

[17] J.-M. Valin, F. Michaud, J. Rouat, and D. Létourneau, “Robust sound source localization using a microphone array on a mobile robot,” in *Intelligent Robots and Systems. Proceedings. IEEE/RSJ International Conference on*, vol. 2. IEEE, 2003, pp. 1228–1233.

[18] J.-M. Valin, F. Michaud, B. Hadjou, and J. Rouat, “Localization of simultaneous moving sound sources for mobile robot using a frequency-domain steered beamformer approach,” in *Robotics and Automation. Proceedings. IEEE International Conference on*, vol. 1. IEEE, 2004, pp. 1033–1038.

[19] Y. Sasaki, S. Kagami, and H. Mizoguchi, “Multiple sound source mapping for a mobile robot by self-motion triangulation,” in *Intelligent Robots and Systems. IEEE/RSJ International Conference on*. IEEE, 2006, pp. 380–385.

[20] E. Mumolo, M. Nolich, and G. Vercelli, “Algorithms for acoustic localization based on microphone array in service robotics,” *Robotics and Autonomous Systems*, vol. 42, no. 2, pp. 69–88, 2003.

[21] R. Roden, N. Moritz, S. Gerlach, S. Weinzierl, and S. Goetze, “On sound source localization of speech signals using deep neural networks,” in *Deutsche Jahrestagung für Akustik (DAGA)*, 2015, pp. 1510–1513.

[22] S. Adavanne, A. Politis, and T. Virtanen, “Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network,” in *26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 1462–1466.

[23] X. Xiao, S. Zhao, X. Zhong, D. L. Jones, E. S. Chng, and H. Li, “A learning-based approach to direction of arrival estimation in noisy and reverberant environments,” in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*. IEEE, 2015, pp. 2814–2818.

[24] R. Takeda and K. Komatani, “Discriminative multiple sound source localization based on deep neural networks using independent location model,” in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 603–609.

[25] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[26] R. Roy and T. Kailath, “ESPRIT – estimation of signal parameters via rotational invariance techniques,” *Acoustics, Speech, and Signal Processing. IEEE Transactions on*, vol. 37, no. 7, pp. 984–995, 1989.

[27] S. Argentieri and P. Danes, “Broadband variations of the music high-resolution method for sound source localization in robotics,” in *Intelligent Robots and Systems. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 2009–2014.

[28] F. Asono, H. Asoh, and T. Matsui, “Sound source localization and signal separation for office robot” jijo-2”,” in *Multisensor Fusion and Integration for Intelligent Systems. Proceedings. IEEE/SICE/RSJ International Conference on*. IEEE, 1999, pp. 243–248.

[29] C. T. Ishi, O. Chatot, H. Ishiguro, and N. Hagita, “Evaluation of a MUSIC-based real-time sound localization of multiple sound sources in real noisy environments,” in *Intelligent Robots and Systems. IEEE/RSJ International Conference on*. Institute of Electrical and Electronics Engineers, 2009, pp. 2027–2032.

[30] K. Nakamura, K. Nakadai, and G. Ince, “Real-time super-resolution sound source localization for robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 694–699.

[31] ——, “Real-time super-resolution sound source localization for robots,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, p. 694–699.

[32] K. Okutani, T. Yoshida, K. Nakamura, and K. Nakadai, “Outdoor auditory scene analysis using a moving microphone array embedded in a quadrocopter,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, p. 3288–3293.

[33] T. Ohata, K. Nakamura, T. Mizumoto, T. Taiki, and K. Nakadai, “Improvement in outdoor sound source detection using a quadrotor-embedded microphone array,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep 2014, p. 1902–1907.

[34] G. Chardon, “A block-sparse music algorithm for the localization and the identification of directive sources,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, p. 3953–3957.

[35] R. Takeda and K. Komatani, “Noise-robust music-based sound source localization using steering vector transformation for small humanoids,” *Journal of Robotics and Mechatronics*, vol. 29, no. 1, p. 26–36, Feb 2017.

[36] K. Nakamura, K. Nakadai, F. Asano, Y. Hasegawa, and H. Tsujino, “Intelligent sound source localization for dynamic environments,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2009, p. 664–669.

[37] J. Ramırez, J. C. Segura, C. Benıtez, A. De La Torre, and A. Rubio, “Efficient voice activity detection algorithms using long-term speech information,” *Speech communication*, vol. 42, no. 3-4, pp. 271–287, 2004.

[38] K. Hoshiba, K. Nakadai, M. Kumon, and H. G. Okuno, “Assessment of music-based noise-robust sound source localization with active frequency range filtering,” *Journal of Robotics and Mechatronics*, vol. 30, no. 3, p. 426–435, 2018.