

Lost in Tracking Translation: A Comprehensive Analysis of Visual SLAM in Human-Centered XR and IoT Ecosystems

YASRA CHANDIO, University of Massachusetts Amherst, United States

KHOTSO SELIALIA, University of Massachusetts Amherst, United States

JOSEPH DEGOL, Steg AI, United States

LUIS GARCIA, University of Utah, United States

FATIMA M. ANWAR, University of Massachusetts Amherst, United States

Advancements in tracking algorithms have empowered nascent applications across various domains, from steering autonomous vehicles to guiding robots to enhancing augmented reality experiences for users. However, these algorithms are application-specific and do not work across applications with different types of motion; even a tracking algorithm designed for a given application does not work in scenarios deviating from highly standard conditions. For example, a tracking algorithm designed for robot navigation inside a building will not work for tracking the same robot in an outdoor environment. To demonstrate this problem, we evaluate the performance of the state-of-the-art tracking methods across various applications and scenarios. To inform our analysis, we first categorize algorithmic, environmental, and locomotion-related challenges faced by tracking algorithms. We quantitatively evaluate the performance using multiple tracking algorithms and representative datasets for a wide range of Internet of Things (IoT) and Extended Reality (XR) applications, including autonomous vehicles, drones, and humans. Our analysis shows that no tracking algorithm works across different applications and scenarios within applications. Ultimately, using the insights generated from our analysis, we discuss multiple approaches to improving the tracking performance using input data characterization, leveraging intermediate information, and output evaluation.

1 INTRODUCTION

Tracking systems are fundamental to immersive Extended Reality (XR) applications, facilitating accurate and real-time navigation and mapping that are crucial for creating immersive and interactive experiences [12, 52]. However, various challenges must be addressed for accurate tracking, particularly in human-centered scenarios like XR [19, 69]. Tracking systems in XR face additional challenges due to human factors such as unpredictable movements, inter-individual variability, contextual factors, cognitive load, occlusions from body parts, physical safety concerns, adaptive requirements, and the need for real-time interaction. These elements introduce layers of unpredictability and complexity, further complicating the tracking process. These challenges are intertwined, involving (1) the environment’s complexity, (2) various locomotion demands, and (3) the inherent limitations of sensing and tracking systems. As a result, while tracking methods are often presented as generic, their performance significantly varies across different environments, locomotion scenarios, and application settings, such as drones [18], autonomous vehicles [40], robotics [114], and other human-centered [36, 68, 102] and non-human-centered environments [6, 64].

To understand these challenges, it is essential to examine the specific factors contributing to the complexity of broader tracking systems and how human factors add to their complexity. First, the complexity of the environment can vary with the number of objects, lighting conditions, occlusions, weather, reflective surfaces, and scene changes. For instance, tracking in a crowded urban setting with changing lighting and reflective surfaces is particularly difficult. In XR applications, this complexity is heightened by the unpredictability of human interactions and the dynamic nature of

Authors’ addresses: Yasra Chandio, University of Massachusetts Amherst, United States; Khotso Selialia, University of Massachusetts Amherst, United States; Joseph DeGol, Steg AI, United States; Luis Garcia, University of Utah, United States; Fatima M. Anwar, University of Massachusetts Amherst, United States.

the environment. Additionally, humans can seamlessly transition between different environments, such as walking from a room to a corridor to the outdoors, without a break. This continuous movement across varied settings introduces additional challenges for tracking systems, as they must constantly adapt to new conditions and maintain accuracy. Second, locomotion differs across applications. Vehicles, robots, and humans move differently, each posing unique challenges. In human-centric applications, such as XR, abrupt movements can cause blurred images. Even in-vehicle navigation typically involves fewer abrupt movements, maintaining tracking accuracy is difficult due to varying speeds and accelerations in parking lots, urban areas, and highways [64]. Tracking human movement adds another layer of complexity, as humans frequently stop, walk, and change speeds unpredictably, making consistent speed maintenance challenging. Third, sensors such as IMU sensors, depth cameras, and RGB cameras each have specific issues [30]. IMU sensors can drift over time [5, 97], depth cameras struggle with lighting and reflective surfaces [46], and RGB cameras are affected by lighting variations. In XR applications, these sensor limitations are compounded by the need to integrate data from multiple sources in real-time [100].

To overcome these challenges, prior work has developed tracking algorithms that leverage various computational approaches. For example, traditional SLAM methods heavily depend on carefully engineered features and manually designed system components [20, 73, 81, 122]. These methods often lack robustness, meaning they struggle to maintain accuracy and reliability in dynamic and diverse real-world scenarios where conditions can vary significantly. Factors such as changing lighting conditions, moving objects, and varying environmental textures can degrade their performance. Conversely, end-to-end learning approaches [14, 105, 118] learn system components directly from data, which can lead to improved adaptability. However, these approaches can also face robustness issues, as they may fail to generalize when encountering unfamiliar situations or environments not represented in their training data. Hybrid approaches [53, 94, 120] aim to enhance overall performance by combining traditional and learning-based methods, leveraging both strengths. While this improves the average case performance, it often sacrifices the best-case performance the individual approaches might achieve.

To comprehensively address these challenges, it is important not only to evaluate tracking systems within XR environments but also to compare their performance against other application domains, such as autonomous vehicles and drones. Evaluating tracking methods across these varied domains provides a broader perspective on the strengths and weaknesses of different approaches. Autonomous vehicles and drones present unique challenges, such as high-speed movement and indoor-outdoor environmental variability, which can inform improvements in XR tracking systems. By understanding how these systems perform in different contexts, we can derive insights that contribute to developing more robust and versatile tracking solutions that can be applied across multiple domains, including but not limited to XR. Additionally, it is crucial to examine how these algorithms behaved in their original use cases [28, 64, 101, 103, 116] before XR became prominent. Understanding their foundational performance and limitations in traditional applications will provide a deeper insight into their adaptability and potential enhancements needed for XR environments.

This paper aims to address these challenges by systematically understanding the challenges, technical requirements bottlenecks, and potential solution directions needed to enhance tracking performance in XR and beyond. In doing so, we make the following contributions:

- (1) **Taxonomy of challenges.** We categorize the algorithmic, environmental, and locomotion-related challenges tracking systems face and their impact on XR applications. This taxonomy provides a structured overview of the difficulties inherent in visual SLAM tracking by highlighting the specific issues that need to be addressed to improve tracking performance in various human-in-loop and other Internet of Things (IoT) systems.

- (2) **Charting tracking performance.** We quantitatively evaluate the performance of state-of-the-art tracking algorithms across three distinct datasets, each representing a different application domain, environment, motion, and tracking target with unique complexities, including representative IoT systems like autonomous vehicles and drones and human-in-the-loop systems such as XR.
- (3) **Dataset characterization.** Building on observations from our quantitative evaluation across traditional, end-to-end learning-based, and hybrid tracking systems, we conduct a preliminary proof of concept data characterization. This analysis highlights the importance of understanding how dataset properties impact tracking performance and identifies potential adaptive solutions for specific environments and use cases.

Unlike existing surveys that focus on specific applications or isolated aspects of tracking systems, our comprehensive evaluation empirically examines a broader range of scenarios and system types. This approach systematically presents challenges and performance bottlenecks across diverse contexts, providing a robust foundation for developing adaptable and reliable tracking solutions. These insights are especially valuable for XR applications, where tracking systems must adapt to the unpredictability of human behavior. By addressing current challenges and conducting proof-of-concept case studies, this paper serves as both a reference point for researchers and a springboard for future innovations in Visual SLAM tracking in XR and beyond.

2 BACKGROUND AND MOTIVATION

2.1 XR Tracking vs. Other CPS Systems

XR tracking systems differ from other cyber-physical systems like autonomous vehicles, drones, and robotics due to their need to integrate virtual and real-world elements in real-time [90]. Each of these systems presents distinct tracking challenges, but XR systems face additional complexities related to human interaction [11, 13, 82] and environmental variability [37]. Autonomous vehicles rely heavily on sensors like GPS, LiDAR, and cameras to navigate and avoid obstacles in dynamic environments [64]. While real-time data processing is crucial in both XR and vehicle systems, XR tracking demands more precision and low latency to maintain user immersion and visual coherence between virtual and real-world elements [87, 107]. Unlike vehicle systems, which only focus on navigating roads and traffic [64], XR systems must blend virtual objects with the real world in a visually coherent manner, necessitating accurate spatial understanding [47, 57, 88, 107] and low latency to maintain immersion [87].

Similarly, drone navigation involves flight control, stabilization, and obstacle avoidance, often in outdoor environments with varying weather conditions and terrains [32]. Drones utilize GPS, IMUs, and cameras to perform tasks like localization and mapping. The primary challenges in drone tracking include maintaining stability, managing energy consumption, and ensuring safety in uncontrolled airspaces [89]. In contrast, XR tracking systems must handle more complex human interactions and diverse environments [3, 44, 66]. For example, XR applications often require users to transition seamlessly between different settings [49], such as moving from a room to a corridor to an outdoor space [62, 69], necessitating quick adaptation to varying lighting conditions [34], occlusions [70], and reflective surfaces [108]. Robotic tracking systems, used in applications ranging from industrial automation [78] to service robots in healthcare [114] and hospitality [115], rely on sensors like LiDAR, cameras, and ultrasonic sensors to navigate and interact with their environment. While robotic and XR tracking share similarities in sensor usage, the key difference lies in the nature of interaction. XR tracking demand higher precision and real-time processing [55]. Additionally, XR systems must account for the unpredictability of human behavior [47], such as sudden movements and changes in speed [69, 106], adding complexity not typically encountered in robotic tracking [86].

2.2 Unique Challenges in XR Tracking

Imagine a user exploring an XR-guided tour in a museum [98]. As they approach a detailed exhibit, they make quick, abrupt movements to get a closer look, causing the tracking system to lose accuracy momentarily. The user then moves through a corridor with mixed lighting conditions, further challenging the system. Exiting into the courtyard, the bright sunlight causes reflections and shadows that the depth cameras must adjust to. Finally, the user walks into a park, where moving objects like trees and other visitors introduce occlusions, requiring the system to recalibrate constantly. This example scenario illustrates the complexity of developing robust and accurate XR tracking systems capable of handling dynamic environments, varied locomotion [112] and a range of human factors, including:

Unpredictable Movements: Human actions are often abrupt and erratic, such as sudden turns or rapid gestures, which can disrupt tracking accuracy. Users can make sudden turns, quick gestures, or rapid changes in walking speed, which can momentarily disrupt the tracking system’s accuracy. XR systems must quickly adapt to these changes to maintain an immersive experience [19].

Inter-individual Variability: Users interact differently with XR systems, depending on their familiarity with the technology, physical abilities, and personal preferences [25]. XR tracking must accommodate this variability to deliver a consistent and intuitive user experience [68].

Environmental Variability: XR systems operate in diverse environments—ranging from well-lit indoor spaces to outdoor areas with changing weather conditions. This variability demands constant recalibration without disrupting user interaction [49, 112]. XR applications often require real-time adaptation to changes in the environment or user activity, placing demands on system responsiveness [109].

Cognitive Load and Real-Time Processing: Complex interactions in XR applications can increase cognitive load, requiring tracking systems to be intuitive and minimally intrusive to avoid adding strain [42]. Low latency is critical to XR immersion [41]. Efficient algorithms that manage real-time data processing, feature extraction, and environmental mapping are necessary to avoid discomfort or disorientation.

Occlusions from Body Parts: In XR, users’ hands, arms, or other body parts can occlude sensors, interrupting tracking. The system must recalibrate quickly to regain accuracy [70].

Physical Safety Concerns: XR systems must maintain situational awareness to prevent collisions or unsafe interactions with real-world objects [35], ensuring users’ physical safety during immersive experiences [24]. This also involves mitigating risks from internal system errors and external attacks [21, 23], which can compromise the system’s accuracy or lead to unsafe scenarios [106].

These challenges emphasize the complexity of XR tracking, which must integrate human factors, environmental variability, and real-time data processing to ensure accurate, immersive experiences. Algorithms must handle environmental changes [27], manage scale [105] and depth perception [31], and perform efficient, real-time feature extraction [2] and association [2, 26].

2.3 Contributions Beyond Related Work

Previous comparative analyses of SLAM tracking methods often focus on specific methodologies or application domains, typically limited to one application such as robotics, drones, or autonomous vehicles. For instance, [51, 103] primarily explore visual odometry methods that utilize end-to-end learning, but within narrow, domain-specific contexts. In contrast, our analysis takes a more holistic approach, focusing on human-centered tracking in XR environments and highlighting the fundamental differences between conventional tracking algorithms for IoT/CPS systems and

Table 1. *Summary of studies conducting a comparative analysis of tracking methods. Terminologies: deep learning, empirical, method-level, dataset-level, sequence-level, sample-level, and environmental representation.*

Reference	SLAM /VO	Deep Learning	Empirical	Environmental	Generalizability			
					Method	Dataset	Sequence	Sample
[6, 38]	VO							
[51, 103]	VO	✓	✓					
[1, 61, 77]	SLAM		✓			✓		
[1, 75, 116]	SLAM		✓		✓			
[7–9, 29, 78, 99, 110]	SLAM				✓			
[10, 15, 33, 43, 48, 61, 63, 92]	SLAM							
[19, 28, 58, 59, 64, 65, 91, 93]	Both	✓		✓				
[17, 30, 45, 66, 80, 85]	Both			✓				
[39, 56, 67, 71, 84, 117]	Both		✓	✓				
Ours	Both	✓	✓	✓	✓	✓	✓	✓

XR-based applications. By evaluating performance across three diverse datasets, two representing IoT/CPS use cases (drones, cars) and one representing an XR use case, we provide a broader perspective on the unique challenges faced in human-in-the-loop tracking systems and other domains. This comparative approach reveals distinct subtleties across different application domains, offering insights that previous studies and surveys have not comprehensively addressed, as summarized in Table 1.

First, we construct a *taxonomy designed to classify challenges systematically* with a exploration of *algorithmic, environmental, and locomotion* challenges in SLAM (§3). Unlike traditional SLAM comparisons, we explore how these factors interact within dynamic, unpredictable human-centered environments, such as those encountered in XR. We emphasize the interaction between environment and movement and its impact on methods. We bridge the gap between traditional SLAM and contemporary end-to-end learning techniques, offering a unified perspective on their evolution and interrelation. Our approach combines both qualitative insights and quantitative evaluations, leading to practical recommendations and future research directions for XR-based tracking systems.

Second, we evaluate the performance of state-of-the-art visual tracking algorithms across applications and datasets §4, analyzing four hierarchical dimensions: method, dataset, sequence, and sample levels. We explore trade-offs between SLAM approaches, from classical to deep learning models, in human-centered XR environments. At the dataset and sequence levels, we examine how environmental conditions and transitions between different settings affect tracking accuracy. At the sample level, we assess challenges like lighting changes and unpredictable user movements. We also compare SLAM, deep learning, and hybrid methods, and analyze the impact of human factors on tracking performance.

Third, we propose three strategies to enhance SLAM systems in dynamic environments (§5): input profiling, intermediate insights, and output evaluation. Input profiling tailors tracking to specific environmental conditions, while intermediate insights allow for real-time adjustments. Output evaluation refines performance by learning from error patterns, improving system robustness and adaptability across diverse scenarios.

2.4 Hypothesis Statement

We hypothesize that *most tracking algorithms are designed for highly standard operating conditions for specific applications. As a result, they perform poorly across applications and scenarios within a given application in real-world settings.*

3 VISUAL SLAM: CHARTING THE UNCHARTED TERRITORIES

In this section, we first present an overview of the state-of-the-art tracking approaches that either leverage traditional visual Simultaneous Localization and Mapping (SLAM) methods, end-to-end learning approaches, or their combination. We also detail our key contribution of conceptualizing and devising a taxonomy of challenges. The goal of analyzing the tracking methods and categorizing their challenges is to guide our evaluation of tracking approaches in Section 4.

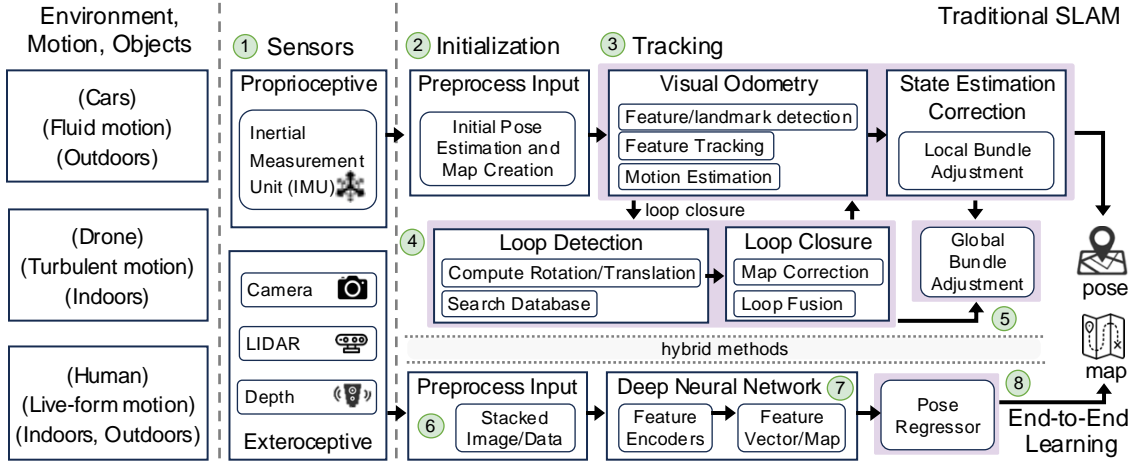


Fig. 1. An overview of traditional SLAM- and end-to-end learning-based tracking methods and their components.

3.1 Visual SLAM Tracking

SLAM is a technique used by a wide range of subjects, such as XR headsets, robots, and autonomous vehicles, to build a map of an unknown environment (mapping) in which they navigate while simultaneously tracking their current location and orientation (localization) [19]. Figure 1 illustrates different components in a SLAM pipeline and various methods used for each step. The state-of-the-art SLAM methods can be broadly classified into three categories: (i) traditional SLAM methods, (ii) end-to-end learning-based methods, and (iii) hybrid methods. These methods differ in how they extract and process information from the sensor data. For example, traditional methods may extract features from images or estimate motion based on pixel movement, while learning-based methods may use raw pixel data. Below, we discuss the tasks performed by a tracking pipeline and outline how different SLAM methods perform these tasks.

3.1.1 Visual SLAM Components and Pipelines. The SLAM pipelines track various types of objects (human wearing an XR headset, drone, robot, car) that have idiosyncratic motion patterns (live-form, turbulent, fluid) in various environments (indoor/outdoor, urban/rural), as shown in Figure 1. While additional combinations of these attributes exist, we focus on these examples as they cover most scenarios where visual SLAM tracking is employed.

- (1) The first example is of humans wearing headsets in XR environments. Humans have a live-form motion as they can make complex movements in many directions as often as they want. Humans experience the XR environment in both indoor and outdoor settings.
- (2) The second example is of a drone in indoor settings, where they keep track of inventory, monitor an industrial plant, or track people across a factory floor. They have turbulent locomotion forms as they must navigate complex environments and avoid collisions with other objects or in-operation machinery.
- (3) The third example is of an autonomous car on the street or a highway, where they generally navigate the environment in a streamlined and fluid motion, i.e., their motion is often predictable.

Given the three example scenarios, we next discuss the visual SLAM pipeline, outlining each task. We outline the different steps in the pipeline when using the traditional SLAM method, based on deep learning-based end-to-end methods or hybrid methods that leverage both approaches. The first step of a tracking pipeline involves data collection from the environment, which is shared across SLAM categories.

- ① A SLAM pipeline starts with data collected from sensors that observe the object and the environment. There are two types of sensors employed in SLAM applications. One type consists of proprioceptive sensors that observe a phenomenon produced and perceived within a subject. Examples of such sensors include the Inertial Measurement Unit (IMU). The second type is exteroceptive sensors, which observe stimuli external to the subject. The example of such sensors include cameras, LIDAR, and depth sensors [30].

Sensors are part of device hardware, such as XR headsets, and they pass the sensed data to the SLAM pipeline. Next, we describe the several key stages of a traditional SLAM pipeline. Traditional methods also differ depending on whether they only track motion, such as visual odometry, or also keep a global environmental map. We use the latter method for subsequent discussion of traditional SLAM pipelines.

- ② In this first step of the traditional SLAM pipeline, data from all sensors is cleaned and processed to estimate the initial pose of the object (such as a human wearing a headset in XR) and create an initial map of the environment (such as a dimly lit room or the outdoor environment) in which the object navigates.
- ③ The initial pose and map, alongside the preprocessed data, are passed to the tracking module, which is the most important stage of the SLAM pipeline. The first step of the tracking stage leverages visual odometry to detect features or landmarks in the environment. These features can be key points, descriptors, or other significant patterns. Visual odometry also tracks the features/landmarks over time and uses these features to estimate motion, i.e., pose over time [20, 73, 122]. Keyframe-based methods identify and use key points or distinct points in the image for its operation, which can be used for matching, tracking, or reconstruction. The tracking stage is also responsible for state estimation correction, which is performed by the local bundle adjustment module by minimizing the difference between the observed position of points and their estimated position.
- ④ Another key component is the loop closure detection module, which determines whether the object has returned to a place it has visited before. The loop detection module keeps track of previously visited places in a database. Since the object may not reach a previously visited location from the same direction and angle, the loop detection module uses various rotations and translations when matching the visited locations in the database. Recognizing a previously visited place enables map correction and ensures the trajectory's long-term consistency [54].
- ⑤ Finally, the global bundle adjustment module adjusts camera poses and 3D point positions to minimize the sum of re-projection errors across all camera views. This module performs the same task as the local bundle adjustment but at the global level. The final output of the SLAM pipeline is a map of the environment in which the object navigates and the object's pose within that map.

The traditional SLAM approaches rely on detailed physical models of the world to exploit the sequential nature of the observations of the environment. While this approach works well in stable environments, it can struggle in dynamic environments with many moving objects. Recent work has shown that deep learning-based approaches can be helpful in various aspects of the SLAM pipeline. Recent advances in deep learning have enabled end-to-end learning methods that map the raw sensor data to the desired output (the pose and the map) using neural networks for processing images, detecting features, estimating depth, and performing other related tasks. These direct learning-based methods work on the raw pixel intensities of images rather than on extracted features, enabling a dense environment model [64]. They are especially valuable in texture-less regions or when capturing dense information. An end-to-end learning-based approach removes the need to manually design and tune the different stages of the SLAM pipeline.

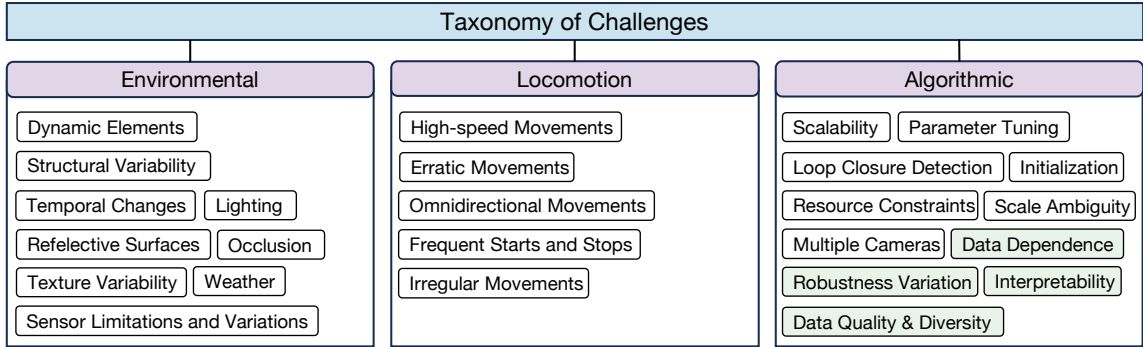


Fig. 2. A taxonomy of challenges tracking algorithms face. The end-to-end learning challenges are highlighted in green.

- ⑥ The first step of the learning-based tracking pipeline preprocesses the raw data from various sensors. The deep learning approaches excel at extracting information from the image, high-level features, and their representations, which are then fed as input into the deep neural network-based model.
- ⑦ The preprocessed data from each sensor is generally fed to sensor-specific feature encoders. These feature encoders generate high-level representations of the data. In some end-to-end learning pipelines, the outputs of sensor-specific feature encoders may be fed to another encoder to generate a combined high-level representation. The high-level representations are called feature vectors, representing the environment and state of the tracked object within the environment. The feature vectors are then passed to the next step of the pipeline.
- ⑧ The combined feature vectors encapsulating the high-level representations from the neural network are then fed into a pose regressor. The pose regressor is also a neural network trained to predict the pose of the object based on the feature vectors. The combined feature vectors can also generate a map of the environment as well.

The hybrid tracking methods replace one or more components of the traditional SLAM pipeline with learning-based methods. This could involve replacing components like feature extraction, data association, mapping, loop closure, and pose estimation. The choice of component to replace depends on the application’s specific requirements, the available data, and the computational resources. Hybrid approaches aim to leverage the explicit performance guarantees of traditional SLAM methods, which are often based on well-understood mathematical principles, and learning-based methods, which can learn complex mappings from data and be robust to environmental changes [83].

3.2 Taxonomy of Challenges

The unique human factors of XR, combined with the inherent complexities of SLAM, pose a broad set of intertwined challenges. To navigate this complex landscape, we categorize the challenges into environmental, locomotion, and algorithmic challenges. Figure 2 presents our taxonomy of challenges, whose different categories we next discuss.

3.2.1 Environmental Challenges. In this category, we include challenges that arise from the number and dynamics of elements in the environment, their perceptibility, and the perception capabilities of the sensors. The ability of the tracking systems heavily depends on the number of dynamic elements in the environment, the temporal characteristics of the environment, and its structural variability [60, 119, 121]. For example, tracking methods employed by autonomous vehicles perform better in highway settings with fewer elements that are also more predictable than in a chaotic city street environment. It is important to note that humans wearing XR headsets face environmental elements with unique

characteristics and broader spatiotemporal dynamics than traditional tracking applications. Humans in XR scenarios are potentially exposed to a broader range of environmental elements with different locomotion characteristics, such as pets, humans in the vicinity, inanimate objects, bicycles on the sidewalk, and vehicles on the road. On the other hand, a drone monitoring an industrial facility will likely not come across a teenager on a bicycle using her phone. Similarly, a car on a city street or a highway will not encounter a toddler fighting a pet.

The ability of an environment and its elements to be sensed is one of the most significant factors in determining the accuracy of the tracking systems. For example, lighting variations can destabilize a tracking system's consistency as light changes the image characteristics [92]. Similarly, reflective surfaces and occlusions can impede tracking algorithms by presenting optical illusions and obstructing the field of view, which may appear as an object exiting the scene. The environment's textural characteristics are crucial in tracking features over time, and textural variations can result in poor tracking performance. Diurnal and seasonal changes to the environment can also impact accuracy, especially for learning-based methods, if they are not trained using a representative dataset. Finally, an environment's ability to be sensed significantly varies with the weather as clouds, snow, and fog yield lighting changes, reflective surfaces, and occlusions [17]. Unfortunately, XR systems are exposed to environments that significantly vary in their characteristics, which determine how easy it is to sense objects. For example, humans effortlessly move from dimly lit bedrooms to streets with occlusions to hiking paths with ample sunlight or snow reflections.

Finally, the ability of the sensors to accurately sense the environmental elements adds another layer of complexity. Sensors' abilities to sense the environment vary based on their specifications, such as the frame rate they can support, their focal length, and sensor sizes. The sensor's capabilities can vary across application scenarios and impact the tracking accuracy. In addition, even if two sensors are the same based on all the metrics mentioned above, the sensor can have variations due to manufacturing defects or natural variations in the materials used to synthesize them [4, 30, 85].

3.2.2 Locomotion-based Challenges. The movement of the objects being tracked significantly impacts the accuracy of a tracking approach. The movement of an object can be categorized based on its speed, predictability, directionality, and regularity. Tracking objects at high speeds or with significant variations in speed significantly impacts the accuracy of the tracking systems. For example, tracking humans in XR is challenging, as humans adapt their speed depending on the situation and can stay still, walk, jog, or run. The erratic movements in all possible directions, i.e., omnidirectional movements, change the system state in an unpredictable and significant way, impacting the accuracy of tracking [72, 112]. Finally, these are magnified in XR scenarios involving humans whose motion is characterized by frequent starts and stops [96]. Some of the tracked objects, such as drones and humans, can also have irregular movement patterns.

3.2.3 Algorithmic Challenges. The fields of tracking systems using SLAM- and odometry-based have made significant strides over the years. However, numerous aspects of tracking algorithms must consistently adapt to handle the new applications in increasingly complex environments. The accuracy of the tracking algorithm is dependent on the system initialization, which determines the accuracy of downstream operations [51]. Handling the environmental dynamics requires parameter tuning, which involves determining how frequently to adjust the parameters and the scale of changes to be made. As various characteristics of the XR environment can change, such as the size of objects in the environment, the algorithms must be robust across varied scenarios [92] that XR tracking systems may face.

Another key challenge is scaling tracking algorithms to expanded maps and complex environments, which requires significant computational resources. For example, Loop Closure Detection requires identifying previously visited locales, which scales as the environment expands [54]. These issues are exacerbated in XR environments that pose the additional challenges of constrained resources [84]. Additionally, XR headsets often have multiple cameras, which introduces

Table 2. *Summary of state-of-the-art tracking methods used in the evaluation.*

Algorithm	SLAM	VO	Deep Learning	Key frame-based	Feature-based	Direct
ORBSLAM3-stereo	✓			✓	✓	
ORBSLAM3-mono	✓			✓	✓	
VINS-Fusion	✓	✓				
DSM	✓					✓
DROIDSLAM	✓		✓	✓		✓
SfmLearner	✓	✓	✓			✓
KP3D		✓	✓	✓	✓	
Tartanvo		✓	✓	✓	✓	
DFVO		✓	✓	✓	✓	
Deepvo		✓	✓			

synchronization and calibration challenges [46, 78, 111]. These issues impact the quality and consistency of the available data to the algorithms and the data used to train the models. The performance significantly degrades if input data distribution significantly shifts. Also, the robustness of these methods may vary across application and environment scenarios. Finally, unlike traditional SLAM methods, learning-based methods are black boxes and offer little to no interpretability. For reliable tracking, it is crucial to understand and solve multifaceted challenges for applications, including autonomous driving, robotics, and mixed reality.

4 VISUAL SLAM: A COMPREHENSIVE ANALYSIS IN XR AND IOT ECOSYSTEMS

In this section, we compare the performance of state-of-the-art visual tracking algorithms outlined in Section 4.1.1 across the applications and datasets described in Section 4.1.2 using the metrics presented in Section 4.1.3. In presenting our findings, we answer the following questions.

- (1) How do algorithms compare in their method-level, dataset-level, sequence-level, and sample-level performance?
- (2) How does the choice of SLAM components, such as DL vs. traditional, impact end-to-end tracking performance?
- (3) How do SLAM, DL, and Hybrid methods compare in tackling environmental, locomotion, and algorithmic challenges?
- (4) How do human factors prevalent in XR impact the performance of the tracking algorithms?

4.1 Methodology

This section presents our methodology for quantitative comparative analysis of visual SLAM tracking methods.

4.1.1 State-of-the-Art Tracking Algorithms. We select a representative set of state-of-the-art tracking algorithms that utilize various SLAM components and pipeline approaches that we described in Section 3.1 to analyze visual SLAM performance comprehensively. Table 2 lists the tracking methods we choose and the computational techniques they use.

In traditional SLAM methods, we choose ORB-SLAM3 [20] that leverages Oriented FAST and Rotated BRIEF (ORB) [76] algorithm for feature extraction, tracking, and mapping. We use both monocular and stereo versions of ORB-SLAM3. The other two SLAM methods are VINS-Fusion [74], and Direct Sparse Mapping (DSM) [122] that use visual odometry (VO) and direct methods, respectively. VINS-Fusion employs unique feature processing to integrate visual information with inertial data. DSM does not rely on feature extraction but directly operates on pixel values.

We picked several hybrid methods that combine learning components with traditional methods. DROID-SLAM [95] utilizes deep neural networks for depth prediction, camera pose estimation, and loop closure. It uniquely uses both keyframe detection and direct methods. SfmLearner [120] combines SLAM, VO, and deep learning to directly learn structures from motion (SfM) [80] without needing explicit feature extraction. KP3D [94] emphasizes keyframe and feature-based methods, leveraging self-supervised deep learning to improve VO task performance. TartanVO [105] also uses deep learning for VO and incorporates keyframe detection and feature-based techniques. DFVO [113] combines deep learning and feature-based VO to merge neural networks with traditional feature extraction and matching.

In our analysis, DeepVO [104] is the only pure deep learning-based end-to-end VO system, and it does not rely even on traditional keyframe detection or explicit feature extraction.

4.1.2 Datasets and Application Scenarios. We select three datasets that map to the three sample combinations of the environment, object, and motion, shown in Figure 1. While there can be other possible combinations, these three use cases represent the most common tracking applications and the most widely used datasets for these applications.

- (1) **KITTI [40]** is a benchmarking dataset that covers urban *outdoor environments* captured from a car with *fluid fast speed motions*. The dataset includes 22 color and grayscale stereo sequences with IMU and depth point clouds, of which 11 sequences are provided with ground truth using a Velodyne laser scanner and a high-precision GPS.
- (2) **EuRoC [18] Micro Aerial Vehicle (MAV)** covers *indoor environments* across two sites of varying structural, textural, and motion (fast/slow) difficulties. It includes machine halls and rooms with flat and structured environments. It is captured using AscTec Firefly hex-rotor MAV equipped with two stereo cameras and IMU with ground truth, making it ideal for testing applications with *turbulent motion*.
- (3) **HoloSet [22]** is collected using a Mixed Reality headset - Microsoft HoloLens2 [50] worn by a human user in both *indoor and outdoor environments*. It provides data from four grayscale, two depth, and one color cameras with IMU and ground truth. It captures *live-form motion* with substantial human factors in data collection reflecting typical human movements, interactions, and occlusions.

4.1.3 Performance Metrics for Evaluation. We use the quality of the trajectory estimated by the tracking method to evaluate and compare performance. While an accurate trajectory does not imply a good map or error-free operation, it is the most commonly used metric to measure the accuracy of SLAM methods. We use two widely used trajectory metrics for our quantitative analysis:

- **Absolute Trajectory Error (ATE)** measures the difference between the translation parts of two trajectories after aligning them into a common reference frame [79]. Mathematically, if T_{gt} is the ground truth trajectory and T_{est} is the estimated trajectory, then:

$$ATE = \sqrt{\frac{1}{N} \sum_{i=1}^N \|T_{gt}(i) - T_{est}(i)\|^2} \quad (1)$$

- **Relative Pose Error (RPE)** measures the difference between relative transformations at time instances i and $i + k$ for different values of k . If $\Delta T_{gt}(i, i + k)$ and $\Delta T_{est}(i, i + k)$ are the relative transformations for ground truth and estimated trajectories respectively, then:

$$RPE = \sqrt{\frac{1}{N-k} \sum_{i=1}^{N-k} \|\log(\Delta T_{gt}(i, i+k)^{-1} \Delta T_{est}(i, i+k))\|_{\mathcal{F}}^2} \quad (2)$$

where \log denotes the matrix logarithm and $\|\cdot\|_{\mathcal{F}}$ is the Frobenius norm [16]. This metric is independent of the reference frame, but when the scale of the map is unknown (e.g., monocular mapping), scale alignment needs to be performed before comparing trajectories using RPE [79].

- **Summary Metrics.** In addition, we define three summary metrics that allow us to compare the performance of algorithms across sequences and datasets. The three metrics include (i) the number of sequences for which an algorithm yields the lowest error, (ii) the average error across sequences, and (iii) the coefficient of variation (CoV) in error across sequences, computed as the standard deviation over the mean.

Table 3. ATE [m] and RPE [m] on EuRoC (Note: (-) ATE above 100m or RPE above 1, (xx) complete failure).

Algorithm	ATE, RPE										
	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203
ORB-SLAM3 (S)	0.24, 0.005	3.11, 0.002	2.54, 0.024	2.53, 0.014	2.62, 0.002	0.49, 0.003	0.57, 0.001	0.64, 0.009	0.67, 0.007	0.84, 0.004	0.22, 0.001
ORB-SLAM3 (M)	4.41, 0.029	4.99, 0.034	4.57, 0.080	2.03, 0.010	2.29, 0.011	1.01, 0.011	1.26, 0.029	1.29, 0.030	1.31, 0.008	0.90, 0.013	1.59, 0.035
VINS-Fusion	3.91, 0.056	3.97, 0.060	2.68, 0.099	2.57, 0.042	3.08, 0.047	1.11, 0.022	0.77, 0.027	0.32, 0.009	1.26, 0.017	1.13, 0.034	1.16, 0.012
DSM	6.24, -	5.38, -	6.80, -	6.04, -	7.05, -	7.10, -	6.48, -	6.35, -	6.21, -	8.53, -	9.06, -
DROID-SLAM	2.94, 0.073	4.25, 0.074	1.86, 0.145	4.16, 0.074	3.36, 0.075	1.01, 0.024	1.63, 0.041	12.16, 0.020	1.41, 0.031	1.20, 0.054	1.43, 0.092
SfMLearner	4.49, 0.673	4.76, 0.750	3.45, -	5.50, 0.459	5.46, 0.550	1.65, 0.524	1.84, 0.675	1.90, 0.882	2.04, -	2.11, 0.869	2.11, 0.885
KP3D	0.22, 0.064	0.21, 0.084	0.20, 0.158	0.18, 0.053	0.25, 0.025	0.37, 0.022	0.54, 0.021	0.71, 0.021	0.48, 0.049	1.07, 0.070	0.39, 0.033
TartanVO	1.67, 0.003	1.62, 0.003	2.97, 0.009	2.37, 0.005	2.15, 0.004	0.54, 0.001	0.69, 0.006	0.53, 0.003	1.10, 0.004	1.37, 0.009	1.16, 0.006
DFVO	-,-	-,-	-,-	39.1, 0.160	60.9, 0.054	6.35, 0.006	-,-	-,-	24.7, 0.021	-,-	-,-
DeepVO	1.67, 0.091	1.59, 0.109	1.65, 0.176	1.56, 0.122	1.48, 0.116	1.84, 0.028	2.11, 0.021	1.71, 0.039	2.02, 0.029	2.18, 0.098	1.85, 0.075

Table 4. ATE [m] and RPE [m] on KITTI. Note: DSM failed on all sequences.

Algorithm	ATE, RPE										
	00	01	02	03	04	05	06	07	08	09	10
ORB-SLAM3 (S)	4.80, 0.022	11.2, 0.061	9.59, 0.039	4.59, 0.016	3.30, 0.016	4.72, 0.013	4.60, 0.019	xx, xx	7.27, 0.031	6.80, 0.028	6.43, 0.021
ORB-SLAM3 (M)	15.6, 0.151	10.1, 0.073	17.5, 0.101	8.94, 0.078	2.72, 0.090	5.94, 0.080	16.8, 0.247	11.6, 0.143	5.22, 0.077	13.1, 0.130	15.8, 0.136
VINS-Fusion	-,-	-,-	-,-	14.4, 0.139	32.6, 0.454	73.5, 0.400	38.1, 0.383	26.6, 0.212	41.2, 0.216	53.1, 0.406	39.6, 0.258
DSM	xx, xx	xx, xx	xx, xx	xx, xx	xx, xx	xx, xx	xx, xx	xx, xx	xx, xx	xx, xx	xx, xx
DROID-SLAM	-,-	-,-	-,-	-,-	-,-	-,-	-,-	85.5, 2.305	-,-	-,-	-,-
SfMLearner	-,-	-,-	-,-	-,-	-,-	-,-	-,-	91.4, -	-,-	-,-	-,-
KP3D	15.2, 0.074	47.6, 0.682	34.2, 0.134	3.04, 0.061	1.94, 0.116	15.6, 0.080	4.36, 0.055	3.87, 0.043	13.4, 0.071	9.07, 0.075	10.2, 0.069
TartanVO	85.8, 0.361	48.2, 0.329	-,-	2.69, 0.046	2.30, 0.068	54.9, 0.206	6.96, 0.067	14.7, 0.114	65.4, 0.293	34.9, 0.156	13.1, 0.098
DFVO	-,-	-,-	-,-	23.1, 0.280	93.9, -	78.3, 0.320	-,-	40.4, 0.190	89.4, 0.230	-,-	-,-
DeepVO	-,-	-,-	-,-	-,-	-,-	-,-	-,-	90.9, -	-,-	-,-	-,-

Table 5. ATE [m] and RPE [m] on HoloSet. Note: ORBSLAM3 and DSM failed on all sequences.

Algorithm	ATE, RPE					
	campus-center-seq1	campus-center-seq2	suburbs-jog-seq1	suburbs-jog-seq2	suburbs-seq1	suburbs-seq2
ORB-SLAM3 (S)	xx, xx	xx, xx	xx, xx	xx, xx	xx, xx	xx, xx
ORB-SLAM3 (M)	xx, xx	xx, xx	xx, xx	xx, xx	xx, xx	xx, xx
VINS-Fusion	14.2, 0.565	13.2, 0.002	26.1, 1.000	2.68, 0.267	-,-	4.05, 0.102
DSM	xx, xx	xx, xx	xx, xx	xx, xx	xx, xx	xx, xx
DROID-SLAM	-,-	15.9, 0.524	32.8, 1.797	37.3, 6.384	-,-	72.9, 0.787
SfMLearner	24.7, -	17.4, -	33.6, -	38.4, -	-,-	79.3, -
KP3D	1.40, 0.005	1.36, 0.007	1.18, 0.005	1.03, 0.004	10.3, 0.008	1.91, 0.002
TartanVO	8.11, 0.037	15.3, 0.072	7.92, 0.045	4.77, 0.030	21.5, 0.091	45.1, 0.084
DFVO	-,-	-,-	-,-	95.2, 1.645	-,-	-,-
DeepVO	25.0, 0.577	16.9, 1.163	33.0, 0.716	37.1, 2.141	-,-	77.7, 2.734

Table 6. Algorithm Performance Summary Across Datasets Computed Using Data in Table 3, Table 4, and Table 5.

Algorithm	EuRoC			KITTI			HoloSet		
	Top	Avg ATE (m)	CoV	Top	Avg ATE (m)	CoV	Top	Avg ATE (m)	CoV
ORB-SLAM3 (S)	3	1.03	0.85	5	6.23	0.38	xx	xx	xx
ORB-SLAM3 (M)	0	2.64	0.50	0	10.72	0.47	xx	xx	xx
VINS-Fusion	0	2.49	0.49	0	39.24	0.52	1	12.81	0.75
DSM	0	6.56	0.14	xx	xx	xx	xx	xx	xx
DROID-SLAM	0	3.02	1.04	0	-	-	0	39.72	0.67
SfMLearner	0	3.27	0.47	0	-	-	0	38.68	0.62
KP3D	7	0.45	0.58	3	14.94	0.91	5	2.53	1.39
TartanVO	1	1.54	0.45	2	35.03	0.84	0	17.11	0.91
DFVO	0	32.26	0.57	1	61.70	0.54	0	95.20	-
DeepVO	0	1.77	0.13	0	-	-	0	37.94	0.61

4.2 Head-to-Head Performance Comparison at All Levels

We run all tracking methods on all datasets for a comparative quantitative analysis. All methods, except ORB-SLAM3 (stereo), use monocular images from the datasets. We post-process the results for all methods to calibrate the scales and recover poses. We report ATE as a measure of global consistency and RPE as a measure of local trajectory accuracy

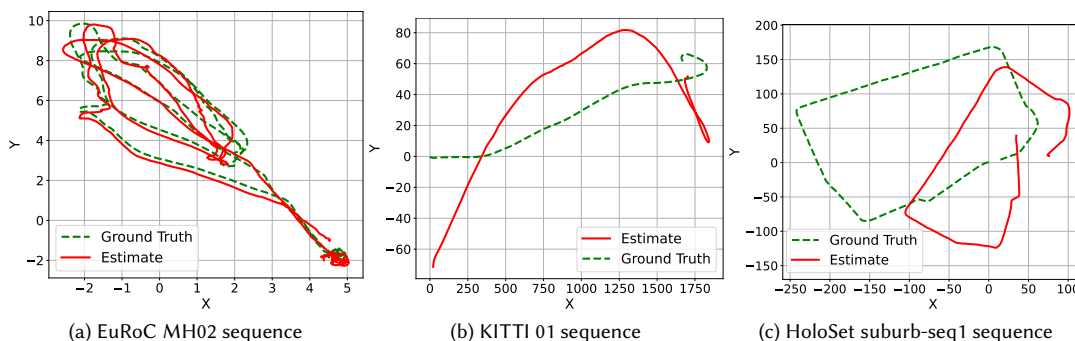


Fig. 3. **Qualitative examples from representative datasets with an estimated trajectory from TartanVO with ground truth.** in Table 3 for EuRoC, Table 4 for KITTI, and Table 5 for HoloSet. We provide the summary results in Table 6. If an algorithm fails on a given sequence, we mark the failure as “xx, xx”. If an algorithm runs but yields an ATE value above 100m or RPE of above 1, we mark that as “-, -”.

1 – Method-level Question: Are there clear winners in a head-to-head comparison across tracking methods? To answer this question, we primarily analyze the EuRoC dataset results from Table 3 and Table 6, a dataset on which all methods run, and most have errors within reasonable bounds. KP3D performs the best of 7 out of the 11 sequences, and the next best method is ORB-SLAM3 (S), which beats KP3D on 3 of the remaining four sequences. KP3D and ORB-SLAM3 (S) are the top two based on the average ATE values of 0.45m and 1.03m, respectively. However, most methods have a very low ATE across sequences: 4 algorithms have errors less than 2m, and only DVFO has an error above 7m.

Key Takeaway. No single tracking method consistently outperforms others, though most methods have low errors.

2 – Dataset-Level Question: Do tracking methods show robustness to variations in dataset characteristics? To explore this, we analyzed the performance of algorithms across three distinct datasets, as shown in Table 6. Interestingly, the best-performing algorithm was not the same across the datasets. For the KITTI dataset, ORB-SLAM3 (S) emerged as the strongest, performing best on 5 of the 11 sequences, while KP3D followed closely with 3 sequences. However, when applied to the HoloSet dataset, ORB-SLAM3 (S) failed completely, unable to generate trajectories for any sequence. This pattern was reinforced by examining ATE values and the coefficient of variation (CoV), indicating that strong performance on one dataset does not necessarily translate to others, particularly when faced with different environmental conditions, motion dynamics, and object types.

We also visually demonstrate the performance of a sample algorithm across sequences from different datasets. In Figure 3, TartanVO’s performance is illustrated across three datasets, showing significant variation across sequences within each dataset. While it performs reasonably well in the EuRoC MH02 sequence with structured, textured surfaces, its accuracy drops notably in KITTI’s outdoor high-speed sequence, where low texture and fast motion introduce more complexity. The HoloSet suburban walk sequence further highlights this inconsistency, where TartanVO struggles with scale estimation and trajectory alignment due to the absence of loop closure detection.

Key Takeaway. Tracking methods generally lack robustness when faced with diverse environmental, motion, and object characteristics across datasets, indicating variability in performance based on the dataset context.

3 – Sequence-Level Question: Do algorithms show consistent performance across sequences within a single dataset? We analyzed the results across sequences to explore this, focusing on the data in Table 3 and Table 6. Among the tracking methods, DeepVO demonstrates the most consistent performance, with the lowest coefficient of variation (CoV) across sequences at 0.13. Its average ATE of 1.77m is relatively low, positioning it among the better-performing

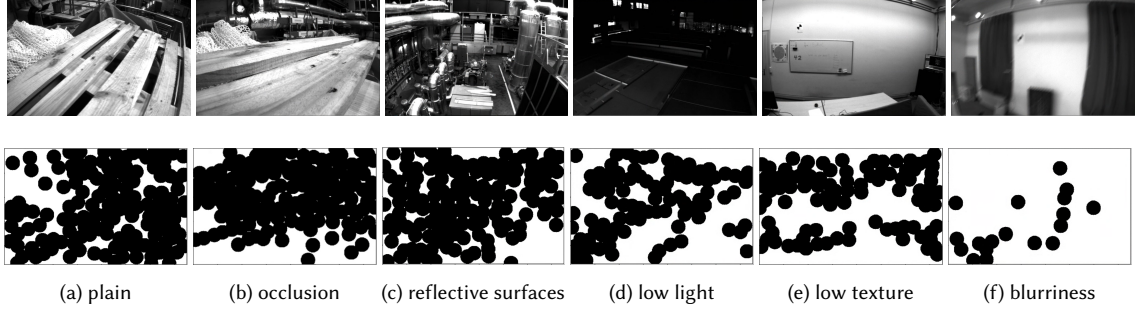


Fig. 4. *Example samples from EuRoC [18] dataset, varied environmental conditions (top) and their corresponding feature (•) mask (bottom) generated with VINS-Fusion [74] feature extractor.*

methods despite not being the top performer for any sequence. In contrast, top-performing methods like KP3D and ORB-SLAM3 (S) exhibit higher CoV values of 0.58 and 0.85, indicating more variability in performance across sequences.

TartanVO’s performance is illustrated across three datasets, showing significant variation across sequences within each dataset. While it performs reasonably well in the EuRoC MH02 sequence with structured, textured surfaces, its accuracy drops notably in KITTI’s outdoor high-speed sequence, where low texture and fast motion introduce more complexity. The HoloSet suburban walk sequence further highlights this inconsistency, where TartanVO struggles with scale estimation and trajectory alignment due to the absence of loop closure detection.

Key Takeaway. *Even though some tracking methods achieve low error for a dataset, this does not necessarily translate to consistent performance across all sequences.*

4 – Sample-level Question: Can the methods handle the variations in the samples within sequence without sample-level fine-tuning? Despite the good performance of some visual SLAM methods under “ideal conditions”, there is no clear winner on *method*, *dataset*, and *sequence* levels. Our evaluation has unveiled several *algorithmic*, *environmental*, and *locomotion* challenges that SLAM methods face on different levels of the pipeline. To investigate further, we evaluate *sample-level* attributes. We took six samples from EuRoC sequences (MH01, MH03, V203): one baseline vanilla sample and five samples with common visual features essential for data characterization in SLAM [1, 77]. These features include occlusion, lighting, reflective surface, image motion blur, and texture variability. We tracked the feature of each sample with VINS-Fusion [74] feature extractor as shown in Figure 4.

We observe that the number of features ranges from 127 per frame to a mere 17 features per frame. While certain attributes might favor the algorithms, leading to reliable tracking (as shown in Figure 4a), others can significantly strain the system, as seen in Figure 4f. Any sudden movement (causing blurry image), low lighting, low textured environment, or occlusions can hinder the methods’ view of certain features. Whereas shiny reflective surfaces can create false features, compromising tracking accuracy. A method’s ability to consistently recognize and adapt to these varying attributes without manual intervention or fine-tuning signifies its robustness.

Key Takeaway. *The disparity in results across the sequences, samples, and visual features indicates that achieving a truly generalized method for handling the broad spectrum of sample attributes remains an open question. A robust and domain-adaptive SLAM can be developed by understanding data at various levels, from method down to sample level.*

4.3 Impact of Choices in Visual SLAM Components and Pipelines

To analyze how the choice of SLAM components, such as deep learning (DL) versus traditional methods, impacts end-to-end tracking performance, we examine the results from Table 6. The performance of traditional, hybrid, and

deep learning approaches is evaluated based on their average ATE, the number of sequences for which they perform the best (Top), and the consistency of their performance as indicated by the CoV.

Traditional Methods. Traditional SLAM approaches, including ORB-SLAM3 (S) and ORB-SLAM3 (M), generally perform well on specific datasets, particularly on KITTI and EuRoC. ORB-SLAM3 (S), for instance, is the top performer on five sequences in KITTI and three sequences in EuRoC. However, these methods struggle with robustness across datasets; for example, ORB-SLAM3 (S) fails to work on HoloSet entirely. The CoV for these algorithms is relatively low on KITTI but higher for EuRoC, indicating more variability in performance across sequences within the EuRoC dataset.

Hybrid Methods. Hybrid approaches, which combine traditional and deep learning components, tend to outperform traditional and DL-only methods on overall performance and consistency. KP3D, one of the top hybrid algorithms, has the lowest average ATE on EuRoC (0.45m) and performs well on KITTI, although with a higher CoV of 0.91. These hybrid methods dominate in average ATE and are the top performers in multiple sequences across different datasets, showing the most balanced performance. However, hybrid methods also struggle on HoloSet, with higher ATE and more variability compared to their performance on EuRoC and KITTI.

Deep Learning Methods. The deep learning-based method, DeepVO, stands out for its consistency across sequences within a single dataset, demonstrated by the lowest CoV of 0.13 on EuRoC. However, its end-to-end performance in terms of average ATE is not as competitive, with values such as 1.77m on EuRoC and 37.94m on HoloSet, indicating that while it performs consistently, its overall error is higher than most hybrid and traditional approaches. Additionally, deep learning methods tend to struggle on datasets with more complex environmental or motion variations, as indicated by the large ATE on HoloSet.

Key Takeaway. *Hybrid methods exhibit the best overall performance, consistently achieving low ATE across multiple datasets and sequences. They outperform traditional and deep learning methods in terms of accuracy and robustness. While deep learning approaches offer more consistency within sequences, they generally produce higher errors, suggesting that traditional and hybrid methods are still more effective for end-to-end tracking in varied conditions.*

4.4 Performance in Tackling Environmental, Locomotion, and Algorithmic Challenges

To compare how SLAM, deep learning (DL), and hybrid methods tackle environmental, locomotion, and algorithmic challenges, we analyze their performance across the datasets (EuRoC, KITTI, and HoloSet) summarized in Table 6, in addition to the individual results for datasets. These datasets capture a range of conditions, from indoor to outdoor environments, controlled settings to dynamic scenes, and varying frame rates and resolutions. Each method’s ability to adapt to these challenges reveals insights into the strengths and weaknesses of different computational approaches.

Traditional Methods. SLAM methods like ORB-SLAM3 (S) and ORB-SLAM3 (M) perform well in controlled environments but struggle as environmental and motion complexity increases. On EuRoC’s easier sequences, such as MH01, ORB-SLAM3 (S) achieves a low ATE of 0.24m, aided by EuRoC’s compact resolution and controlled environment, which facilitates smoother loop closures and reduces scalability challenges. However, as sequences become more turbulent (e.g., V202), ORB-SLAM3 (S)’s ATE increases to 0.84m, reflecting the limitations of traditional methods in more complex indoor environments. In KITTI, while SLAM methods benefit from high-quality calibration and ground truth data, broader variability in lighting and environmental factors challenges their robustness. For instance, ORB-SLAM3 (S) records an ATE of 4.80m in sequence 00, but as environmental complexity increases in sequence 02, the ATE jumps to 11.2m. This highlights SLAM’s struggle with outdoor settings where rapid changes in lighting and environment occur. On HoloSet, SLAM methods fail entirely, and they are unable to handle the unpredictable, multi-directional motion of

the human-controlled XR headset. Both ORB-SLAM3 (S) and ORB-SLAM3 (M) are unable to produce valid trajectories, reflecting their difficulty in handling high frame rates and the uncontrolled dynamics of human motion. The lack of fine-tuning for HoloSet’s novel environmental conditions and higher frame rates further exacerbates their poor performance.

Hybrid Methods. Hybrid methods demonstrate superior adaptability across indoor and outdoor environments, effectively balancing precision and flexibility to handle varied conditions. For instance, KP3D achieves an ATE of 0.21m in EuRoC’s MH01 sequence and continues to perform well in the challenging V203 sequence with an ATE of 0.39m. The combination of feature extraction techniques and loop closure enables hybrid methods to handle both the compact resolution and high frame rate of EuRoC, maintaining low error rates even in turbulent drone motion. In KITTI, hybrid methods continue to outperform SLAM approaches, handling fluid car motion and complex environmental fluctuations more effectively. KP3D achieves an ATE of 14.9m in sequence 04, demonstrating resilience to the broader variability in outdoor conditions that challenge traditional methods. Their sophisticated feature extraction allows hybrid methods to manage the scene complexity and outdoor lighting fluctuations that typically degrade SLAM’s performance. In HoloSet, hybrid methods like KP3D excel in managing human-centered environments, with an ATE of 10.4m in campus-center-seq1. Although the ATE values increase due to the dynamic, uncontrolled environment and higher frame rates, hybrid methods remain the most robust option, able to handle conditions involving unpredictable human motion better. Their adaptability across domains – from indoor corridors (EuRoC) to urban streets (KITTI) to suburban areas (HoloSet) – positions hybrid methods as the most versatile and reliable for complex tracking tasks.

Deep Learning Methods. DL methods exhibit consistency across sequences but generally incur higher error rates, especially in environments with significant variability. While DL methods provide consistent performance, their ATE values are higher than hybrid methods. For instance, in MH01, DeepV0 records an ATE of 1.67m – consistent but less accurate compared to KP3D’s 0.21m. This indicates that DL methods are less precise in handling turbulent motion within controlled indoor environments. On KITTI, DL methods struggle more with outdoor complexity. For example, DeepV0 cannot produce results for most sequences (denoted by “–” in the table), highlighting the difficulty of applying DL methods to rapidly changing outdoor environments, particularly when lighting conditions and scene complexity fluctuate. On the HoloSet dataset, DL methods perform better than SLAM but still lag behind hybrid methods. For example, DeepV0 records an ATE of 33.0m in suburb-jog-seq1, significantly higher than hybrid methods like KP3D. While DL methods can handle unpredictable human motion, their error rates remain higher than hybrid methods, reflecting their limitations in highly dynamic, human-centered environments.

Key Takeaway. *Hybrid methods consistently outperform SLAM and DL approaches by balancing adaptability and accuracy across diverse environments, from controlled indoor settings to dynamic outdoor scenes. While SLAM methods excel in structured environments, they struggle in more complex scenarios, such as human-controlled movements in HoloSet, where they often fail. DL methods offer consistency but at the cost of higher errors, particularly in outdoor and unpredictable environments. Overall, hybrid methods demonstrate superior versatility, handling various challenges with lower error rates, making them the most robust option for complex tracking tasks.*

4.5 Effect of Human Factors in XR

The HoloSet dataset introduces significant challenges related to human factors, particularly the unpredictable and multi-directional motion associated with a human wearing an XR headset. This variability in movement, combined with changing environments from indoors to suburban outdoors, presents a unique set of challenges for tracking methods.

Traditional Methods. SLAM methods, such as ORB-SLAM3 (S) and ORB-SLAM3 (M), struggle to handle the dynamic nature of human-controlled motion. These methods rely heavily on stable, predictable environments and predefined trajectories, which are difficult to maintain when the human’s head movements introduce abrupt and irregular changes in position and angle. Consequently, SLAM methods fail entirely in HoloSet, unable to produce any valid trajectories due to the unpredictable nature of human-controlled XR interactions.

Hybrid Methods. Hybrid methods, like KP3D, demonstrate much better resilience to the human factors inherent in XR. While their ATE values increase compared to more controlled environments, they still manage reasonable performance in HoloSet. For instance, KP3D achieves an ATE of 10.4m in `campus-center-seq1`, demonstrating that hybrid approaches can handle the combination of human motion and environmental changes more effectively than SLAM methods. The flexibility of hybrid algorithms, which combine traditional feature extraction with learning-based components, allows them to adapt to the erratic, multi-directional head movements typical in XR environments.

Deep Learning Methods. DL methods, such as DeepV0, also show an ability to track human motion better than SLAM methods but with less precision than hybrid approaches. In HoloSet’s `suburb-jog-seq1`, DeepV0 records an ATE of 33.0m, significantly higher than KP3D’s performance. While DL methods benefit from their inherent flexibility in handling varying motions, their higher error rates suggest they are less adept at managing rapid, unpredictable changes in orientation and environment caused by human interaction in XR scenarios.

Key Takeaway. *Human factors in XR environments, such as unpredictable head movements and changing positions, severely degrade the performance of SLAM methods, which are not adaptable enough to handle such dynamics. Hybrid methods offer the best balance, maintaining reasonable accuracy by leveraging both traditional and learning-based techniques to adapt to erratic human-controlled movements. DL methods perform better than SLAM approaches but exhibit higher error rates, indicating that while they are more adaptable, they lack the precision needed for highly dynamic XR environments.*

5 DISCUSSION: POTENTIAL APPROACHES TO IMPROVING VISUAL SLAM

In this section, informed by our analysis from [Section 4](#), we propose three key strategies to improve tracking methods for SLAM systems, particularly in dynamic environments such as human-centered XR and IoT applications. These strategies focus on leveraging different levels of available information—input profiling, intermediate insights, and output evaluation—to enhance the robustness and adaptability of SLAM systems across diverse domains.

5.1 Input Profiling

By understanding the input data’s nuances, tracking methods or systems can preemptively adjust tracking strategies. In dynamic environments, if the system is under transition, such as moving from indoor to urban outdoor environments, dataset characterization can provide crucial insights into the variations in lighting, dynamics, and textures. Such insights can be invaluable for tweaking tracking methods and ensuring the system undergoes seamless domain transitions. Input profiling involves assessing the data characteristics at various levels: dataset, sequence, and sample levels.

1 – Dataset-level Profiling. This involves an overarching dataset analysis to determine the general conditions, such as lighting, texture, and motion profiles. Profiling the entire dataset can help tracking methods anticipate typical challenges (e.g., indoor and outdoor transitions) and allow for application-specific manual offline fine-tuning of the parameters.

2 – Sequence-Level Profiling. Sequence-level profiling enables systems to tailor tracking approaches to specific environmental conditions and motion dynamics of each sequence. As detailed in [Table 7](#) and [Table 8](#), each dataset

Table 7. *Dataset Characterization for EuRoC, KITTI, and HoloSet: Scene, Motion, and Lighting Characteristics*

Dataset	Sequence Name	Agent Type	Scene Type	Motion	Light	Extra Information
EuRoC	MH01 (easy)	Drone	Indoor	Medium	Bright	Good texture
	MH02 (easy)	Drone	Indoor	Medium	Bright	Good texture
	MH03 (medium)	Drone	Indoor	Fast	Bright	-
	MH04 (difficult)	Drone	Indoor	Fast	Dark	-
	MH05 (difficult)	Drone	Indoor	Fast	Dark	-
	V101 (easy)	Drone	Indoor	Slow	Bright	-
	V102 (medium)	Drone	Indoor	Fast	Bright	-
	V103 (difficult)	Drone	Indoor	Fast	Medium	Motion blur
	V201 (easy)	Drone	Indoor	Slow	Bright	-
	V202 (medium)	Drone	Indoor	Fast	Bright	-
V203 (difficult)	Drone	Indoor	Fast	Medium	Motion blur	
KITTI	00	Car	outdoor	Slow	Medium	Residential streets, evening, dusk, shadows
	01	Car	Outdoor	Fast	Bright	Highway, daytime
	02	Car	Outdoor	Medium	Medium	Roads and streets, dusk, shadows
	03	Car	Outdoor	Medium	Medium	Roads and streets, dusk, shadows
	04	Car	Outdoor	Fast	Medium	Roads, shadows
	05	Car	Outdoor	Slow	Medium	Residential streets, evening, dusk, shadows
	06	Car	Outdoor	Medium	Dark	Night/dusky grey, industrial buildings
	07	Car	Outdoor	Medium	Medium	Campus downtown, dusk
	08	Car	Outdoor	Slow	Bright	Residential streets, morning, shadows
	09	Car	Outdoor	Fast	Bright	Highway, daytime
10	Car	Outdoor	Slow	Medium	Residential streets in evening, dusk, shadows	
HoloSet	Campus Center (seq1)	Human	Indoor	Slow	Medium	Stairs, humans
	Campus Center (seq2)	Human	Indoor	Slow	Medium	Stairs, humans
	Suburbs Jog (seq1)	Human	Outdoor	Fast	Bright	Trees, parked cars, road
	Suburbs Jog (seq2)	Human	Outdoor	Fast	Bright	Trees, parked cars, road
	Suburb Walk (seq1)	Human	Outdoor	Medium	Bright	Trees, parked cars, road
	Suburb Walk (seq2)	Human	Outdoor	Medium	Bright	Trees, parked cars, road

presents sequences with unique challenges, including variations in motion (fast or slow), lighting (bright or dark), and scene complexity (structured indoor environments or dynamic outdoor scenes). By leveraging this information, tracking algorithms can be optimized for the unique characteristics of each sequence in an online or offline manner.

In the EuRoC dataset, for instance, sequences like MH01 and MH02 are easier, featuring medium motion in bright, well-textured indoor environments. Our texture profiling shows that MH01 and MH02 have high texture scores (0.269 and 0.218) and low percentages of low-light conditions, making them favorable for tracking. Algorithms can optimize their performance by focusing on structured feature extraction and loop closure in these sequences, benefiting from the good lighting and texture availability. In contrast, sequences like MH04 and MH05 introduce significant difficulty with fast drone motion in dark indoor scenes, reflected in their high low-light percentages (83.7% and 84.4%) and moderate texture scores. In these cases, tracking systems need to rely more heavily on inertial data or enhance feature extraction in low-light conditions. Additionally, sequences like V103 and V203 introduce motion blur, posing further challenges. Algorithms can improve performance by detecting these conditions in real time and adjusting tracking strategies, such as applying motion compensation techniques or increasing reliance on temporal smoothness to reduce drift.

The KITTI dataset presents outdoor, car-based sequences where motion and lighting vary significantly between residential streets, highways, and industrial areas. For instance, KITTI sequence 01 involves fast motion on a highway in bright daylight, where tracking methods must be optimized for high-speed motion, ensuring features are tracked reliably despite rapid changes in the scene. In contrast, sequences like KITTI 06 are set in dark, industrial environments with medium motion, where our profiling shows low texture and low-light conditions, necessitating adjustments in feature extraction sensitivity. For residential street sequences like KITTI 00 and KITTI 05, tracking algorithms need to handle medium lighting and shadows caused by objects like parked cars and trees. Sequence-level profiling in KITTI helps systems adapt tracking techniques for each sequence’s environmental attributes, improving robustness.

Table 8. *Profiling Lighting and Texture on EuRoC.*

Sequence	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203
Texture score (norm)	0.269	0.218	0.123	0.238	0.237	0.154	0.117	0.163	0.218	0.197	0.090
Low Lighting Profile (%)	26.7	29.0	19.2	83.7	84.4	3.4	2.1	26.2	8.7	11.4	24.7

The HoloSet dataset introduces human-centered sequences with both indoor and outdoor settings. These sequences feature a human wearing an XR headset, resulting in slower and more unpredictable motion than the drone or car-based sequences in EuRoC and KITTI. For the Campus Center sequences, motion is slow and set indoors with medium lighting. Here, systems can focus on identifying and tracking key features in cluttered scenes with human movement and stairs while adjusting to the slower motion. In contrast, suburb jog sequences involve fast motion in bright, outdoor environments with trees, parked cars, and roads. Profiling these sequences shows the need for systems to anticipate rapid changes in scene structure and motion while handling outdoor lighting challenges. Like KITTI, environmental elements like trees introduce occlusion challenges, and tracking algorithms can prioritize high-texture regions while compensating for motion blur and abrupt human movements.

By tailoring tracking methods to the specific conditions of each sequence—whether it involves fast motion, complex geometry, or challenging lighting—algorithms can dynamically adjust their processing techniques. For example:

- **Fast motion** (e.g., KITTI 01, HoloSet Suburbs Jog seq1) requires robust feature tracking algorithms that account for rapid changes in the environment, potentially leveraging inertial data to maintain stability.
- **Low light** sequences (e.g., EuRoC MH05, KITTI 06) benefit from adaptive feature extraction methods that adjust sensitivity to contrast and compensate for the lack of visual information in darker environments.
- **Motion blur** (e.g., EuRoC V203) and **occlusions** (e.g., KITTI 00) require algorithms to compensate by predicting feature movement based on prior frames or relying on temporal coherence for smoother tracking.

By implementing sequence-level profiling, SLAM systems can better navigate the varied environmental and motion challenges posed by different datasets, leading to improved tracking accuracy and robustness across diverse domains like indoor XR environments and outdoor IoT applications.

3 – Sample-Level Profiling. This involves real-time analysis of individual frames or segments of a sequence to understand changes in lighting, texture, or motion. For example, in sequences with sudden lighting transitions or occlusions, real-time profiling can adjust feature extraction methods dynamically, preventing the loss of tracking due to momentary visual obstructions or environmental changes.

By systematically profiling the input at different levels, SLAM systems can dynamically adapt to varying scene characteristics, improving robustness and accuracy across domains.

5.2 Intermediate Information

The second approach involves leveraging intermediate values and insights during the tracking process, much like the approach discussed in the nFEX paper [26]. Intermediate information, such as feature tracking confidence or feature density in a region, can be used to adjust the tracking algorithm mid-process dynamically.

Feature Extraction Feedback. SLAM systems can use intermediate feedback about feature quality and density to improve tracking accuracy. For instance, if the system detects that feature tracking is failing due to low texture or occlusion, it can switch to alternate tracking strategies, such as using inertial data, predicting motion based on previous frames, or increasing the maximum number of features tracked.

Adaptive Sampling: By analyzing intermediate outputs, the system can adjust the frequency of feature extraction or the regions of interest for tracking. This approach is particularly useful for scenes with varying complexity, such as those that move from structured environments with predictable motion (EuRoC’s indoor sequences) to unstructured, complex environments (HoloSet’s human-centered suburban sequences).

Intermediate information enables a more responsive and adaptive tracking system, improving accuracy by making real-time adjustments based on evolving scene characteristics.

5.3 Output Evaluation

The final approach involves analyzing the tracking system’s outputs, particularly errors, and using this information to refine and improve its performance. By evaluating output errors, the system can make post-process adjustments or inform future tracking sessions.

Error Correction. By evaluating discrepancies between the estimated trajectory and ground truth (as shown in the qualitative TartanVO analysis in Figure 3), systems can identify error patterns specific to certain environments or conditions. For example, if a system consistently fails to track correctly in low-texture environments, it can reweight the importance of visual features versus inertial data.

Trajectory Feedback Loops. Systems can use output evaluation to refine future predictions. For example, if a system consistently underestimates scale in outdoor environments (as seen in the HoloSet sequences), it can introduce corrections based on prior tracking errors, allowing the system to adjust its internal model of the scene over time.

Cross-Domain Adaptation. By analyzing output errors across different domains (e.g., indoor to outdoor transitions in EuRoC vs. KITTI or the human-centric scenes in HoloSet), systems can gain insights into how to improve domain adaptability. Output evaluation helps to pinpoint weaknesses that may not be evident during training but become apparent in real-world, cross-domain deployments.

By systematically evaluating the outputs, SLAM systems can continuously improve their tracking accuracy, refining their methods based on real-world data and improving their resilience across different environments.

6 CONCLUSION AND FUTURE WORK

Our study is unique because it goes beyond mere qualitative comparisons of datasets and prior work using single metrics from traditional studies. Instead, we conduct a comprehensive analysis to evaluate the multiple tracking methods across a wide range of datasets. Our research effort in devising the taxonomy of challenges and performing comparative analysis provides key insights into the workings of SLAM pipelines. We show that understanding and embracing data diversity across application scenarios, sequences, and samples can improve the robustness of future SLAM methods applicable to XR context. While significant strides have been made in SLAM research, many unresolved challenges still exist. By leveraging the insights and recommendations provided in this work, we envision future tracking that is adaptable and generalizable across domains in real-world scenarios.

REFERENCES

- [1] Islam Ali and Hong Zhang. 2022. Are we ready for robust and resilient slam? a framework for quantitative characterization of slam datasets. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- [2] Mounir Amraoui, Rachid Latif, Abdelhafid Elouardi, and Abdelouahed Tajer. 2019. Features Extractors Evaluation Based V-SLAM Applications. In *4th World Conference on Complex Systems (WCCS)*.
- [3] Su-Yong An, Jeong-Gwan Kang, Lae-Kyoung Lee, and Se-Young Oh. 2010. SLAM with salient line feature extraction in indoor environments. In *11th International Conference on Control Automation Robotics & Vision*.

- [4] Alexander Andreopoulos and John K Tsotsos. 2011. On sensor bias in experimental methods for comparing interest-point, saliency, and recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 1 (2011).
- [5] Fatima M. Anwar, Luis Garcia, Xi Han, and Mani Srivastava. 2019. Securing Time in Untrusted Operating Systems with TimeSeal. In *IEEE Real-Time Systems Symposium (RTSS)*.
- [6] Mohammad OA Aqel, Mohammad H Marhaban, M Iqbal Saripan, and Napsiah Bt Ismail. 2016. Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus* 5 (2016), 1–26.
- [7] Saba Arshad and Gon-Woo Kim. 2021. Role of deep learning in loop closure detection for visual and lidar slam: A survey. *Sensors* 21, 4 (2021), 1243.
- [8] Josep Aulinas, Yvan Petillot, Joaquim Salvi, and Xavier Lladó. 2008. The SLAM problem: a survey. *Artificial Intelligence Research and Development* (2008), 363–371.
- [9] Rana Azzam, Tarek Taha, Shoudong Huang, and Yahya Zweiri. 2020. Feature-based visual simultaneous localization and mapping: A survey. *SN Applied Sciences* 2 (2020), 1–24.
- [10] Tim Bailey and Hugh Durrant-Whyte. 2006. Simultaneous localization and mapping (SLAM): Part II. *IEEE robotics & automation magazine* 13, 3 (2006), 108–117.
- [11] Ayush Bhargava, Jeffrey W Bertrand, Anand K Gramopadhye, Kapil C Madathil, and Sabarish V Babu. 2018. Evaluating multiple levels of an interaction fidelity continuum on performance and learning in near-field training simulations. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (2018).
- [12] Mark Billinghurst, Adrian Clark, Gun Lee, et al. 2015. A survey of augmented reality. *Foundations and Trends® in Human-Computer Interaction* 8, 2-3 (2015), 73–272.
- [13] Frank Biocca, Jin Kim, and Yung Choi. 2001. Visual Touch in Virtual Environments: An Exploratory Study of Presence, Multimodal Interfaces, and Cross-Modal Sensory Illusions. *Presence: Teleoperators and Virtual Environments* 10, 3 (06 2001), 247–265.
- [14] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. 2018. CodeSLAM—learning a compact, optimisable representation for dense visual SLAM. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2560–2568.
- [15] Jaime Boal, Alvaro Sánchez-Miralles, and Alvaro Arranz. 2014. Topological simultaneous localization and mapping: a survey. *Robotica* 32, 5 (2014).
- [16] Albrecht Böttcher and David Wenzel. 2008. The Frobenius norm and the commutator. *Linear algebra and its applications* 429, 8-9 (2008), 1864–1885.
- [17] Guillaume Bresson, Zayed Alsayed, Li Yu, and Sébastien Glaser. 2017. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles* 2, 3 (2017).
- [18] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. 2016. The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research* 35, 10 (2016), 1157–1163.
- [19] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. 2016. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. 32, 6 (2016).
- [20] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. 2021. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM. *IEEE Transactions on Robotics* 37, 6 (2021).
- [21] Yasra Chandio and Fatima M. Anwar. 2020. Poster: Spatiotemporal Security in Mixed Reality systems. In *18th ACM Conference on Embedded Networked Sensor Systems (SenSys)*.
- [22] Yasra Chandio, Noman Bashir, and Fatima M Anwar. 2022. HoloSet-A Dataset for Visual-Inertial Pose Estimation in Extended Reality: Dataset. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. 1014–1019.
- [23] Yasra Chandio, Noman Bashir, and Fatima M. Anwar. 2024. Stealthy and Practical Multi-modal Attacks on Mixed Reality Tracking. In *IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR)*.
- [24] Yasra Chandio, Victoria Interrante, and Fatima M. Anwar. 2024. Balancing Presence And Safety Using Reaction Time In Mixed Reality. In *Workshop at IEEE International Symposium on Mixed and Augmented Reality (SafeAR@ISMAR)*.
- [25] Yasra Chandio, Victoria Interrante, and Fatima M. Anwar. 2024. Human Factors at Play: Understanding the Impact of Conditioning on Presence and Reaction Time in Mixed Reality. *IEEE Transactions on Visualization and Computer Graphics* 30, 5 (2024), 2400–2410.
- [26] Yasra Chandio, Momin Ahmed Khan, Khotso Selialia, Luis Antonio Garcia, Joseph DeGol, and Fatima M. Anwar. 2024. A Neurosymbolic Approach To Adaptive Feature Extraction In SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- [27] Changhao Chen, Chris Xiaoxuan Lu, Bing Wang, Niki Trigoni, and Andrew Markham. 2021. DynaNet: Neural Kalman dynamical model for motion estimation and prediction. *IEEE Transactions on Neural Networks and Learning Systems* 32, 12 (2021), 5479–5491.
- [28] Changhao Chen, Bing Wang, Chris Xiaoxuan Lu, Niki Trigoni, and Andrew Markham. 2020. A survey on deep learning for localization and mapping: Towards the age of spatial machine intelligence. *arXiv preprint arXiv:2006.12567* (2020).
- [29] Weifeng Chen, Guangtao Shang, Aihong Ji, Chengjun Zhou, Xiyang Wang, Chonghui Xu, Zhenxiong Li, and Kai Hu. 2022. An Overview on Visual SLAM: From Tradition to Semantic. *Remote Sensing* 14, 13 (2022).
- [30] TJ Chong, XJ Tang, CH Leng, Mohan Yogeswaran, OE Ng, and YZ Chong. 2015. Sensor technologies and simultaneous localization and mapping (SLAM). *Procedia Computer Science* 76 (2015), 174–179.
- [31] Catherine Diaz, Michael Walker, Danielle Albers Szafrir, and Daniel Szafrir. 2017. Designing for depth perceptions in augmented reality. In *IEEE international symposium on mixed and augmented reality (ISMAR)*.
- [32] Louis Dressel and Mykel J. Kochenderfer. 2019. Hunting Drones with Other Drones: Tracking a Moving Radio Target. *IEEE International Conference on Robotics and Automation (ICRA)*.

- [33] Hugh Durrant-Whyte and Tim Bailey. 2006. Simultaneous localization and mapping: part I. *IEEE robotics & automation magazine* 13, 2 (2006), 99–110.
- [34] Austin Erickson, Kangsoo Kim, Gerd Bruder, and Gregory F. Welch. 2020. Exploring the Limitations of Environment Lighting on Optical See-Through Head-Mounted Displays. In *Symposium on Spatial User Interaction*.
- [35] Chao Fan, Weike Ding, Kun Qian, Hao Tan, and Zihan Li. 2024. Cueing Flight Object Trajectory and Safety Prediction Based on SLAM Technology. *Journal of Theory and Practice of Engineering Science* 4, 05 (2024), 1–8.
- [36] Chenglong Fu, Qiang Zeng, and Xiaojiang Du. 2021. Hawatcher: Semantics-aware anomaly detection for appified smart homes. In *USENIX Security*.
- [37] Pascal Fua and Vincent Lepetit. 2007. Vision based 3D tracking and pose estimation for mixed reality. In *Emerging technologies of augmented reality: Interfaces and design*. IGI Global, 1–22.
- [38] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. 2015. Visual simultaneous localization and mapping: a survey. *Artificial intelligence review* 43 (2015), 55–81.
- [39] Bharath Garigipati, Nataliya Strokina, and Reza Ghabcheloo. 2022. Evaluation and comparison of eight popular Lidar and Visual SLAM algorithms. In *2022 25th International Conference on Information Fusion (FUSION)*. IEEE, 1–8.
- [40] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research* 32, 11 (2013), 1231–1237.
- [41] Mar Gonzalez-Franco, Rodrigo Pizarro, Julio Cermeron, Katie Li, Jacob Thorn, Windo Hutabarat, Ashutosh Tiwari, and Pablo Bermell-Garcia. 2017. Immersive mixed reality for manufacturing training. *Frontiers in Robotics and AI* 4 (2017), 3.
- [42] C Shawn Green and Daphne Bavelier. 2003. Action video game modifies visual selective attention. *Nature* 423, 6939 (2003), 534–537.
- [43] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. 2010. A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine* 2, 4 (2010), 31–43.
- [44] Shiyi Guo, Zheng Rong, Shuo Wang, and Yihong Wu. 2022. A LiDAR SLAM with PCA-based feature extraction and two-stage matching. *IEEE Transactions on Instrumentation and Measurement* 71 (2022).
- [45] Abhishek Gupta and Xavier Fernando. 2022. Simultaneous Localization and Mapping (SLAM) and Data Fusion in Unmanned Aerial Vehicles: Recent Advances and Challenges. *Drones* 6, 4 (2022).
- [46] Georg Halmetschlager-Funek, Markus Suchi, Martin Kampel, and Markus Vincze. 2018. An empirical evaluation of ten depth cameras: Bias, precision, lateral noise, different lighting conditions and materials, and multiple sensor setups in indoor environments. *IEEE Robotics & Automation Magazine* 26, 1 (2018), 67–77.
- [47] Tianyi Hu, Fan Yang, and Tim Scargill Maria Gorlatova. 2024. Apple vs. Meta: A Comparative Study on Spatial Tracking in SOTA XR Headsets. In *2nd ACM Workshop on Mobile Immersive Computing, Networking, and Systems (ImmerCom)*.
- [48] Shoudong Huang and Gamini Dissanayake. 2016. A critique of current developments in simultaneous localization and mapping. *International Journal of Advanced Robotic Systems* 13, 5 (2016), 1729881416669482.
- [49] Patrick Hübner, Kate Clintworth, Qingyi Liu, Martin Weinmann, and Sven Wursthorn. 2020. Evaluation of HoloLens tracking and depth sensing for indoor mapping applications. *Sensors* 20, 4 (2020), 1021.
- [50] Microsoft Inc. 2020. HoloLens 2. <https://www.microsoft.com/en-us/hololens/hardware>. (2020). [Online; accessed September 2024].
- [51] Eunju Jeong, Jaun Lee, and Pyojin Kim. 2021. A Comparison of Deep Learning-Based Monocular Visual Odometry Algorithms. In *Asia-Pacific International Symposium on Aerospace Technology*. Springer, 923–934.
- [52] Georg Klein and David Murray. 2007. Parallel tracking and mapping for small AR workspaces. In *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*.
- [53] Jatavallabhula Krishna Murthy, Soroush Saryazdi, Ganesh Iyer, and Liam Paull. 2020. gradSLAM: Dense SLAM meets automatic differentiation. In *arXiv preprint arXiv:1910.10672*.
- [54] Yasir Latif, Cesar Cadena, and José Neira. 2013. Robust loop closing over time. In *Proc. Robotics: Science Systems*. 233–240.
- [55] K. Lebeck, K. Ruth, T. Kohno, and F. Roesner. 2017. Securing Augmented Reality Output. *IEEE S&P*.
- [56] Ao Li, Han Liu, Jinwen Wang, and Ning Zhang. 2022. From timing variations to performance degradation: Understanding and mitigating the impact of software execution timing in slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- [57] Mingyang Li and Anastasios Mourikis. 2013. High-precision, Consistent EKF-based Visual-Inertial Odometry. *Journal of Robotics Research* (2013).
- [58] Ruihao Li, Sen Wang, and Dongbing Gu. 2018. Ongoing evolution of visual SLAM from geometry to deep learning: Challenges and opportunities. *Cognitive Computation* 10 (2018), 875–889.
- [59] Shaopeng Li, Daqiao Zhang, Yong Xian, Bangjie Li, Tao Zhang, and Chengliang Zhong. 2022. Overview of deep learning application on visual SLAM. *Displays* (2022).
- [60] Yanyan Li, Raza Yunus, Nikolas Brasch, Nassir Navab, and Federico Tombari. 2021. RGB-D SLAM with structural regularities. In *IEEE International Conference on Robotics and Automation (ICRA)*.
- [61] Yuanzhi Liu, Yujia Fu, Fengdong Chen, Bart Goossens, Wei Tao, and Hui Zhao. 2021. Simultaneous localization and mapping related datasets: A comprehensive survey. *arXiv preprint arXiv:2102.04036* (2021).
- [62] Benjamin C Lok. 2004. Toward the merging of real and virtual spaces. *Commun. ACM* 47, 8 (2004).

- [63] Andréa Macario Barros, Maugan Michel, Yoann Moline, Gwenolé Corre, and Frédérick Carrel. 2022. A comprehensive survey of visual slam algorithms. *Robotics* 11, 1 (2022), 24.
- [64] Stefan Milz, Georg Arbeiter, Christian Witt, Bassam Abdallah, and Senthil Yogamani. 2018. Visual slam for automated driving: Exploring the applications of deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 247–257.
- [65] Saad Mokssit, Daniel Bonilla Licea, Bassma Guermah, and Mounir Ghogho. 2023. Deep Learning Techniques for Visual SLAM: A Survey. *IEEE Access* 11 (2023), 20026–20050.
- [66] Kirill Muravyev and Konstantin Yakovlev. 2022. Evaluation of RGB-D SLAM in Large Indoor Environments. In *International Conference on Interactive Collaborative Robotics*. Springer, 93–104.
- [67] Luigi Nardi, Bruno Bodin, M Zeeshan Zia, John Mawer, Andy Nisbet, Paul HJ Kelly, Andrew J Davison, Mikel Luján, Michael FP O’Boyle, Graham Riley, et al. 2015. Introducing SLAMBench, a performance and accuracy benchmarking methodology for SLAM. In *IEEE international conference on robotics and automation (ICRA)*.
- [68] Steven Ngo, Dave DeAngelis, and Luis Garcia. 2022. Modeling Human-Cyber Interactions in Safety-Critical Cyber-Physical/Industrial Control Systems. In *IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*.
- [69] Linus Nwankwo and Elmar Rueckert. 2023. Understanding Why SLAM Algorithms Fail in Modern Indoor Environments. In *Advances in Service and Industrial Robotics*, Tadej Petrič, Aleš Ude, and Leon Žlajpah (Eds.). Springer Nature Switzerland, Cham, 186–194.
- [70] Jonathan Petit, Bas Stottelaar, Michael Feiri, and Frank Kargl. 2015. Remote Attacks on Automated Vehicles Sensors: Experiments on Camera and Lidar. *Black Hat Europe* 11 (2015).
- [71] David Prokhorov, Dmitry Zhukov, Olga Barinova, Konushin Anton, and Anna Vorontsova. 2019. Measuring robustness of Visual SLAM. In *2019 16th International Conference on Machine Vision Applications (MVA)*. IEEE, 1–6.
- [72] Mark Pupilli and Andrew Calway. 2006. Real-time visual slam with resilience to erratic motion. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, Vol. 1. IEEE, 1244–1249.
- [73] Tong Qin, Peiliang Li, and Shaojie Shen. 2018. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics* 34, 4 (2018).
- [74] Tong Qin and Shaojie Shen. 2018. Online Temporal Calibration for Monocular Visual-Inertial Systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- [75] John G Rogers, Jason M Gregory, Jonathan Fink, and Ethan Stump. 2020. Test your slam! the sub-tunnel dataset and metric for mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*.
- [76] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. 2011. ORB: An efficient alternative to SIFT or SURF. In *International Conference on Computer Vision*. 2564–2571.
- [77] Sajad Saeedi, Eduardo DC Carvalho, Wenbin Li, Dimos Tzoumanikas, Stefan Leutenegger, Paul HJ Kelly, and Andrew J Davison. 2019. Characterizing visual localization and mapping datasets. In *International Conference on Robotics and Automation (ICRA)*.
- [78] Sajad Saeedi, Michael Trentini, Mae Seto, and Howard Li. 2016. Multiple-robot simultaneous localization and mapping: A review. *Journal of Field Robotics* 33, 1 (2016), 3–46.
- [79] Marta Salas, Yasir Latif, Ian D Reid, and J Montiel. 2015. Trajectory alignment and evaluation in SLAM: Horns method vs alignment on the manifold. In *Robotics: Science and Systems Workshop: The problem of mobile sensors*. sn, 1–3.
- [80] Muhamad Risqi U Saputra, Andrew Markham, and Niki Trigoni. 2018. Visual SLAM and structure from motion in dynamic environments: A survey. *ACM Computing Surveys (CSUR)* 51, 2 (2018), 1–36.
- [81] Thomas Schops, Torsten Sattler, and Marc Pollefeys. 2019. Bad slam: Bundle adjusted direct rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 134–144.
- [82] Valentin Schwind, Pascal Knierim, Nico Haas, and Niels Henze. 2019. Using Presence Questionnaires in Virtual Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*.
- [83] Raluca Scona, Mariano Jaimez, Yvan R Petillot, Maurice Fallon, and Daniel Cremers. 2018. Staticfusion: Background reconstruction for dense rgb-d slam in dynamic environments. In *IEEE international conference on robotics and automation (ICRA)*.
- [84] Sofiya Semenova, Steven Y Ko, Yu David Liu, Lukasz Ziarek, and Karthik Dantu. 2022. A quantitative analysis of system bottlenecks in visual SLAM. In *Proceedings of the 23rd Annual International Workshop on Mobile Computing Systems and Applications*. 74–80.
- [85] Myriam Servières, Valérie Renaudin, Alexis Dupuis, and Nicolas Antigny. 2021. Visual and visual-inertial slam: State of the art, classification, and experimental benchmarking. *Journal of Sensors* 2021 (2021), 1–26.
- [86] Richard Skarbez, Frederick P Brooks, and Mary C Whitton. 2020. Immersion and coherence: Research agenda and early results. *IEEE Transactions on Visualization and Computer Graphics* 27, 10 (2020).
- [87] Mel Slater, Vasilis Linakis, Martin Usoh, and Rob Kooper. 1996. Immersion, Presence and Performance in Virtual Environments: An Experiment with Tri-Dimensional Chess. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*.
- [88] Ivo Služanović, Mihael Liskij, Ante Derek, and Ivan Martinović. 2020. Tap-pair: Using spatial secrets for single-tap device pairing of augmented reality headsets. In *ACM CDASP*.
- [89] Yunmok Son, Hocheol Shin, Dongkwan Kim, Youngseok Park, Juhwan Noh, Kibum Choi, Jungwoo Choi, and Yongdae Kim. 2015. Rocking drones with intentional sound noise on gyroscopic sensors. In *USENIX Security*.

- [90] Maximilian Speicher, Brian D. Hall, and Michael Nebeling. 2019. What is Mixed Reality?. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*.
- [91] Muhammad Sualah and Gon-Woo Kim. 2019. Simultaneous localization and mapping in the epoch of semantics: a survey. *International Journal of Control, Automation and Systems* 17 (2019), 729–742.
- [92] Hamid Taheri and Zhao Chun Xia. 2021. SLAM; definition and evolution. *Engineering Applications of Artificial Intelligence* 97 (2021), 104032.
- [93] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. 2017. Visual SLAM algorithms: A survey from 2010 to 2016. *IPSP Transactions on Computer Vision and Applications* 9, 1 (2017), 1–11.
- [94] Jiexiong Tang, Rares Ambrus, Vitor Guizilini, Sudeep Pillai, Hanme Kim, Patric Jensfelt, and Adrien Gaidon. 2020. Self-Supervised 3D Keypoint Learning for Ego-Motion Estimation. In *Conference on Robot Learning (CoRL)*.
- [95] Zachary Teed and Jia Deng. 2021. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in Neural Information Processing Systems* 34 (2021), 16558–16569.
- [96] Yang Tian, Victor Gomez, and Shugen Ma. 2015. Influence of two SLAM algorithms using serpentine locomotion in a featureless environment. In *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 182–187.
- [97] Timothy Trippel, Ofir Weisse, Wenyuan Xu, Peter Honeyman, and Kevin Fu. 2017. WALNUT: Waging Doubt on the Integrity of MEMS Accelerometers with Acoustic Injection Attacks. In *IEEE European Symposium on Security and Privacy (EuroS&P)*.
- [98] Mariapina Trunfio, Timothy Jung, and Salvatore Campana. 2022. Mixed reality experiences in museums: Exploring the impact of functional elements of the devices on visitors’ immersive experiences and post-experience behaviours. In *Information & Management*, Vol. 59. Elsevier.
- [99] Konstantinos A Tsintotas, Loukas Bampis, and Antonios Gasteratos. 2022. The revisiting problem in simultaneous localization and mapping: A survey on visual loop closure detection. *IEEE Transactions on Intelligent Transportation Systems* 23, 11 (2022).
- [100] Dorin Ungureanu, Federica Bogo, S. Galliani, Pooja Sama, Xin Duan, Casey Meekhof, Jan Stühmer, Thomas J. Cashman, Bugra Tekin, Johannes L. Schönberger, Pawel Olszta, and Marc Pollefeys. 2020. HoloLens 2 Research Mode as a Tool for Computer Vision Research. *arXiv preprint arXiv:2008.11239* (2020).
- [101] J. Valente, K. Bahirat, K. Venechanos, A.A. Cardenas, and P. Balakrishnan. 2019. Improving the Security of Visual Challenges. *ACM Transactions on Cyber-Physical Systems (TCPS)* (2019).
- [102] Petr Vávra, Jan Roman, P. Zonča, Peter Ihnát, M. Nèmec, Kumar Jayant, Nagy Habib, and Ahmed El-Gendi. 2017. Recent Development of Augmented Reality in Surgery: A Review. *Journal of Healthcare Engineering* (2017).
- [103] Ke Wang, Sai Ma, Junlan Chen, Fan Ren, and Jianbo Lu. 2022. Approaches, Challenges, and Applications for Deep Visual Odometry: Toward Complicated and Emerging Areas. *IEEE Transactions on Cognitive and Developmental Systems* 14, 1 (2022).
- [104] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. 2017. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *IEEE international conference on robotics and automation (ICRA)*.
- [105] Wenshan Wang, Yaoyu Hu, and Sebastian Scherer. 2020. TartanVO: A Generalizable Learning-based VO. (2020).
- [106] Chris Warin and Delphine Reinhardt. 2022. Vision: Usable Privacy for XR in the Era of the Metaverse. In *European Symposium on Usable Security (EuroUSEC)*.
- [107] Carolin Wienrich, Philipp Komma, Stephanie Vogt, and Marc E. Latoschik. 2021. Spatial Presence in Mixed Realities—Considerations About the Concept, Measures, Design, and Experiments. *Frontiers in Virtual Reality* 2 (2021).
- [108] Li Yan, Xiao Hu, Leyang Zhao, Yu Chen, Pengcheng Wei, and Hong Xie. 2022. DGS-SLAM: A Fast and Robust RGBD SLAM in Dynamic Environments Combined by Geometric and Semantic Information. *Remote Sensing* 14, 3 (2022), 795.
- [109] Mathew Yarossi, Madhur Mangalam, Stephanie Naufel, and Eugene Tunik. 2021. Virtual Reality as a Context for Adaptation. *Frontiers in Virtual Reality* 2 (2021).
- [110] Georges Younes, Daniel Asmar, Elie Shammas, and John Zelek. 2017. Keyframe-based monocular SLAM: design, survey, and future directions. *Robotics and Autonomous Systems* 98 (2017), 67–88.
- [111] Qian-Qian Yu, Yi-Yang Wang, Ke-Qi Fan, and Yu-Hui Zheng. 2021. Domain Adaptive Visual Tracking with Multi-scale Feature Fusion. In *2021 IEEE International Conference on Progress in Informatics and Computing (PIC)*.
- [112] Wenhao Yu, Deepali Jain, Alejandro Escontrela, Atil Iscen, Peng Xu, Erwin Coumans, Sehoon Ha, Jie Tan, and Tingnan Zhang. 2021. Visual-locomotion: Learning to walk on complex terrains with vision. In *5th Annual Conference on Robot Learning*.
- [113] H. Zhan, C. S. Weerasekera, J. W. Bian, and I. Reid. 2020. Visual Odometry Revisited: What Should Be Learnt?. In *IEEE International Conference on Robotics and Automation (ICRA)*.
- [114] Hui Zhang, Li Zhu Liu, He Xie, Yiming Jiang, Jian Zhou, and Yaonan Wang. 2022. Deep Learning-Based Robot Vision: High-End Tools for Smart Manufacturing. *IEEE IMM*.
- [115] Mingming Zhang, Xingxing Zuo, Yiming Chen, Yong Liu, and Mingyang Li. 2021. Pose estimation for ground robots: On manifold representation, integration, reparameterization, and optimization. *IEEE Transactions on Robotics* 37, 4 (2021).
- [116] Shishun Zhang, Longyu Zheng, and Wenbing Tao. 2021. Survey and Evaluation of RGB-D SLAM. *IEEE Access* 9 (2021).
- [117] Yipu Zhao, Justin S Smith, Sambhu H Karumanchi, and Patricio A Vela. 2020. Closed-loop benchmarking of stereo visual-inertial SLAM systems: Understanding the impact of drift and latency on tracking accuracy. In *IEEE International Conference on Robotics and Automation (ICRA)*.
- [118] H. Zhou, B. Ummenhofer, and T. Brox. 2018. DeepTAM: Deep Tracking and Mapping. In *European Conference on Computer Vision (ECCV)*.

- [119] Huizhong Zhou, Danping Zou, Ling Pei, Rendong Ying, Peilin Liu, and Wenxian Yu. 2015. StructSLAM: Visual SLAM with Building Structure Lines. *IEEE Transactions on Vehicular Technology* 64, 4 (2015).
- [120] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. 2017. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1851–1858.
- [121] Danping Zou, Yuanxin Wu, Ling Pei, Haibin Ling, and Wenxian Yu. 2019. StructVIO: visual-inertial odometry with structural regularity of man-made environments. *IEEE Transactions on Robotics* 35, 4 (2019).
- [122] Jon Zubizarreta, Iker Aguinaga, and Jose Maria Martinez Montiel. 2020. Direct Sparse Mapping. *IEEE Transactions on Robotics* (2020).