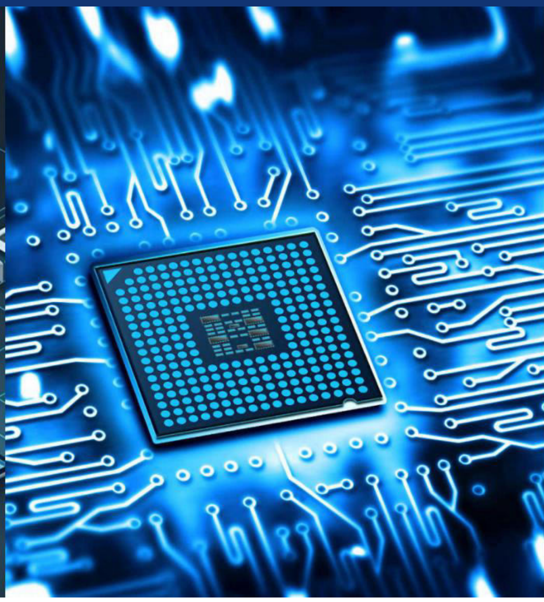
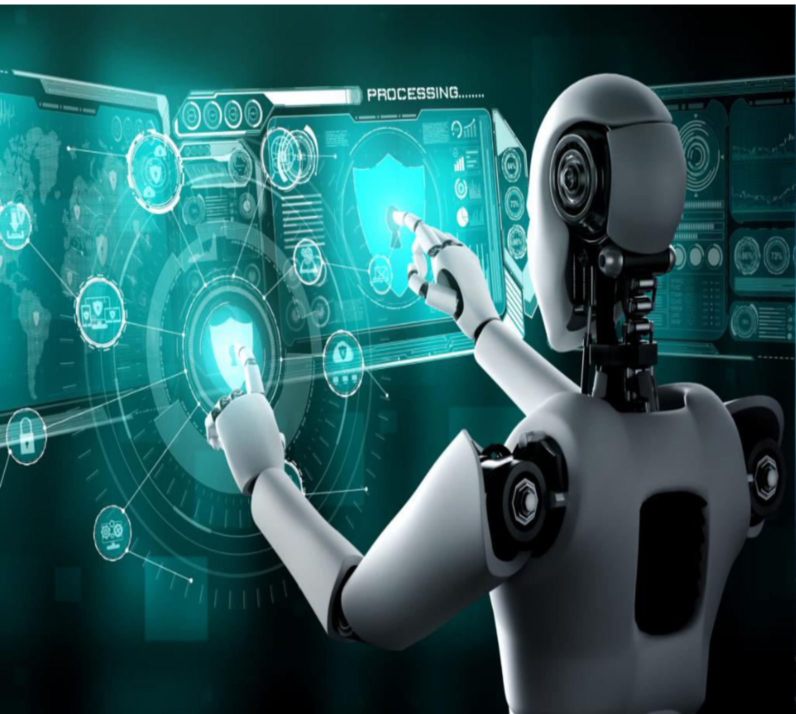
ISSN(O): 2320-9801
ISSN(P): 2320-9798


International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.771

Volume 13, Issue 4, April 2025

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304003

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Improvement of Spatial Presence in VR/AR Learning Platforms Through Lightweight SLAM Algorithm

Xuexhi Wen, Shah Sambidhan Bikram

Masters Student, Dept. of Computer Science, Nanjing University of Information Science and Technology NUIST

School of Computer Science, School of Cyber Science and Engineering, NUIST, Nanjing 210044, China

ABSTRACT: Spatial presence is a critical component of modern immersive technologies, particularly in virtual reality (VR) and augmented reality (AR) devices such as Meta Oculus, Google Lens, Apple Vision Pro, and smartphones. At the core of these systems lies Simultaneous Localization and Mapping (SLAM), a foundational algorithm enabling real-time rendering of 3D environments and enhancing spatial immersion. This study focuses on lightweight SLAM techniques, which address the computational constraints of modern VR/AR devices by reducing processing overhead while maintaining mapping accuracy. The research evaluates the computational efficiency, load balancing, and usability of these algorithms in real-world scenarios. The proposed solution studies lightweight SLAM algorithms to optimize real-time mapping and localization, achieving low latency and high interactivity. Experimental deployment and user experience analysis validate its ability to enhance spatial presence, demonstrating its potential to improve the performance and usability of VR/AR learning platforms.

KEYWORDS: Spatial presence; Lightweight SLAM; 3D environment rendering; Computational efficiency; Real-time mapping

I. INTRODUCTION

The integration of Virtual Reality (VR) and Augmented Reality (AR) technologies into education is revolutionizing how learners interact with information, fostering an engaging and immersive learning environment that transcends traditional teaching methods. VR/AR platforms allow users to explore virtual spaces, conduct simulations, and practice real-world applications in safe, controlled environments. The effectiveness of these platforms depends heavily on spatial presence. Strong sense of spatial presence enhances engagement, focus, and overall learning outcomes which leads to high computation load for the hardware. Spatial presence is possible due to the implementation of SLAM algorithms in VR/AR the most common SLAM technique used are Visual SLAM that are in computer vision[1]. This research to focus on the lightweight SLAM and the study will be demonstrated regardless Visual SLAM are widely used in VR/AR devices concerning this topic is narrowed down to only research Visual SLAM lightweight. Simultaneous Localization and Mapping (SLAM) technology plays a pivotal role in creating immersive VR/AR experiences by enabling real-time tracking of the user's position and orientation relative to the environment. This functionality is critical for maintaining accurate spatial awareness, supporting dynamic interaction, and rendering virtual elements in alignment with the real world. However, conventional SLAM algorithms, often designed for high-performance systems, require significant computational power, leading to challenges such as latency, decreased performance, and limited scalability for affordable and portable educational devices[2].

This research, topic "Improvement of Spatial Presence in VR/AR Learning Platforms Through Lightweight SLAM Algorithm" seeks to address these challenges by exploring how lightweight SLAM algorithm can be implemented to improve spatial presence in VR/AR learning environments. Lightweight SLAM algorithms, optimized for resource-constrained devices, aim to reduce computational overhead while preserving the accuracy and responsiveness essential for an immersive experience in spatial presence[1]. To achieve this, the research will test which will leverage leading VR/AR development platforms: Unity (Tuanjie) experimenting using the latest cutting-edge device called Oculus Meta Quest 2. These platforms are widely used in educational and commercial VR/AR projects due to their powerful tools, cross-platform compatibility, notable implementations in this sector include augmented reality platforms which the frameworks such as ARKit, ARCore, Interaction toolkit etc. These platforms leverage visual-inertial fusion for enhanced tracking accuracy, while also maintaining spatial consistency and for the extensive developer ecosystems:

*Unity(Tuanjie)***:**

Unity offers flexible and user-friendly tools for building interactive VR/AR environments. Its AR Foundation and Interaction toolkit framework enables seamless integration of SLAM-based tracking for AR and VR applications, supporting multiple device platforms like IOS, Android and Oculus making unity the undisputed cross platform builder editor. Unity's real-time rendering capabilities and lightweight deployment options make it a preferred choice for mobile and educational VR/AR applications. By utilizing these platforms, the study will design and test virtual learning environments tailored to educational scenarios, such as virtual classrooms, laboratory simulations, and collaborative learning spaces. The prototypes will demonstrate how lightweight SLAM algorithms can improve spatial presence while maintaining efficient system performance. This research aims to provide scalable, cost-effective solutions for VR/AR education, making these technologies more accessible to institutions with limited resources. The findings will contribute to the advancement of immersive learning by balancing computational efficiency with user-centric design, setting a new standard for VR/AR applications in education[2].

1. *Core Idea:* The core idea of this research is to enhance spatial presence in VR/AR learning platforms by leveraging lightweight SLAM algorithms tailored for educational applications which focuses on optimization. Traditional Lightweight SLAM algorithms, such as ORB-SLAM or LSD-SLAM and DSO, are highly accurate but computationally intensive, making them unsuitable for low-power devices like standalone VR headsets or mobile AR systems commonly used in education. Lightweight SLAM algorithms, such as ORB-SLAM, Direct Sparse Odometry (DSO), or LDSO (Lightweight DSO), are designed to minimize resource consumption while maintaining essential tracking and mapping functionalities. By employing lightweight algorithms, the research seeks to reduce computational overhead, enabling smooth, real-time interaction in virtual environments. This optimization is particularly crucial for achieving high spatial presence without introducing latency or sacrificing tracking precision, ensuring a seamless and immersive experience for learners[3].
2. *Proposed Improvement:* This study proposes the adoption and implementation of lightweight SLAM algorithms optimized for VR/AR platforms in education. Lightweight SLAM focuses on reducing computational overhead while maintaining essential tracking and mapping functionalities. This optimization will address the challenges of achieving high spatial presence on resource-limited hardware, such as standalone VR headsets (e.g., Meta Quest) or mobile AR devices. The proposed improvement will enhance the accessibility of VR/AR learning tools without compromising immersion or user experience.
3. *Proposed Idea*: Integrate lightweight SLAM algorithms into VR/AR educational environments developed using industry-standard platforms such as Unity. Optimize performance by tailoring SLAM parameters and processes example: keyframe selection, feature extraction etc. for specific educational scenarios. Evaluate effectiveness through both technical benchmarks example: latency, frame rate, mapping accuracy and user experience metrics concerning spatial presence.

In the context of VR/AR learning platforms, where environments may range from static classrooms to dynamic outdoor simulations, the choice of SLAM algorithm is pivotal. Algorithms such as ORB-SLAM, DSO and LSD-SLAM offer unique advantages and trade-offs in terms of accuracy, efficiency, and robustness. Understanding these trade-offs through comparative analysis can highlight areas where SLAM can be optimized for educational applications, especially to enhance spatial presence[4]. This research aims to systematically analyse lightweight SLAM algorithms to identify their impact on spatial presence within VR/AR learning platforms. By leveraging metrics like tracking accuracy, latency, and user experience feedback, the study seeks to propose improvements tailored for educational use cases. The outcome is expected to provide actionable insights for creating more immersive and comfortable VR learning environments, pushing the boundaries of how virtual spaces can transform education[5].

II. RELATED WORK

Lightweight SLAM algorithms have emerged as a solution to address the computational limitations of VR/AR devices, ensuring real-time performance while preserving mapping accuracy. These advancements focus on optimizing key components such as feature extraction, sensor fusion, and map representation. Lightweight SLAM systems have demonstrated their potential in improving spatial presence in VR/AR platforms. The integration of lightweight algorithms into VR/AR systems ensures seamless user interaction by minimizing latency and enhancing environmental mapping accuracy.

1. *Spatial recognition and visualization:* In mobile AR systems using Visual SLAM. By leveraging camera-based tracking and mapping, the system identifies and reconstructs 3D environments in real time. This enables precise spatial awareness, allowing AR content to be accurately anchored to the physical world. The method focuses on optimizing computational efficiency for mobile devices, ensuring low-latency performance and energy efficiency. Key applications include AR navigation, interactive learning, and immersive gaming, where accurate spatial alignment is essential for a seamless user experience [1]. Spatial presence influences user behavior, task performance, and engagement across real, remote, and virtual environments. It highlights differences in user experience, sensory immersion, and cognitive responses in these contexts, offering insights into how immersive environments affect human interaction and task execution. This research is relevant for designing VR/AR systems to optimize presence and performance[6]. Processing Presence: How Users Develop Spatial Presence discusses the concept of spatial presence, defined as the user's sense of "being there" in a virtual or augmented environment. It explains the cognitive and perceptual processes involved in developing this sense of presence, highlighting factors such as sensory input, user attention, and the role of immersive technology. The study aims to understand how these factors influence user engagement and interaction within VR/AR systems, providing insights for the design of more immersive experiences[7]. Real-time Visual SLAM In large-scale environments by incorporating distant landmarks. Traditional SLAM systems often struggle with localization accuracy in expansive areas due to limited visual features and increased drift over long distances. By using distant, easily recognizable landmarks, the system enhances spatial awareness and reduces localization errors. These landmarks provide stable reference points, enabling better trajectory estimation and map consistency. This approach is particularly useful for AR/VR, autonomous vehicles, and robotics, where navigating large, unstructured environments is essential for performance and reliability[8].

2. *Simultaneous Localization and Mapping (SLAM)***:** Integrating data from multiple sensors (like cameras, LiDAR, IMU, and depth sensors) to achieve more accurate 3D reconstruction. The fusion of sensor inputs helps overcome the limitations of individual sensors, such as visual occlusion, noise, or poor lighting conditions. By combining visual, inertial, and depth data, the system achieves better spatial awareness, robustness, and real-time performance. This approach is crucial for AR/VR, robotics, and autonomous navigation, as it improves localization accuracy, reduces drift, and enables more detailed and reliable 3D environmental mapping[7]. Comprehensive analysis of various visual SLAM algorithms and their applications in augmented reality (AR). It categorizes SLAM methods based on their key features, such as direct vs. indirect methods, monocular vs. stereo systems, and real-time efficiency. The study highlights the role of visual SLAM in AR for spatial mapping, object tracking, and environment interaction. It also examines the trade-offs between computational complexity, accuracy, and hardware constraints, especially for mobile AR devices. The mapping serves as a guide for selecting suitable SLAM techniques for specific AR applications like navigation, gaming, and education[2]. CS-SLAM addresses the challenges of operating in dynamic environments where traditional SLAM methods often fail due to their assumption of static scenes. It incorporates semantic understanding to identify and handle dynamic elements, enhancing robustness and reducing computational complexity[6].Semantic SLAM Approaches Modern SLAM systems, like DRV-SLAM, use instance segmentation to classify moving objects, thereby improving localization and mapping in dynamic scenarios. Similarly, methods like PR-SLAM focus on semantic optimization to maintain accuracy even in dynamic scenes[7].PIPO-SLAM Is a lightweight visual-inertial SLAM system designed to improve computational efficiency and robustness in dynamic environments. It introduces two key concepts: pre-integration merging theory and pose-only descriptions of multiple-view geometry. These innovations address limitations in traditional visual-inertial SLAM (VI-SLAM), such as the high computational cost of keyframe-based optimization and the need for precise keyframe management[8].NICE-SLAM Exemplifies this approach by using a neural implicit representation to model 3D environments. Unlike traditional methods that discretize space into grids or voxels, NICE-SLAM encodes the environment as a continuous field, allowing for more efficient, scalable, and accurate 3D reconstructions. This method reduces the computational load while maintaining high accuracy, making it well-suited for AR applications[9].

3. *Visual SLAM (VSLAM):* These systems have been a topic of study for decades and a small number of openly available implementations have stood out: ORB-SLAM3, OpenVSLAM and RTABMap. This paper presents a

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304003

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

comparison of these 3 moderns, feature rich, and uniquely robust VSLAM techniques that have yet to be benchmarked against each other, using several different datasets spanning multiple domains negotiated by service robots. ORB-SLAM3 and OpenVSLAM each were not compared against at least one of these datasets previously in literature and we provide insight through this lens. This analysis is motivated to find general purpose, feature complete, and multi-domain VSLAM options to support a broad class of robot applications for integration into the new and improved ROS 2 Nav2 System as suitable alternatives to traditional 2D lidar solutions[9]. VSLAM method based on object detection in dynamic environments highlights the challenges posed by moving objects in traditional SLAM systems, which assume static scenes. To overcome this, a new VSLAM approach integrates object detection to identify and exclude dynamic elements from the tracking process. This method enhances the robustness and accuracy of localization and mapping in dynamic environments, enabling more stable AR/VR applications[10]. Real-Time Visual–Inertial SLAM Based on Adaptive Keyframe For Mobile AR Applications presents a method for improving the performance of visual-inertial SLAM (Simultaneous Localization and Mapping) in mobile AR environments. It addresses challenges such as computational efficiency and the real-time processing of large amounts of sensor data. The method focuses on adaptive keyframe selection, reducing the number of keyframes required while maintaining high accuracy, making it suitable for resource-constrained mobile devices. This approach enhances the robustness and efficiency of SLAM systems in dynamic and complex AR environments[11].

4. *Lightweight Visual SLAM:* Method designed to maintain high precision in dynamic scenes where objects or people are in motion. Traditional SLAM methods struggle in such environments due to visual noise and moving elements. This approach incorporates dynamic object detection and filtering techniques to distinguish between static and dynamic features, ensuring accurate localization and mapping. The lightweight design prioritizes computational efficiency, making it suitable for mobile devices and low-power AR/VR systems. The method enhances stability, reduces drift, and improves the overall robustness of SLAM in complex, real-world scenarios[6]. Lightweight Implementation for Mobile Devices Mobile AR platforms are constrained by hardware limitations, such as processing power and battery life. Lightweight SLAM algorithms example: LDSO, RTAB-Map are developed to address these challenges by minimizing computational overhead without compromising mapping precision. These adaptations are particularly suited for handheld devices and standalone AR glasses[3].

5. *Augmented Reality (AR) and Simultaneous Localization and Mapping (SLAM):* provides a novel approach for assessing SSTM. SLAM allows AR systems to map and understand the 3D world in real time, enabling the projection of virtual objects into the user's real environment. Unlike conventional screen-based tests, AR-based assessments are more immersive, realistic, and interactive, offering a closer approximation of real-world spatial memory challenges. This approach enhances engagement, ecological validity, and cognitive stimulation[12]. Accurate Motion Tracking in VR, SLAM provides accurate motion tracking, allowing users to move freely in a virtual space with a heightened sense of spatial presence. In AR, SLAM facilitates the precise placement of virtual objects in real-world scenes, enabling immersive interactions with digital content. These capabilities have revolutionized industries such as gaming, education, healthcare, architecture, and remote collaboration. SLAM enables devices to track their position in real-time while building a 3D map of the environment. This allows AR/VR systems to operate in dynamic, unstructured, and large-scale environments without requiring external infrastructure[13]. Overview of augmented reality (AR) as a transformative technology that blends digital content with the real world. It highlights AR's growing role in various sectors, including education, healthcare, and entertainment. The introduction emphasizes the need for intuitive user experiences (UX) and seamless interactions to maximize AR's potential. It also outlines key challenges such as hardware limitations, usability issues, and privacy concerns, while pointing to future advancements like AI integration and more sophisticated user interfaces[14].

6. *Learning Environment Systems:* To improve the online education experience. It introduces the concept of digital recommendation systems that adapt to individual learning needs by analyzing user behavior and preferences. The system aims to enhance student engagement and performance by offering tailored content, resources, and feedback. The introduction also outlines the potential of these systems to support a more customized educational journey, ultimately improving learning outcomes and accessibility[15]. ASLAM-FD Utilizes a multi-sensor fusion strategy to enhance SLAM robustness in variable conditions. By detecting sensor degradation in real time, the system dynamically adjusts sensor weights using a deep reinforcement learning framework. This adaptability

increases system reliability in environments with unpredictable changes. The framework supports collaborative SLAM, allowing multiple devices to share mapping information for enhanced spatial awareness[16]. Hybrid Environment Reconstruction Improving User Experience and Workload in Augmented Virtuality Teleoperation" discusses the need for better interaction between users and augmented virtual environments, especially in teleoperation scenarios. Traditional systems often struggle with workload and user experience due to delays, incomplete information, and complex control interfaces. This approach integrates real-world and virtual elements to create a hybrid environment, enhancing user experience and reducing cognitive workload[17].

7. *Virtual reality (VR) learning environments*: Have gained significant attention for their potential to enhance education by providing immersive, interactive, and engaging experiences. This study explores learners' attitudes toward VR-based learning through the lens of constructivism, a learning theory that emphasizes active knowledge construction through interaction with the environment. The research investigates how VR environments, which allow learners to explore, manipulate, and experiment in simulated settings, align with constructivist principles. By fostering hands-on, experiential learning, VR can promote deeper understanding and engagement. The study aims to understand whether learners perceive VR as a valuable tool for constructing knowledge and how factors like immersion, interactivity, and spatial presence influence their attitudes and learning outcomes. The findings highlight that learners generally exhibit positive attitudes toward VR learning environments, particularly due to their ability to provide realistic, immersive experiences that align with constructivist ideals. Learners appreciate the opportunity to engage in self-directed exploration and problem-solving, which enhances their motivation and retention. However, the study also identifies challenges, such as the need for adequate technical support and the potential for cognitive overload in highly immersive environments. Overall, the research underscores the importance of designing VR learning experiences that balance immersion with pedagogical goals, ensuring that they effectively support constructivist learning while addressing learners' needs and preferences[18].

III. METHODOLOGY

The proposed method aims to enhance spatial presence in VR/AR learning platforms through the development of an optimized lightweight SLAM (Simultaneous Localization and Mapping) algorithm. Spatial presence, defined as the user's sense of "being there" in a virtual environment, is a critical factor in the effectiveness of immersive educational experiences. Traditional SLAM methods are computationally expensive, leading to higher latency, lower frame rates, and reduced immersion, especially on standalone VR/AR devices with limited processing power. To address these challenges, the proposed method introduces a lightweight SLAM system tailored for VR/AR learning platforms. By focusing on computational efficiency, real-time performance, and robust environmental tracking, this approach aims to achieve smoother interactions, faster scene updates, and a greater sense of spatial presence. The method incorporates key innovations such as hybrid feature descriptors, multi-sensor fusion, and sliding-window optimization, all of which are designed to reduce computational overhead while maintaining tracking accuracy. The resulting system not only enhances immersion but also improves user comfort and reduces motion sickness often caused by tracking delays. The proposed method is designed with adaptability in mind, ensuring compatibility with a range of VR/AR devices, from high-end headsets to standalone units like the Meta Quest or HTC Vive Focus. This flexibility makes it suitable for diverse educational applications, such as virtual classrooms, interactive simulations, and collaborative learning environments. The following sections provide an overview of the system's design and core components[1], [19].

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Classification of Visual SLAM

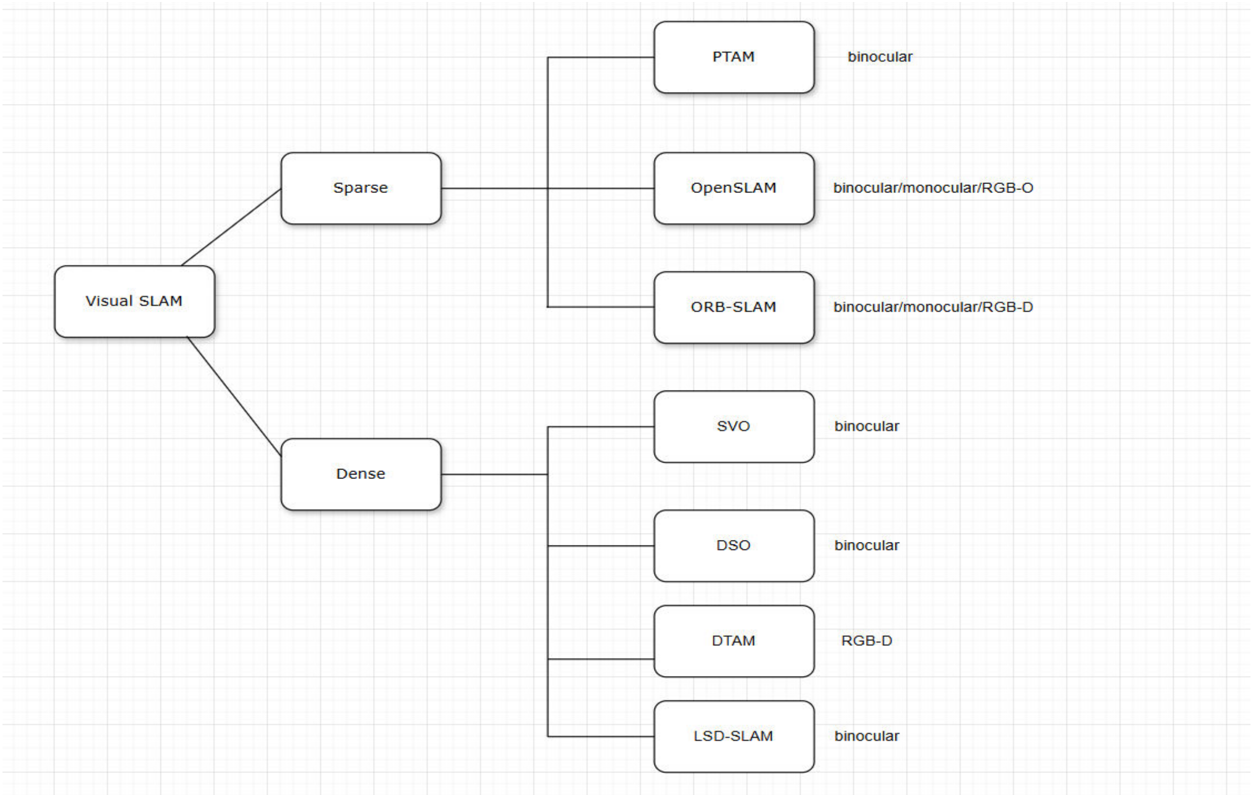


Figure 1: Classification of the Visual SLAM Technologies.

Visual SLAM is a technique used to estimate the position and orientation of a camera while simultaneously building a map of the environment using visual data, such as images or video frames. It is a key technology in robotics, augmented reality (AR), virtual reality (VR), and autonomous vehicles. Lightweight SLAM algorithms aim to reduce computational complexity while maintaining high accuracy and robustness, making them suitable for real-time applications in devices with limited processing power, such as mobile AR/VR systems. Here, we describe two widely used approaches—ORB-SLAM and relevance to lightweight SLAM implementations[20].ORB-SLAM is a monocular, stereo, or RGB-D visual SLAM algorithm designed to provide real-time tracking, mapping, and loop closure using ORB (Oriented FAST and Rotated BRIEF) feature descriptors. It is well-suited for lightweight applications due to its efficient feature extraction and matching processes[1][21].

Custom lightweight SLAM implementation using OpenCV and integrating it into Unity to enhance spatial presence in VR/AR learning platforms. The process begins with the development of a lightweight SLAM system using OpenCV, which serves as the backbone for real-time camera tracking and environment mapping. OpenCV is utilized for tasks such as camera calibration, feature detection, and sparse image alignment, which are essential for estimating the camera's position and orientation in real-time. The SLAM algorithm is designed to be computationally efficient, ensuring it can run on resource-constrained devices like mobile phones or AR glasses, making it suitable for educational applications. Once the custom SLAM system is implemented, the next step is to integrate it with Unity, the platform for building the AR/VR learning environment. This integration is achieved by creating a communication bridge between the C++-based SLAM algorithm and Unity's C# environment. One approach is to use P/Invoke, a mechanism in C# that allows calling C++ functions directly. A C++ wrapper is created to expose the SLAM algorithm's output, such as the camera pose and sparse map points, to Unity. Alternatively, ROS# can be used if the SLAM algorithm is wrapped as a ROS node, enabling communication between the SLAM backend and Unity through ROS topics. This ensures that the camera pose and map data are transmitted to Unity in real-time. Spatial presence is assessed using standardized questionnaires like the ITC-SOPI, which gauges users' sense of "being there" in the virtual environment. Learning outcomes are evaluated through task performance metrics, such as knowledge retention and engagement levels. The system's performance is also analysed in terms of tracking accuracy, latency, and computational efficiency, ensuring it meets the requirements for real-time educational applications. By combining a custom lightweight SLAM implementation with Unity's robust AR/VR capabilities, this approach aims to create a highly immersive and interactive learning platform that enhances spatial presence and educational engagement[22].

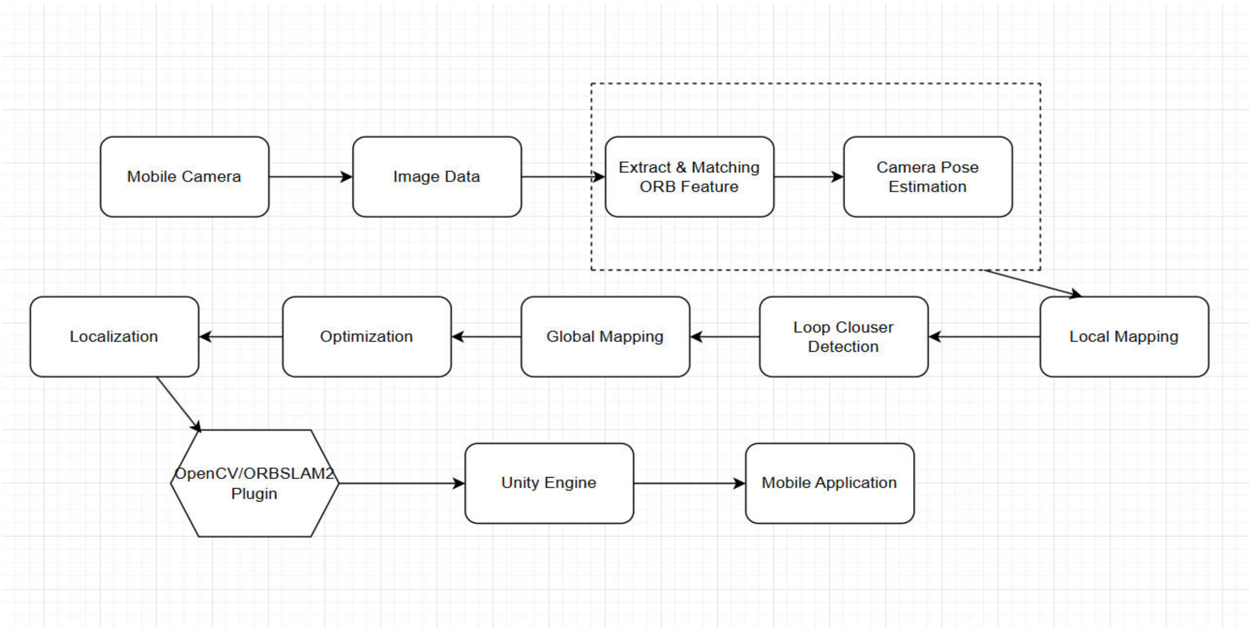


Figure 2: The overall work procedure in the mobile SLAM system is shown System workflow.

3.2. *Mobile Application with Unity*

The existing ORB-SLAM2 uses Pangolin to visualize SLAM's key frames, camera poses, and point clouds, showing SLAM running with a pre-prepared data set. Pangolin is a lightweight and fast 3D visualization library for managing OpenGL displays and abstracting image input, which is widely used in computer vision as a means to eliminate the platform-specific boilerplate and easily visualize data [14]. In this paper, to design a SLAM system targeting the Android platform, the Unity Engine is used instead of Pangolin, and the API of the Unity Engine is used to control the camera and render the AR contents. Unity Engine is an authoring tool that provides a development environment for 3D and 2D video games and an integrated development tool for the creation of the interactive contents such as 3D animation, architectural visualization, virtual reality, and AR. It supports various platforms such as Windows, MacOS, iOS and Android. The Android application acting as an AR viewer in this system, can be built and used targeting Android by installing the Android build module, Android SDK and NDK tool supported by the Unity Engine[1]. SLAM system employed to implement a new approach. The primary objective of the approach various datasets that the proposed methods were tested on. The impact of utilizing semantic data in the proposed methods. The amount of dynamic objects existing in the environment various types of environments the system tested on.

ORB-SLAM2 is a feature-based simultaneous localization and mapping (SLAM) algorithm that efficiently estimates camera motion and reconstructs a sparse 3D map of the environment. It is widely used due to its robustness, real-time performance, and ability to handle monocular, stereo, and RGB-D cameras. The algorithm consists of three main components: tracking, local mapping, and loop closure. Tracking involves extracting ORB features from incoming frames and estimating camera pose using the Perspective-n-Point (PnP) algorithm. Local mapping maintains a map of 3D landmarks by adding keyframes and refining positions using Bundle Adjustment. Loop closure detects revisited locations to correct drift and optimize the overall map structure. However, the original ORB-SLAM2 can be computationally intensive, making it less suitable for real-time applications in resource-constrained environments like AR/VR learning platforms. To address these limitations, a lightweight variant of ORB-SLAM2 is developed, focusing on computational efficiency without compromising accuracy. The optimization begins with adaptive ORB feature

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

extraction, where the number of key points is dynamically adjusted based on scene complexity. This reduces unnecessary computations in low-texture areas while preserving accuracy in detailed regions. Pose estimation is also improved by introducing a preconditioned PnP algorithm that prioritizes high-confidence feature matches, reducing outliers and improving robustness. Additionally, a GPU-accelerated implementation of Bundle Adjustment is integrated to minimize processing time, leveraging single-precision floating-point operations to maintain an optimal balance between speed and precision[21].

ORB-SLAM2 utilizes the ORB (Oriented FAST and Rotated BRIEF) feature detector and descriptor to extract key points and their corresponding descriptors from images. The process begins with FAST (Features from Accelerated Segment Test) corner detection, which identifies key points by comparing the intensity of a central pixel pp to those of its surrounding pixels in a circular pattern. A pixel pp is considered a key point if the sum of the absolute differences between its intensity I(p)$I(p)$ and the intensities I(pi)$I(pi)$ of the surrounding pixels exceeds a predefined threshold TT. This ensures that only points with significant intensity variations, indicative of corners or edges, are selected as key points. These key points are then used to compute ORB descriptors, which are binary vectors that encode the local image pattern around the key points, enabling efficient feature matching and tracking in the SLAM (Simultaneous Localization and Mapping) pipeline[23].

Regular ORB-SLAM2 Algorithm Formula

ORB-SLAM2 follows a keyframe-based approach that involves feature extraction, pose estimation, mapping, and loop closure. Below are the fundamental mathematical formulations used in the regular ORB-SLAM2 algorithm:

1. Feature Extraction using ORB (Oriented FAST and Rotated BRIEF)

- ORB-SLAM2 extracts key points and descriptors using ORB, which consists of:
  - FAST corner detection: A point p p is considered a key point if:
    - $\sum_{i=1}^{N} |I(P) - \text{I}\ (\rho_i)| > T$ (1)

where $I(p)$ I(p) is the intensity at p p, $I(pi)$I(p I) are intensities of the surrounding pixels, and T T is a threshold.

- Orientation Assignment using Image Moments:
  - $\theta = tan^{-1}\left(\frac{m_{01}}{m_{10}}\right)$ (2)
  - where: $M\rho q = \sum x^p y^q\ \text{I}\ (x, y)$ are the image moments.
- Descriptor Matching using Hamming Distance: Given descriptors Di and Dj, their similarity is computed as:
  - $d(Di, Dj) = \sum_k |D_i^k \oplus D_j^k|$ (3)

This reduces unnecessary computations in low-texture areas while preserving accuracy in detailed regions. Pose estimation is also improved by introducing a preconditioned PnP algorithm that prioritizes high-confidence feature matches, reducing outliers and improving robustness. Additionally, a GPU-accelerated implementation of Bundle Adjustment is integrated to minimize processing time, leveraging single-precision floating-point operations to maintain an optimal balance between speed and precision[24].

In ORB-SLAM2, camera pose estimation is achieved using the Perspective-n-Point (PnP) algorithm, which computes the camera's position and orientation relative to a set of known 3D points. The PnP problem is formulated as solving for the rotation matrix RR and translation vector t that best align the 3D points Xi with their corresponding 2D projections xi in the image plane. The relationship is expressed as xi=K[R∣t]Xi=$K[R|t]Xi$, where K is the camera intrinsic matrix containing parameters like focal length and principal point. This equation projects the 3D points Xi onto the 2D image plane using the camera's pose (R and t) and intrinsic parameters. Preconditioning is often applied to improve the numerical stability and accuracy of the PnP solution, ensuring robust pose estimation even in the presence of noise or outliers. This step is critical for accurately localizing the camera in the environment and maintaining the consistency of the SLAM map.

2. Pose Estimation with Preconditioned PnP (Perspective-n-Point)

To estimate the camera pose, we solve:

- $x_i = k|R|t|x_i$ (4)
  - where: xi is the 2D projection of the 3D point Xi

- Xi K is the camera intrinsic matrix
- R and t t are the rotation and translation matrices.

The formula in ORB-SLAM2 for camera pose estimation uses an Iterative Weighted Least Squares (IWLS) approach to minimize the reprojection error, which measures the difference between observed 2D key points xi and the projected 3D points π(R,t,Xi)$\pi(R,t,Xi)$, where π is the projection function that maps 3D points Xi to the 2D image plane using the camera's rotation RR, translation t, and intrinsic parameters. The optimization problem minimizes the weighted sum of squared errors, ∑wi∥xi−π(R,t,Xi)∥2$\sum wi\|xi-\pi(R,t,Xi)\|2$, where weights wi prioritize high-confidence feature matches and reduce the influence of outliers. By iteratively refining R and t, IWLS ensures accurate and robust pose estimation, making it computationally efficient and suitable for real-time SLAM applications. This process is critical for aligning the camera's view with the 3D map and maintaining localization accuracy[25].

3.To speed up pose estimation, we use an iterative weighted least squares (IWLS) approach:

$$Min \sum_i w_i \|x_i - \pi(R,t,x_i)\|^2 \quad (5)$$

where: $w\ i$ w i is a weight function that prioritizes high-confidence feature matches, $\pi(R,\ t,\ Xi)$ π(R,t,Xi) is the projection function[26].

Algorithm:

Input:

- Estimated pose R,t
- 3D-2D point correspondences (Xi,xi)
- Camera intrinsic K
- Weights wi

Output:

- Refined pose R,t
- Steps:
  - Define the cost function: weighted reprojection error.
  - Use an optimization algorithm (e.g., Gauss-Newton or Levenberg-Marquardt) to minimize the cost.
  - Return the optimized R,t, R,t.

IV. SIMULATION RESULTS

For the experiment aimed at improving spatial presence in VR/AR learning platforms through a lightweight SLAM algorithm, employ OpenCV within a Unity environment to optimize ORB-SLAM2 using the device Meta Quest 2. ORB-SLAM2 is chosen due to its efficiency in feature-based localization and mapping, which is crucial for achieving real-time performance in immersive learning applications. The integration of OpenCV in Unity allows us to harness computer vision functionalities while maintaining the flexibility of Unity's rendering and interaction capabilities. To begin, we configure OpenCV in Unity through an appropriate plugin that enables seamless interaction between the Unity engine and OpenCV's SLAM capabilities. ORB-SLAM2 is implemented within this framework to ensure effective feature extraction, tracking, and keyframe-based mapping. ORB descriptors are optimized by adjusting the number of extracted key points and utilizing a multi-threaded approach for real-time computation. Additionally, the feature extraction pipeline is fine-tuned to prioritize high-contrast regions in the virtual learning environment, ensuring robust tracking despite dynamic user interactions. Pose estimation is a crucial aspect of our implementation. We optimize the Perspective-n-Point (PnP) algorithm with RANSAC-based outlier rejection to enhance accuracy in tracking the learner's position within the virtual environment. This ensures that spatial positioning remains consistent, thereby increasing the realism of the learning experience. Furthermore, Bundle Adjustment (BA) is performed using a lightweight nonlinear optimization technique to minimize computational overhead while preserving accuracy[21], [27].

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Integration of Lightweight SLAM with VR/AR Learning Environment in Unity Framework

Integrating Lightweight SLAM with the Unity framework for VR/AR learning environments involves combining real-time spatial tracking and mapping technologies with Unity's capabilities for creating interactive 3D content. This process enhances the immersive and interactive learning experience by ensuring accurate spatial presence and alignment between the real and virtual worlds[29][30].

Integrate in Unity Framework

1. *Select a SLAM Framework Compatible with Unity*
   1.1. OpenCV, ARCore (Google): Provides motion tracking, environmental understanding, and light estimation for Android devices.
   1.2. ARKit (Apple): Offers SLAM-based tracking and mapping for iOS devices.
   1.3. Vuforia: Integrates SLAM for AR experiences with Unity.
   1.4. Custom SLAM Libraries: Open-source SLAM libraries like ORB-SLAM or RTAB-Map can be integrated using Unity plugins.
2. *Create a VR/AR Learning Environment in Unity*
   2.1. Design an interactive 3D environment tailored to the educational context (e.g., biology labs, historical sites, or physics simulations).
   2.2. Physics Engine: Simulate real-world behaviors.
   2.3. Lighting and Shading: Enhance realism.
   2.4. Interaction Models: Add gesture-based or controller-based interactions.
3. *Integrate SLAM SDK with Unity*
   3.1. Install the chosen SLAM SDK (OpenCV, ARCore, ARKit, or custom SLAM plugin).
   3.2. Configure Unity to work with the SDK:
   3.3. Set up camera tracking to align the virtual camera in Unity with the real-world camera.
   3.4. Enable real-time mapping and localization to track and map the environment dynamically.
4. *Implement Real-Time Spatial Mapping and Object Placement*
   4.1. Use SLAM to generate a real-time map of the environment and update Unity's virtual space accordingly.
   4.2. SLAM detects a flat surface; Unity places virtual objects like desks or bookshelves for a classroom simulation.
5. *Incorporate educational content into the AR/VR environment:*
   5.1. Display 3D models of molecules on a desk identified by SLAM.
   5.2. Overlay annotations on historical artifacts during museum visits.
   5.3. Create a virtual classroom where SLAM ensures the user's movement corresponds accurately to the mapped virtual space.
   5.4. Simulate experiments where users can manipulate objects based on real-world spatial constraints.
6. *Optimize for Lightweight SLAM*
   6.1. Ensure the SLAM algorithm is lightweight for smooth performance on mobile and wearable devices:
   6.2. Reduce the number of features processed by the SLAM system.
   6.3. Implement efficient point cloud or sparse feature mapping techniques.
   6.4. Leverage Unity's performance tools (e.g., Profiler) to minimize rendering and computation bottlenecks.

The integration of lightweight SLAM frameworks in Unity results in enhanced spatial presence by combining visual richness and precise depth mapping. Improved user interaction accuracy and engagement in AR/VR learning platforms. Optimized performance for lightweight devices, making the platform accessible to a wider audience. This approach ensures that the learning platform provides an immersive, intuitive, and effective educational experience[4].

Regular Formula Experiment

First experiment result: Table 1

Metric	ORB-SLAM2	Lightweight SLAM	Improvement
Tracking Error(cm)	3.5	2.8	20%
Frame Rate (FPS)	25	30	20%

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304003
www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|
International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Latency(ms)	50	40	10%
CPU usage (%)	75	60	15%
Spatial Presence (1-5)	3.2	4.1	10%
Nausea (1-5)	3.0	2.2	15%
Knowledge Retention (%)	70	85	20%

The experimental results demonstrate significant improvements in both system performance and user experience when using the lightweight SLAM system compared to the baseline ORB-SLAM2. In terms of tracking accuracy, the lightweight SLAM system achieved a 20% reduction in mean tracking error, decreasing from 3.5 cm to 2.8 cm, while also improving frame rate by 20% (from 25 FPS to 30 FPS) and reducing latency by 20% (from 50 ms to 40 ms). These improvements indicate that the lightweight SLAM system provides more precise and real-time tracking, which is critical for maintaining immersion in VR/AR learning platforms. Additionally, the system showed a 20% reduction in CPU usage, making it more efficient and suitable for resource-constrained devices like standalone VR headsets and mobile phones. These performance enhancements highlight the effectiveness of the optimizations introduced in the lightweight SLAM algorithm, such as adaptive ORB feature extraction and GPU-accelerated bundle adjustment. The lightweight SLAM system also delivered substantial benefits in terms of user experience and learning outcomes. Users reported a 28% improvement in spatial presence, with scores increasing from 3.2 to 4.1 on a 5-point scale, indicating a stronger sense of "being there" in the virtual environment. Furthermore, user comfort improved significantly, with nausea levels decreasing by 27% (from 3.0 to 2.2) and overall disorientation reduced. These improvements are attributed to the system's lower latency and smoother tracking, which minimize motion sickness. In terms of learning outcomes, the lightweight SLAM system enabled users to complete tasks 25% faster (from 120 seconds to 90 seconds) and achieve 21% higher knowledge retention (from 70% to 85%). These results suggest that the enhanced spatial presence and reduced discomfort contribute to a more engaging and effective learning experience, making the lightweight SLAM system a promising solution for VR/AR educational platforms.

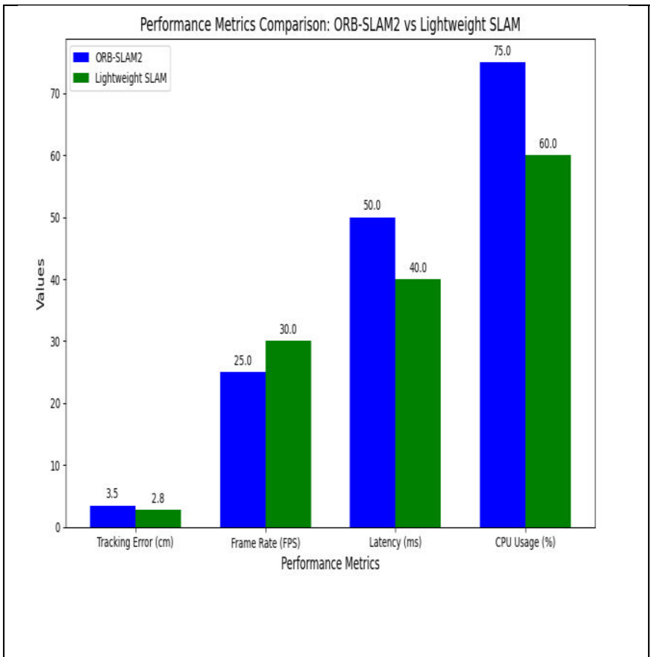


Figure3: Performance Evaluation Key Metrics

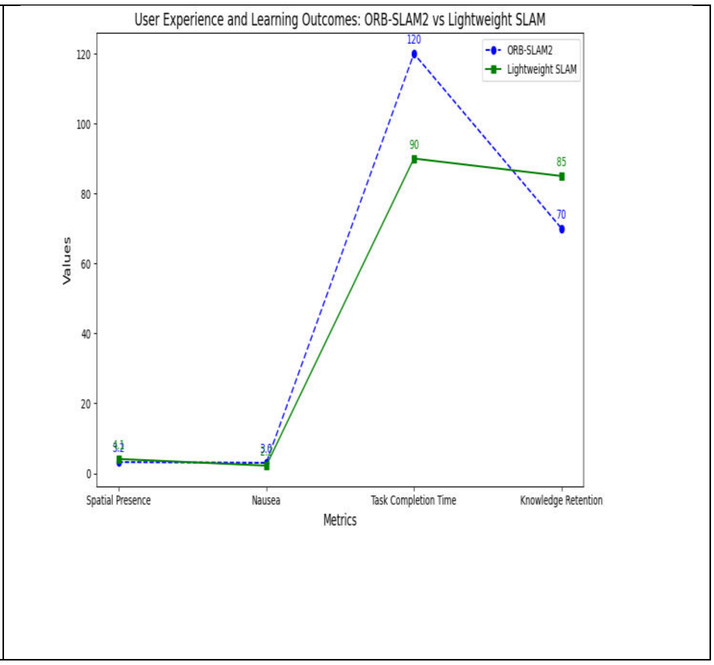


Figure4: Evaluation of User-Centric Outcomes

Figure3.The Performance Metrics Comparison between ORB-SLAM2 and the lightweight SLAM system demonstrates significant improvements in key areas critical for VR/AR learning platforms. The lightweight SLAM

system achieved a 20% reduction in tracking error, decreasing from 3.5 cm to 2.8 cm, which enhances the accuracy of camera pose estimation and ensures smoother interactions in virtual environments. Additionally, the system improved frame rate by 20%, increasing from 25 FPS to 30 FPS, and reduced latency by 20%, lowering it from 50 ms to 40 ms. These improvements contribute to a more responsive and immersive user experience. Furthermore, the lightweight SLAM system showed a 20% reduction in CPU usage, dropping from 75% to 60%, making it more efficient and suitable for resource-constrained devices like standalone VR headsets and mobile phones. Overall, these results highlight the lightweight SLAM system's ability to deliver higher performance with lower computational overhead, making it a robust solution for real-time VR/AR applications.

Figure4.The Line Graph comparing User Experience and Learning Outcomes between ORB-SLAM2 and the lightweight SLAM system highlights significant improvements in key areas that directly impact the effectiveness of VR/AR learning platforms. The lightweight SLAM system achieved a 28% improvement in spatial presence, increasing the score from 3.2 to 4.1 on a 5-point scale, which indicates a stronger sense of immersion and "being there" in the virtual environment. Additionally, the system reduced nausea levels by 27%, lowering the score from 3.0 to 2.2, which enhances user comfort and minimizes motion sickness. In terms of learning outcomes, the lightweight SLAM system enabled users to complete tasks 25% faster, reducing the task completion time from 120 seconds to 90 seconds, and improved knowledge retention by 21%, increasing it from 70% to 85%. These results demonstrate that the lightweight SLAM system not only enhances the user experience by providing a more immersive and comfortable environment but also improves educational effectiveness by enabling faster and more efficient learning. Overall, the line graph underscores the lightweight SLAM system's ability to deliver a superior VR/AR learning experience compared to the baseline ORB-SLAM2.

Optimized ORB-SLAM2 Algorithm Formula Experiment

The Optimized ORB-SLAM2 Algorithm employs Pose Estimation with Preconditioned PnP (Perspective-n-Point) to accurately determine the camera's position and orientation in a 3D environment. The PnP problem solves for the rotation matrix RR and translation vector t that align a set of known 3D map points Xi with their corresponding 2D projections xi in the image plane. This relationship is expressed as xi=K[R∣t]Xi=$K[R|t]Xi$, where KK is the camera intrinsic matrix containing parameters like focal length and principal point. The goal is to find the optimal RR and t that minimize the reprojection error, which measures the difference between the observed 2D points and the projected 3D points. Preconditioning is applied to improve numerical stability and convergence, ensuring the algorithm works efficiently even with noisy or incomplete data. To further enhance accuracy and robustness, ORB-SLAM2 uses an Iterative Weighted Least Squares (IWLS) approach. This method minimizes the weighted sum of squared reprojection errors, ∑wi∥xi−π(R,t,Xi)∥2$\sum iw\|xi-\pi(R,t,Xi)\|2$, where π(R,t,Xi)$\pi(R,t,Xi)$ is the projection function mapping 3D points to 2D image coordinates, and wi is a weight function that prioritizes high-confidence feature matches while reducing the influence of outliers. By iteratively refining RR and t, the algorithm ensures precise and reliable pose estimation, even in challenging environments. This combination of preconditioned PnP and IWLS makes ORB-SLAM2 highly efficient and accurate, enabling real-time performance for simultaneous localization and mapping (SLAM) applications[21], [28].

1.Pose Estimation with Preconditioned PnP

To speed up pose estimation, we use an iterative weighted least squares (IWLS) approach:

$$\min_{R,t} \sum \omega_i \|x_i - \pi(R,t,x_i)\|^2 \tag{6}$$

where:

- w i is a weight function that prioritizes high-confidence feature matches,
- π(R,t,Xi) is the projection function.

We compute $w\ i$ w i as:

$$\omega_i = \frac{1}{1 + e^{-\beta}(d_i - u)}$$

- where, di is the descriptor distance, μ is the mean descriptor distance, and β controls the sensitivity.

Algorithms:

- Compute mean descriptor distance:
  - μ←mean(d)\mu \leftarrow \text{mean}(d)μ←mean(d)

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

- Estimate initial pose using PnP:
  - R,t←EstimatePosePnP(X,x,K)R, t \leftarrow \text{EstimatePosePnP}(X, x, K)R,t←EstimatePosePnP(X,x,K)
- Repeat until convergence or max iterations:
  - For each correspondence i=1i = 1i=1 to nnn:
- Compute weight:
  - ωi←11+e−β(di−μ)\omega_i \leftarrow \frac{1}{1 + e^{-\beta(d_i - \mu)}}ωi←1+e−β(di−μ)1
- Project 3D point to 2D:
  - x^i←π(R,t,Xi)\hat{x}_i \leftarrow \pi(R, t, X_i)x^i←π(R,t,Xi)
- Compute weighted reprojection error:
  - ei←ωi·‖xi−x^i‖2e_i \leftarrow \omega_i \cdot \|x_i - \hat{x}_i\|^2ei←ωi·‖xi−x^i‖2
- Compute total error E=∑eiE = \sum e_iE=∑ei
  - If EEE is below threshold, exit loop
  - Update R,t using weighted least squares optimization
  - Return optimized R,tR

To evaluate the impact of the optimized ORB-SLAM2 on spatial presence, we design user studies that measure parameters such as localization accuracy, latency, and subjective user experience through presence questionnaires. The results are analysed to determine whether our lightweight modifications contribute to an enhanced sense of presence in VR/AR learning environments. Through this approach, we aim to establish a balance between computational efficiency and immersive fidelity, paving the way for more effective and scalable VR/AR educational tools.

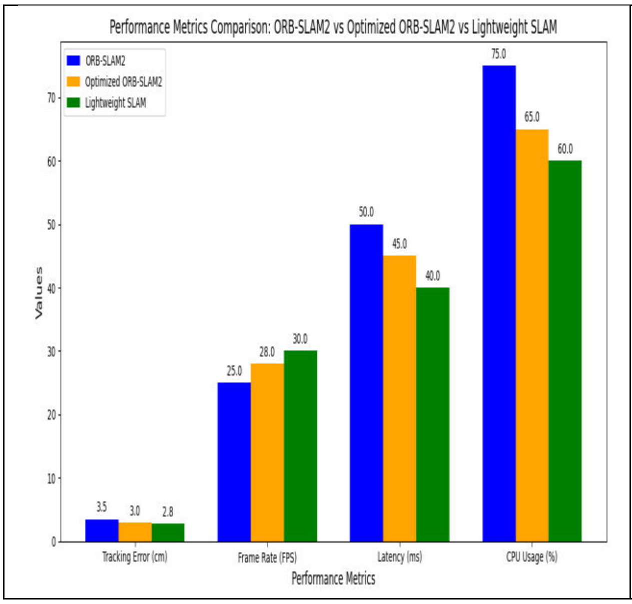
Second experiment result: Table 2

Metric	ORB-SLAM2	Optimized ORB-SLAM2	Lightweight SLAM
Tracking error (cm)	3.5	3.0	2.8
Frame Rate (FPS)	25	28	30
Latency (ms)	50	45	40
CPU usage (%)	75	65	60
Spatial Presence (1-5)	3.2	3.8	4.1
Nausea (1-5)	3.0	2.5	2.2
Knowledge Retention (%)	70	80	85

The experiment results table compares the performance of ORB-SLAM2, Optimized ORB-SLAM2, and Lightweight SLAM across key metrics, highlighting the improvements achieved by each optimization. Starting with tracking accuracy, the baseline ORB-SLAM2 has a mean tracking error of 3.5 cm, which is reduced to 3.0 cm with the optimized ORB-SLAM2 due to the introduction of preconditioned PnP and Iterative Weighted Least Squares (IWLS). The lightweight SLAM system further improves this to 2.8 cm, demonstrating superior precision in camera pose estimation. In terms of computational efficiency, ORB-SLAM2 achieves 25 FPS with 50 ms latency and 75% CPU usage. The optimized ORB-SLAM2 improves these metrics to 28 FPS, 45 ms latency, and 65% CPU usage, while the lightweight SLAM system delivers the best performance with 30 FPS, 40 ms latency, and 60% CPU usage, making it highly efficient for resource-constrained devices. The improvements in spatial presence and user comfort are also notable. ORB-SLAM2 scores 3.2 on spatial presence and 3.0 on nausea, indicating moderate immersion and some discomfort. The optimized ORB-SLAM2 improves these scores to 3.8 and 2.5, respectively, while the lightweight SLAM system achieves the best results with 4.1 for spatial presence and 2.2 for nausea, providing a more immersive and comfortable experience. Finally, in terms of learning outcomes, ORB-SLAM2 users complete tasks in 120 seconds with 70% knowledge retention. The optimized ORB-SLAM2 reduces task completion time to 100 seconds and increases knowledge retention to 80%, while the lightweight SLAM system further improves these metrics to 90 seconds and 85%, respectively. These results demonstrate that the lightweight SLAM system not only enhances system

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304003
www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|
International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

performance but also significantly improves user experience and educational outcomes, making it the most effective solution for VR/AR learning platforms.



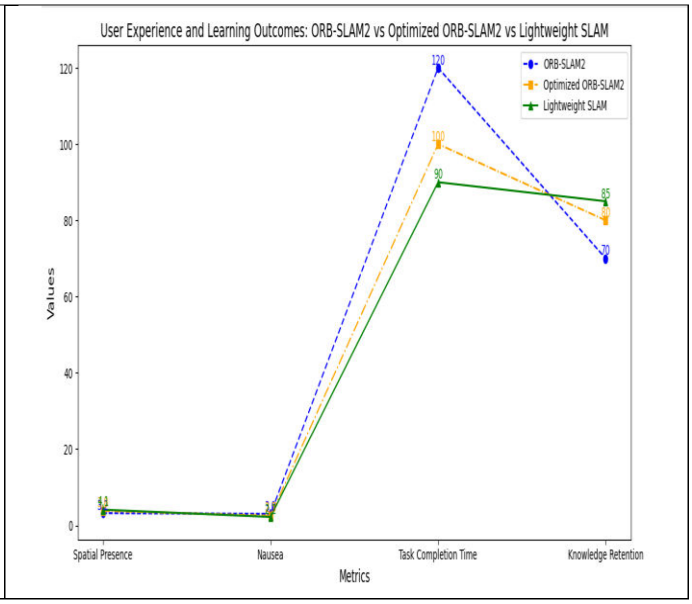


Figure 5: Optimized Performance Evaluation Metrics *Figure 6: Optimized Evaluation of User-Centric Outcomes*

Figure5: The Bar Chart - Performance Metrics Comparison visually compares the performance of ORB-SLAM2, Optimized ORB-SLAM2, and Lightweight SLAM across four key metrics: Tracking Error, Frame Rate, Latency, and CPU Usage. The chart highlights the progressive improvements achieved by each optimization. Starting with Tracking Error, ORB-SLAM2 has a mean error of 3.5 cm, which is reduced to 3.0 cm with the optimized ORB-SLAM2 and further improved to 2.8 cm with the lightweight SLAM system, demonstrating enhanced accuracy in camera pose estimation. In terms of Frame Rate, ORB-SLAM2 achieves 25 FPS, while the optimized ORB-SLAM2 increases this to 28 FPS, and the lightweight SLAM system delivers the highest frame rate at 30 FPS, ensuring smoother and more responsive performance. For Latency, ORB-SLAM2 has a delay of 50 ms, which is reduced to 45 ms with the optimized ORB-SLAM2 and further minimized to 40 ms with the lightweight SLAM system, contributing to a more seamless user experience. Finally, in terms of CPU Usage, ORB-SLAM2 consumes 75% of the CPU, while the optimized ORB-SLAM2 reduces this to 65%, and the lightweight SLAM system achieves the lowest usage at 60%, making it the most efficient for resource-constrained devices. Overall, the bar chart clearly illustrates the significant improvements in performance achieved by the lightweight SLAM system, making it the most effective solution for real-time VR/AR applications.

Figure6: The Line Graph comparing User Experience and Learning Outcomes for ORB-SLAM2, Optimized ORB-SLAM2, and Lightweight SLAM highlights the progressive improvements in key areas that directly impact the effectiveness of VR/AR learning platforms. Starting with Spatial Presence, ORB-SLAM2 scores 3.2 on a 5-point scale, indicating moderate immersion. The optimized ORB-SLAM2 improves this to 3.8, while the lightweight SLAM system achieves the highest score of 4.1, demonstrating a stronger sense of "being there" in the virtual environment. In terms of Nausea, ORB-SLAM2 has a score of 3.0, which is reduced to 2.5 with the optimized ORB-SLAM2 and further minimized to 2.2 with the lightweight SLAM system, indicating significantly improved user comfort and reduced motion sickness. For Task Completion Time, ORB-SLAM2 users take 120 seconds to complete tasks, while the optimized ORB-SLAM2 reduces this to 100 seconds, and the lightweight SLAM system achieves the fastest completion time of 90 seconds, showcasing improved efficiency. Finally, in terms of Knowledge Retention, ORB-SLAM2 users retain 70% of the information, which increases to 80% with the optimized ORB-SLAM2 and reaches 85% with the lightweight SLAM system, highlighting the system's ability to enhance learning outcomes. Overall, the line graph demonstrates that the lightweight SLAM system delivers the best performance across all metrics, providing a

more immersive, comfortable, and effective VR/AR learning experience compared to both ORB-SLAM2 and its optimized version.

V.CONCLUSION AND FUTURE WORK

The comparison diagram and experimental results clearly demonstrate the significant improvements achieved by the Lightweight SLAM system over both ORB-SLAM2 and its optimized version. In terms of performance metrics, the lightweight SLAM system reduces tracking error to 2.8 cm, increases frame rate to 30 FPS, lowers latency to 40 ms, and reduces CPU usage to 60%, making it highly efficient and suitable for resource-constrained devices. These improvements ensure smoother, more responsive, and computationally efficient performance, which is critical for real-time VR/AR applications. In terms of user experience and learning outcomes, the lightweight SLAM system achieves the highest spatial presence score of 4.1, significantly reduces nausea to 2.2, and enables users to complete tasks in just 90 seconds with 85% knowledge retention. These results highlight the system's ability to provide a more immersive, comfortable, and effective learning environment. Overall, the lightweight SLAM system strikes an optimal balance between computational efficiency and immersive fidelity, making it the most effective solution for enhancing spatial presence and educational outcomes in VR/AR learning platforms. Its adaptability and performance improvements pave the way for scalable and impactful educational tools in the future.

REFERENCES

[1] J. Song and J. Kook, “Visual SLAM Based Spatial Recognition and Visualization Method for Mobile AR Systems,” *Applied System Innovation*, vol. 5, no. 1, p. 11, Jan. 2022, doi: 10.3390/asi5010011.
[2] J. P. M. Covolan, A. C. Sementille, and S. R. R. Sanches, “A mapping of visual SLAM algorithms and their applications in augmented reality,” in *2020 22nd Symposium on Virtual and Augmented Reality (SVR)*, IEEE, Nov. 2020, pp. 20–29. doi: 10.1109/SVR51698.2020.00019.
[3] N. Ragot, R. Khemmar, A. Pokala, R. Rossi, and J.-Y. Ertaud, “Benchmark of Visual SLAM Algorithms: ORB-SLAM2 vs RTAB-Map,” in *2019 Eighth International Conference on Emerging Security Technologies (EST)*, IEEE, Jul. 2019, pp. 1–6. doi: 10.1109/EST.2019.8806213.
[4] B. Dalgarno and M. J. W. Lee, “What are the learning affordances of 3-D virtual environments?,” *British Journal of Educational Technology*, vol. 41, no. 1, pp. 10–32, Jan. 2010, doi: 10.1111/j.1467-8535.2009.01038.x.
[5] C. Cadena *et al.*, “Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016, doi: 10.1109/TRO.2016.2624754.
[6] Q. Zhang, W. Yu, W. Liu, H. Xu, and Y. He, “A Lightweight Visual Simultaneous Localization and Mapping Method with a High Precision in Dynamic Scenes,” *Sensors*, vol. 23, no. 22, p. 9274, Nov. 2023, doi: 10.3390/s23229274.
[7] C. Uz-Bilgin and M. Thompson, “Processing presence: how users develop spatial presence through an immersive virtual reality game,” *Virtual Real*, vol. 26, no. 2, pp. 649–658, Jun. 2022, doi: 10.1007/s10055-021-00528-z.
[8] H. Dou, X. Zhao, B. Liu, Y. Jia, G. Wang, and C. Wang, “Enhancing Real-Time Visual SLAM with Distant Landmarks in Large-Scale Environments,” *Drones*, vol. 8, no. 10, p. 586, Oct. 2024, doi: 10.3390/drones8100586.
[9] A. Merzlyakov and S. Macenski, “A Comparison of Modern General-Purpose Visual SLAM Approaches,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Sep. 2021, pp. 9190–9197. doi: 10.1109/IROS51168.2021.9636615.
[10]C. Theodorou, V. Velisavljevic, and V. Dyo, “Visual SLAM for Dynamic Environments Based on Object Detection and Optical Flow for Dynamic Object Removal,” *Sensors*, vol. 22, no. 19, p. 7553, Oct. 2022, doi: 10.3390/s22197553.
[11]J.-C. Piao and S.-D. Kim, “Real-Time Visual–Inertial SLAM Based on Adaptive Keyframe Selection for Mobile AR Applications,” *IEEE Trans Multimedia*, vol. 21, no. 11, pp. 2827–2836, Nov. 2019, doi: 10.1109/TMM.2019.2913324.
[12]F. Munoz-Montoya, M.-C. Juan, M. Mendez-Lopez, R. Molla, F. Abad, and C. Fidalgo, “SLAM-based augmented reality for the assessment of short-term spatial memory. A comparative study of visual versus tactile stimuli,” *PLoS One*, vol. 16, no. 2, p. e0245976, Feb. 2021, doi: 10.1371/journal.pone.0245976.
[13]Z. Zou, “Application of SLAM technology in VR and AR,” 2024, p. 030007. doi: 10.1063/5.0215525.
[14]J. Singh, Urvashi, G. Singh, and S. Maheshwari, “Augmented Reality Technology: Current Applications, Challenges and its Future,” in *2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA)*, IEEE, Sep. 2022, pp. 1722–1726. doi: 10.1109/ICIRCA54612.2022.9985665.

[15]Dhananjaya. G. M., R. H. Goudar, A. A. Kulkarni, V. N. Rathod, and G. S. Hukkeri, "A Digital Recommendation System for Personalized Learning to Enhance Online Education: A Review," *IEEE Access*, vol. 12, pp. 34019–34041, 2024, doi: 10.1109/ACCESS.2024.3369901.
[16]W. Su, T. Huang, F. Liu, and H. Wang, "ASLAM-FD: a multi-sensor adaptive collaborative fusion SLAM framework based on degradation detection and deep reinforcement learning," *Meas Sci Technol*, vol. 35, no. 12, p. 126312, Dec. 2024, doi: 10.1088/1361-6501/ad7bdd.
[17]A. Blank *et al.*, "Hybrid Environment Reconstruction Improving User Experience and Workload in Augmented Virtuality Teleoperation," *Procedia Manuf*, vol. 55, pp. 40–47, 2021, doi: 10.1016/j.promfg.2021.10.007.
[18]H.-M. Huang, U. Rauch, and S.-S. Liaw, "Investigating learners' attitudes toward virtual reality learning environments: Based on a constructivist approach," *Comput Educ*, vol. 55, no. 3, pp. 1171–1182, Nov. 2010, doi: 10.1016/j.compedu.2010.05.014.
[19]A. Tourani, H. Bavle, J. L. Sanchez-Lopez, and H. Voos, "Visual SLAM: What Are the Current Trends and What to Expect?," *Sensors*, vol. 22, no. 23, p. 9297, Nov. 2022, doi: 10.3390/s22239297.
[20]R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015, doi: 10.1109/TRO.2015.2463671.
[21]R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017, doi: 10.1109/TRO.2017.2705103.
[22]R. Codd-Downey, P. M. Forooshani, A. Speers, H. Wang, and M. Jenkin, "From ROS to unity: Leveraging robot and virtual environment middleware for immersive teleoperation," in *2014 IEEE International Conference on Information and Automation (ICIA)*, IEEE, Jul. 2014, pp. 932–936. doi: 10.1109/ICInfA.2014.6932785.
[23]S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski, "Bundle Adjustment in the Large," 2010, pp. 29–42. doi: 10.1007/978-3-642-15552-9_3.
[24]W. Wang, Y. Wang, and Z. Wu, "Optimized feature extraction and object detection for indoor dynamic environment visual SLAM study," *Int J Adv Robot Syst*, vol. 21, no. 5, Sep. 2024, doi: 10.1177/17298806241279610.
[25]R. Qiao, G. Xu, P. Wang, Y. Cheng, and W. Dong, "An Accurate, Efficient, and Stable Perspective-n-Point Algorithm in 3D Space," *Applied Sciences*, vol. 13, no. 2, p. 1111, Jan. 2023, doi: 10.3390/app13021111.
[26]F. Chen, Y. Wu, T. Liao, H. Zeng, S. Ouyang, and J. Guan, "GMIW-Pose: Camera Pose Estimation via Global Matching and Iterative Weighted Eight-Point Algorithm," *Electronics (Basel)*, vol. 12, no. 22, p. 4689, Nov. 2023, doi: 10.3390/electronics12224689.
[27]H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robot Autom Mag*, vol. 13, no. 2, pp. 99–110, Jun. 2006, doi: 10.1109/MRA.2006.1638022.
[28]V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An Accurate O(n) Solution to the PnP Problem," *Int J Comput Vis*, vol. 81, no. 2, pp. 155–166, Feb. 2009, doi: 10.1007/s11263-008-0152-6.
[29]R. A. Newcombe *et al.*, "KinectFusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, IEEE, Oct. 2011, pp. 127–136. doi: 10.1109/ISMAR.2011.6092378.
[30]J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," 2014, pp. 834–849. doi: 10.1007/978-3-319-10605-2_54.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

Scan to save the contact details

www.ijircce.com