Intelligent Systems in Motion:

A Comprehensive Review on Multi-Sensor Fusion and Information Processing From Sensing to Navigation in Path Planning

Yiyi Cai, School of Electronic and Information Engineering, South China University of Technology, China

Tuanfa Qin, The Guangxi Key Laboratory of Multimedia Communications and Network Technology, Guangxi University, China*

Yang Ou, School of Computer and Electronic Information, Guangxi University, China

Rui Wei, School of Computer and Electronic Information, Guangxi University, China

ABSTRACT

Simultaneous localization and mapping (SLAM) serves as a cornerstone in autonomous systems and has seen exponential growth in its roles, particularly in facilitating advanced path planning solutions. One emerging avenue of research that is rapidly evolving is the incorporation of multi-sensor fusion techniques to enhance SLAM-based path planning. The paper initiates with a thorough review of various sensor types and their attributes before covering a broad spectrum of both traditional and contemporary algorithms for multi-sensor fusion within SLAM. Performance evaluation metrics pertinent to SLAM and sensor fusion are explored. A special focus is laid on the interconnected roles and applications of multi-sensor fusion in SLAM-based path planning, discussing its significance in navigation scenarios as well as addressing challenges such as computational burden and real-time implementation. This paper sets the stage for future developments in creating more robust, resilient, and efficient SLAM-based path planning systems enabled by multi-sensor fusion.

KEYWORDS

Multi-Sensor Fusion, Simultaneous Localization and Mapping, SLAM-Based Path Planning

INTRODUCTION

Autonomous systems, particularly in the domain of robotics and unmanned vehicles, have seen substantial growth over the past few decades. Central to the autonomy of these systems is their ability to understand and navigate through their environment. Two technologies have been crucial in achieving this: Simultaneous Localization and Mapping (SLAM), and path planning algorithms.

While SLAM allows these systems to localize themselves within an unknown environment while concurrently mapping it (S. Wang, Wu, & Zhang, 2019), path planning algorithms help in determining the most efficient route from a starting point to a destination (Xuemin et al., 2018).

DOI: 10.4018/IJSWIS.333056
*Corresponding Author

However, the data used for SLAM and path planning often come from various sensors, each with their own strengths and limitations. For example, while LiDAR provides high-resolution distance measurements, it may struggle in foggy or dusty conditions. In contrast, radar can operate well in adverse weather but might not offer the same level of detail (Xu et al., 2022). This is where multi-sensor fusion plays a pivotal role.

Multi-sensor fusion involves the integration of data from different types of sensors to create a more robust, comprehensive, and accurate representation of the environment (D.J. Yeong, Velasco-Hernandez, et al., 2021). It allows for the pooling of sensor strengths while mitigating their individual limitations. Over the years, various multi-sensor fusion techniques, such as Kalman Filters, Particle Filters, and Bayesian Networks, have been developed to combine heterogeneous sensor data effectively (Narjes & Asghar, 2019).

The synergy of multi-sensor fusion and path planning within the SLAM framework opens up new avenues for enhanced navigation and safety. By fusing sensor data, SLAM algorithms can generate more accurate maps, and path planning algorithms can make more informed decisions. This is of paramount importance in dynamic and uncertain environments where real-time decision-making is critical (Y. Zhao et al., 2022).

Significance of the Survey

The integration of multi-sensor fusion with SLAM-based path planning is an area of burgeoning research, spurred by the rising demands of various applications including autonomous vehicles, robotics (Liang, 2020), and augmented reality. While individual papers, articles, and reports have addressed aspects of sensor fusion or path planning within the SLAM framework (Khan et al., 2021; Joachim et al., 2016), there is a conspicuous lack of comprehensive reviews that synthesize these interconnected domains.

In their article, Xiang et al. (2023) investigate the challenges and inconsistencies in multi-sensor fusion processes in autonomous driving, proposing an innovative taxonomy dividing fusion perception strategies into symmetric and asymmetric fusions with detailed subcategories. However, they also underscore the limitations in the current AD perception's reliability, highlighting challenges in environmental perception capability and in the robustness of data-driven methods, especially in extreme situations like blind areas.

X. Wang et al. (2023) delivered an exhaustive review of multi-sensor fusion 3D object detection networks in the context of autonomous driving, specifically focusing on the integration of LiDAR and cameras. Their review provides insights into popular datasets and assessment metrics. Despite offering a comprehensive analysis, this article is limited because it does not extensively explore solutions beyond the technical and measurement aspects. It also does not address the real-world applicability and integration challenges of these multi-sensor fusion systems in varied driving environments.

Another review by Wu et al. (2022) discusses the limitations of vision-based environmental perception technologies. It emphasizes the importance of multi-sensor fusion for enhanced adaptability and performance in complex, unstructured conditions, offering insights into various application scenarios, datasets, and sensor fusion methods. However, the paper does not provide detailed, practical implementations or case studies to showcase the real-world efficacy and adaptation of these multi-sensor fusion technologies in dynamically changing and challenging environments.

Likewise, Harun et al. (2022) delve into the intricate challenges of obstacle detection in Unmanned Autonomous Vehicles. They advocate for the integration of multi-sensor fusion technology to surmount the limitations inherent in utilizing a single type of sensor due to varied obstacle characteristics and ambient conditions. A limitation of this study is its overarching focus on providing a framework for sensor selection without giving in-depth analysis or empirical data on the performance efficacy of these sensor configurations in diverse, real-time application scenarios.

In another article, Chghaf et al. (2022) primarily examine the implementation of SLAM in autonomous vehicles, focusing on the integration and efficiency of onboard sensors, specifically camera-based and LiDAR-based systems. They also explore hardware-software co-design to optimize performance in real-time environments. However, their research is constrained by the complexity of algorithmic integration and real-time processing requirements, indicating a need for further exploration and optimization in these areas to enhance the performance and reliability of autonomous navigation systems.

The preceding papers and studies focus predominantly on methodologies related to multi-sensor fusion and signal or data processing within the realms of SLAM and automated navigation, applied in distinct contexts. Nonetheless, it is imperative to distinguish between SLAM and automated navigation as intertwined yet distinct domains. The intersection of these fields is prominently noted in path planning, a critical component in automated navigation, intricately enhanced by SLAM. There is a discernible gap in the literature with regard to the comprehensive exploration of multi-sensor fusion through the lens of SLAM-based path planning. This encompasses a holistic examination of sensor types, performance metrics, data fusion methodologies, and their merits, constraints, and applicability across varied environmental settings. This review aims to address this lacuna by encapsulating the symbiotic relationship between multi-sensor fusion and SLAM in the augmentation of automated navigation, carving out a nuanced narrative that intertwines these complex, multifaceted elements.

Our survey provides a comprehensive review of multi-sensor fusion techniques specifically adapted for SLAM and studies how they have been utilized to enhance path planning algorithms. By examining a wide range of sensor types, fusion methodologies, and path planning algorithms under the umbrella of SLAM, this survey serves as a one-stop reference, facilitating a deeper understanding of the current state-of-the-art, challenges, and future research directions. The paper aims not only to serve academic researchers but also to offer insights for engineers and stakeholders in industries ranging from robotics to transportation and beyond (Gollan et al., 2018).

Contributions of the Paper

The primary focus of this survey is to offer a comprehensive review of multi-sensor fusion techniques as they are specifically applied in the realm of SLAM for path planning. The survey covers a range of sensor types including, but not limited to, LiDAR, radar, cameras, and IMUs. It also investigates various fusion methodologies like Kalman Filters, Particle Filters, and Bayesian Networks. The target audience for this paper spans academic researchers, engineers, and industry stakeholders in the fields of robotics, autonomous vehicles, and navigation systems. While this review incorporates seminal works and key contributions up to the current year, it does not extend to real-time implementation or commercial applications of these technologies.

This paper distinguishes itself by delving into the intricate dynamics of multi-sensor fusion, particularly accentuating its pivotal role within SLAM for path planning. It manifests a nuanced evaluation, threading through an array of sensor technologies including LiDAR, radar, cameras, and IMUs, offering insights into their individual and combined efficacies. One of the salient features of this review is its in-depth analysis of diverse fusion methodologies. Kalman Filters, Particle Filters, and Bayesian Networks are meticulously explored, laying bare their operational mechanisms, strengths, and limitations. This paper casts light on the underlying principles, algorithms, and computational paradigms that propel these fusion methods to the forefront of modern SLAM and automated navigation ecosystems.

Sensor Types and Their Roles in Multi-Sensor Fusion and SLAM-Based Path Planning

This section offers an analytical review of core sensor technologies, including Cameras, LiDAR, Inertial Measurement Units (IMUs), and odometry, and their integration in Simultaneous Localization and Mapping (SLAM) and path planning. Informed by a range of referenced works, this section seeks to delve deep into these technologies, elaborating not only on their conceptual foundations but also on their practical applications, results, and inherent limitations.

We will explore these topics in the following ways:

1. Multifaceted Exploration of Sensor Technologies: We commence with an incisive exploration of each sensor type including cameras, LiDAR, IMU (Inertial Measurement Units) and odometry. These sub-sections provide exhaustive conceptual insights, each drawing from a rich corpus of academic contributions to elucidate the operational principles, applications, and constraints of these sensors in SLAM and path planning.
2. Conceptual Insights and Practical Implications: Though primarily conceptual, the discussions are anchored in the practical implications of each sensor type. We draw extensively on referenced works to bring to the fore the results and limitations associated with each technology, offering readers a comprehensive view that intertwines theoretical principles with real-world applications and challenges.
3. Innovations and Challenges: Our review meticulously highlights existing innovative algorithms and methods, emphasizing how they strive to address the identified challenges and limitations of sensor technologies. Each algorithm and method is assessed based on its contribution to enhancing the accuracy, efficiency, and reliability of SLAM and path planning systems.
4. Comprehensive Review: Our aim is to offer a detailed, balanced overview that transcends the theoretical foundations to encapsulate the practical results and persisting challenges. By harmonizing conceptual insights with real-world implications, this section aspires to be a valuable resource for researchers, aiding in the identification of areas ripe for further exploration and innovation.

Commonly Used Sensors: Characteristics and Comparative Analysis

Through the lens of accuracy, computational load, and environmental suitability, the different sensor types exhibit distinct advantages and limitations. Cameras, for instance, excel in providing rich, high-resolution data suitable for complex tasks such as semantic mapping and object recognition. However, their performance can be compromised in fluctuating lighting conditions, and the computational burden is non-negligible. On the other hand, LiDAR sensors generate precise 3D environmental models but are substantially affected by weather conditions and are typically more expensive. IMUs offer rapid sampling rates and are less sensitive to environmental conditions, but they suffer from cumulative errors over time. Odometry-based solutions are computationally efficient, but they are often prone to drift and are highly dependent on the quality of wheel-ground interaction. Table 1 shows a brief comparative analysis of sensors in SLAM.

Introduction to Cameras

Typical Cameras in SLAM

- Monocular Cameras: These are single-lens cameras that capture two-dimensional image data. Monocular cameras are often used for their lightweight characteristics and low computational burden. They are particularly popular in mobile robotics and drone-based SLAM (Mur-Artal et

Table 1. Brief comparative analysis of sensor types in SLAM

Sensor Type	Accuracy	Computational Load	Environmental Suitability
Camera	High	Moderate to High	Moderate
LiDAR	Very High	High	Low to Moderate
IMU	Moderate	Low	High
Odometry	Low	Low	Moderate

al., 2015; García et al., 2016). However, they often require additional movement to resolve scale ambiguity and cannot directly measure depth, which can be a limitation in certain applications.

- Binocular (Stereo) Cameras: Stereo cameras consist of two horizontally aligned lenses that capture the environment from slightly different angles, allowing for depth perception through stereo triangulation. These are often employed in applications where direct depth measurement is essential, such as in autonomous driving (Geiger et al., 2013; Kostavelis et al., 2016). However, they often come with increased computational costs and are typically larger and heavier than monocular cameras.

RGB-D Cameras: These cameras capture both color (RGB) and depth (D) information, often using infrared projectors to estimate the depth of each pixel. They have been extensively used in indoor SLAM applications like robotic vacuum cleaners and navigation aids for visually impaired individuals (Henry et al., 2014; S. Tang et al., 2016). The direct depth measurement capability significantly simplifies the mapping process, but it comes at the cost of higher computational requirements and limited outdoor usability due to infrared interference.

Each type of camera brings its own set of advantages and disadvantages to SLAM applications, influencing choices based on computational resources, environmental conditions, and specific task requirements. Whether utilized alone or in combination with other sensing modalities, cameras remain an indispensable tool in the landscape of SLAM and navigation technologies.

General Advantages and Limitations of Cameras in SLAM and Navigation

Cameras stand as one of the most widely employed sensing modalities in SLAM due to their ability to capture rich, high-resolution data that offers an intricate understanding of the environment. This richness allows the implementation of more complex algorithms for tasks such as object recognition, semantic mapping, and other higher-level functions crucial for advanced navigation scenarios (S. Yang et al., 2015; X. Yu et al., 2022). With improvements in sensor technology, modern cameras can even capture data across multiple spectrums, providing additional information layers that can be invaluable in specific applications like nighttime navigation or foggy conditions.

However, the advantages of data richness and high resolution come at the cost of computational complexity. The processing of image data, particularly in the context of real-time operations, requires robust computational resources for tasks such as image preprocessing, feature extraction, and data association. As SLAM algorithms become increasingly sophisticated to make use of this rich data, the computational burden can grow exponentially, affecting the real-time feasibility of SLAM systems.

In terms of environmental limitations, cameras are significantly impacted by lighting conditions. While they excel in well-lit, controlled environments, their performance degrades under low-light or variable lighting conditions. These limitations can cause challenges in feature extraction and matching, which are essential components of Visual SLAM (VSLAM) algorithms (L. Yu, Yang, & Yang, 2016). Moreover, rapid changes in lighting, such as moving from a dark space to a brightly lit area, can lead to temporary blindness, affecting SLAM performance. Weather conditions also impact the performance of cameras. For instance, fog, rain, and snow can scatter light and obscure vision, affecting the clarity and reliability of the data captured (Mohammed et al., 2020). This requires additional preprocessing steps to correct for the effect of these environmental limitations, thus further increasing the computational load.

Overall, while cameras offer a wide range of benefits for SLAM and navigation applications, it is imperative to weigh these advantages against the constraints imposed by varying environmental conditions and computational demands. As such, they are often used in combination with other sensor types to build a more resilient and robust SLAM system.

Introduction to LiDAR

LiDAR (Light Detection and Ranging) operates on the principle of sending laser beams toward a target and measuring the time it takes for the light to return after reflecting off the object. This time-of-flight data is then converted into distance metrics, which can be utilized to construct three-dimensional point clouds representing the environment (George & George, 2004). The technology has found broad applications across various sectors, but its role in SLAM (Simultaneous Localization and Mapping) is particularly noteworthy. In SLAM systems, LiDAR serves multiple functions: First, it contributes to localization by identifying unique geometric features in the environment that can be tracked over time. Second, it plays a crucial role in mapping by providing high-fidelity, three-dimensional representations of the environment, which are critical for tasks such as obstacle avoidance and navigation.

Typical Types of LiDAR in SLAM

In terms of types of LiDAR used in SLAM applications, several options are common:

- Mechanical Rotating LiDAR: These sensors employ a rotating head that emits laser beams, covering a 360-degree field of view. Their high coverage makes them suitable for applications that require comprehensive environmental mapping, such as autonomous vehicles (Gong et al., 2023).
- Solid-State LiDAR: Offering a more robust and compact design by eliminating moving parts, solid-state LiDARs generally provide a narrower field of view but are highly valued in applications where mechanical durability is essential.
- Flash LiDAR: Flash LiDAR systems use a single pulse to illuminate a scene and capture the return signal in a two-dimensional array. They offer fast data acquisition but are generally less accurate compared to mechanical or solid-state LiDARs (N. Li et al., 2022).

LiDAR technology has progressively evolved, and its integration into SLAM systems has been a subject of intensive research, focusing on improving accuracy, reducing computational load, and enhancing real-time performance (Meyer et al., 2022; L. Li et al., 2017).

General Advantages and Limitations of LiDAR

LiDAR (Light Detection and Ranging) has become an increasingly popular sensor in SLAM and navigation systems due to its capability for generating high-resolution 3D maps of the environment. The strength of LiDAR lies in its precision and reliability, offering detailed spatial information that can be critical for obstacle detection, collision avoidance, and path planning. This sensor's utility has been particularly noteworthy in the development of autonomous vehicles, where accurate real-time mapping is essential (Mohanan & Ambuja, 2018).

However, one of the key drawbacks of LiDAR is the high cost associated with acquiring and maintaining these high-resolution systems. Additionally, LiDAR can generate vast amounts of data, requiring robust computational resources for real-time processing and map building (Zhu et al., 2019). This becomes even more complex when LiDAR data is used in conjunction with other sensor modalities for multi-sensor fusion, which significantly increases the computational load.

Another important limitation of LiDAR sensors is their environmental sensitivity. Unlike cameras, LiDAR is relatively less sensitive to lighting variations, making it suitable for night-time operations or scenarios where lighting conditions can change dramatically (Burdziakowski & Bobkowska, 2021). However, the technology can be highly sensitive to weather conditions. Factors like rain, snow, and fog can significantly reduce the sensor's range and data reliability by scattering the laser beams (Zang et al., 2019). Moreover, performance can be affected by dust and other airborne particles, leading to sensor noise and erroneous readings.

Despite its susceptibility to certain environmental conditions, its high spatial resolution and reliable data capture make LiDAR an attractive choice for various SLAM applications. To mitigate its limitations, LiDAR is often used in tandem with other sensors, such as cameras or radar, to create a more comprehensive and resilient system for SLAM and navigation.

Introduction to Inertial Measurement Unit (IMU)

How IMU Works in SLAM and Navigation

An Inertial Measurement Unit (IMU) is an electronic device that measures and reports a body's specific force, angular velocity, and often the magnetic field surrounding the body, by using a combination of accelerometers, gyroscopes, and sometimes magnetometers. IMUs are typically characterized by their sampling frequency, which can range from low-frequency units used in hobbyist applications to high-frequency units suitable for scientific and industrial applications, often reaching up to 1,000 Hz or more (R. Li, et al., 2014). The basic architecture of an IMU generally includes a three-axis accelerometer to measure linear accelerations, a three-axis gyroscope to measure angular velocities, and in more advanced configurations, a three-axis magnetometer to measure magnetic fields. The sensor fusion algorithm combines these measurements to provide an estimate of the device's orientation and motion in three-dimensional space.

In the realm of SLAM and path planning, IMUs serve as valuable adjuncts to other sensing modalities, offering real-time data that can complement the information obtained from cameras, LiDAR, or other sensors. When used in conjunction with cameras, IMUs can assist in tasks like feature tracking and pose estimation, particularly in fast-motion scenarios where optical methods may falter (Fu et al., 2021). In setups involving LiDAR, the IMU can provide critical information about the vehicle or device orientation, helping to correct for pitch and roll that could otherwise distort the 3D point cloud generated by the LiDAR. By providing these capabilities, IMUs significantly enrich the data pool available for sensor fusion algorithms, making them indispensable components in modern SLAM and navigation systems (Aslam et al., 2020).

General Advantages and Limitations of IMU

There are several important advantages to using IMUs, such as:

- High-Frequency Data: One of the most notable advantages of IMUs is their ability to provide high-frequency data, sometimes reaching up to 1,000 Hz or more. This enables them to capture fast changes in orientation and position that slower sensors might miss, thus improving the temporal resolution of the SLAM system.
- Indoor and GPS-Denied Environments: IMUs are highly valuable in environments where other sensors like GPS are unreliable or unavailable. They function well in indoor settings, tunnels, and urban canyons, thus ensuring uninterrupted navigation capabilities.
- Low Cost and Power Efficiency: Compared to other sensor technologies like high-end LiDAR and radar systems, IMUs are generally more cost-effective and power-efficient. This makes them particularly suitable for small-scale robots, drones, or any system with limited power resources (Martinelli, 2012).
- Sensor Fusion Flexibility: IMUs are often fused with other sensor data to provide a robust navigation solution. They can be integrated seamlessly with cameras for visual-inertial odometry or with LiDAR for more accurate 3D mapping and localization.
- Reduced Computational Complexity: While processing IMU data does require some computational effort, it is generally less computationally intensive than processing the high-dimensional data from sensors like LiDAR or cameras. This is particularly beneficial for real-time SLAM applications where computational resources may be limited (Saraf et al., 2023).

However, there are also some limitations to using IMUs in SLAM or path planning, such as:

- Drift Error: IMUs suffer from drift error, which accumulates over time and can significantly compromise the system's accuracy. This issue is particularly pronounced in lower-quality units and usually necessitates the fusion with other sensor data for correction (Narasimhappa et al., 2020).
- Noise Sensitivity: The raw data from IMUs can be noisy, leading to inaccuracies in state estimation. Various filtering techniques, such as Kalman or particle filters, are commonly applied to mitigate this issue, albeit at the cost of additional computational complexity (Narasimhappa et al., 2018).

Introduction to Odometry

Odometry is a method used in robotics for estimating a robot's position relative to a starting point. The term is often associated with wheel encoders, but more broadly, odometry data can also come from other types of motion sensors. Odometry plays a crucial role in SLAM as it provides an initial, albeit noisy, estimate of the robot's motion between sensor readings. It acts as a "local mapping" source that works alongside other "global mapping" sensors such as LiDAR, cameras, and IMUs, to form a complete navigation solution (Chaudhari et al., 2019; S. Yu et al., 2016; Tian et al., 2017). In the context of SLAM, odometry data, which is often collected through wheel encoders or other kinematic measurements, provides quick and continuous estimates of the robot's position. These estimates are then refined using other sensor data to improve the accuracy and reliability of the SLAM system (Yin et al., 2020; Takamura et al., 2021). Odometry data can be particularly useful when the robot navigates through areas where other sensors have difficulty operating, such as under overhangs, around corners, or through tunnels.

Typical Types of Odometry

The typical types of odometry include:

- Wheel Odometry: Wheel odometry is among the most direct forms of odometry, utilizing wheel encoders to measure the rotation of the wheels in order to compute the robot's relative position. This method is commonly employed for wheeled robots. While it is generally straightforward, it can be prone to errors, especially when wheel slippage occurs on slippery or uneven surfaces (Ouyang et al., 2021).
- Visual Inertial Odometry (VIO): This combines visual information (from cameras) and inertial data (from IMUs) to estimate the robot's pose and velocity. VIO tends to offer improved robustness over pure visual or inertial methods alone because it integrates the strengths of both sensor types. Visual cues can offer absolute position estimates while the IMU provides high-frequency updates. The integration, however, often requires sophisticated algorithms and calibration techniques to achieve optimal results (L. Yu, Qin, et al., 2023).
- LiDAR Inertial Odometry (LIO): Similar in concept to VIO, LIO combines data from LiDAR sensors with inertial measurements from an IMU. The dense point clouds from LiDAR provide spatial understanding, while the IMU fills in the gaps with high-frequency pose estimates. The fusion of these sensors offers the potential for accurate pose estimation in challenging environments, especially in scenarios where visual cues might be sparse or unreliable (Shi et al., 2023; J. Tang, Zhang, et al., 2023).
- Acoustic Odometry: Uses sound waves to estimate position, often found in underwater applications (Ahmed et al., 2022; Iqbal et al., 2021).

General Advantages and Limitations of Odometry

The incorporation of odometry into SLAM architectures offers a series of merits and challenges, notably when amalgamated with other sensing mechanisms like cameras and LiDAR. Odometry is praised for its ability to provide real-time, low-latency data, a feature that becomes crucial for the dynamic decision-making process in multifaceted environments (Chen et al., 2022). This quick data acquisition is particularly advantageous when odometry is used in conjunction with other sensor types. For instance, the combination of visual data from cameras and odometric information—commonly referred to as Visual-Inertial Odometry (VIO)—has shown a marked reduction in drift errors and a significant enhancement in feature recognition capabilities for extended navigation tasks (Takamura et al., 2021). Likewise, when odometry is paired with LiDAR data (LiDAR-Inertial Odometry, or LIO), it compensates for scenarios where LiDAR information may be sparse or unreliable, thereby enhancing the robustness of the system (J. Tang, Zhang, et al., 2023).

Yet, the benefits of odometry are not without accompanying limitations. One of the most prevalent challenges is the issue of error accumulation. Odometry systems are notorious for inherent drift, which, if not corrected by external references, leads to cumulative navigational errors (Agostinho et al., 2022). The effectiveness of wheel-based odometry is further influenced by the quality of surface traction, making it less reliable on terrains with inconsistent frictional properties (Ouyang et al., 2021). Additionally, the introduction of multi-sensor configurations like VIO and LIO also renders the system susceptible to inaccuracies stemming from calibration errors, complicating the initial setup and maintenance (Forster et al., 2017).

In summary, while odometry brings resource efficiency and low-latency advantages, especially when fused with other sensing modalities, it requires scrupulous calibration and is prone to drift and environmental limitations. Thus, the utility of odometry in SLAM and navigation systems calls for a judicious evaluation of both its capabilities and constraints.

ALGORITHMS AND METHODOLOGIES FOR MULTI-SENSOR FUSION IN SLAM-BASED PATH PLANNING

The effectiveness of any SLAM-based path planning system is deeply contingent upon the underlying algorithms and methodologies employed for sensor fusion. The sensor data, whether it originates from cameras, LiDAR, IMUs, or odometry systems, needs to be appropriately fused to generate a coherent and accurate understanding of the environment. This fused data forms the basis for any subsequent path planning algorithms, ensuring that the system's navigation is both reliable and efficient.

In this section, we aim to provide a comprehensive review of the algorithms and methodologies commonly employed for multi-sensor fusion in the context of SLAM-based path planning. We start by examining traditional fusion algorithms, which have been foundational in the development of early SLAM systems and continue to be used in various forms. Subsequently, we discuss the rise of modern, often computational-intensive (Sharma et al., 2022) approaches that leverage advanced techniques such as deep learning (Lv et al., 2022). Further, we delve into application-specific methodologies, which have been tailored for particular operational scenarios like urban navigation (C. Yang, 2022), indoor mapping, and drone navigation. Finally, we present a comparative analysis, highlighting the key metrics for evaluation and discussing benchmarking studies that help us understand the trade-offs and considerations involved in choosing a fusion method.

Traditional Fusion Algorithms

Kalman Filters

One of the earliest and most widely used methods for sensor fusion in SLAM-based path planning is the Kalman Filter. Originating from control theory, Kalman Filters offer a recursive way to estimate the state of a system based on noisy observations (Kalman, 1960). They have been effectively applied in various SLAM systems to combine information from different sensor types, offering advantages in computational efficiency and real-time processing (Smith et al., 1990). Kalman Filters are computationally less intensive and well-suited for systems where real-time data processing is essential. They also work well when the system model and noise characteristics are linear or can be linearized. Their primary drawback is their inability to handle non-linear system models and noise characteristics effectively, which led to the development of Extended Kalman Filters and Unscented Kalman Filters as improvements (Julier & Uhlmann, 1997).

Kalman Filters operate in a two-step process: prediction and update (Kalman, 1960; Maybeck, 1979). In the prediction phase, the filter uses the previous state to predict the next state of the system. This prediction incorporates the system's dynamics and control inputs. The general mathematical representation for this prediction is:

$$x^{k|k+1} = Ax^{k-1|k-1} + Bu_{k} \tag{1}$$

where:

- $x^{k|k+1}$ is the predicted state,
- A is the state transition model,
- $x^{k-1|k-1}$ is the previous state estimate, and
- Bu_{k} is the control input, where B is the control-input model and u_{k} is the control vector.

In the update phase, the Kalman Filter refines its predicted state by incorporating a new measurement. The basic formula for this update phase is:

$$x^{k|k} = x^{k|k-1} + K\left(z_{k} - Hx^{k|k-1}\right) \tag{2}$$

where:

- $x^{k|k}$ is the updated state estimate,
- z_{k} is the actual measurement,
- H is the measurement model, and
- K is the Kalman Gain, calculated as:

$$K = P_{k|k-1}H_{T}\left(HP_{k|k-1}H_{T} + R\right)_{-1} \tag{3}$$

In Equation 3, is the predicted covariance and is the measurement noise covariance.

These two phases continue iteratively, allowing the system to update its belief about the state of the system continually, incorporating new measurements and predictions in a statistically optimal way.

Particle Filters

Particle Filters, or Sequential Monte Carlo methods, are another class of state estimation algorithms widely used in SLAM (L. Zhang et al., 2009; T. Li et al., 2010). Unlike Kalman Filters, Particle Filters can handle non-linear system models and non-Gaussian noise, making them more flexible but computationally more demanding (L. Zhang et al., 2009). This flexibility to handle non-linear and non-Gaussian systems is the principal advantage of Particle Filters. They have been applied in complex environments where the dynamics are not easily modeled by linear equations (Vishak et al., 2017). The computational intensity of Particle Filters makes them less suited for real-time applications, especially when the number of particles required for accurate state estimation is high.

The Particle Filter algorithm involves a probabilistic approach to estimate the state of a system, typically a Markov process. Algorithm 1 is a brief pseudocode for the Particle Filter algorithm.

Bayesian Networks

Though not commonly employed for the core functionality of SLAM, Bayesian Networks are utilized for complex decision-making subproblems and multi-sensor fusion. Bayesian Networks provide a graphical model to represent the probabilistic relationships among a set of variables. They have been used in some SLAM systems to model the interdependencies between various sensors and the environment (J. Zhang et al., 2019). The main advantage of Bayesian Networks is their ability to model complex relationships and dependencies between multiple variables, offering a more nuanced approach to sensor fusion (F. Yang et al., 2021). Like Particle Filters, Bayesian Networks are computationally intensive and may not be suitable for real-time applications. Moreover, constructing a meaningful Bayesian Network for a complex system can be a challenging task.

A Brief Comparison of Traditional Fusion Algorithms

Traditional sensor fusion algorithms have their own set of advantages and limitations, shaped by the complexities of SLAM-based path planning and the specific challenges posed by the environmental

Algorithm 1. Particle filter

```
Input:
N: Number of particles
f(x_t, u_t, w_t): State transition model
p(z_k | x_k): Likelihood function for observations
u_1, u_2, …, u_T: Control inputs
z_1, z_2, …, z_T: Observations
x_0: Initial state estimate

1. Initialization:
for i = 1 to N do
x[i]_0 = Sample from initial state distribution p(x_0)
w[i]_0 = 1/N
end for
2. for k = 1 to T do
2.1 Prediction:
for i = 1 to N do
w[i]_k = w[i]_{k-1}
x[i]_(k | k-1} = f(x[i]_{k-1 | k-1}, u_k, Sample from process noise)
end for
2.2 Update(weighting):
for i = 1 to N do
w[i]_k = w[i]_k * p(z_k | x[i]_{k | k-1})
end for
2.3 Normalization:
total_weight = Sum(w[i]_k for i = 1 to N)
for i = 1 to N do
w[i]_k = w[i]_k / total_weight
end for
2.4 Resampling:
Resample N particles {x[i]_{k | k}} from {x[i]_{k | k-1}} with probability w[i]_k
2.5 Estimation:
x_hat_k = Sum(w[i]_k * x[i]_{k | k} for i = 1 to N)
return x_hat_k for k = 1 to T
```

dynamics and sensor characteristics. Let's investigate the use cases, advantages, and disadvantages of each:

I. Kalman Filters.
   1. Use Cases: Widely used in linear or near-linear systems where the noise is Gaussian. Variants like the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) are adapted for non-linear systems.
   2. Advantages: Computationally efficient while providing real-time state estimation, which is crucial for SLAM applications.
   3. Disadvantages: The most significant disadvantage of Kalman Filters is their assumption of linear relationships, limiting their applicability in systems with non-linear characteristics.

II. Particle Filters.
   1. Use Cases: They are more versatile than Kalman filters and are capable of handling non-linear, non-Gaussian systems. Often used in scenarios where the system is too complex for Kalman Filters.
   2. Advantages: Can model more complex distributions and systems.
   3. Disadvantages: Computationally more expensive than Kalman Filters, especially as the state space's dimensionality increases.

III. Bayesian Networks.
   1. Use Cases: Generally used in more complex decision-making problems where the relationships between various variables need to be explicitly modeled. Not as commonly used for the core functionality of SLAM but could be used for specific sub-problems like sensor fusion, decision making, or diagnostics.
   2. Advantages: Can handle a wide variety of data types and distributions, making it versatile for complex systems.
   3. Disadvantages: Computational cost can be high, especially as the network grows. Also, constructing an accurate Bayesian Network may require expert knowledge.

Comparative Analysis of State Estimation Techniques in SLAM. A comparative analysis of Kalman Filters, Particle Filters, and Bayesian Networks in the context of SLAM is shown in Table 2. Each technique is evaluated based on computational complexity, the nature of the assumed noise distribution, applicability to non-linear systems, real-time suitability, and prevalence in SLAM literature.

Modern Approaches

Technological advancement has paved the way for contemporary approaches in the areas of SLAM and path planning that significantly diverge from traditional methods. These so-called "modern

Table 2. Comparative analysis of state estimation techniques in SLAM

Criteria	Kalman Filters	Particle Filters	Bayesian Networks
Computational Complexity	Low	High	Very High
Assumed Noise Distribution	Gaussian	Non-Gaussian	Flexible
Applicability to Non-linear Systems	Limited; variants like EKF and UKF available	High	High
Real-time Suitability	High	Limited by computational expense	Limited by network complexity
Common Use in SLAM	Very Common	Common	Less Common

approaches” are characterized by their emphasis on computational intelligence, the application of machine learning algorithms, and more complex optimization techniques. These developments have been stimulated by the exigencies of dealing with high-dimensional, noisy, and uncertain data in real-world scenarios. This section aims to shed light on these novel methodologies by focusing on two key areas: Deep Learning-based Fusion, and Sensor Fusion using Graph SLAM.

Deep Learning-Based Fusion

The advent of deep learning technologies has ushered in a paradigm shift in the landscape of multi-sensor fusion for SLAM and path planning. By harnessing the data-driven power of neural networks, researchers have circumvented some of the most vexing limitations associated with traditional, rule-based algorithms (Saleem et al., 2023; J.L. Yu et al., 2020). This remarkable transition offers an innovative pathway to tackle complex environments where conventional algorithms often falter. In essence, what qualifies these methodologies as “modern” is their reliance on machine learning techniques to automatically discern intricate patterns in sensor data, thereby negating the need for explicit programming.

A typical deep learning-based multi-sensor fusion system capitalizes on Convolutional Neural Networks (CNNs) for processing high-dimensional data (D. Li et al., 2019), such as images, while Recurrent Neural Networks (RNNs) handle sequences of sensor data with temporal dependencies. In a unified architecture, raw sensor data are channeled into neural network layers where high-level feature extraction occurs automatically. These extracted features are subsequently fused either at the feature level or the decision level, serving the dual purpose of optimizing data utilization and enhancing the accuracy of SLAM processes (J. Tang, Folkesson, & Jensfelt, 2018).

Deep learning-based fusion exhibits marked efficacy in complex scenarios that readily challenge rule-based counterparts. These include navigating cluttered or dynamically evolving environments and applications demanding a semantic understanding of the environment (Guebli & Belkhir, 2021). Its utility is notably prominent in applications like autonomous vehicles, where the system must adapt to a wide variety of conditions on the fly (Darapaneni, N. et al., 2021).

Several seminal works have fortified the intellectual underpinning of this field. For instance, the study by X. Zhao et al. (2020) was a pioneering effort in fusing camera and LiDAR data specifically for autonomous driving applications. Similarly, J. Tang, Folkesson, & Jensfelt (2018) designed an innovative architecture incorporating CNNs and RNNs, setting a benchmark for indoor navigation solutions. Teng et al. (2021) ventured into the realm of decision-level fusion, thereby broadening the scope and applicability of deep learning-based sensor fusion techniques.

While the versatility of deep learning-based fusion is laudable, it is not devoid of shortcomings. Its computational intensity and data-dependency pose significant challenges. Moreover, the opaqueness inherent in the “black-box” nature of deep learning algorithms becomes especially pertinent in scenarios where explicability is non-negotiable, such as in safety-critical applications. In summary, deep learning-based fusion methods have carved a niche for themselves in the complex terrains of SLAM and multi-sensor systems, offering promising solutions at the expense of computational complexity and data requirements. Table 3 offers a brief comparative analysis of both types of typical deep learning-based multi-sensor fusion systems.

Sensor Fusion Using Graph SLAM

Graph SLAM, standing for Graph-based Simultaneous Localization and Mapping, serves as a pivotal technique for tackling the formidable SLAM problem by transforming it into an optimization issue. Developed as a probabilistic framework, Graph SLAM offers an integrated structure for multi-sensor fusion, where different types of sensory measurements—be it from LiDAR, cameras, or IMUs—are amalgamated into a singular graph-based representation. This leads to more nuanced, robust, and accurate solutions to the complex problems inherent in SLAM applications.

Table 3. Comparative analysis of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) in the context of SLAM

Criteria	Convolutional Neural Networks (CNNs)	Recurrent Neural Networks (RNNs)
Architecture	Composed of convolutional layers, activation layers, and often pooling layers.	Sequential architecture featuring loops that allow for temporal dynamic behavior.
Key Strengths	Effective for spatial hierarchies and identifying local patterns.	Good at capturing temporal dynamics and context in data.
Key Weaknesses	Limited ability to handle temporal dynamics.	Suffer from vanishing and exploding gradient problems, making them hard to train on long sequences.
Typical Use in SLAM	- Feature extraction from images - Semantic understanding of scenes - Object recognition	- Predicting future states of dynamic objects - Learning temporal dependencies in sensor data
Specific Scenarios in SLAM	- Object and landmark recognition - Scene segmentation - Depth estimation	- Time-series data fusion (e.g., IMU readings) - Predicting future positions of dynamic objects
Representative Papers	CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction (Tateno et al., 2017)	DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks (S. Wang, Clark, et al., 2017)

The foundational element of Graph SLAM is a graph in which each node embodies either a robot pose at a particular instance or a significant environmental landmark. Edges between these nodes stand for spatial or temporal constraints that are derived from sensor measurements. As sensors like LiDARs and cameras produce data, these measurements are used to generate or update the constraints, which are modeled as edges in the graph. Through iterative optimization algorithms, the graph is adjusted in such a way that the accumulated error across these constraints is minimized. In essence, this optimization process resolves the uncertainties in both localization and mapping.

When it comes to applicability, Graph SLAM excels in scenarios where the sensor data may be sporadic, noisy, or originate from multiple sources. Given its capacity for consolidating diverse types of information into a singular framework, it shows exemplary performance in dynamic and complex settings. Its capability to efficiently solve large-scale optimization problems makes it apt for large mapping applications, where traditional methods might falter.

The trajectory of research in Graph SLAM has been marked by several significant milestones that have fundamentally shaped its current state. One of the seminal works in this area was by Thrun & Montemerlo (2006), which shifted the perspective of SLAM from being primarily a filtering problem to an optimization problem. This shift provided the foundational framework for a multitude of subsequent advancements. Building on this, Kümmerle, Grisetti, et al. (2011) introduced the g2o library, a landmark development that revolutionized the optimization algorithms in Graph SLAM. The library significantly broadened the applicability of Graph SLAM by making it efficient enough for large-scale mapping and high-speed applications.

The field then saw further enrichment with the inclusion of semantic data in the mapping process. Hornung et al. (2013) explored how semantic labeling of landmarks and environments could make SLAM solutions not just geometric but also semantically meaningful. Around the same time, the issue of loop closure detection, a notoriously challenging aspect of SLAM, was robustly addressed by Cadena et al. (2016). Their work presented algorithms for reliably identifying loop closures, which significantly enhanced map accuracy.

Subsequently, the utilization of 3D LiDAR point clouds in Graph SLAM became a focus of research. Bosse & Zlot's (2009) paper laid the groundwork for this by extending Graph SLAM techniques to assimilate 3D point cloud data, significantly improving both environmental perception and mapping accuracy. As the field matured, the introduction of new optimization algorithms like SE-Sync by Dong et al. (2015) offered a certifiable solution to the Graph SLAM problem. This work signaled the maturity the field had achieved, indicating its readiness for more complex applications. This readiness was further corroborated by Roumeliotis et al. in 2016, who demonstrated the real-time capabilities of Graph SLAM in multi-robot scenarios. Finally, the concept of lifelong SLAM, where robots continually update the map over extended periods, was notably explored by M. Zhao et al. in 2021, marking a significant step towards more adaptive and long-lasting SLAM solutions.

While robust and scalable, Graph SLAM is not without its shortcomings. The computational burden associated with solving the optimization problem can be considerable, especially as the map grows in size. Furthermore, the integration of measurements from diverse sensors introduces challenges in data association and timing synchronization. These challenges need to be adeptly managed to maintain the accuracy and reliability of the SLAM process. The future research trajectory in the realm of Graph SLAM aims at several fronts. One avenue is the quest for making the algorithm more adaptive to various types of sensors, including newer, less-conventional ones. Further, the effective incorporation of semantic data into the graph for enhanced context-awareness is also a focal point of ongoing research. Last but not least, reducing the computational complexity for real-time applications remains a crucial objective.

Comparative Analysis

In this section, we focus on a systematic comparative analysis of the diverse sensor types and sensor fusion techniques presented in the preceding sections. Our primary objectives include evaluating these technologies based on several metrics, discussing benchmarking studies that validate their effectiveness, and exploring various trade-offs and considerations essential for their deployment in SLAM applications.

Evaluation Metrics

Evaluation metrics serve as the backbone of any comparative analysis, providing a standardized set of criteria against which different technologies can be assessed. In the context of SLAM and sensor fusion, typical metrics involve accuracy, computational complexity, robustness, and scalability. Accuracy is paramount for ensuring reliable mapping and localization, and it often varies significantly between sensors like cameras, LiDAR, IMUs, and odometry systems (Kümmerle, Steder, & Dornhege, 2009). Computational complexity is another vital factor, especially in real-time applications where resource constraints can be severe (Jiang et al., 2019). Robustness encompasses the system's ability to maintain performance under varying environmental conditions, including lighting, weather, and motion dynamics (X. Li et al., 2023). Scalability reflects how well the system can adapt to larger or more complex environments, a criterion that is increasingly important in modern SLAM research (Asaad & Maghdid, 2021).

Table 4 aims to summarize the metrics crucial for evaluating different technologies in SLAM applications. For instance, while accuracy is a crucial factor for all, it is of paramount importance in LiDAR systems. On the other hand, computational complexity is a significant concern for camera-based systems but is less so for Kalman Filters. By presenting these evaluation metrics in a tabular form, we facilitate a more straightforward comparison, thus aiding the decision-making process for researchers and practitioners alike.

Benchmarking Studies

Benchmarking is an indispensable approach to evaluating the performance and utility of various multi-sensor fusion techniques in SLAM. Comparing these techniques on standardized datasets allows for an objective assessment of their capabilities. Several studies have set the benchmarks in this field, thereby providing comprehensive analysis platforms.

Table 4. Analysis of evaluation metrics across different sensor types and fusion methods

Evaluation Metric	Importance in Cameras	Importance in LiDAR	Importance in IMUs	Importance in Odometry	Importance in Kalman Filters	Importance in Particle Filters
Accuracy	High	Very High	Medium	High	High	High
Computational Complexity	High	Medium	Low	Medium	Low	High
Robustness	Medium	Low	High	Medium	High	Medium
Scalability	Medium	High	High	Low	Medium	Low

Note. When a metric is "high," it is of significant importance for this sensor type or fusion method. When a metric is "medium," it is important but is not a primary concern, and when a metric is "low," it is of lesser importance for this sensor type or fusion method.

Table 5. Summary of standard datasets in SLAM sensor fusion benchmarking

Dataset	Source of Data	Characteristics	Environments	Common Metrics Used
KITTI	Automotive	High-resolution, large-scale	Urban, Highway	RMSE, Mahalanobis distance
TUM RGB-D	Indoor cameras	RGB-D data, moderate scale	Indoor, Office	RMSE, Drift rate
EuRoC MAV	Aerial vehicles	High-dynamic range	Indoor, Aerial	ATE, RPE

Note. RMSE = Root Mean Square Error; ATE = Absolute Trajectory Error; RPE = Relative Pose Error

Standard Datasets

Commonly used datasets for benchmarking include KITTI (Geiger et al., 2013), TUM RGB-D (Sturm et al., 2012), and the EuRoC MAV Dataset (Sturm et al., 2012). These datasets provide a variety of test environments, such as urban landscapes, indoor spaces, and aerial environments, offering a robust challenge to the accuracy and reliability of sensor fusion algorithms. The goal is to establish a standardized basis for comparing how different sensor fusion methods perform under analogous conditions (Alatise & Hancke, 2020; Fung et al., 2017). Table 5 outlines key characteristics, typical environments, and commonly used metrics for evaluation in KITTI, TUM RGB-D, and EuRoC MAV datasets.

Criteria for Comparison

The criteria for comparing sensor fusion methods generally include accuracy, computational cost, robustness, and scalability, as shown in Table 6. Each of these criteria is rigorously tested on standardized datasets to ensure a fair comparison.

APPLICATION OF MULTI-SENSOR FUSION IN SLAM-BASED PATH PLANNING

In the contemporary landscape of robotics and autonomous systems, path planning remains a cornerstone for operational efficacy. While the essence of path planning—determining the

Table 6. Description of criteria in SLAM and navigation systems

Criteria	Description
Accuracy	Measures the closeness of the computed values to the ground truth.
Computational Cost	Evaluates the required computational resources for algorithm execution.
Robustness	Assesses the algorithm's performance in challenging conditions, such as noisy data or dynamic environments.
Scalability	Tests the algorithm's ability to handle increasing data sizes and complexities.

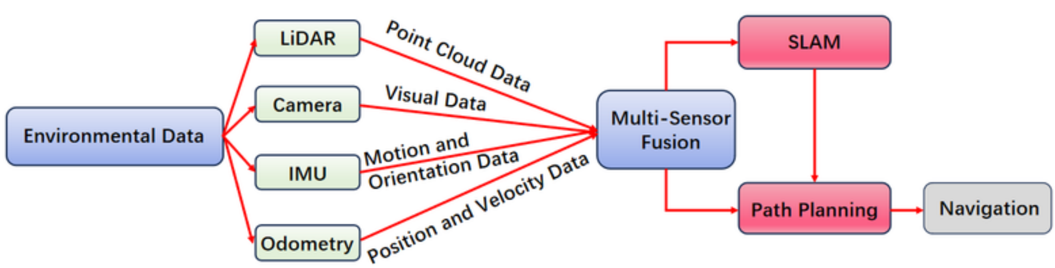
most efficient and safe trajectory between points A and B—seems straightforward, its real-world implementation is fraught with complexities. This is further exacerbated when the environment is unknown or dynamic, necessitating real-time mapping and localization for effective navigation. This is where Simultaneous Localization and Mapping (SLAM) comes into play. However, the limitations of using single-sensor data in SLAM are rapidly uncovered when applied to the multidimensional complexities of real-world environments. Thus, multi-sensor fusion emerges as a critical enabler in enhancing the robustness and accuracy of SLAM-based path planning.

The integration of multi-sensor fusion in SLAM-based path planning represents a pivotal advancement, addressing various limitations inherent to single-sensor systems. By amalgamating diverse data sources, be it LiDAR, cameras, IMUs, or odometry, a more comprehensive environmental understanding is achieved. This enriched environmental model not only improves the SLAM algorithm's accuracy but also robustly informs path planning algorithms, thereby enabling safer and more efficient navigation.

Figure 1 encapsulates the streamlined progression of data and the procedural steps pivotal to the integration of multi-sensor fusion within SLAM-based path planning frameworks for autonomous systems. The various components of the figure are explained below:

1. Data Gathering: The workflow commences with the acquisition of diverse environmental data through an array of sensors including Cameras, LiDAR, IMUs, and Odometry. Each category of sensors is instrumental in capturing a specific type of data. Namely,
   - Cameras: Capture rich visual data.
   - LiDAR: Collects point cloud data for measuring distances.
   - IMUs: Record motion and orientation data.
   - Odometry: Measures positional and velocity data.
2. Multi-Sensor Fusion: These diverse datasets, rich in their unique attributes, converge at the Multi-Sensor Fusion stage, where they are combined to provide a more detailed and accurate representation of the environment.
3. SLAM Process: The fused data feeds into the SLAM process, supporting the creation of real-time maps and ensuring accurate localization in different environments.
4. Path Planning: The Path Planning stage utilizes the outputs from both the multi-sensor fusion and SLAM to determine optimal navigation paths. The focus here is on ensuring that the paths are both efficient and safe, given the available environmental and localization data.
5. Navigation: At the Navigation stage, these planned paths are put into action, enabling the autonomous system to move effectively and safely through the specified environment. Each stage is interconnected, highlighting the importance of multi-sensor data and its processing in enhancing the overall efficacy of autonomous navigation systems.

Figure 1. Multi-sensor fusion workflow in SLAM-based path planning for autonomous navigation



Interconnected Roles of Multi-Sensor Fusion in SLAM, Navigation, and Path Planning

The integration of multiple sensors is not merely a technical enhancement but a paradigm shift that significantly impacts various aspects of robotics, particularly SLAM (Simultaneous Localization and Mapping), navigation, and path planning. This section aims to elucidate the interconnected roles that multi-sensor fusion plays in these crucial areas.

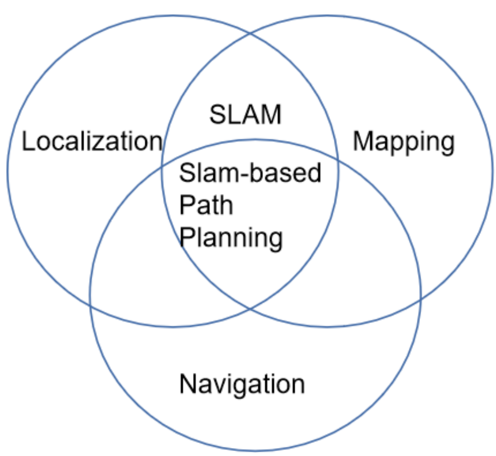
Figure 2 shows the interrelationships between SLAM, navigation, and path planning. The components of the figure include:

- SLAM and Navigation: SLAM underpins the effectiveness of autonomous navigation by generating a comprehensive and adaptive map, concurrently localizing the vehicle within this dynamically constructed environment. The accuracy and robustness of SLAM's mapping and localization directly correlate with the navigation system's ability to execute precise and adaptive movements. Without reliable SLAM outputs, navigation is effectively incapacitated, lacking the situational awareness necessary for informed movement.
- SLAM and Path Planning: Path planning's efficacy is intricately linked to the quality of the map produced by SLAM. The identification of obstacles, safe zones, and other pivotal environmental elements is predicated on the granularity and accuracy of these maps. Enhanced mapping outcomes, facilitated by advanced SLAM algorithms, directly contribute to the evolution of adaptive and optimized paths, ensuring not only efficient navigation but also heightened safety protocols.
- Navigation and Path Planning: Navigation encompasses the overarching process of transiting from an origin to a destination, wherein path planning serves as an essential subset. Path planning algorithms are instrumental in delineating optimal trajectories, ensuring that navigation is not only purposeful but is executed with maximal efficiency and safety. The synergy between these elements is emblematic of the nuanced interplay that characterizes autonomous systems' operational dynamics.

Multi-Sensor Fusion and SLAM

SLAM's primary goal is to construct a map of an environment while simultaneously keeping track of the agent's location within that environment. While single-sensor systems can perform these tasks

Figure 2. Interrelationships between SLAM, navigation, and path planning



to a certain extent, their capabilities are often limited by the sensor's inherent weaknesses, such as sensitivity to environmental conditions or computational load. Multi-sensor fusion mitigates these limitations by integrating data from various sensors like cameras, LiDAR, IMUs, and odometry systems. This integration not only improves the system's robustness but also enhances the accuracy and richness of the constructed maps. For example, the fusion of LiDAR and camera data can provide both geometric and semantic information, making the map more informative for both humans and machines (Feng et al., 2021).

Multi-Sensor Fusion and Navigation

Navigation is the process of safely and efficiently moving an agent from one point to another. Effective navigation requires a combination of localization, obstacle avoidance, and path planning. Multi-sensor fusion significantly boosts the agent's perception capabilities, enabling better decision-making in real-time. For instance, the fusion of IMU and GPS data can provide a more reliable estimate of the agent's current location, thereby facilitating more accurate and efficient navigation routes (Elghazaly et al., 2023).

Multi-Sensor Fusion and Path Planning

Before delving into the role of multi-sensor fusion in path planning, it is essential to distinguish it from navigation. While navigation is a broader term that encompasses the entire process of moving from one location to another, path planning is a subset of navigation. Specifically, path planning involves determining the most efficient route from point A to point B, given a set of environmental constraints. It works in tandem with navigation, providing the "how" to the "where" that navigation aims to reach.

In this context, multi-sensor fusion significantly enhances the path planning process. While traditional path planning algorithms often rely on static, pre-constructed maps and have limited adaptability to real-time changes, SLAM-based path planning is more dynamic. Here, multi-sensor fusion becomes invaluable. By amalgamating real-time data from various sensors like LiDAR, IMUs, and cameras, the system gains a more comprehensive understanding of the environment. This allows for the detection of dynamic obstacles and changing conditions, which traditional path planning methods may not account for. Consequently, it enables safer and more efficient routes, reducing the risk of collision and optimizing energy consumption (Liu et al., 2023).

Types of Applications for Multi-Sensor Fusion in SLAM-Based Path Planning

The incorporation of multi-sensor fusion techniques into SLAM-based path planning has gained immense traction, driving transformative applications across various verticals. By synergistically coupling accurate mapping, robust navigation, and efficient path planning, multi-sensor fusion has served as a catalyst for the development and deployment of increasingly complex robotic and autonomous systems. This section elucidates the myriad applications where this integrated approach is instrumental.

Autonomous Vehicles

The proliferation of autonomous vehicles (AVs) has instigated a paradigm shift in urban transportation and logistics, setting the stage for smarter, safer, and more efficient systems. Within the architecture of AVs, multi-sensor fusion plays an indispensable role by assimilating data from a variety of sensors such as LiDAR, cameras, radar, and Inertial Measurement Units (IMUs). This facilitates the generation of highly accurate maps and real-time path planning. Also, a novel algorithm (Chui et al., 2022) using a modified AODV routing protocol in VANETS Cloud has been proposed to alert trailing vehicles when the leading one slows down, aiming to enhance safety by increasing the reaction time of drivers in low visibility conditions. The multimodal nature of sensor fusion ensures that AVs can adeptly navigate through intricate urban landscapes, tackling challenges like dynamically moving obstacles, varying weather conditions, and intricate traffic patterns (Yeong, Barry, & Walsh, 2020). Such robustness is a testament to the transformative potential of multi-sensor fusion in the automotive industry.

Robotics

In both industrial and service robotics, the imperatives of task efficiency and operational reliability have underscored the need for sophisticated navigation and planning algorithms. Here, multi-sensor fusion augments SLAM-based path planning by amalgamating data from a spectrum of sensors like visual cameras, inertial sensors, and sometimes even tactile sensors. This enriched perception allows robots to perform diverse tasks such as inventory management in warehouses, environmental monitoring in hazardous zones, and search-and-rescue missions in unstructured terrains (An et al., 2023).

Augmented Reality (AR) and Virtual Reality (VR)

In the realm of Augmented Reality (AR) and Virtual Reality (VR), user experience is directly influenced by the system's ability to offer seamless and realistic simulations. Multi-sensor fusion significantly enhances these virtual experiences by fusing data from accelerometers, gyroscopes, and optical cameras. This provides extremely accurate motion tracking and spatial awareness, which is paramount not just in gaming but also in specialized training simulations for sectors such as healthcare, military, and industrial manufacturing (Macario et al., 2022).

Aerospace and Maritime Applications

Unmanned Aerial Vehicles (UAVs) and autonomous maritime systems present another frontier for multi-sensor fusion in SLAM-based path planning. These platforms often operate in GPS-denied or GPS-unreliable domains, necessitating the fusion of an array of sensors for safe and effective navigation. This includes, but is not limited to, acoustic sonars, magnetic compasses, and visual-inertial systems. The implementation of multi-sensor fusion ensures that these systems can adapt to rapidly changing environmental conditions, thereby offering superior operational flexibility (Muhammad et al., 2021).

Smart Infrastructure and IoT

The smart cities of the future will inevitably be powered by intricate networks of interconnected devices and systems (Memos et al., 2018). Here, sensor fusion techniques, integrated with SLAM-based navigation algorithms, will be instrumental in orchestrating complex tasks like real-time traffic management, automated waste collection, and public safety surveillance. By leveraging a diverse set of sensors connected through IoT networks, these smart infrastructure systems stand to gain in terms of both efficiency and reliability (Tonga et al., 2022; Plageras et al., 2018). Vijayakumar, Rajkumar, & Deborah (2022) introduce a refined passive-awake assistant methodology, which is integral in enhancing energy conservation during wireless transmissions. When synergized with IoT technologies, this approach markedly optimizes power utilization, extending the endurance and efficiency of electric vehicles, especially in scenarios demanding extended operational durations.

Typical Algorithms and Methodologies for SLAM-Based Path Planning

The intersection of SLAM and path planning has provided fertile ground for a rich variety of algorithms and methodologies. These are further enhanced when combined with multi-sensor fusion, which contributes additional layers of reliability and robustness to these approaches. Here, we review the primary algorithms and methodologies that have garnered attention in both academia and industry.

Graph-Based Methods

Graph-based algorithms serve as one of the cornerstones in SLAM-based path planning, primarily due to their computational efficiency and flexibility in handling complex spaces. This section delves into two seminal graph-based methods, namely, Dijkstra's algorithm and the A* algorithm, highlighting their operational mechanisms, applicability, and inherent limitations within the realm of SLAM-based path planning.

Dijkstra's Algorithm

Dijkstra's algorithm, first introduced by Edsger W. Dijkstra in 1956, aims to find the shortest path from a source vertex to all other vertices in a weighted graph. In the context of SLAM-based path planning, Dijkstra's algorithm constructs a graph from the map generated by SLAM, where the vertices represent specific points in the environment, and the edges correspond to navigable paths between them. The weights can signify the cost, distance, or any other metric that represents the "expense" of a particular path. Some important features of this algorithm are:

- Operational Mechanism: Dijkstra's algorithm uses a priority queue to keep track of vertices based on their cumulative distances from the source vertex. It iteratively updates the distances and selects the vertex with the smallest known distance, effectively expanding the known graph. The algorithm is mathematically expressed and executed iteratively, evaluating the cost: $d\left(v\right) = \min\{d\left(v\right), d\left(u\right) + w\left(u,v\right)\}$, where $d\left(v\right)$ is the current shortest distance from the source to vertex v, $d(u)$ is the distance from the source to a neighbor vertex u, and $w(u,v)$ is the edge weight between u and v.
- Applicability: This algorithm is especially effective in environments where the cost of traversing between nodes is deterministic and known in advance. It has been extensively utilized in static mapping scenarios (Ramesh et al., 2023).
- Limitations: The algorithm's primary drawback in SLAM contexts is its inability to adapt effectively to real-time changes in the environment. The graph would need frequent updates to ensure accurate path planning, adding computational overhead to the system.

A* Algorithm. The A* (A-star) algorithm was introduced in 1968 by Peter Hart, Nils Nilsson, and Bertram Raphael as an extension of Dijkstra's algorithm. It integrates heuristic information into the search, thereby accelerating the path-finding process significantly. Some important features of this algorithm are:

- Operational Mechanism: Similar to Dijkstra's algorithm, A* also constructs a graph from the SLAM-generated map. What distinguishes A* is its use of a heuristic function, which estimates the cost from the current vertex to the goal, guiding the search process to explore more promising paths. The efficiency of the algorithm is encapsulated in its cost function: $f\left(n\right) = g\left(n\right) + h\left(n\right)$, where $g\left(n\right)$ represents the exact cost of the path from the starting point to any vertex n, and $h\left(n\right)$ is the heuristic estimate of the cost from vertex n to the goal. Algorithm 2 is a pseudocode representation of the A* algorithm:
  Applicability: Due to its heuristic-based approach, A* is suitable for a wider range of scenarios, including those where the environment is partially known or subject to changes, making it a popular choice in more dynamic SLAM applications (Khlif et al., 2022).
- Limitations: The performance of A* is heavily dependent on the quality of the heuristic function. A poor heuristic can reduce the algorithm to inefficient or even exhaustive search. Moreover, like Dijkstra's, it, too, requires graph updates for real-time applicability, although to a lesser extent.

Algorithm 2. A* path planning

```
Initialize:
OpenList = {s} // Nodes to be evaluated,s is the start node
CloseList = { } // Nodes already evaluated
G(s)=0 // Cost from start node to itself is zero
H(s)=heuristic_cost(s,g) // Estimated cost from start node s to goal node
F(s)=G(s)+ H(s) // Total estimated cost of path through node s
Loop:
While OpenList is not empty do
n=node in OpenList having lowest F(n) // Node with lowest total estimated cost
if n is goal node(n=g) then
return Reconstruct path // Goal found, reconstruct path from start node to goal
Remove n from OpenList
Add n to CloseList
for each neighbor m of n do
if m is in CloseList then
continue // Skip node already evaluated
tentative_G_cost =G(n)+distance(n,m) // Cost from start node to neighbor m through n
if m is not in OpenList or tentative_G_cost < G(m) then
G(m) =tentative_G_cost // Update cost if lower cost found
H(m) =heuristic_cost(m,g) // Heuristic cost from neighbor m to node g
F(m) = G(m)+ H(m) // Total estimated cost of path through neighbor m
if m is not in OpenList then
Add m to OpenList // Add neighbor to nodes to be evaluated
return null // No path exists
```

The choice between Dijkstra's algorithm and the A* algorithm for path planning often depends on specific requirements and constraints of the application in question. However, A* is generally more popular in many contemporary path planning scenarios for a couple of reasons, such as efficiency and flexibility. A* typically finds the shortest path more quickly than Dijkstra's algorithm due to its heuristic function, which guides the algorithm's search towards the goal. This makes A* more suitable for real-time applications. Meanwhile, the heuristic function in A* can be tailored for particular kinds of problem domains, making it more adaptable.

Sampling-Based Methods

Sampling-based methods, such as Rapidly-Exploring Random Trees (RRT) and Probabilistic Roadmaps (PRM), have been increasingly adopted for their effectiveness in high-dimensional spaces and dynamic environments (Agrawa et al., 2022). These algorithms are particularly well-suited for robotics and autonomous vehicles, and their performance can be substantially improved when complemented by multi-sensor fusion. Both Rapidly-Exploring Random Trees (RRT) and Probabilistic Roadmaps (PRM) are widely used sampling-based methods in path planning and are considered foundational techniques. Each has its own merits and drawbacks and is often selected based on the specific requirements of a given application. Table 7 provides a comparative analysis of RRT and PRM.

The ensuing analysis gives precedence to a more detailed exposition of the RRT (Rapidly-Exploring Random Trees) over PRM (Probabilistic Roadmaps) for several compelling reasons. While both algorithms have been instrumental in solving high-dimensional problems, RRT's efficiency in real-time and dynamic environments, coupled with its adaptability, positions it as a focal point of interest. PRM's pre-computation and storage of a roadmap, though effective for multiple queries, can be cumbersome in environments requiring real-time responsiveness. In contrast, RRT's on-the-fly computation not only caters to dynamic and unpredictable terrains but also aligns with the evolving requisites of autonomous systems' navigation where adaptability and rapid response are paramount. The RRT algorithm mainly explores the robot's configuration space through random sampling, then

Table 7. Comparative analysis of RRT and PRM in path planning

Criteria	RRT	PRM
Environment	Suited for complex and unknown environments	Best in known environments
Real-time Capability	Good; Suitable for real-time applications	Limited, unless roadmap is precomputed
Path Optimality	Generally produces sub-optimal paths	More likely to find optimal paths, especially with post-processing
Computational Overhead	Lower, especially in real-time scenarios	Potentially higher due to roadmap precomputation
Multiple Queries	Not well-suited, as a new tree must be generated each time	Well-suited, as a precomputed roadmap can be reused
Dimensionality	Handles higher-dimensional spaces effectively	Can struggle with higher-dimensional spaces
Preprocessing Required	No	Yes
Dynamic Environments	Better suited due to real-time adaptability	Less suited, unless roadmap can be efficiently updated
Complexity	O(n log n) for basic implementations	Varies depending on nearest-neighbor search and local planning
Application Scenarios	Online planning, dynamic environments, robotics	Motion planning in robotics, computer-aided design, simulations

constructs an acyclic graph based on the sampling points, and finally finds the path from the starting point to the end point. The main features of RRT are:

- Operating mechanism: The core mechanism of Rapid Exploration Random Tree (RRT) is to randomly sample in the robot's configuration space and quickly build a search tree based on these sampling points. The basic steps are:
    1) Start building the search tree with the initial position of the robot as the root node.
    2) Randomly sample a point in the configuration space.
    3) Find the node closest to the point in the tree.
    4) Expand the tree from the nearest node found in the direction of the sampling point. This step is controlled by a predefined step size.
    5) If the expansion encounters no obstacles, add the newly generated node to the tree.
    6) Repeat the above steps until certain termination conditions are met, such as the tree reaching a certain size or finding a path to the goal.

    The algorithm can be represented by the pseudocode shown in Algorithm 3:

- Applicability: The advantage of the RRT algorithm is that it can effectively handle high-dimensional configuration spaces and complex environmental structures. Because it uses random sampling, RRT is not limited by the dimensionality of the configuration space, making it ideal for handling complex, multi-dimensional dynamic environments. RRT is also particularly suitable for application scenarios that need to provide solutions quickly, because it can quickly build solutions within a limited number of iterations.
- Limitations: Although the RRT algorithm has many advantages, it also has some limitations. First, due to its stochastic nature, RRT is not guaranteed to find the optimal path. The resulting path may circumvent obstacles but is not necessarily the shortest or fastest path. Secondly, the exploration efficiency of RRT is unbalanced in open areas and narrow channel areas. In open areas, RRT may over-explore, while in narrow passages or complex areas, more iterations may be required to find a feasible path.

Algorithm 3. Function build RRT

Input: Initial point, goal point, obstacles, max_iterations, step_size
Output: RRT.tree

```
tree.init(initial point)
for i =1 to max_iterations do
//sampleRandomPoint is a function that randomly samples a point in the configuration space
random_point = sampleRandomPoint()

//Finding the node in the tree thatis nearest to the random point
nearest_node = tree.nearestNode(random_point)

//extend is a function that extends the tree towards the random point from the nearest node, within the limits of a predefined step size
New_node = extend(nearest_node, random_point, step_size)

//collision is a function to check if the new node or the path to the new node collides with any obstacles
if not collision(new_node, obstacles) then
//If no collision, adding the new node to the tree
tree.addNode(new_node)
//Checking if the new node is close enough to the goal
if new_node is close to goal_point then
//If goal is reached, returning the tree
return tree
//Returning the built RRT tree after max_iterations if the goal is not reached
return tree
```

Reinforcement Learning-Based Methods

Reinforcement learning (RL) methods have emerged as a promising frontier in SLAM-based path planning (Hafeez et al., 2020). These algorithms utilize a learning agent to interact with the environment and make decisions that maximize a given reward function (Vijayakumar & Rajkumar, 2022). While the field is still nascent, RL-based SLAM has demonstrated exceptional adaptability and efficacy, particularly in uncertain and complex scenarios. Several recent papers have begun to explore the integration of deep reinforcement learning with multi-sensor fusion to further augment system capabilities (Y. Li et al., 2022; Sun et al., 2021; Cvitić et al., 2021). While there are a few DRL methods commonly employed in path planning, Deep Q-Networks (DQN) is one of the most typical representatives.

DQN is an extension of the traditional Q-Learning algorithm, equipped with a deep neural network to approximate the Q-function. Deep Q-Networks (DQNs) are popular in path planning because they can efficiently handle high-dimensional state and action spaces, learning optimal policies even in complex environments. DQNs offer an innovative confluence of Q-Learning and Deep Neural Networks to approximate the action-value function $Q\left(s,a\right)$ .The overarching aim is to find an optimal policy that maximizes the expected sum of discounted rewards.

Let θ be the parameters of the primary Q-network and θ^{-} be the parameters of the target Q-network. The objective function $L\left(\theta\right)$ aims to minimize the temporal difference error and is represented as:

$$L\left(\theta\right) = E[\left(Q\left(s,a;\theta\right) - \left(r + \gamma_{a'} maxQ\left(s',a';\theta^{-}\right)\right)\right)_{2}] \quad (4)$$

where γ denotes the discount factor.

The algorithmic steps are as follows:

1. Initialization Phase: Initialize the primary Q-network with randomly generated weights θ , and synchronize the target Q-network weights θ^{-} with these initial values.
2. Action Selection via ϵ-Greedy Strategy: An action a is chosen from the current state s by employing an ϵ -greedy policy predicated on the Q-network, represented as:

$$a = \begin{cases} random\, action & with\, probability \epsilon \\ \arg max_{a} Q\left(s, a; \theta\right) & with\, probability 1 - \epsilon \end{cases} \tag{5}$$

3. Environment Interactivity: Implement the chosen action a , consequently observing the immediate reward r and the resultant state s'
4. Experience Replay Mechanism: Accumulate the tuple $\left(s, a, r, s'\right)$ in the experience replay buffer D .
5. Stochastic Mini-batch Sampling: A mini-batch is randomly culled from the replay buffer D denoted as $\left(s, a; r, s'\right)$.
6. Target Q-Value Computation: For each sampled tuple, the target Q-value y_{i} is computed using the following equation:

$$y_{i} = r_{i} + \gamma_{i} max Q\left(s_{i'}, a'; \theta^{-}\right) \tag{6}$$

7. Backpropagation and Weight Update: Utilize stochastic gradient descent or variants thereof to minimize the loss $\mathcal{L}\left(\theta\right)$ and update θ .
8. Target Network Synchronization: At intervals of c steps, the target Q-network parameters θ^{-} are updated to align with the current Q-network parameters θ .
9. Iterative Refinement: Repeat steps 2–8 until a specified termination criterion is fulfilled.

By amalgamating the robustness of Q-Learning with the function approximation capabilities of deep neural networks, DQNs adeptly handle high-dimensional state and action spaces, providing a comprehensive framework for complex reinforcement learning scenarios. Algorithm 4 shows the pseudocode for the DQN algorithm:

Hybrid Methods

Hybrid methods amalgamate features from multiple categories of algorithms to spawn more versatile and adaptive solutions. For instance, Walid et al. (2023) proposed a method that combines elements of graph-based and sampling-based techniques to efficiently navigate across both structured and unstructured environments. The combination of deep learning and genetic algorithms can be used in path planning. Deep learning models can be trained to understand complex environments and make initial path predictions. These initial predictions could then be optimized using a genetic algorithm (Ilyas et al., 2022), which is excellent for searching through a large solution space to find an optimal or near-optimal solution.

Algorithm 4. Deep Q-network(DQN)

```
Initialize:
 Q-network with parameters ¸  // Main Q-network
 Target Q-network with parameters ¸ ^- ⟵ ¸  // Target Q-network for stability
Experience replay buffer D // Buffer to store experience tuples (s,a,r,s')
 Initialize µ (exploration rate) // Initial value of exploration rate
for episode = 1 to M do // M: total number of episodes
Initialize state s
for t=1 to T do // T: maximum steps per episode
// Epsilon-Greedy Action Selection
if random() < µ then
Choose a random action a
else
 Choose action a=argmax_a Q(s,a; ¸ )
end if
// Perform Actiono
Execute action a,obeserve reward r and next state s'
// Store Transition
Store (s,a,r,s') in D
// Experience Replay
Sample a mini-batch of transitions (s_j,a_j,r_j,s_j') from D
// Compute Q-Learning Targets
for each sampled transition j do
if s_j' is terminal then
y_j= r_j
else
 y_j= r_j+ ³  max_a' Q(s_j',a', ¸ ^-) // ³ : discount factor
end if
end for
// Update Q-network by minimizing the loss
 Perform a gradient descent step on (y_j-Q(s_j,a_j; ¸ ))^2 with respect to ¸
// Update Target Network
if t mod C==0 then //C:frequency of target network updates
 Update target network: ¸ ^- ⟵ ¸
end if
// Update State
 s ⟵ s'
end for // End of inner loop
end for //End of outer loop
```

CONCLUSION

This review systematically examines the integration and nuances of multi-sensor fusion within the specialized arena of SLAM-based path planning. We have delineated an array of methodologies and technologies, shedding light on the pivotal role that synergistic interactions between sensor fusion and SLAM play in enhancing the efficacy of autonomous navigation systems.

However, amidst the advancements, clear challenges and research voids have been identified. A pressing concern is the computational intensity associated with real-time applications. Current methodologies, though refined, still grapple with latency, underscoring an urgent need for further innovations to enhance computational efficiencies. The question of algorithm robustness in varied and dynamic environments remains a significant research frontier. Existing solutions have exhibited limitations in offering consistent, reliable performance across a spectrum of environmental conditions. This accentuates a research gap that, when addressed, could substantially bolster the performance metrics of these systems.

The intricate interplay between SLAM, navigation, and path planning is a domain where deeper insights are necessary. An exhaustive exploration into their interconnected dynamics could unveil optimization strategies to enhance the integrative output and efficiency of autonomous systems. Future trajectories are likely to be characterized by the emergence of adaptive algorithms imbued with machine learning capabilities to offer responsive adjustments to environmental variables. The evolution of these adaptive systems is not just a technological imperative but a stepping stone to attaining unprecedented levels of accuracy and efficiency. In this multifaceted domain, collaboration and interdisciplinary integration emerge as vital. Insights and innovations from computer vision, robotics, and machine learning are instrumental in sculpting the forthcoming phase of development in multi-sensor fusion for SLAM-based path planning.

In summary, while this review presents a comprehensive overview, it also serves as a precursor to future research. The identified gaps and future directions are not mere observations but explicit calls to action for the academic and research community. Each presents an opportunity for innovative pursuits, poised to advance the frontiers of knowledge and application in this evolving field. We anticipate a future where these identified gaps transform into nodes of breakthrough, heralding an era of enhanced, efficient, and precise SLAM-based path planning.

AUTHOR NOTE

Yiyi Cai is a doctoral student in Electronics and Communications at South China University of Technology. Her research interests include SLAM and multi-sensor fusion.

Tuanfa Qin is a professor and doctoral supervisor at Guangxi University. His research interests include the theory and technology of wireless multimedia communication.

Yang Ou is a graduate student in Electronics and Communications at Guangxi University. His research interests include robot multi-data fusion.

Rui Wei is a graduate student in Electronics and Communications at Guangxi University. His research interests include the Internet of Vehicles.

The authors would like to thank the editor and anonymous reviewers for their contributions towards improving the quality of this paper. Data used to support the findings of this study are included within the paper. We have no conflicts of interest to disclose. Our work was supported by a grant from the NSFs of China (Grant no. 62361003).

REFERENCES

Agostinho, L. R., Ricardo, N. M., Pereira, M. I., Hiolle, A., & Pinto, A. M. (2022). A practical survey on visual odometry for autonomous driving in challenging scenarios and conditions. *IEEE Access : Practical Innovations, Open Solutions*, *10*, 72182–72205. doi:10.1109/ACCESS.2022.3188990

Agrawal, R., Singh, B., & Kumar, R. (2022). Classical approaches for mobile robot path planning: A review. *2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS),* 400–405. doi:10.1109/ICCCIS56430.2022.10037620

Ahmed, Z., Ayaz, M., Hijji, M. A., Abbas, M. Z., & Rahim, A. (2022). AUV-based efficient data collection scheme for underwater linear sensor networks. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–19. doi:10.4018/IJSWIS.299858

Alatise, M. B., & Hancke, G. P. (2020). A review on challenges of autonomous mobile robot and sensor fusion methods. *IEEE Access : Practical Innovations, Open Solutions*, *8*, 39830–39846. doi:10.1109/ACCESS.2020.2975643

An, X., Wu, C., Lin, Y., Lin, M., Yoshinaga, T., & Ji, Y. (2023). Multi-robot systems and cooperative object transport: Communications, platforms, and challenges. *IEEE Open Journal of the Computer Society*, *4*, 23–36. doi:10.1109/OJCS.2023.3238324

Andreas, P., Plageras, K. E., Psannis, C. S., Wang, H. X., & Gupta, B. B. (2018). Efficient IoT-based sensor BIG Data collection–processing and analysis in smart buildings. *Future Generation Computer Systems*, *82*, 349–357. doi:10.1016/j.future.2017.09.082

Asaad, S. M., & Maghdid, H. S. (2021). A comprehensive review of indoor/outdoor localization solutions in IoT era: Research challenges and future perspectives. *Computer Networks*, *212*, 109041. doi:10.1016/j.comnet.2022.109041

Aslam, M. S., Aziz, M. I., Naveed, K., & Zaman, U. K. (2020). An RPLiDAR based SLAM equipped with IMU for autonomous navigation of wheeled mobile robot. *2020 IEEE 23rd International Multitopic Conference (INMIC),* 1–5. doi:10.1109/INMIC50486.2020.9318133

Bosse, M., & Zlot, R. (2009). Continuous 3D scan-matching with a spinning 2D laser. *2009 IEEE International Conference on Robotics and Automation*, 4312–4319. doi:10.1109/ROBOT.2009.5152851

Burdziakowski, P., & Bobkowska, K. (2021). UAV photogrammetry under poor lighting conditions—Accuracy considerations. *Sensors (Basel)*, *21*(10), 3531. doi:10.3390/s21103531 PMID:34069500

Burri, M., Nikolic, J., Gohl, P., Thomas, S., Joern, R., Sammy, O., Markus, W. A., & Roland, S. (2016). The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, *35*(10), 1157–1163. doi:10.1177/0278364915620033

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., & Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, *32*(6), 1309–1332. doi:10.1109/TRO.2016.2624754

Chaudhari, T., Jha, M., Bangal, R., & Chincholkar, G. (2019). Path planning and controlling of omni-directional robot using Cartesian odometry and PID algorithm. *2019 International Conference on Computing, Power and Communication Technologies (GUCON),* 63–68.

Chen, Z., Qi, Y., Zhong, S., Feng, D., Chen, Q., & Chen, H. (2022). SCL-SLAM: A scan context-enabled LiDAR SLAM using factor graph-based optimization. *2022 IEEE International Conference on Unmanned Systems (ICUS),* 1264–1269. doi:10.1109/ICUS55513.2022.9987005

Chghaf, M., Rodríguez, S., & Ouardi, A. E. (2022). Camera, LiDAR and multi-modal SLAM systems for autonomous ground vehicles: *A survey. Journal of Intelligent & Robotic Systems*, *105*(1), 105. doi:10.1007/s10846-022-01582-8

Chui, K. T., Kochhar, T. S., Chhabra, A., Singh, S. K., Singh, D., Peraković, D., Ammar, A., & Arya, V. (2022). Traffic accident prevention in low visibility conditions using vanets cloud environment. *International Journal of Cloud Applications and Computing*, *12*(1), 1–21. doi:10.4018/IJCAC.313572

Cvitić, I., Peraković, D., Periša, M., & Brij, G. (2021). Ensemble machine learning approach for classification of IoT devices in smart home. *International Journal of Machine Learning and Cybernetics*, *12*(11), 3179–3202. doi:10.1007/s13042-020-01241-0

Darapaneni, N., Raj, P., Paduri, A. R., Anand, E., Rajarathinam, K., Eapen, P. T., & Krishnamurthy, S. (2021). Autonomous car driving using deep learning. *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC),* 29–33. doi:10.1109/ICSCCC51823.2021.9478090

Dong, J., Nelson, E., Indelman, V., Michael, N., & Dellaert, F. (2015). Distributed real-time cooperative localization and mapping using an uncertainty-aware expectation maximization approach. *2015 IEEE International Conference on Robotics and Automation (ICRA),* 5807–5814. doi:10.1109/ICRA.2015.7140012

Elghazaly, G., Frank, R., Harvey, S., & Safko, S. (2023). High-definition maps: Comprehensive survey, challenges, and future perspectives. *IEEE Open Journal of Intelligent Transportation Systems*, *4*, 527–550. doi:10.1109/OJITS.2023.3295502

Engel, J., Sturm, J., & Cremers, D. (2013). Semi-dense visual odometry for a monocular camera. *Proceedings of the IEEE international conference on computer vision,* 1449–1456. doi:10.1109/ICCV.2013.183

Feng, D., Haase, S. C., Rosenbaum, L., Hertlein, H., Gläser, C., Timm, F., Wiesbeck, W., & Dietmayer, K. (2021). Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, *22*(3), 1341–1360. doi:10.1109/TITS.2020.2972974

Forster, C., Zhang, Z., Gassner, M., Werlberger, M., & Scaramuzza, D. (2017). SVO: Semidirect Visual Odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, *33*(2), 249–265. doi:10.1109/TRO.2016.2623335

Fu, B., Han, F. Z., Wang, Y., Jiao, Y., Ding, X. Q., Tan, Q., Chen, L., Wang, M., & Xiong, R. (2021). High-precision multicamera-assisted camera-IMU calibration: Theory and method. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–17. doi:10.1109/TIM.2021.3051726

Fung, M. L., Chen, M. Z. Q., & Chen, Y. H. (2017). Sensor fusion: A review of methods and applications. *2017 29th Chinese Control and Decision Conference (CCDC),* 3853–3860. doi:10.1109/CCDC.2017.7979175

García, S., López, M. E., Barea, R., Bergasa, L. M., Gómez, A., & Molinos, E. J. (2016). Indoor SLAM for micro aerial vehicles control using monocular camera and sensor fusion. *2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC),* 205–210. doi:10.1109/ICARSC.2016.46

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, *32*(11), 1231–1237. doi:10.1177/0278364913491297

George, S., & George, V. (2004). Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, *59*(1–2), 85–101. doi:10.1016/j.isprsjprs.2004.05.004

Gollan, B., Haslgruebler, M., Ferscha, A., & Heftberger, J. (2018). Making sense: Experiences with multi-sensor fusion in industrial assistance systems. *5th International Conference on Physiological Computing Systems.* doi:10.5220/0007227600640074

Gong, S., Shi, C., Zhang, H., Lu, H., Zeng, Z., & Chen, X. (2023). RSS-LIWOM: Rotating solid-state LiDAR for robust LiDAR-Inertial-Wheel odometry and mapping. *Remote Sensing (Basel)*, *15*(16), 4040. doi:10.3390/rs15164040

Guebli, W., & Belkhir, A. (2021). Inconsistency detection-based LOD in smart homes. *International Journal on Semantic Web and Information Systems*, *17*(4), 56–75. doi:10.4018/IJSWIS.2021100104

Hafeez, F., Sheikh, U. U., Alkhaldi, N., Garni, H. A., Arfeen, Z., & Khalid, S. A. (2020). Insights and strategies for an autonomous vehicle with a sensor fusion innovation: A fictional outlook. *IEEE Access : Practical Innovations, Open Solutions*, *8*, 135162–135175. doi:10.1109/ACCESS.2020.3010940

Harun, M. H., Abdullah, S. S., Aras, M., & Bahar, M. B. (2022). Sensor fusion technology for unmanned autonomous vehicles (UAV): A review of methods and applications. *2022 IEEE 9th International Conference on Underwater System Technology: Theory and Applications (USYS),* 1–8.

Henry, P., Krainin, M., Herbst, E., Ren, X., & Fox, D. (2014). RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In O. Khatib, V. Kumar, & G. Sukhatme (Eds.), *Experimental Robotics: Springer Tracts in Advanced Robotics* (p. 79). Springer. doi:10.1007/978-3-642-28572-1_33

Hesch, J. A., Kottas, D. G., Bowman, S. L., & Roumeliotis, S. I. (2012). Observability-constrained vision-aided inertial navigation. *University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab. Tech. Rep*, *1*, 6.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, *34*(3), 189–206. doi:10.1007/s10514-012-9321-0

Hu, X., Chen, L., Tang, B., Cao, D., & He, H. (2018). Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles. *Mechanical Systems and Signal Processing*, *100*, 482–500. doi:10.1016/j.ymssp.2017.07.019

Ilyas, Q. M., Ahmad, M., Rauf, S., & Irfan, D. (2022). RDF query path optimization using hybrid genetic algorithms: Semantic web vs. data-intensive cloud computing. *International Journal of Cloud Applications and Computing*, *12*(1), 1–16. doi:10.4018/IJCAC.2022010101

Iqbal, S., Hussain, I., Sharif, Z., Qureshi, K. H., & Jabeen, J. (2021). Reliable and energy-efficient routing scheme for underwater wireless sensor networks (UWSNs). *International Journal of Cloud Applications and Computing*, *11*(4), 42–58. doi:10.4018/IJCAC.2021100103

Jiang, X., Yu, H., Hoy, M., & Dauwels, J. (2019). Robust linear-complexity approach to full SLAM problems: Stochastic variational Bayes inference. *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall),* 1–5. doi:10.1109/VTCFall.2019.8891361

Joachim, C., Thomas, R., & Tobias, K. (2016). An evidential approach to SLAM, path planning, and active exploration. *International Journal of Approximate Reasoning*, *73*, 1–26. doi:10.1016/j.ijar.2016.02.003

Julier, S. J., & Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. *Proc. AeroSense: The 11th Int. Symp. on Aerospace/Defense Sensing, Simulation and Controls*, 182–193.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, *82D*(1), 35–45. doi:10.1115/1.3662552

Khan, M. S. A., Hussian, D., Ali, Y., Rehman, F. U., Aqeel, A. B., & Khan, U. S. (2021). Multi-sensor SLAM for efficient navigation of a mobile robot. *2021 4th International Conference on Computing & Information Sciences (ICCIS),* 1–5. doi:10.1109/ICCIS54243.2021.9676374

Khlif, N., Khraief, N., & Belghith, S. (2022). *Reinforcement learning for mobile robot navigation: An overview. 2022 IEEE Information Technologies & Smart Industrial Systems*. doi:10.1109/ITSIS56166.2022.10118362

Kostavelis, I., Boukas, E., Nalpantidis, L., & Gasteratos, A. (2016). Stereo-based visual odometry for autonomous robot navigation. *International Journal of Advanced Robotic Systems*, *13*(1), 21. Advance online publication. doi:10.5772/62099

Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., & Burgard, W. (2011). G2o: A general framework for graph optimization. *2011 IEEE International Conference on Robotics and Automation,* 3607–3613. doi:10.1109/ICRA.2011.5979949

Kümmerle, R., Steder, B., Dornhege, C., Ruhnke, M., Grisetti, G., Stachniss, C., & Kleiner, A. (2009). On measuring the accuracy of SLAM algorithms. *Autonomous Robots*, *27*(4), 387–407. doi:10.1007/s10514-009-9155-6

Li, D., Deng, L., Gupta, B. B., Wang, H., & Choi, C. (2019). A novel CNN based security guaranteed image watermarking generation scenario for smart city applications. *Information Sciences*, *479*, 432–447. doi:10.1016/j.ins.2018.02.060

Li, L., Yang, M., Bing, W., & Wang, C. (2017). *An overview on sensor map based localization for automated driving. 2017 Joint Urban Remote Sensing Event*. doi:10.1109/JURSE.2017.7924575

Li, N., Ho, C. P., Xue, J., Lim, L. W., Chen, G., Fu, Y. H., & Lee, L. Y. T. (2022). A progress review on solid-state LiDAR and nanophotonics-based LiDAR sensors. *Laser & Photonics Reviews*, *16*(11), 2100511. doi:10.1002/lpor.202100511

Li, R., Liu, J., Zhang, L., & Hang, Y. (2014). *LIDAR/MEMS IMU integrated navigation (SLAM) method for a small UAV in indoor environments. 2014 DGON Inertial Sensors and Systems*. doi:10.1109/InertialSensors.2014.7049479

Li, T., Sun, S., & Duan, J. (2010). Monte Carlo localization for mobile robot using adaptive particle merging and splitting technique. *The 2010 IEEE International Conference on Information and Automation,* 1913–1918 doi:10.1109/ICINFA.2010.5512017

Li, X., Song, R., Fan, J., Liu, M., & Wang, F. (2023). Development and testing of advanced driver assistance systems through scenario-based systems engineering. *IEEE Transactions on Intelligent Vehicles*, *8*(8), 3968–3973. doi:10.1109/TIV.2023.3297168

Li, Y., Chen, R. Z., Niu, X. J., Zhuang, Y., Gao, Z. Z., Hu, X., & El, S. N. (2022). Inertial sensing meets machine learning: Opportunity or challenge? *IEEE Transactions on Intelligent Transportation Systems*, *23*(8), 9995–10011. doi:10.1109/TITS.2021.3097385

Liang, X. Y., Bai, X., & Zhang, S. H. (2020, April 17). Review of three-dimensional environment information perception and reconstruction methods for mobile robot based on multi-sensor fusion. *Proc. SPIE 11455, Sixth Symposium on Novel Optoelectronic Detection Technology and Applications*, 114557V. doi:10.1117/12.2565371

Liu, L. X., Wang, X., Yang, X., Liu, H. J., Li, J. P., & Wang, P. F. (2023). Path planning techniques for mobile robots: Review and prospect. *Expert Systems with Applications*, *227*, 120254. doi:10.1016/j.eswa.2023.120254

Lv, L., Wu, Z., Zhang, L., Gupta, B. B., & Tian, Z. (2022). An edge-AI based forecasting approach for improving smart microgrid efficiency. *IEEE Transactions on Industrial Informatics*, *18*(11), 7946–7954. doi:10.1109/TII.2022.3163137

Macario, B. A., Michel, M., Moline, Y., Corre, G., & Carrel, F. A. (2022). Comprehensive survey of visual SLAM algorithms. *Robotics (Basel, Switzerland)*, *11*(1), 24. doi:10.3390/robotics11010024

Martinelli, A. (2012). Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Transactions on Robotics*, *28*(1), 44–60. doi:10.1109/TRO.2011.2160468

Maybeck, P. S. (1979). *Stochastic Models, Estimation, and Control*. Academic Press., doi:10.1109/TAC.1983.1103336

Meyer, L., Strobl, K. H., & Triebel, R. (2022). The probabilistic robot kinematics model and its application to sensor fusion. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* 3263–3270. doi:10.1109/IROS47612.2022.9981399

Mohammed, A. S., Amamou, A., Ayevide, F. K., Kelouwani, S., Agbossou, K., & Zioui, N. (2020). The perception system of intelligent ground vehicles in all weather conditions: A systematic literature review. *Sensors (Basel)*, *20*(22), 6532. doi:10.3390/s20226532 PMID:33203155

Mohanan, M. G., & Ambuja, S. (2018). A survey of robotic motion planning in dynamic environments. *Robotics and Autonomous Systems*, *100*, 171–185. doi:10.1016/j.robot.2017.10.011

Muhammad, A., Ali, M. A. H., & Shanono, I. H. (2021). A review: On intelligent mobile robot path planning techniques. *2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE),* 53–58, doi:10.1109/ISCAIE51753.2021.9431819

Mur-Artal, R., Montiel, J. M. M., & Tardós, J. D. (2015). ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, *31*(5), 1147–1163. doi:10.1109/TRO.2015.2463671

Narasimhappa, M., Mahindrakar, A. D., Guizilini, V. C., Terra, M. H., & Sabat, S. L. (2018). An improved Sage Husa adaptive robust Kalman filter for de-noising the MEMS IMU drift signal. *2018 Indian Control Conference (ICC),* 229–234. doi:10.1109/INDIANCC.2018.8307983

Narasimhappa, M., Mahindrakar, A. D., Guizilini, V. C., Terra, M. H., & Sabat, S. L. (2020). MEMS-based IMU drift minimization: Sage Husa adaptive robust Kalman filtering. *IEEE Sensors Journal*, *20*(1), 250–260. doi:10.1109/JSEN.2019.2941273

Narjes, D., & Asghar, G. (2019). Variational Bayesian adaptive Kalman filter for asynchronous multirate multi-sensor integrated navigation system. *Ocean Engineering*, *174*, 108–116. doi:10.1016/j.oceaneng.2019.01.012

Ouyang, M., Cao, Z., Guan, P., Li, Z., Zhou, C., & Yu, J. (2021). Visual-gyroscope-wheel odometry with ground plane constraint for indoor robots in dynamic environment. *IEEE Sensors Letters*, *5*(3), 1–4. doi:10.1109/LSENS.2021.3057088 PMID:36789370

Ramesh, A., Suseendhar, P., & Venugopal, E., & S. P. (2023) An overview of navigation algorithms for unmanned aerial vehicle. *2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT),* 724–727. doi:10.1109/IDCIoT56793.2023.10053496

Saleem, H., Malekian, R., & Munir, H. (2023). Neural network-based recent research developments in SLAM for autonomous ground vehicles: A review. *IEEE Sensors Journal*, *23*(13), 13829–13858. doi:10.1109/JSEN.2023.3273913

Saraf, A., Moon, S., & Madotto, A. (2023). A survey of datasets, applications, and models for IMU sensor signals. *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW),* 1–5. doi:10.1109/ICASSPW59220.2023.10193365

Sharma, P., Raj, B., & Gill, S. S. (2022). Spintronics based non-volatile MRAM for intelligent systems: Memory for intelligent systems design. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–16. doi:10.4018/IJSWIS.310056

Shi, P., Zhu, Z., Sun, S., Zhao, X., & Tan, M. (2023). Invariant extended Kalman filtering for tightly coupled LiDAR-inertial odometry and mapping. *IEEE/ASME Transactions on Mechatronics*, *28*(4), 2213–2224. doi:10.1109/TMECH.2022.3233363

Smith, R., Self, M., & Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, 167–193.

Sturm, J., Burgard, W., & Cremers, D. (2012). Evaluating egomotion and structure-from-motion approaches using the TUM RGB-D benchmark. *Proc. of the Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RJS International Conference on Intelligent Robot Systems (IROS)*, 13.

Sun, H., Zhang, W., Yu, R., & Zhang, Y. (2021). Motion planning for mobile robots—Focusing on Deep Reinforcement Learning: A systematic review. *IEEE Access : Practical Innovations, Open Solutions*, *9*, 69061–69081. doi:10.1109/ACCESS.2021.3076530

Takamura, T., Shimizu, S., Murakami, R., Carfi, A., & Mastrogiovanni, F. (2021). A visual odometry for wide angle Fovea sensor SLAM. *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society,* 1–6. doi:10.1109/IECON48115.2021.9589175

Tang, J., Folkesson, J., & Jensfelt, P. (2018). Geometric correspondence network for camera motion estimatio. *IEEE Robotics and Automation Letters*, *3*(2), 1010–1017. doi:10.1109/LRA.2018.2794624

Tang, J., Zhang, X., Zou, Y., Li, Y., & Du, G. (2023). A high-precision LiDAR-inertial odometry via Kalman Filter and Factor Graph Optimization. *IEEE Sensors Journal*, *23*(11), 11218–11231. doi:10.1109/JSEN.2023.3260636

Tang, S., Zhu, Q., Chen, W., Darwish, W., Wu, B., Hu, H., & Chen, M. (2016). Enhanced RGB-D mapping method for detailed 3D indoor and outdoor modeling. *Sensors (Basel)*, *16*(10), 1589. doi:10.3390/s16101589 PMID:27690028

Tateno, K., Tombari, F., Laina, I., & Navab, N. (2017). CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 6565–6574. https://doi.org/ doi:10.1109/CVPR.2017.695

Teng, S., Chen, G., Liu, Z., Cheng, L., & Sun, X. (2021). Multi-sensor and decision-level fusion-based structural damage detection using a one-dimensional convolutional neural network. *Sensors (Basel)*, *21*(12), 3950. doi:10.3390/s21123950 PMID:34201143

Thrun, S., & Montemerlo, M. (2006). The Graph SLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, *25*(5-6), 403–429. doi:10.1177/0278364906065387

Tian, D., Zou, F., Xu, F., & Di, P. (2017). 3D laser odometry for a mobile robot platform. *2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER),* 423–426. doi:10.1109/CYBER.2017.8446268

Tonga, P. A., Said Ameen, Z., Mubarak, A. S., & Al-Turjman, F. (2022). A review on on device privacy and machine learning training. *2022 International Conference on Artificial Intelligence in Everything (AIE),* 679–684. doi:10.1109/AIE57029.2022.00133

Vasileios, A., Memos, K. E., Psannis, Y. I., Byung, G. K., & Gupta, B. B. (2018). An efficient algorithm for media-based surveillance system (EAMSuS) in IoT smart city framework. *Future Generation Computer Systems*, *83*, 619–628. doi:10.1016/j.future.2017.04.039

Vijayakumar, P., & Rajkumar, S. C. (2022). Deep reinforcement learning-based pedestrian and independent vehicle safety fortification using intelligent perception. *International Journal of Software Science and Computational Intelligence*, *14*(1), 1–33. doi:10.4018/IJSSCI.291712

Vijayakumar, P., Rajkumar, S. C., & Deborah, L. J. (2022). Passive-awake energy conscious power consumption in smart electric vehicles using cluster type cloud communication. *International Journal of Cloud Applications and Computing*, *12*(1), 1–14. doi:10.4018/IJCAC.297108

Vishak, P., Sudheesh, P., & Jayakumar, M. (2017). A survey on nonlinear applications of modified particle filter. *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET),* 1059–1063. doi:10.1109/WiSPNET.2017.8299924

Walid, J., Ihssane, B., & Nabil, E. A. (2023). Autonomous vehicles motion planning techniques: Recent advances and future trends. *AIP Conference Proceedings*, *2814*(1), 040021. doi:10.1063/5.0148620

Wang, S., Clark, R., Wen, H., & Trigoni, N. (2017). DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks. *2017 IEEE International Conference on Robotics and Automation (ICRA),* 2043–2050. doi:10.1109/ICRA.2017.7989236

Wang, S., Wu, Z., & Zhang, W. (2019). An overview of SLAM. In Y. Jia, J. Du, & W. Zhang (Eds.), *Proceedings of 2018 Chinese Intelligent Systems Conference. Lecture Notes in Electrical Engineering* (pp. 528). Springer. doi:10.1007/978-981-13-2288-4_64

Wang, X., Li, K., & Chehri, A. (2023). Multi-sensor fusion technology for 3D object detection in autonomous driving: A review. *IEEE Transactions on Intelligent Transportation Systems.* https://api.semanticscholar.org/CorpusID:263179557

Wu, J., Gao, J., Yi, J., Liu, P., & Xu, C. (2022). Environment perception technology for intelligent robots in complex environments: A review. *2022 7th International Conference on Communication, Image and Signal Processing (CCISP),* 479–485.

Xiang, C., Feng, C., Xie, X., Shi, B., Lu, H., Lv, Y., Yang, M., & Niu, Z. (2023). Multi-sensor fusion and cooperative perception for autonomous driving: A review. *IEEE Intelligent Transportation Systems Magazine*, *15*(5), 36–58. doi:10.1109/MITS.2023.3283864

Xu, X., Zhang, L., Yang, J., Cao, C., Wang, W., Ran, Y., Tan, Z., & Luo, M. (2022). A review of multi-sensor fusion SLAM systems based on 3D LIDAR. *Remote Sensing (Basel)*, *14*(12), 2835. doi:10.3390/rs14122835

Yang, C. (2022). A path-clustering driving travel-route excavation. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–16. doi:10.4018/IJSWIS.306750

Yang, F., Shi, L., & Zheng, L. (2021). Event-triggered globally sequential fusion estimation for clustered wireless sensor networks with variational Bayesian. *IEEE Sensors Journal*, *21*(18), 20273–20282. doi:10.1109/JSEN.2021.3092888

Yang, S., Yi, X., Wang, Z., Wang, Y., & Yang, X. (2015). Visual SLAM using multiple RGB-D cameras. *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO),* 1389–1395. doi:10.1109/ROBIO.2015.7418965

Yeong, D. J., Barry, J., & Walsh, J. (2020). A review of multi-sensor fusion system for large heavy vehicles off road in industrial environments. *2020 31st Irish Signals and Systems Conference (ISSC),* 1–6. doi:10.1109/ISSC49989.2020.9180186

Yeong, D. J., Velasco-Hernandez, G., Barry, J., & Walsh, J. (2021). Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors (Basel)*, *21*(6), 2140. doi:10.3390/s21062140 PMID:33803889

Yin, J., Zhang, Y., & Li, X. (2020). Added the odometry optimized SLAM loop closure detection system. *2020 5th International Conference on Control, Robotics and Cybernetics (CRC),* 216–220. doi:10.1109/CRC51253.2020.9253497

Yu, J. L., Su, Y. C., & Liao, Y. F. (2020). The path planning of mobile robot by neural networks and hierarchical reinforcement learning. *Frontiers in Neurorobotics*, *14*, 1662–5218. doi:10.3389/fnbot.2020.00063 PMID:33132890

Yu, L., Qin, J., Wang, S., Wang, Y., & Wang, S. (2023). A tightly coupled feature-based visual-inertial odometry with stereo cameras. *IEEE Transactions on Industrial Electronics*, *70*(4), 3944–3954. doi:10.1109/TIE.2022.3176304

Yu, L., Yang, E., & Yang, B. (2016). AFE-ORB-SLAM: Robust monocular VSLAM based on adaptive FAST threshold and image enhancement for complex lighting environments. *Journal of Intelligent & Robotic Systems*, *105*(2), 26. doi:10.1007/s10846-022-01645-w

Yu, L., Yang, E., & Yang, B. (2022). AFE-ORB-SLAM: Robust Monocular VSLAM Based on Adaptive FAST Threshold and Image Enhancement for Complex Lighting Environments. *Journal of Intelligent & Robotic Systems*, *105*(2), 105. doi:10.1007/s10846-022-01645-w

Yu, S. L., Tsou, D. W., & Juang, J. C. (2016). Comparison of odometry techniques for vehicle navigation. *2016 International Automatic Control Conference (CACS),* 19-24. doi:10.1109/CACS.2016.7973877

Yu, X., Zhou, B., Chang, Z., Qian, K., & Fang, F. (2022). MMDF: Multi-Modal Deep Feature based place recognition of mobile robots with applications on cross-scene navigation. *IEEE Robotics and Automation Letters*, *7*(3), 6742–6749. doi:10.1109/LRA.2022.3176731

Zang, S., Ding, M., Smith, D., Tyler, P., Rakotoarivelo, T., & Kaafar, M. A. (2019). The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car. *IEEE Vehicular Technology Magazine*, *14*(2), 103–111. doi:10.1109/MVT.2019.2892497

Zhang, J., Yue, H., Wu, X., & Chen, W. (2019). A brief review of Bayesian belief network. *2019 Chinese Control And Decision Conference (CCDC),* 3910–3914. doi:10.1109/CCDC.2019.8832649

Zhang, L., Meng, X., & Chen, Y. (2009). Unscented transform for SLAM using Gaussian mixture model with particle filter. *2009 International Conference on Electronic Computer Technology,* 12–17. doi:10.1109/ICECT.2009.98

Zhao, M., Xin, G., Le, S., Qin, B. X., Shi, X. S., Gim, H., & Sun, G. H. (2021). A general framework for lifelong localization and mapping in changing environment. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* 3305–3312. doi:10.1109/IROS51168.2021.9635985

Zhao, X., Sun, P., Xu, Z., Min, H., & Yu, H. (2020). Fusion of 3D LIDAR and camera data for object detection in autonomous vehicle applications. *IEEE Sensors Journal*, *20*(9), 4901–4913. doi:10.1109/JSEN.2020.2966034

Zhao, Y., Xiong, Z., Zhou, S., Wang, J., Zhang, L., & Campoy, P. (2022). Perception-aware planning for active SLAM in dynamic environments. *Remote Sensing (Basel)*, *14*(11), 2584. doi:10.3390/rs14112584

Zhu, Y., Xue, B., Zheng, L., Huang, H., Liu, M., & Fan, R. (2019). Real-time, environmentally-robust 3D LiDAR localization. *2019 IEEE International Conference on Imaging Systems and Techniques (IST),* 1–6. doi:10.1109/IST48021.2019.9010305

APPENDIX

Summary of symbols and acronyms

Symbol/Acronym	Meaning/Explanation
SLAM	Simultaneous Localization and Mapping
LiDAR	Light Detection and Ranging
AD	Autonomous Driving
IMU	Inertial Measurement Units
RGB-D	Red Green Blue-Depth
3D	Three Dimensions
GPS	Global Positioning System
VIO	Visual Inertial Odometry
LIO	Lidar Inertial Odometry
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
CNNs	Convolutional Neural Networks
RNNs	Recurrent Neural Networks
AVs	Autonomous Vehicles
AODV	Ad hoc On-Demand Distance Vector
VANETS	Vehicular Ad-Hoc Networks
AR	Augmented Reality
VR	Virtual Reality
UAVs	Unmanned Aerial Vehicles
IoT	Internet of Things
RRT	Rapidly-Exploring Random Trees
RPM	Probabilistic Roadmaps
RL	Reinforcement Learning
DQN	Deep Q-Networks

Yiyi Cai, Ph.D. student of Electronics and communications. Studied in South China University of Technology. Her research interests include SLAM and multi-sensor fusion.

Tuanfa Qin, Professor, Doctoral supervisor. Worked in Guangxi University. His research interests include theory and technology of wireless multimedia communication.

Ou Yang, master student of electronics and communications, studied in Guangxi University. Research interest is robot multi-data fusion.

Wei Rui, master student of electronics and communications, studied in Guangxi University. Research interest is internet of vehicles.