Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 21, No. 1, 2026

A Visual SLAM-Based Framework for Indoor Hovering Robots in 3D Rescue Missions with Dynamic Obstacle Avoidance

Dania Madeleina Otoka Niabanga [1, *], Amir Ali Mokhtarzadeh[1], Ervan Novt Tanggono[2], Gloire Orianne Kifouria[1], Josepha Fansi Nguietchuan[1], Israel Ntumba Mbala[3]

[1] Faculty of Computer Science and Software Engineering, Huaiyin Institute of Technology, Huai'an City, Jiangsu, China
[2] Faculty of Mechanical and Materials Engineering, Huaiyin Institute of Technology, Huai'an City, Jiangsu, China
[3] Faculty of Automation, Huaiyin Institute of Technology, Huai'an City, Jiangsu, China

*** Corresponding author**

Abstract: This paper presents an intelligent and energy-efficient UAV framework for indoor rescue missions in GPS-denied environments. The proposed system integrates visual-inertial SLAM (Simultaneous Localization and Mapping), dynamic obstacle avoidance, and adaptive path planning using lightweight embedded sensors. At its core, the design fuses monocular camera data with inertial measurements through ORB-SLAM3, enhanced by 3D depth sensing from the STMicroelectronics VL53L9 dToF LiDAR. A deep learning module based on YOLOv5 and semantic mapping enables the UAV to detect and navigate around dynamic obstacles, while a novel risk-based path planner evaluates cost maps in real time to optimize safety and energy consumption. The architecture is evaluated in simulated disaster environments, demonstrating improved localization accuracy, reduced power usage, and superior obstacle avoidance over conventional SLAM-based and planning baselines. The results support the framework's viability for deployment in real-world rescue missions requiring intelligent aerial autonomy.

Keywords: UAV; Indoor rescue; Visual-inertial SLAM; Dynamic obstacle avoidance; Adaptive path planning; Deep learning; LiDAR; GPS-denied environments.

1. Introduction

Natural disasters, such as the earthquakes in Syria and Turkey in 2023, have become more frequent. These events can cause severe damage and make it hard for rescue teams to reach people in need. In some cases, the areas are too dangerous or inaccessible due to factors such as collapsed buildings or hazardous gases in the air [1]. This makes it essential to explore new methods for supporting search and rescue efforts, especially indoors, where space is tight and conditions are unpredictable. One promising solution is the use of flying robots, also known as hovering robots or UAVs (Unmanned Aerial Vehicles), that can move through cluttered environments without the need for wheels or tracks. However, since GPS signals are often unreliable or completely unavailable indoors, these robots need advanced systems to help them understand their location and surroundings [2].

A method that has shown great potential is visual SLAM (Simultaneous Localisation and Mapping), which enables the robot to use onboard cameras to create a map of the environment while tracking its position simultaneously [3]. In recent studies, researchers have combined visual SLAM with deep learning techniques to enable these robots to identify and avoid moving obstacles, such as debris or people, which is crucial in disaster scenarios [4]. While these systems are beneficial, indoor environments still present numerous challenges, including poor lighting, repetitive patterns, and constantly changing layouts, all of which can impact the robot's performance [5]. To address these issues, newer designs are incorporating additional sensors and more sophisticated algorithms to enhance navigation and safety. In this paper, we introduce a system that combines visual SLAM and real-time obstacle avoidance for indoor hovering robots, aimed at supporting rescue missions in complex and hazardous spaces. The goal is to provide a simple but effective robotic platform that can safely navigate and support human efforts during emergencies. However, few existing studies tackle the combined challenges of energy efficiency, semantic reasoning, and dynamic obstacle adaptation within a single lightweight UAV framework suitable for indoor rescue missions, this paper fills that gap by combining visual-inertial SLAM, low-power LiDAR sensing, and deep-learning-based semantic navigation into a unified, energy-efficient system.

2. Literature Review

In recent years, the use of robots in indoor rescue operations has become a significant area of research, particularly as traditional ground-based robots exhibit numerous limitations in disaster environments. For example, Birk et al. [6] noted

that wheeled and tracked robots often struggle to navigate collapsed buildings or debris-filled areas, making it more difficult to reach victims quickly. Because of this, researchers have begun investigating unmanned aerial vehicles (UAVs), also known as flying robots, which can hover and move freely in tight and cluttered spaces. Tzoumanikas et al. [7] supported this shift by reviewing SLAM techniques specifically designed for UAVs, explaining that these systems must be both lightweight and efficient due to the hardware limitations of drones. In a similar direction, Tardioli et al. [8] developed autonomous flying strategies for confined indoor environments, emphasising how hovering robots can navigate tight hallways where humans or ground robots cannot operate.

2.1. Evolution of SLAM for UAV Localization

A Major challenge these robots face is how to locate

themselves and move around without GPS, which does not work well indoors. This has made visual SLAM (Simultaneous Localisation and Mapping) a key technology. Zhou et al. [9] provided a helpful overview of visual SLAM systems, explaining how they utilise onboard cameras to enable robots to create maps and track their location simultaneously. Cadena et al. [10] noted that SLAM has advanced significantly over the years and is now more adept at handling complex, real-world environments, particularly in robotics. Fuentes-Pacheco et al. [11] also reviewed different types of SLAMs, highlighting that visual SLAM is especially useful in indoor rescue settings because it is smaller, cheaper, and more adaptable than sensor-heavy systems, such as LiDAR.Although visual SLAM is promising, it still presents challenges in indoor spaces. Xiong et al. [12] described common difficulties such as poor lighting, repetitive wall patterns, and moving people, which can confuse the robot's map. These problems can impact the robot's ability to move safely and accurately. To address this, Qin et al. [13] proposed combining camera data with IMUs (Inertial Measurement Units) so that even if the camera loses track of its surroundings, the robot can still estimate its movement. Their system, VINS-Mono, has improved the accuracy with which robots track themselves in challenging situations, such as fast motion or visual disturbances.Several SLAM algorithms have been developed to address these challenges. Mur-Artal and Tardós [14] introduced ORB-SLAM2, a well-known SLAM system that uses either one or two cameras to map and localise the robot. It is popular because it operates in real-time and can handle a wide range of indoor scenes. Engel et al. [15] developed LSD-SLAM, which works slightly differently by directly comparing images instead of matching individual points. This approach performs well in environments with a high level of visual texture. Later, Campos et al. [16] improved upon these methods with ORB-SLAM3, which combines camera and motion data and also enables the robot to maintain multiple maps, making it more reliable in larger or dynamic spaces.

2.2. Dynamic Obstacle Detection and Avoidance

Mapping alone, however, is insufficient for real-world rescue missions, where dynamic obstacles such as moving debris or victims often appear unpredictably. Fu et al. [17] explored the combination of SLAM with dynamic obstacle detection, demonstrating that a robot can adjust its path as it navigates through changing environments. Building on that, Chen et al. [18] utilised deep learning to instruct robots on distinguishing between fixed objects, such as walls, and moving hazards, like people or debris. Their method helps robots stay safe and avoid accidents during their missions.

2.3. Learning-Based Decision-Making and Path Planning

Beyond perception and mapping, autonomous decision-making is vital for UAVs to perform independent missions. Dharmadhikari et al. [19] used deep learning to train UAVs to make quick navigation decisions when encountering new obstacles. This makes the robots more flexible in situations they haven't seen before. Similarly, Tai et al. [20] explored reinforcement learning, which enables robots to learn how to navigate through a space by trial and error in a virtual setting and then apply that knowledge in real-world rescue environments. This is especially helpful when the robot doesn't have a pre-made map. Finally, Zhou et al. [21] developed a system that integrates SLAM with motion planning using LiDAR, demonstrating another approach for UAVs to navigate safely in complex environments, albeit with the requirement for heavier sensors.

2.4. Research Trend and Contribution Gap

Altogether, existing studies demonstrate that numerous researchers are working to enhance the performance of autonomous robots in indoor rescue scenarios. From lightweight mapping and obstacle detection to learning-based planning, the trend is shifting toward more intelligent and adaptable flying robots. Even though challenges remain, especially in unpredictable disaster environments, the combination of SLAM, perception, and planning systems offers a strong path forward in building robots that can truly help save lives in emergency scenarios.

In light of the limitations observed in traditional ground-based rescue robots and the growing capabilities of mobile autonomous systems, this paper proposes a framework specifically designed for indoor hovering UAVs operating in GPS-denied, cluttered environments. The system integrates visual-inertial SLAM for robust pose estimation, real-time dynamic obstacle detection using deep learning, and semantic risk-aware path planning enhanced by low-power 3D LiDAR sensing. Leveraging lightweight embedded sensors such as the VL53L9 direct Time-of-Flight module and an onboard IMU, the framework fuses geometric and semantic data to support energy-efficient, autonomous navigation. Built upon open-source SLAM infrastructure and adaptive planning logic, the proposed architecture demonstrates a practical and low-cost solution for autonomous UAV exploration and obstacle avoidance in disaster response scenarios.

3. Methodology

This paper introduces a modular and energy-efficient framework designed for autonomous UAV navigation in indoor rescue missions. The core methodology integrates visual SLAM-based localization, real-time semantic obstacle avoidance, and adaptive, energy-aware planning. The system architecture and evaluation are framed within a simulation environment that emulates typical disaster scenarios [22].

3.1. Simulation Environment

We developed and evaluated the roaming robot in ROS 2 Humble (Ubuntu 22.04) using the standard visualization and simulation tools available in the ROS ecosystem.

1) Gazebo Classic (invoked via gazebo_ros).
2) RViz2 for model verification and qualitative visualization.
3) robot_state_publisher and joint_state_publisher_gui to broadcast TF and interactively set joint states.
4) Robot geometry and kinematics are defined in URDF with all meshes referenced via package:// URIs. For simulation, the URDF includes a Gazebo <gazebo reference="base_link"><gravity>0</gravity></gazebo> tag so the robot hovers (no gravity-induced fall) and can be spawned at a controlled height.

All launches and parameters are supplied as Python launch files; the URDF and launch sources are included in the accompanying ROS 2 package to allow exact replication.

3.2. World geometry and dimensions

Simulations were conducted in Gazebo Classic's world

with random obstacles, which comprises a flat ground plane and a sky dome. Although the plane is effectively unbounded, we constrained the operating workspace around the origin to a *10m×10m×5m (X–Y–Z)* axis-aligned region for analysis and visualization. The world origin is at the ground plane, and the robot was spawned at *(x,y,z) = (0, 0, 1.0 m)*, such that the bottom of the robot is approximately *1.0 m* above the ground (hovering). The default world lighting and atmospheric settings were used; no additional obstacles or structures were placed unless otherwise stated.

3.3. Robot model and frames

The robot model is a "drone-like rover" exported from CAD to URDF. The root frame is *base_link*; rotor links are *rotor_1* to *rotor_4* connected by continuous joints (spin axis aligned with local +Z). For numerical stability and to avoid unrealistic energy growth in free spin, each rotor joint includes light viscous damping (damping = 0.01, friction = 0). Meshes are STL assets placed under the package's *meshes/* directory and resolved by Gazebo/RViz through the *package://* URI scheme. [23]

3.4. Physics and numerical settings

Unless otherwise noted, Gazebo Classic defaults were used:

1) Gravity (world): 0,0,−9.81 m/s² (default), but disabled for the robot via the <gravity>0</gravity> model tag to maintain a constant hover.
2) Time step/integrator: max_step_size = 0.001s (1 ms); real-time update rate ≈ 1000 Hz; target real-time factor ≈ 1.0.
3) Contact/collision: default ODE parameters; collision geometry follows visual geometry unless simplified collision meshes are provided.
4) Initial conditions: zero linear and angular velocity; spawn pose as above.

These choices ensure numerically stable hovering without a controller while keeping the simulation real-time on a commodity workstation. When gravity is enabled, the same integrator and step sizes are retained; only the model gravity flag is toggled.

3.5. Visualization and verification

RViz2 was used to verify link frames, mesh alignment, and joint axes prior to simulation. The RobotModel display consumed the same robot_description parameter as Gazebo to guarantee consistency. A fixed frame of base_link and a 1 m grid were used to confirm spawn height and to visually bracket the 10×10×5m workspace.

Table 1. launch and parameter

Component	Launch file (Package)	Command	Key parameters/notes
RViz2	display.launch.py (rovering_robot)	ros2 launch rovering_robot display.launch.py	Fixed frame: base_link; RobotModel reads robot_description from URDF; 1 m grid for visual reference
Gazebo	gazebo.launch.py (rovering_robot)	ros2 launch rovering_robot gazebo.launch.py	World: random.world; spawn pose: (x, y, z) = (0, 0, 1.0 m); model-level gravity: disabled (<gazebo reference="base_link"><gravity>0</gravity></gazebo>); physics defaults: max_step_size = 0.001s, update rate ≈ 1000 Hz, real-time factor ≈ 1.0; initial linear/angular velocities = 0.

To improve depth sensing and robustness in visually degraded indoor environments, the UAV platform integrates the STMicroelectronics VL53L9 direct Time-of-Flight (dToF) 3D LiDAR sensor. The VL53L9 operates over a 48×48 region of interest (ROI), producing 2,304 independent depth measurements per frame. This dense matrix of distance readings effectively generates a low-resolution, high-speed depth map that supports real-time navigation, particularly in environments where visual SLAM alone is insufficient due to poor lighting, dust, or smoke.

Although the VL53L9 sensor is not commercially available, its behavior was modeled virtually in the simulation using publicly released STMicroelectronics datasheet parameters. This approach allowed realistic emulation of its depth-sensing characteristics in indoor environments

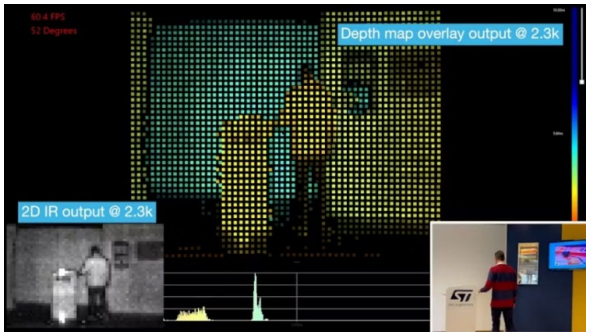

Figure 1. STMicroelectronics VL53L9 direct Time-of-Flight (dToF) 3D LiDAR sensor.

The sensor employs **single-photon avalanche diode (SPAD)** technology to capture depth data based on the **round-trip time-of-flight (ToF)** of laser pulses. The distance to a target object is calculated using the formula:

$$D = \frac{c \cdot \Delta t}{2} \quad \textbf{(1)}$$

Where D represents the measured distance, c is the speed of light, and Δt is the time taken for the laser pulse to travel to the target and back. The division by two accounts for the round trip. Equation 1 underpins the direct time-of-flight principle, enabling significant depth estimation in environments where cameras may fail due to dust, smoke, or poor lighting. By integrating this low-power LiDAR, the UAV achieves reliable 3D perception while maintaining energy efficiency, a critical factor for missions with limited battery capacity.

This sensor supports 2304 ranging zones and produces dense 3D depth maps that improve perception in smoke, darkness, or visual ambiguity. These inputs are fused with camera and IMU data to form the UAV's perception backbone.

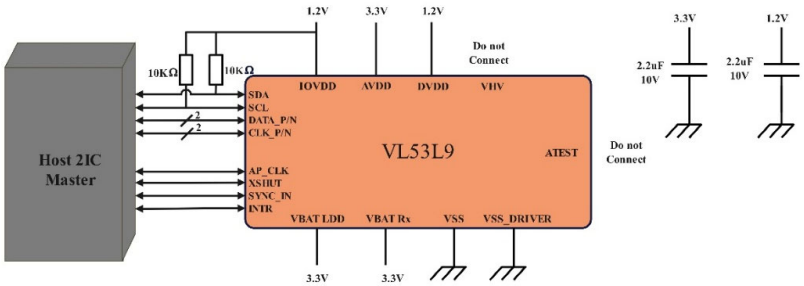

Figure 2. Dense 3D depth map generated by the VL53L9 LiDAR sensor illustrating spatial perception in indoor environments.

Unlike conventional mechanical 3D LiDAR systems, the VL53L9 is optimized for **low-power, embedded applications**, making it ideal for UAV-based indoor rescue. Its **energy consumption** is approximately **20–30 mW during active scanning**, significantly extending mission duration in power-constrained scenarios. Moreover, the sensor features a **wide field of view (~63° × 63°)**, enabling it to capture spatial geometry in tight corridors and obstructed environments without the need for stereoscopic or depth cameras.

For localization, the paper employs ORB-SLAM3 which fuses monocular camera input with IMU data to estimate the UAV's trajectory and build a sparse 3D feature map. The SLAM outputs are dynamically cross-validated using the VL53L9's depth map to reduce error drift. In instances of high velocity or feature loss, the system applies a visual-inertial fusion strategy using nonlinear optimization to minimize the reprojection error across consecutive frames. The cost function used in the back-end optimization is given by:

$$\min_{x} \sum_{i=1}^{n} \rho(\|z_i - h(x_i)\|^2) \quad (2)$$

In this formulation, z denotes the observed 2D feature point in the image, and h(x) is the observation model that projects a 3D landmark into 2D image coordinates based on the estimated state vector x. The robust loss function ρ(·) (such as Huber) is applied to reduce the effect of outliers, which are common in disaster environments due to occlusions, dust, or moving objects. This equation was chosen because it minimizes the discrepancy between the measured feature positions and the predicted ones, thereby refining pose estimation. By using h(x), the system can incorporate both the geometric projection model and the current state estimate, making the optimization more resilient to noise and disturbances compared to purely geometric formulations.

To detect obstacles, the system uses a deep learning-based perception module that incorporates YOLOv5 for object classification and localization. The object detection module employs YOLOv5s (small variant), optimized with TensorRT on an NVIDIA Jetson Xavier NX for real-time inference. Building on these detections, a semantic risk map assigns each grid cell a weighted cost that considers object class, proximity, and motion. During navigation, a trajectory safety index measures safety by calculating the ratio of safe to total path points. Simultaneously, an energy-aware cost assesses the total power consumption per mission segment and is integrated into the overall optimization goal to balance energy, safety, and time. The model is pre-trained on relevant indoor datasets and optimised using TensorRT for real-time inference. To compensate for dynamic scenes and moving hazards, a Kalman filter tracks the object trajectory based on successive detection frames. The fused obstacle detection is then used to update a semantic occupancy grid with risk labels. Depth readings from the VL53L9 LiDAR improve spatial awareness, allowing the UAV to estimate object proximity and shape even in poorly lit or texture-deficient environments. A context-aware algorithm drives the navigation and trajectory planning system termed the Dynamic Semantic Cost Mapping and Path Prioritisation Algorithm (DSCMPPA). The algorithm generates a real-time risk cost map where each grid cell's score C_{ij} is calculated as a linear combination of three contributing factors:

$$c_{i,j=}\alpha_s S_{i,j}+\alpha_d D_{i,j}+\alpha_m M_{i,j} \quad (3)$$

In this formulation, Sij represents the semantic risk (e.g., humans, debris, or walls), Dij is the inverse distance to the UAV, and Mij is the motion penalty associated with moving hazards. The coefficients αs, αd, and αmare tunable weights reflecting the relative importance of safety, distance, and motion. This formula was chosen because it enables the UAV to make risk-aware decisions, prioritizing safety when navigating close to humans or moving debris. Unlike traditional geometric path planning, this semantic approach integrates contextual understanding into the navigation process, which is vital in rescue missions.

The local planner finds the lowest-cost path on this dynamic map using a graph search algorithm like Dijkstra or A*, creating trajectories that actively steer clear of high-risk, moving objects. To save energy, the system uses several strategies within an energy-aware AI framework. First, perception modules such as YOLOv5 and VL53L9 are triggered only based on the system's state: if the UAV's speed goes over a specific limit or SLAM confidence drops, these modules turn on; otherwise, they stay off. Second, path planning includes an energy cost function:

$$E_{path} = \sum_{i=1}^{n} P_i \cdot \Delta t_i \quad (4)$$

Pi is the UAV's instantaneous power at step i (including propulsion, avionics, and any sensor/compute loads), and Δti is the duration of that step. This discrete sum is the standard Riemann approximation of the continuous energy integral E = ∫P(t)dt. It was chosen because it naturally captures **time-varying power** due to throttle changes (hover, climb, aggressive turns), **payload/compute spikes** (e.g., YOLOv5 inference bursts), and **sensor scheduling** (e.g., LiDAR on/off). It also generalizes the simpler segment expression E=P·T: if a path segment is modeled with a single constant-power interval, the sum reduces to that product. Using Epath allows the planner to evaluate competing trajectories based on their actual battery impact, and to incorporate this term directly into your multi-objective cost function (with weights for safety/time/energy), ensuring that paths that are safer but

excessively power-hungry, or fast but power-spiky, are correctly penalized.

The final cost function that the planner seeks to minimise is:

$$TotalCost\beta_1 \cdot C_{risk} + \beta_2 \cdot E_{path} + \beta_3 \cdot T_{mission} \quad (5)$$

These cost functions are directly used to generate the experimental curves presented in Section 4. Crisk quantifies the collision risk associated with obstacles and dynamic hazards, Epath represents the total energy consumption along the planned trajectory, and Tmission denotes the estimated mission duration. The coefficients β1, β2, and β3 are tunable weights that allow the planner to balance safety, energy efficiency, and time sensitivity according to the mission context. This formulation was chosen because rescue scenarios demand adaptability: in high-risk environments, β1 can be increased to prioritize safety, whereas in urgent medical deliveries, β3 may dominate to minimize time. Similarly, in long-duration missions where endurance is critical, β2 ensures energy efficiency. By consolidating these competing objectives into a single cost function, the UAV can dynamically generate trajectories that are both safe and mission-aware, reflecting real-world trade-offs faced in disaster response operations; Trajectories are scored based on a weighted objective function that balances safety, energy cost, and mission time.

Finally, the system is evaluated across three disaster-representative simulated environments involving narrow corridors, human motion, and cluttered debris. Performance metrics include localisation accuracy (mean drift error in meters), obstacle detection precision and recall, average power usage per minute, and trajectory safety index measured via collision avoidance and successful re-planning events. Results are compared against baseline systems using static SLAM-only navigation and continuous-sensing AI to validate the proposed framework's effectiveness in achieving robust, intelligent, and energy-efficient UAV navigation for indoor rescue operations.

To achieve robust real-time localisation under the constraints of indoor rescue operations, this paper employs a tightly-coupled visual-inertial SLAM (VI-SLAM) approach. While monocular visual SLAM alone can offer significant pose estimates in well-lit and feature-rich environments, it is prone to drift or failure in scenarios characterised by rapid motion, occlusions, or lack of visual texture common in disaster-prone indoor environments. The integration of inertial measurements from an onboard IMU substantially enhances the resilience and accuracy of SLAM by providing complementary motion estimates between frames and enabling predictive pose propagation when visual features are sparse or temporarily unavailable.

The VI-SLAM system combines data from a monocular RGB camera and a six-axis IMU using a nonlinear optimisation framework. In this configuration, image frames are processed to extract visual features which are matched across time to infer relative camera motion. Simultaneously, the IMU provides high-frequency readings of linear acceleration and angular velocity, allowing for the integration of pose changes using the principles of rigid body dynamics. The fused pose estimation problem is modelled as a nonlinear least-squares optimisation that minimises both the photometric error from camera observations and the residuals from IMU propagation. The optimisation seeks to determine the UAV's state vector, including position, orientation, velocity, and inertial sensor biases, by minimising the following cost function:

$$\hat{x} = \underset{x}{argmin}\left(\sum_i \rho(\|z_i - h(x_i)\|^2) + \sum_j \|x_j - f(x_{j-1}, u_j)\|^2\right) \quad (6)$$

$\hat{x}$ denotes the optimal state estimate (including position, orientation, velocity, and sensor biases). The first summation minimizes the **reprojection error**, where zi is the observed feature in the image and h(xi) is the observation model projecting a 3D landmark into 2D image coordinates. The robust loss function ρ(·) reduces the effect of outliers caused by noise, occlusion, or dynamic objects. The second summation represents the **IMU motion model residuals**, where xj is the estimated state at time step j, and f(xj−1,uj) is the predicted state propagated from the previous timestep using the IMU control input uj. This term enforces temporal consistency between successive states. Equation 6 was chosen because it explicitly fuses camera and inertial data into a single nonlinear least-squares optimisation. Visual SLAM alone is prone to drift under poor lighting or occlusion, while inertial integration alone accumulates error over time. By combining both, the UAV maintains robust and drift-resistant localisation, ensuring stable navigation in visually degraded or GPS-denied rescue environments.

During visual dropout or motion blur, the system continues to integrate IMU data over short durations to propagate pose estimates using discrete-time kinematic equations. These include position update:

$$P_{t+1} = p_t + V_t\Delta t + \frac{1}{2}(R_t a_t - g)\Delta t^2 \quad (7)$$

And velocity update:

$$V_{t+1} = V_t + (R_t a_t - g)\Delta t^2 \quad (8)$$

These equations (7) and (8) allow for the prediction of future UAV states during periods of feature scarcity, where p and y represent position and velocity, Rt is the rotation matrix, at is the measured acceleration, g is the gravitational vector, and Δt is the time step. These equations were chosen because they allow the UAV to predict its future state when visual information is unavailable. This predictive ability ensures continuity of localization and control, which is essential for stability during critical rescue operations in visually degraded environments.

The complete VI-SLAM pipeline, as illustrated earlier, shows how visual features and inertial readings are co-processed through nonlinear optimisation and fed into a visual-inertial back-end. The result is a robust, drift-resistant pose estimate that supports stable mapping and control during rescue missions. By leveraging this tightly-coupled fusion, the UAV maintains consistent localisation accuracy even in challenging environments where camera-based tracking alone would be unreliable.

This implementation builds upon the visual-inertial mode of ORB-SLAM3, which supports tightly-coupled IMU and visual feature fusion.

3.6. Coupling Between Perception and Planning

To ensure consistency between perception and decision-making, the proposed framework explicitly couples the visual–inertial–LiDAR SLAM module with the DSCMPPA planning algorithm. The SLAM subsystem provides two essential outputs: $\hat{x}_k$ and a local environment representation$\mathcal{M}_{k'}$ enriched with depth information from the VL53L9 LiDAR and semantic labels from the YOLOv5 module. These outputs are directly used by the DSCMPPA

planner to compute risk, energy, and trajectory costs. Specifically, C_{risk} (Eq. 3) is derived from obstacle proximity and semantic classification extracted from $\mathcal{M}_{k'}$, while E_{path} (Eq. 4) depends on trajectory smoothness and control effort, both influenced by localization accuracy. $T_{mission}$ is estimated from the planned trajectory based on the current pose and the structure of the environment. Consequently, the overall cost function (Eq. 5) is indirectly influenced by the quality of SLAM estimation, as improved localization accuracy and depth perception lead to more reliable risk evaluation and more energy-efficient trajectories. This coupling ensures that the observed improvements in trajectory smoothness, risk reduction, and energy efficiency are not solely due to the planning algorithm, but also result from the enhanced perception capabilities of the integrated SLAM system.

4. Results

The proposed UAV navigation framework integrates a tightly-coupled visual-inertial SLAM (VI-SLAM) system, low-power 3D LiDAR sensing, and semantic-aware trajectory planning into a unified modular architecture tailored for indoor search and rescue operations in GPS-denied environments. At the core of this architecture lies the fusion of monocular camera data with high-frequency inertial measurements, complemented by dense depth perception from the VL53L9 direct Time-of-Flight LiDAR sensor. The perception stack is further augmented by a YOLOv5-based deep learning module for object recognition and classification, enabling dynamic obstacle detection under challenging lighting and visibility conditions. The output of the perception system is fed into a dynamic semantic cost mapping algorithm, which updates the UAV's navigation plan in real time by considering environmental risks and energy constraints. The modular nature of the system ensures scalability and flexibility for deployment in diverse indoor environments. **Figure 3** provides a high-level block diagram illustrating the interaction between the SLAM subsystem, LiDAR module, semantic planner, and control loop.

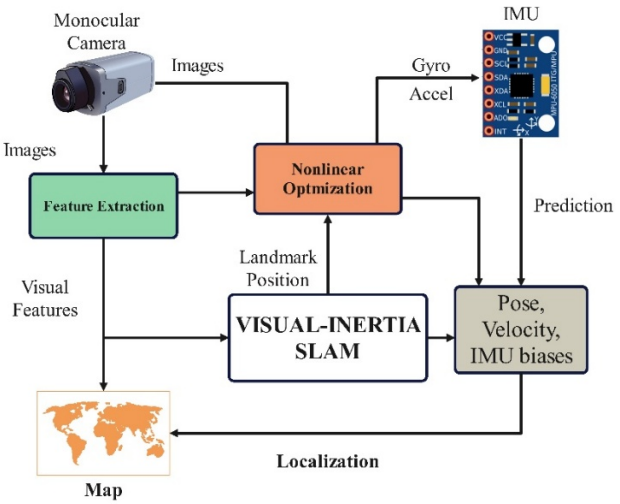

Figure 3. Block diagram of the visual-inertial SLAM system integrated into the UAV navigation framework.

4.1. Planning Evaluation (Trajectory)

To evaluate the path-planning component of the proposed framework, we compared DSCMPPA against a classical A* planner and a deep-learning baseline using the trajectories shown in Figure 4. The three-dimensional trajectory comparison indicates that DSCMPPA follows a smoother and more direct path from start to goal, with fewer abrupt detours than A*. The deep-learning approach also reaches the goal, but its trajectory remains slightly less regular than DSCMPPA.

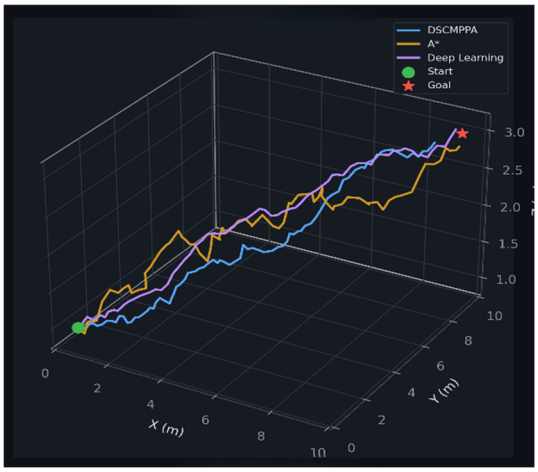

Figure 4. Three-dimensional trajectory comparison of DSCMPPA, A*, and Deep Learning planners from start to goal.

4.2. Planning Evaluation (DSCMPPA vs A* vs Deep Learning)

Risk-aware navigation is one of the central objectives of the proposed method. Figure 5 combines the time-varying risk metric $M(x,t)$ with the multi-run risk-versus-path-length distribution. DSCMPPA maintains the lowest overall risk profile, with most values concentrated below approximately 0.35 and with a lower mean exposure than the other methods. In contrast, A* experiences the highest risk peaks, reaching about 0.65–0.67 during the middle and late stages of the mission, and its repeated runs cluster at longer path lengths with higher average risk. The deep-learning baseline performs better than A* but still exhibits higher average risk than DSCMPPA.

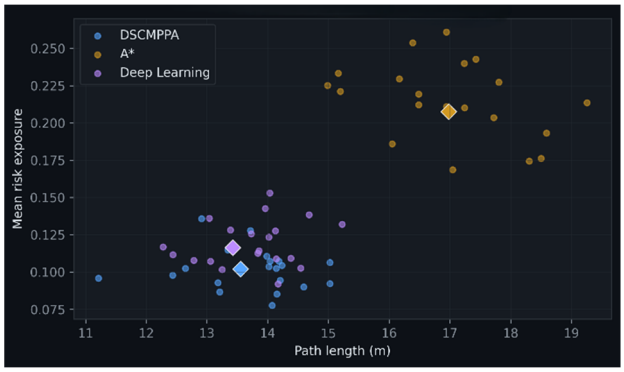

Figure 4.-Risk vs Path Scatter

Figure 5 presents the evolution of risk exposure over normalized mission time for the three planning methods. DSCMPPA consistently maintains the lowest risk profile throughout the mission, with smoother variations and reduced peak values. In contrast, the A* planner exhibits significant risk spikes, particularly during mid-mission phases, indicating vulnerability when navigating near obstacles. The deep learning approach demonstrates moderate performance but still shows higher variability compared to DSCMPPA. These results highlight the effectiveness of the proposed

semantic risk-aware cost function (Eq. 3) in minimizing exposure to hazardous regions during navigation. Figure 6 shows the cumulative energy consumption as a function of normalized mission time, computed using Eq. (4). The DSCMPPA planner achieves the lowest energy consumption, with a gradual and steady increase, indicating efficient motion and reduced control effort. The A* planner consumes more energy due to longer paths and frequent direction changes, while the deep learning approach exhibits the highest energy usage, reflecting less efficient trajectory control. This demonstrates that DSCMPPA effectively incorporates energy-aware optimization, improving UAV endurance.

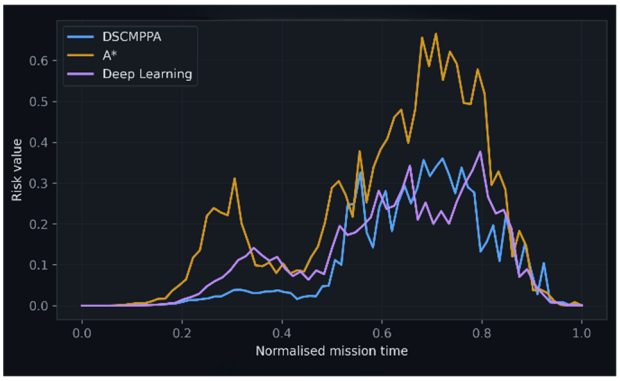

Figure 6.- Dynamic risk exposure over mission time

This figure 5 illustrates the dynamic risk exposure over mission time and the distribution of risk versus path length across multiple runs. The risk values are computed using Eq. (3), which integrates semantic information, obstacle proximity, and motion dynamics. DSCMPPA consistently maintains lower risk levels compared to A* and the deep-learning baseline, demonstrating its ability to prioritize safe navigation in dynamic environments. The clustering of results at lower risk levels further indicates improved robustness and consistency.

4.3. End-to-End System Performance

All planning methods (DSCMPPA, A*, and Deep Learning) are evaluated using the same SLAM-generated map and pose estimation in order to ensure a consistent and fair comparison across all approaches. By relying on a common perception backbone, the experimental setup eliminates variability arising from differences in localization accuracy or environmental representation. This ensures that the observed differences in trajectory quality, risk exposure, and energy consumption are solely attributable to the planning strategies themselves rather than to discrepancies in sensing or mapping. Furthermore, this unified evaluation framework highlights the effectiveness of DSCMPPA in leveraging reliable SLAM outputs to produce safer and more energy-efficient trajectories, while maintaining robustness in complex and dynamic environments. As a result, the end-to-end system performance reflects a true comparison of decision-making capabilities under identical perceptual conditions, reinforcing the validity of the experimental results.

4.4. Cumulative Energy Consumption

Energy efficiency was assessed through the cumulative energy curves and the resource-management visualisations reported in Figure 6. DSCMPPA shows the lowest total energy demand at the end of the mission, finishing near 1.4 J, whereas A* reaches roughly 2.6 J and the deep-learning baseline about 4.4 J. The lower slope of the DSCMPPA curve indicates that the proposed planner avoids energetically expensive manoeuvres and unnecessary path corrections.

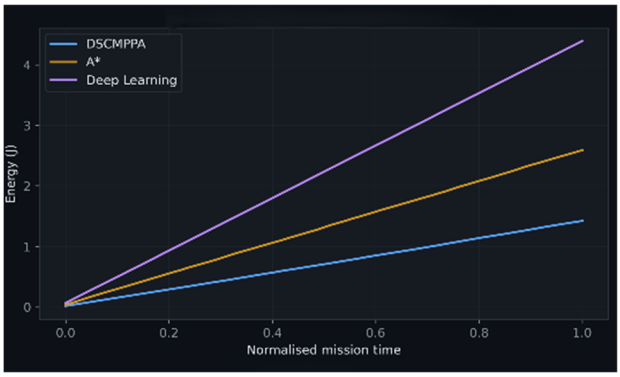

Figure 5.-Energy Over Time

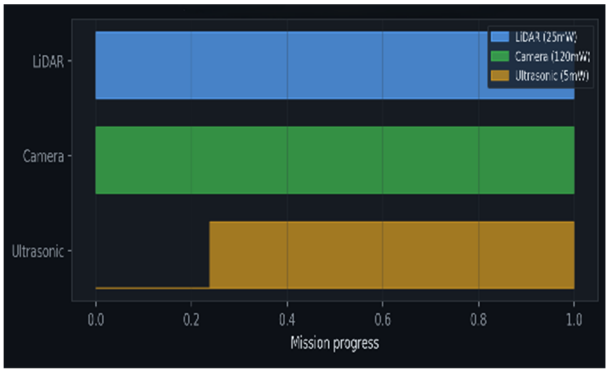

Figure 8.-Sensor Activation

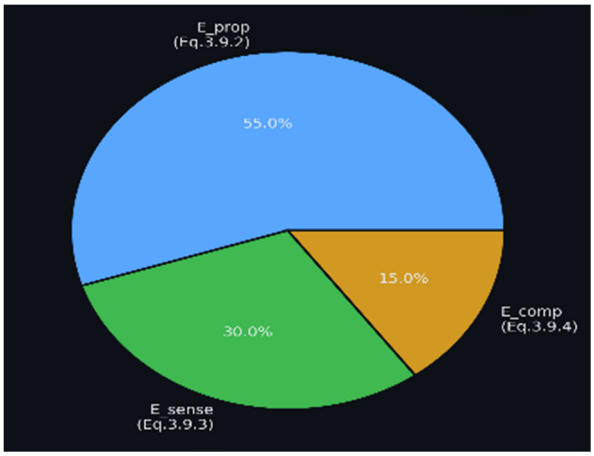

Figure 6.-Energy Breakdown Pie

Figure 6 also shows that propulsion remains the dominant energy contributor (55%), followed by sensing (30%) and computation (15%). In addition, the sensor-activation profile indicates that LiDAR and camera remain active throughout the mission, while ultrasonic sensing is activated only after the early phase, which is consistent with context-dependent sensor management.

Figure 7 presents a normalized comparison of path length, risk exposure, and energy consumption across the three planning methods. DSCMPPA achieves the best overall balance, combining relatively short paths with low risk and minimal energy usage. A* produces the longest and riskiest paths, while the deep learning method achieves short paths but at the cost of significantly higher energy consumption. These results confirm that DSCMPPA provides a more balanced and efficient trade-off between safety, efficiency, and path optimality. Figure 8 shows the relationship between

path length and mean risk exposure across multiple runs. DSCMPPA results are clustered in the lower-left region, indicating both shorter paths and lower risk levels. A* results appear in the upper-right region, reflecting longer paths and higher risk exposure. The deep learning approach lies between the two but still shows greater variability. This distribution demonstrates the robustness and consistency of DSCMPPA in achieving safer and more efficient navigation outcomes. Figure 9 illustrates the evolution of adaptive weights used in the multi-objective cost function. The energy weight gradually increases over time, reflecting the growing importance of energy efficiency as the mission progresses. Conversely, the motion and distance weights slightly decrease, indicating a shift in priority toward endurance. This adaptive behaviour enables the UAV to dynamically balance competing objectives, ensuring safe and energy-efficient navigation under changing mission conditions.

4.5. Multi-Metric Performance and Adaptive Control

A broader comparison of planner behaviour is given in Figure 7, which combines the normalised performance metrics with the evolution of the adaptive weights. DSCMPPA achieves the best overall compromise: it combines a relatively short path length (about 0.80 in normalised terms), the lowest risk exposure (about 0.49), and the lowest energy score (about 0.32). A* records the worst balance, with the longest path and the highest risk exposure, while the deep-learning baseline obtains a short path but at a substantially higher energy cost.

The adaptive-weight plot in Figure 7 shows a gradual increase in the energy weight a_e(t) and a slight decrease in the motion and distance terms as the mission progresses, indicating that the controller increasingly prioritises endurance when the battery state becomes more critical. Taken together, these results show that DSCMPPA is not only effective in lowering risk, but also explicitly designed to preserve endurance while maintaining navigation quality.

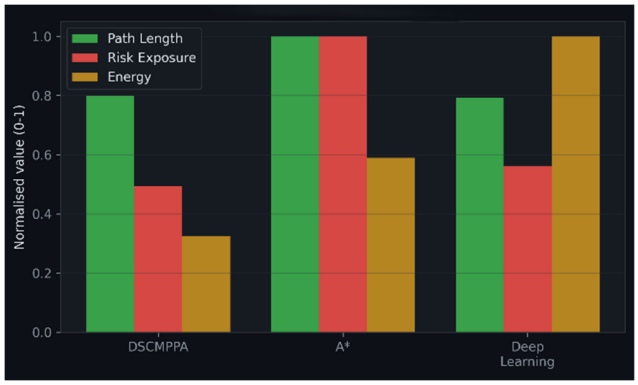

Figure 10.-Bar Metrics

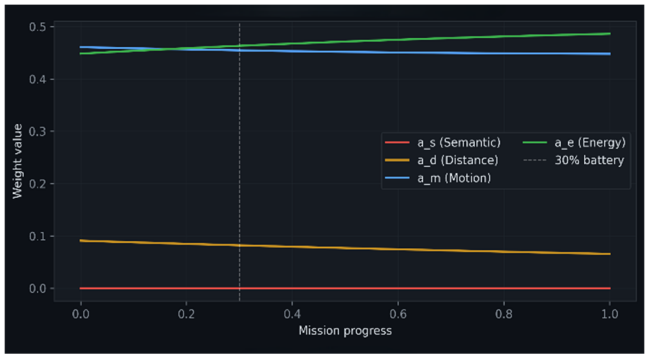

Figure 11.-Adaptive Weights

Figure 10 presents the distribution of energy consumption across different system components. Propulsion accounts for the largest share (55%), followed by sensing (30%) and computation (15%). This distribution highlights that motion control dominates energy usage in UAV systems, while sensing and onboard processing also contribute significantly. These insights justify the integration of energy-aware planning strategies to optimize overall mission efficiency. Figure 11 shows the activation timeline of different sensors during the mission. LiDAR and camera sensors remain active throughout the mission to ensure continuous perception, while the ultrasonic sensor is activated only after the initial phase. This selective activation strategy reduces unnecessary energy consumption while maintaining sufficient environmental awareness. The results demonstrate the effectiveness of context-aware sensing in improving overall system efficiency.

5. Modular Synergy and System-Level Validation

Taken together, the results show that the benefit of the proposed framework does not come from a single module, but from the coordinated interaction between adaptive cost mapping, energy-aware control, and context-sensitive sensing. DSCMPPA consistently dominates the trade-off surface by reducing risk and energy simultaneously without incurring the path-length penalty observed with A*. The adaptive weights explain part of this behaviour: as the mission advances, the energy term becomes more influential while the controller preserves trajectory stability. At the same time, the sensor-activation strategy prevents unnecessary sensing load during the earliest mission stage, supporting a more efficient overall policy.

5.1. Limitations and Future Work

Although the simulation results demonstrate strong performance across key metrics, certain limitations must be acknowledged. First, the system has been tested in controlled simulation environments that, while realistic, may not capture the full variability of real-world disasters, including sensor noise, irregular terrain, and unpredictable airflow patterns. Additionally, the current framework operates in a single-agent mode and does not support inter-UAV communication or collaborative decision-making. Future work will aim to deploy the system on a physical UAV platform for field testing and extend the architecture to support multi-agent cooperation, shared mapping, and coordinated victim localization strategies. Such advancements would further enhance the scalability and effectiveness of the system in large-scale rescue missions.

5.2. Summary of Comparative Performance

To summarise the comparative results, Table 2 reports representative values extracted from the uploaded evaluation figures. DSCMPPA delivers the most balanced performance profile, combining the lowest mean risk and the lowest cumulative energy with a short path length. A* remains simple and effective in reaching the goal, but it does so with longer paths and higher exposure to danger. The deep-learning baseline generates compact trajectories, yet its energy demand is the highest among the tested methods.

These findings confirm that the proposed DSCMPPA-

based planning strategy provides a robust, energy-aware, and safety-oriented solution suitable for UAV navigation in cluttered indoor rescue environments.

Table 2. Comparative planner performance based on trajectory, risk, and energy metrics.

Method	Representative Path Length (m)	Mean Risk Exposure	Final Cumulative Energy (J)	Overall Interpretation
DSCMPPA	≈13.5	≈0.10	≈1.4	Best overall trade-off
A*	≈17.0	≈0.21	≈2.6	Longest and riskiest path
Deep Learning	≈13.4	≈0.12	≈4.4	Short path but highest energy

6. Discussion

The observed improvements are the result of both enhanced perception through visual–inertial–LiDAR fusion and the proposed DSCMPPA planning strategy. The revised results indicate that the main contribution of the proposed framework lies in its ability to balance competing objectives rather than optimise a single metric in isolation. Compared with A* and the deep-learning baseline, DSCMPPA produces trajectories that are shorter than those of A*, substantially less risky than both alternatives on average, and markedly more energy efficient. This is particularly important in indoor rescue operations, where excessive detours, high-risk corridor crossings, and repeated corrective manoeuvres may compromise both safety and mission endurance.The risk curves and the scatter distribution suggest that DSCMPPA is not only better in a single run, but also more stable across multiple runs. Its samples remain concentrated in a low-risk, moderate-path-length region, whereas A* exhibits consistently higher exposure and longer routes. The deep-learning planner narrows the gap in path length, but its higher spread in risk and its strong energy demand reduce its practical attractiveness for battery-constrained UAV platforms.A key differentiator of the proposed approach is its explicit energy-aware adaptation. The cumulative energy curves, the adaptive-weight evolution, and the energy breakdown all indicate that endurance is built directly into the optimisation logic. Instead of treating energy as a secondary outcome, the method progressively increases the importance of the energy term as the mission progresses, which helps explain the lower final consumption of DSCMPPA.Another important observation is that adaptive sensing complements adaptive planning. The sensor-activation profile shows that not all sensing resources must be used uniformly throughout the mission. Maintaining LiDAR and camera activity while delaying ultrasonic activation suggests a pragmatic strategy in which sensing complexity is matched to mission context, thereby supporting efficient resource use without sacrificing environmental awareness.Nevertheless, several limitations remain. The reported values are derived from simulation-based evaluation and from a limited set of comparative planners. Future work should therefore validate the method on a physical UAV platform, test its robustness under real sensor noise and airflow disturbances, and compare it against a broader set of modern learning-based and sampling-based planners.

7. Conclusion

This paper presents a modular UAV navigation framework in which the DSCMPPA planner is evaluated against A* and a deep-learning baseline using trajectory, risk, and energy criteria. The revised results show that the proposed method achieves the best overall trade-off for indoor rescue navigation by combining smooth goal-reaching trajectories with lower exposure to environmental risk and reduced cumulative energy consumption. Quantitatively, DSCMPPA ends the mission with approximately 1.4 J of cumulative energy, compared with about 2.6 J for A* and 4.4 J for the deep-learning baseline. Its mean risk remains clustered near 0.10 in the multi-run analysis, while A* is concentrated around 0.20–0.25 and the deep-learning baseline occupies an intermediate position. In the normalised comparison, DSCMPPA also records the lowest overall energy and risk scores while preserving a short path length. Compared with the alternative planners, the proposed framework offers a more balanced combination of safety, efficiency, and interpretability, making it a promising solution for constrained UAV platforms operating in GPS-denied indoor scenarios.

Future work will focus on hardware implementation, richer multi-agent coordination, and further validation of adaptive sensing and weighting policies under real-world rescue conditions. Overall, the updated results strengthen the conclusion that semantically informed, energy-aware path planning is a practical foundation for autonomous aerial operation in hazardous indoor environments. Across all comparative figures, the same trend is observed: DSCMPPA provides the most favourable compromise between path length, risk exposure, and energy consumption. A* reaches the goal with a clear penalty in route efficiency and safety, while the deep-learning baseline preserves relatively short paths but at a much higher energetic cost. This trade-off analysis is essential for rescue UAVs, because the best planner is not necessarily the one with the shortest path alone, but the one that sustains safe motion and mission endurance simultaneously. The adapted results therefore validate the architectural decision to integrate semantic cost adaptation, energy-aware optimisation, and selective sensor usage into a unified navigation framework. The results confirm that tightly coupling SLAM-based perception with adaptive multi-objective planning is essential for achieving robust, safe, and energy-efficient UAV navigation.

References

[1] Shahmoradi, A., et al. (2023). UAV-assisted search and rescue in disaster management: A review. Drones, 7(2), 64. https://doi.org/10.3390/drones7020064

[2] Gupta, A., & Fernando, X. (2022). Simultaneous Localization and Mapping (SLAM) and Data Fusion in Unmanned Aerial Vehicles: Recent Advances and Challenges. Drones, 6(4), 85. https://doi.org/10.3390/drones6040085. https://doi.org/10.1109/TRO.2016.2624754

[3] Fu, Y., Zhou, X., & Xu, D. (2021). Visual SLAM with dynamic obstacle avoidance in indoor environments. Sensors, 21(6), 2034. https://doi.org/10.3390/s21062034

[4] Yang, L., Ye, J., Zhang, Y., Wang, L., & Qiu, C. (2024). A semantic SLAM-based method for navigation and landing of UAVs in indoor environments. Knowledge-Based Systems, 293, 111693. https://doi.org/10.1016/j.knosys.2024.111693.

[5] Mokhtarzadeh, A. A.; & Yangqing, Z. J. "Human-Robot Interaction and Self-Driving-Car Safety: Integrating Dispositif Networks." In Proc. IEEE Int. Conf. Intelligence and Safety for Robotics (ISR), Shenyang, 2018, 494–499. DOI 10.1109/IISR.2018.8535696.

[6] Birk, A., et al. (2009). Rescue robotics—a crucial milestone on the road to intelligent systems. Advanced Robotics, 23(9), 1051–1068. https://doi.org/10.1163/156855309X452449

[7] Tzoumanikas, D., et al. (2019). A survey of SLAM techniques for UAVs. Drones, 3(4), 75. https://doi.org/10.3390/drones3040075

[8] Tardioli, D., et al. (2019). Autonomous navigation of aerial robots in confined spaces. Sensors, 19(5), 1097. https://doi.org/10.3390/s19051097

[9] Zhou, L., Wang, Z., & Wang, H. (2020). Visual SLAM for indoor environments: A comprehensive survey. Robotics and Autonomous Systems, 125, 103425. https://doi.org/10.1016/j.robot.2019.103425

[10] Cadena, C., et al. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on Robotics, 32(6), 1309–1332. https://doi.org/10.1109/TRO.2016.2624754

[11] Fuentes-Pacheco, J., Ruiz-Ascencio, J., & Rendón-Mancha, J. M. (2015). Visual simultaneous localization and mapping: A survey. Artificial Intelligence Review, 43, 55–81. https://doi.org/10.1007/s10462-013-9405-y

[12] Xiong, Y., Zhou, Y., She, J., & Yu, A. (2025). Collaborative Coverage path planning for UAV swarm for Multi-region Post-Disaster assessment. Vehicular Communications, 100915. https://doi.org/10.1016/j.vehcom.2025.100915Qin, T., Li, P., & Shen, S. (2018). VINS-Mono: A robust and versatile monocular visual-inertial state estimator. IEEE Transactions on Robotics, 34(4), 1004–1020. https://doi.org/10.1109/TRO.2018.2853729

[13] Qin, T., Li, P., & Shen, S. (2018). VINS-Mono: a robust and versatile monocular Visual-Inertial state estimator. IEEE Transactions on Robotics, 34(4), 1004–1020. https://doi.org/10.1109/tro.2018.2853729

[14] Mur-Artal, R., & Tardós, J. D. (2017). ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Transactions on Robotics, 33(5), 1255–1262. https://doi.org/10.1109/TRO.2017.2705103

[15] Engel, J., Schöps, T., & Cremers, D. (2014). LSD-SLAM: Large-scale direct monocular SLAM. ECCV, 834–849. https://doi.org/10.1007/978-3-319-10605-2_54

[16] Campos, C., et al. (2021). ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multi-map SLAM. IEEE Transactions on Robotics, 37(6), 1874–1890. https://doi.org/10.1109/TRO.2021.3075644

[17] Fu, Y., Zhou, X., & Xu, D. (2021). Visual SLAM with dynamic obstacle avoidance in indoor environments. Sensors, 21(6), 2034. https://doi.org/10.3390/s21062034

[18] Chen, X., et al. (2021). Real-time dynamic obstacle detection and tracking for autonomous robots using deep learning. IEEE Access, 9, 71349–71359. https://doi.org/10.1109/ACCESS.2021.3077794

[19] Dharmadhikari, T., Sahu, S. K., & Gandhi, N. (2020). Deep learning-based dynamic obstacle avoidance for UAVs. IEEE Sensors Journal, 20(24), 15281–15289. https://doi.org/10.1109/JSEN.2020.3029122

[20] Tai, L., Paolo, G., & Liu, M. (2017). Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. IROS, 31–36. https://doi.org/10.1109/IROS.2017.8202134

[21] Zhou, Y., Zhou, B., & Wang, J. (2021). LiDAR-based SLAM and motion planning for UAVs in cluttered environments. Sensors, 21(18), 6058. https://doi.org/10.3390/s21186058Bottom of Form

[22] Fei, H.; Mokhtarzadeh, A. A.; Zhou, J.; & Geng, J. "Feature-Fusion Panoramic Segmentation via Deep Neural Networks." Journal of Physics: Conference Series 2467 (2023): 012006. DOI 10.1088/1742-6596/2467/1/012006.

[23] Lin, J.; Gu, Z.; Mokhtarzadeh, A. A.; Chen, X.; Ashim, K.; & Shi, K. "A Fast Humanoid-Robot Arm for Boxing Based on Servo Motors." In Proc. Int. Conf. High-Performance Big Data and Intelligent Systems (HPBD&IS), Macau, 2021, 252–255. DOI 10.1109/HPBDIS53214.2021.9658471.

[24] Song, L.; & Mokhtarzadeh, A. A. "Automatic-Charging Method for Quadruped Robots." Journal of Physics: Conference Series 2467 (2023): 012028. DOI 10.1088/1742-6596/2467/1/012028.

[25] Song, L.; & Mokhtarzadeh, A. A. "Automatic-Charging Method for Quadruped Robots." Journal of Physics: Conference Series 2467 (2023): 012028. DOI 10.1088/1742-6596/2467/1/012028.

[26] Li, S.; Mokhtarzadeh, A. A.; Gao, H.; & Zhang, Y. "Pose-Aware Multi-Position Feature Network for Driver-Distraction Recognition." Journal of Physics: Conference Series 2467 (2023): 012013. DOI 10.1088/1742-6596/2467/1/012013.

[27] Amir Ali Mokhtarzadeh (2023). "Robotics and AI." Nanjing: Nanjing University Press, 2023.