



RESEARCH ARTICLE

A visual SLAM-based lightweight multi-modal semantic framework for an intelligent substation robot

Shaohu Li¹ , Jason Gu², Zhijun Li^{3,4}, Shaofeng Li⁵, Bixiang Guo⁶, Shangbing Gao⁶, Feng Zhao⁶, Yuwei Yang⁷, Guoxin Li⁴  and Lanfang Dong⁸

¹Institute of Advanced Technology, University of Science and Technology of China, Hefei, China

²Department of Electrical and Computer Engineering, Dalhousie University, Halifax, Canada

³School of Mechanical Engineering, Tongji University, Shanghai, China

⁴Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China

⁵State Grid Anhui Electric Power Company Suzhou Power Supply Company, Suzhou, China

⁶State Grid Anhui Electric Power Company Fuyang Power Supply Company, Fuyang, China

⁷Department of Automation, University of Science and Technology of China, Hefei, China

⁸School of Computer Science and Technology, University of Science and Technology of China, Hefei, China

Corresponding author: Zhijun Li; Email: zjli@ieec.org

Received: 27 May 2023; **Revised:** 24 February 2024; **Accepted:** 22 March 2024

Keywords: vSLAM; deep learning; multi-modal framework; substation inspection

Abstract

Visual simultaneous localisation and mapping (vSLAM) has shown considerable promise in positioning and navigating across a variety of indoor and outdoor settings, significantly enhancing the mobility of robots employed in industrial and everyday services. Nonetheless, the prevalent reliance of vSLAM technology on the assumption of static environments has led to suboptimal performance in practical implementations, particularly in unstructured and dynamically noisy environments such as substations. Despite advancements in mitigating the influence of dynamic objects through the integration of geometric and semantic information, existing approaches have struggled to strike an equilibrium between performance and real-time responsiveness. This study introduces a lightweight, multi-modal semantic framework predicated on vSLAM, designed to enable intelligent robots to adeptly navigate the dynamic environments characteristic of substations. The framework notably enhances vSLAM performance by mitigating the impact of dynamic objects through a synergistic combination of object detection and instance segmentation techniques. Initially, an enhanced lightweight instance segmentation network is deployed to ensure both the real-time responsiveness and accuracy of the algorithm. Subsequently, the algorithm's performance is further refined by amalgamating the outcomes of detection and segmentation processes. With a commitment to maximising performance, the framework also ensures the algorithm's real-time capability. Assessments conducted on public datasets and through empirical experiments have demonstrated that the proposed method markedly improves both the accuracy and real-time performance of vSLAM in dynamic environments.

1. Introduction

Simultaneous localisation and mapping (SLAM) has seen widespread application across numerous domains, including intelligent substations [1] and exoskeleton robots [2, 3], to facilitate safe and stable navigation for mobile and manipulative robots undertaking varied tasks. The unstructured nature of substations necessitates that the SLAM technology employed by robots exhibits robust generalisation capabilities to accommodate fluctuations in the environment [4]. Robots designed for intelligent substations are instrumental in executing a myriad of tasks, leveraging functionalities such as map construction [5], autonomous positioning, path planning [6], and the identification and retrieval of equipment.

Laser-based SLAM predominantly relies on lasers to reconstruct environments for positioning and mapping purposes. Nonetheless, 2D laser systems offer insufficient data, while 3D laser systems are

prohibitively expensive. In contrast, visual sensors employed in Visual SLAM (vSLAM) have witnessed significant advancements owing to their cost-effectiveness and the rich semantic information they provide. However, the efficacy of most vSLAM systems is predicated on the assumption of static environments [7], which restricts their overall robustness.

The concept of keyframe-based vSLAM has rapidly evolved, attributed to its minimal computational demand and high precision [8]. This approach focuses on pose optimisation and map creation through the establishment of feature points and keyframes. ORB-SLAM3 [9] exemplifies this, achieving enhanced speed and accuracy in computation by linking keyframes with active maps. In static settings, the correlation between feature points typically exhibits strong geometric consistency. Conversely, in dynamic environments, the clarity of these geometric correlations can diminish. Techniques such as local bundle adjustment (BA) and loop closure detection are employed to reduce the adverse effects of dynamic objects on cross-frame pose estimation. Despite these advancements, the challenge posed by dynamic objects remains, affecting the stability of sophisticated vSLAM systems. The utility of neural networks in improving robotic perception is increasingly recognised as a solution to the challenges presented by dynamic environments [10].

In recent years, the integration of deep learning with simultaneous localisation and mapping has witnessed rapid advancements. Detect-SLAM [11] employs object detection to enhance robustness in dynamic environments, albeit at the cost of reducing the number of available static feature points. Previous literature [12] has demonstrated that instance segmentation can effectively eliminate dynamic feature points, yet this approach significantly compromises real-time performance. Further studies [13–16] have explored the amalgamation of deep learning and clustering techniques to efficaciously remove feature points, although the definition of clustering hyperparameters presents challenges and necessitates reliance on more precise depth maps. Moreover, the integration of multi-view geometry with deep learning, as discussed in refs. [17, 18], overlooks the semantic information pertinent to feature points.

This work introduces novel approaches for the detection of dynamic feature points and the fusion of multiple modules. Initially, an enhanced lightweight neural network is employed to identify dynamic objects through detection frames and segmentation results, subsequently using these results alongside the positional relationships of feature points to generate a mask for dynamic feature points. Tailored improvements, predicated on the characteristics of vSLAM, are implemented within the network to facilitate increased speed and accuracy. Furthermore, a dynamic threshold method is introduced to ascertain the maximal number of features that can be feasibly removed, offering superior adaptability to environmental variations in comparison to the deterministic threshold and movement capability approaches referenced in ref. [11]. Lastly, within the multi-module fusion process, the object detection framework is integrated into the backend local map to validate and adaptively determine the operation of the instance segmentation modules. This method, in contrast to the deterministic operating modes discussed in refs. [13, 14], effectively consolidates individual modules, curtailing overall running time and bolstering stability.

The main contributions of this paper include:

- The development of a lightweight neural network tailored for multimodal semantic vSLAM, featuring a rapid and efficient backbone network structure alongside decoupled headers for enhanced parameter sharing.
- The proposition of a novel framework that adeptly merges deep learning with vSLAM, wherein the concurrent detection and segmentation technique substantially improves both the speed and accuracy of dynamic feature point detection within vSLAM.
- The introduction of a lightweight semantic vSLAM framework capable of automatic environmental adaptation, with experiments on datasets and in real-world scenarios demonstrating its superiority over existing vSLAM methodologies in terms of speed and accuracy.

2. Related work

In scenarios characteristic of substations, the requirement for humans and machines to collaborate necessitates an environment that is frequently dynamic rather than static. Robots are thus required to possess the capability to comprehend complex scenarios and task processes [19, 20]. This necessitates the deployment of vSLAM technologies capable of managing feature points on dynamic objects while simultaneously balancing performance with efficiency.

Multi-view geometric verification methods utilise reprojection errors between successive frames to identify dynamic objects. Zhang et al. [21] employ the residuals of relative positions derived from dense optical flow between consecutive frames for dynamic object segmentation. Dai et al. [22] leverage the relative positional invariance of map points to distinguish moving objects across successive frames. The principal advantage of geometric methods lies in their capacity to deduce the mobility of objects within the environment through the analysis of image relations across different frames, without the need for extensive prior information. However, these methods do not adequately consider the overall semantic relationships between feature points.

Object detection methods differentiate between dynamic and static feature points through the extent of the detection bounding box. MOD-SLAM [23] integrates an object detection module that utilises the results from detection boxes to eliminate dynamic features. Bao et al. [24] introduced a method that initially applies detection results obtained from the detection module, subsequently determining whether to proceed with semantic segmentation based on the quantity of static feature points. Such targeted detection algorithms enhance the robustness of vSLAM in dynamic environments with minimal impact on real-time performance. Nonetheless, the discrepancy between the size of the detection frame and the actual contours of objects constrains the effectiveness of object detection within vSLAM systems.

Semantic segmentation-based approaches remove dynamic feature points based on segmentation results. DynaSLAM [25] combines Mask R-CNN [26] with multi-view geometry to exclude dynamic feature points, albeit at the expense of significantly affecting the real-time performance of vSLAM. DS-SLAM [17] suggests executing geometry checks and Segnet [27] in parallel threads to accurately identify dynamic object masks, though improvements in speed are not markedly evident. Ji et al. [14] proposed a combined approach of semantic segmentation and clustering, wherein dynamic object masks are acquired through segmentation of keyframes using Segnet and clustering on the depth graph with Kmeans [28], achieving notable speed and segmentation accuracy. However, this approach is contingent upon the sensor's ability to accurately gauge depth, and its effectiveness is notably influenced by the settings of hyperparameters.

3. System description

Experimental validation was conducted using the developed substation robotic equipment, as depicted in Fig. 1. This apparatus encompasses a mobile chassis, a hydraulic lifting platform, a robotic arm, several RGB-D cameras, and a central processing unit equipped with an Intel i9-10900X CPU and an NVIDIA Quadro RTX 6000 GPU. The hardware components and their respective functionalities are outlined as follows:

- The mobile chassis affords the substation robot mobility and directional steering capabilities. Equipped with four-wheel steering, the chassis is designed to minimise the turning radius, thereby enhancing manoeuvrability in confined spaces. Furthermore, its dual-drive architecture at the front and rear facilitates navigation over complex terrains.
- The hydraulic lifting platform primarily serves to extend the vertical operational range of the robotic arm, allowing it to reach varying heights across different scenarios. This design feature effectively addresses the constraints associated with working within a singular vertical space.

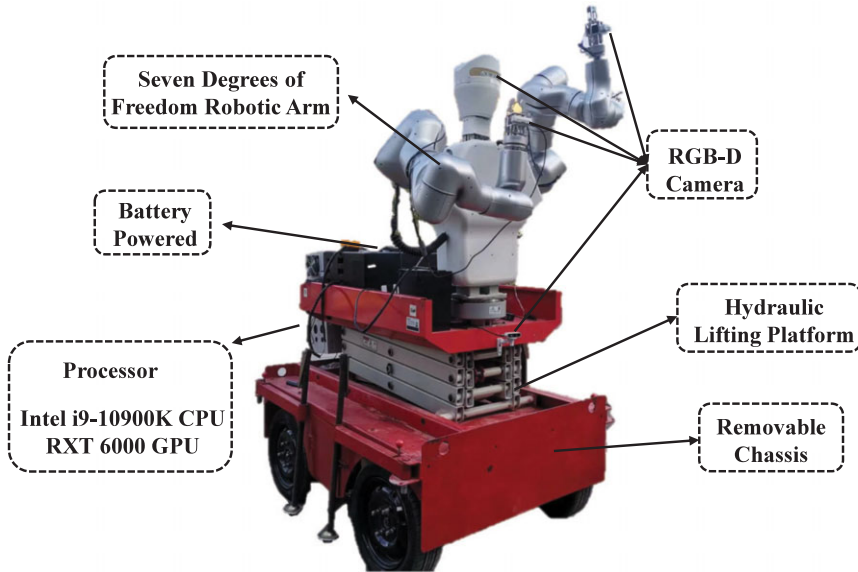


Figure 1. An overview of the smart substation robot platform.

- The robotic arm, featuring seven degrees of freedom, enables precise grasping and manipulation within intricate environments. Additionally, it is outfitted with a terminal rotation motor at its front end, facilitating tasks such as tightening and disassembling components.
- Multiple RGB-D cameras are strategically positioned on various parts of the substation robot to cater to the demands of diverse tasks. The camera located on the robot's head offers a comprehensive view and depth perception for the robotic arm's operational area. The camera attached to the extremity of the arm delivers finer resolution images and depth detail for precise manipulation tasks. Meanwhile, the camera mounted on the mobile chassis provides consistent height and positional imagery for accurate positioning and navigation, ensuring the stability of the external parameter matrix between the chassis and its environment.
- The central processor, boasting significant computational power, orchestrates the substation robot's perception, control, planning, and locomotion. It assimilates and processes data from the robotic arm, the RGB-D cameras, and the chassis, integrating this information to furnish real-time feedback and control over the robot's operations.

4. Method

The methodology proposed herein amalgamates object detection and instance segmentation techniques to eliminate dynamic objects from scenes. Leveraging the profound capability of neural networks to decipher semantic information, this approach efficiently removes a majority of the dynamic feature points. The integration of object detection and instance segmentation methodologies is designed to address the limitations inherent in each individual module. As illustrated in Fig. 2, feature extraction is conducted via the backbone network, with object detection and instance segmentation modules appended to each frame to identify dynamic entities. Within the vSLAM tracking thread, the results from instance segmentation are employed to enhance the precision of pose optimisation. Concurrently, in the local mapping thread, the outcomes of object detection and instance segmentation are amalgamated to refine the delineation of dynamic objects.

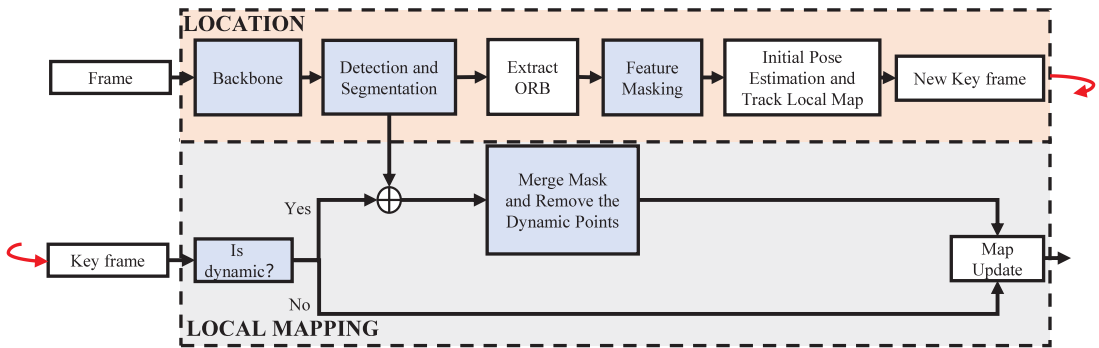


Figure 2. The overall flow chart of the system. The blue blocks are modules we added in ORB-SLAM3.

4.1. Location

Neural networks have been incorporated within the tracking thread to detect and segment dynamic objects. In selecting an appropriate network, both the performance and efficiency of the algorithm were considered, leading to the adoption of YOLOv8 [29] as the primary network. YOLOv8 is capable of conducting both object detection and instance segmentation concurrently, offering improvements in speed and accuracy over previously suggested methods. However, the exigencies of vSLAM in dynamic settings necessitate even faster neural network performance. The original pretraining weights, optimised for the eighty categories of the COCO dataset, include several categories irrelevant to dynamic vSLAM scenarios. Consequently, a subset of 20 pertinent dynamic and static object categories from COCO was chosen for detection. The YOLOv8s-seg model was selected as the foundational model, with proposed enhancements aimed at rendering it more lightweight without compromising algorithmic performance.

4.1.1. Network improvements

The performance of the target detection and instance segmentation network is significantly influenced by its 'Neck' component. Models constructed using an abundance of depthwise separable convolutional layers fail to achieve the desired level of accuracy. This paper proposes the incorporation of GSConv [30] into the 'Neck' of the YOLOv8s-seg network to diminish model complexity while preserving accuracy. Specifically, GSConv is integrated with the VoVGSCSP module, wherein the VoVGSCSP module supersedes the C2f module in the original YOLOv8 architecture. This modification results in a more efficient 'Slim-Neck', thereby enhancing the network's computational cost-effectiveness. The GSConv and VoVGSCSP modules are depicted in Fig. 3.

GSConv aims to minimise the loss of semantic information during the process of channel expansion and feature map reduction. However, its implementation within the backbone network incurs significant computational overhead. The adjustment of channel numbers and feature map dimensions in the 'Neck' section, conversely, is more judicious, thus not excessively augmenting inference time consumption. Given the necessity to fuse feature maps across varying channels, an attention mechanism is requisite. Hence, informed by the findings of [30], GSConv has been integrated into the 'Neck' section of the YOLOv8 architecture, culminating in the formation of the YOLOv8s-seg-Slim-Neck structure. This adaptation not only curtails computational demands but also augments the network's efficacy in both detection and segmentation tasks. Detailed experimental comparisons are provided in the subsequent experimental section.

YoloV8 utilises a decoupled head design, segregating the predictions of classification and bounding boxes into distinct branches. It has been observed that the decoupled head of YoloV8 harbours a relatively large parameter count, which, for network deployment, could potentially impair real-time performance. To mitigate this, parameters of certain layers within the decoupled head are shared, and a

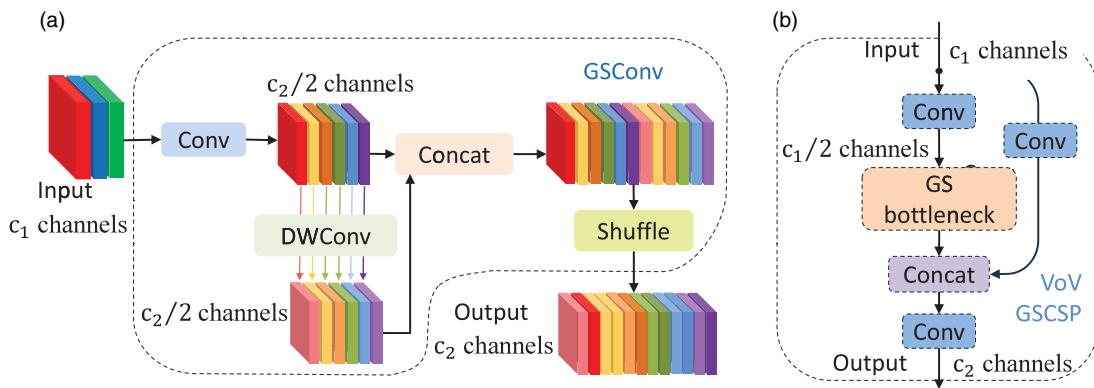


Figure 3. (a) The composition of the GSConv module. (b) The VoVGSCSP module is composed of GSConv.

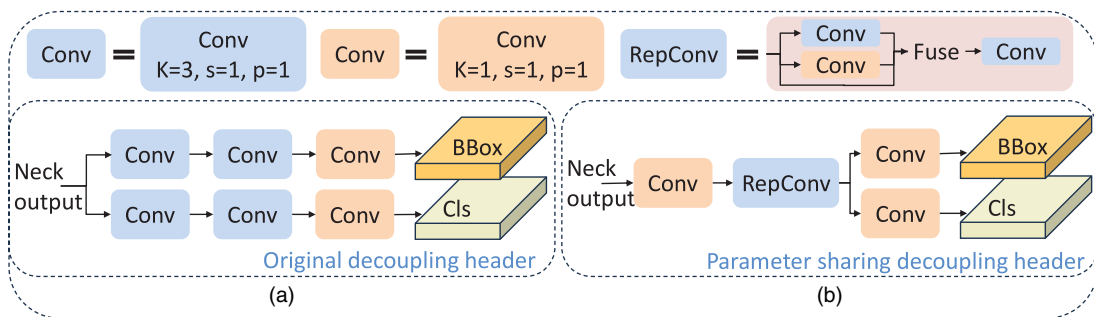


Figure 4. (a) Original decoupling header in yolov8. (b) A decoupling header for parameter sharing has been added to yolov8.

convolutional module endowed with a re-parameterization mechanism (RepConv) [31] is introduced to offset the performance decrement occasioned by the reduction of parameters, as illustrated in Fig. 4.

RepConv represents a convolutional module that reallocates the parameters of its internal convolution through computational fusion. This design is more conducive to hardware compatibility, yet a straightforward network architecture might diminish the network’s feature extraction capability and the path of gradient flow. Consequently, during training, RepConv incorporates multiple branches within the module to enhance the network’s feature extraction capacity. Nevertheless, in the inference phase, these multiple computational modules are consolidated into a single entity to enhance both the efficiency and performance of the model.

For instance segmentation, the network’s output header adopts a simplistic structure. Reductions in convolution layers and channels confer marginal improvements in speed, yet precipitate a considerable diminution in network precision.

Both the slim-neck structural refinement and the implementation of a parameter-sharing decoupled head through RepConv facilitate a substantial reduction in parameters, enhancing the inference speed of YoloV8 without compromising its performance. These enhancements have been amalgamated to formulate the final YOLOv8s-seg-gs-rep network architecture. To further augment network velocity, the TensorRT inference framework is employed during network deployment, thereby expediting the integration of the network with vSLAM. The efficacy of these improvement techniques has been empirically validated within the ablation study section, elucidating the influence of each enhancement on both speed and accuracy.

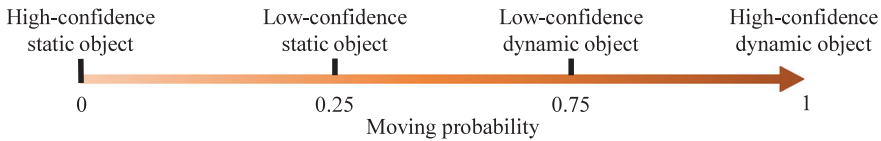


Figure 5. Movement confidence thresholds for different objects.

4.1.2. Posture optimisation

In the context of front-end pose optimisation within vSLAM, the outcomes of instance segmentation are utilised to obscure dynamic feature points. The processed image, courtesy of the refined YOLOv8s-seg network, facilitates the acquisition of segmentation results for currently active objects. These results are subsequently amalgamated into a comprehensive mask map, excluding dynamic feature points from the pose optimisation process.

The segmentation of dynamic objects necessitates the predefinition of such entities based on artificial priors. Nonetheless, the inherent mobility of an object is not a definitive attribute; hence, variable segmented mobility values are ascribed to distinct objects.

The methodology for allocating mobility segments is delineated in Fig. 5. Three intervals are equidistantly distributed between 0 and 1, with the quartile thresholds representing varying probabilities of movement. For instance, a value of 1 denotes a human, indicative of a high probability of dynamic movement; 0.75 pertains to a sports ball, signifying a moderate probability of motion; and 0.25 is attributed to a keyboard, suggesting a minimal likelihood of movement, thereby classifying it as a static object with low confidence.

The presence of dynamic points within an image can significantly consume spatial resources. Utilising a fixed threshold method or a binary mobility capability assignment could result in the failure of optimisation efforts. To circumvent this, four methods of mobility capability assignment alongside dynamic threshold techniques are employed to bolster the algorithm’s robustness. The dynamic threshold approach operates as follows:

$$d_{point} = \begin{cases} static, & \text{if } m \leq m_{th} \\ dynamic, & \text{otherwise} \end{cases}, \tag{1}$$

where d_{point} is a flag of whether the feature points are dynamic. m means the mobility of the category to which the current feature point belongs. m_{th} represents the threshold for judging the dynamics of feature points, which is 0.25 at the beginning. The formula only calculates the dynamics of feature points within the segmentation mask, and those not within the segmentation mask are considered static. If the number of static points does not meet the set minimum number, m_{th} needs to be adaptively increased by 0.1 until it reaches the maximum of 1. If it is not satisfied when m_{th} is 1, dynamic objects may fill the entire picture, and we would give up this dynamic object optimisation.

When the remaining static points meet the needs of attitude estimation, the camera poses $T_{wc} \in SE(3)$ would be estimated and optimised using BA. The camera poses T_{wc} , which comprises the camera’s rotation R and position t . The set of static feature points on the current image is p_c . The coordinate set of the points in the three-dimensional world matched by p_c is P . The optimisation equation is as follows:

$$p_e = argmin \frac{1}{2} \sum_{i=1}^n \|p_{c_i} - \pi T_{wc}(P_i)\|^2, \tag{2}$$

where p_e is the camera pose iterative optimisation error, π is the inherent parameter of the camera, which is the reprojection model from the three-dimensional coordinates to the camera coordinate system, and n is the number of matching feature points between the three-dimensional space and the two-dimensional space. A more accurate camera pose T_{wc} can be obtained by continuously iterating to reduce the reprojection error.

4.2. Local mapping

Enhancing the efficiency and reducing the complexity of neural networks may compromise their performance. To mitigate this issue within the backend of vSLAM, we employ a technique that combines detection bounding boxes and strengthens segmentation outcomes. However, this adjustment need not be applied to every backend keyframe but rather to those keyframes exhibiting significant dynamic characteristics. Consequently, we utilise the outcomes from the front-end detection of dynamic feature point numbers and employ formula 3 to assess the dynamism of the current frame.

$$d_{frame} = \begin{cases} False, & \text{if } m_{cd} \leq 1/N, m_{th} = 0.5 \\ True, & \text{otherwise,} \end{cases} \quad (3)$$

where m_{cd} represents the number of dynamic feature points in the front-end under the threshold $m_{th} = 0.5$, N represents the total number of feature points in the current image, and d_{frame} is a flag of whether the frame is dynamic. Only dynamic keyframes perform joint optimisation of object detection and instance segmentation.

Given the compact nature of the base network model, instance segmentation may yield inaccurate results or fail altogether. To address this challenge, we introduce a strategy whereby the detection and segmentation heads operate in a complementary fashion. Specifically, we juxtapose the outcomes of object detection bounding boxes with those of instance segmentation. Should the ratio s_o , representing the quotient of the instance segmentation area to the object detection bounding box area, fall below a minimal threshold t_o , the instance segmentation result for the current object is deemed unsuccessful, and the bounding box mask is adopted as the definitive mask. This comparative analysis of object detection and instance segmentation results facilitates the calculation of the overlap area, utilising the indices provided by the model's object detection and instance segmentation heads. The mask calculation formula of the jointly optimised dynamic object is as follows:

$$obj_{mask} = \begin{cases} seg_{mask}, & \text{if } s_o > t_o \\ det_{mask}, & \text{if } s_o \leq t_o, \end{cases} \quad (4)$$

where seg_{mask} is the mask obtained from instance segmentation, and det_{mask} is the mask obtained from object detection. t_o is selected as 0.25.

The outcome post-fusion serves as the definitive mask, enabling the removal of dynamic points during the back-projection and optimisation processes. This approach ensures the precise exclusion of keyframes and dynamic points from the map. Whether in map matching or keyframe matching, the influence of dynamic objects is markedly diminished. The algorithm proposed herein adeptly balances speed and performance, demonstrating versatility across varied environments.

5. Experiments and results

5.1. Ablation experiment

In the realm of network lightening, it is imperative to minimise any reduction in model accuracy. Should a module manage to decrease the number of parameters without compromising, or indeed enhancing, the model's accuracy, it would thereby elevate the cost-efficiency of the optimisation process. The selection of network modules must also take into account the compatibility of the network structure with the hardware platform. The GSConv module redefines convolution operations, enriching the network with enhanced gradient flow and informational exchange. The RepConv module, during training, leverages multiple convolution combinations and, throughout inference, employs convolution merging to optimise the utilisation of convolution modules. Furthermore, RepConv exhibits considerable compatibility with hardware. Both modules contribute to rendering the network more efficient and faster. However, considering the YOLOv8-seg model necessitates attention to both detection and segmentation heads, prompting thorough comparative experimentation. Experiments were conducted on 20

Table I. Twenty categories selected from COCO.

Person	Bicycle	Car	Motorcycle	Bird	Cat	Dog
Backpack	Umbrella	Handbag	Suitcase	Sports ball	Bottle	Chair
Tv	Laptop	Mouse	Keyboard	Cell phone	Refrigerator	

Table II. Comparison of combined experiments between different modules.

Model	Size (pixels)	Box mAP (0.5:0.95)	Seg mAP (0.5:0.95)	FLOPs (@640 (G))	Gradients (M)	Params (M)	epoch
yolov8s-seg	640	46.2	39.5	42.7	11.8	11.8	500
yolov8s-seg-rep	640	46.1	39.8	40.8	10.1	12.2	500
yolov8s-seg-gs	640	46.7	40.1	39.4	10.9	10.9	500
yolov8s-seg-gs-rep	640	46.8	40.3	37.4	9.2	10.2	500

specifically chosen categories, as delineated in Table I, to ascertain the efficacy of the proposed module amalgamation.

A suite of evaluative metrics was employed in comparative analyses between the refined and original networks. The dataset for comparison was compiled by extracting relevant data from the original COCO dataset, as per the categories listed in Table I, with the original COCO test set serving as the evaluation benchmark. The comparative outcomes are presented in Table II.

The evaluation criteria used in Table II are as follows: *size_pixels* represents the size of the image input by the network, *mAP* represents the average accuracy of each detection category under a fixed IOU threshold, *mAP_{0.5:0.95}* represents the average of all *mAP* with IOU thresholds from 0.5 to 0.95, which can represent generalisation performance of the network. *Box mAP_{0.5:0.95}* represents the generalisation performance of the detection box, *seg mAP_{0.5:0.95}* represents the generalisation performance of segmentation, and *FLOPs* represents the network required floating point operations are in giga. *Gradients(M)* represents the number of gradients in millions when the network is backpropagated. *Params(M)* represents the total number of parameters owned by the network in millions, and *epoch* represents the number of iterations of the data by the training network.

We found that using GSConv in the neck part to form a slim-neck structure and using the RepConv structure in the parameter sharing decoupling header can achieve higher inference speed and lower parameter amount. At the same time, the network's performance is also optimal compared to other structures.

5.2. Simulation experiment

5.2.1 TUM and EUROC dataset

The EUROC dataset [32] and the TUM RGB-D dataset [33] serve as benchmarks for evaluating the performance of vSLAM across various scenarios. These datasets facilitate a comprehensive assessment of the method proposed herein. Predicated upon the foundational ORB-SLAM3, the proposed algorithm demonstrates comparable performance on datasets characterised by lower levels of mobility. Nevertheless, it substantially surpasses the original ORB-SLAM3 in sequences featuring significant object movement. The focus of our experiments is on sequences with high dynamics, specifically *fr3/walking_**, with outcomes juxtaposed against those obtained with the original ORB-SLAM3 and other contemporary state-of-the-art methodologies.

5.2.2 Evaluation metrics

The evaluation of vSLAM principally involves measuring the discrepancy between the trajectory delineated by the algorithm and the actual trajectory. Absolute trajectory error (ATE) and relative pose error

Table III. Comparison with ORB-SLAM3 using ATE as evaluation metric [m]. Highlight the best results in bold.

Sequence	ORB-SLAM3		Ours	
	RMSE	Std	RMSE	Std
<i>fr3/walking_xyz</i>	0.2686	0.1095	0.0179	0.0084
<i>fr3/walking_static</i>	0.0174	0.0103	0.0081	0.0034
<i>fr3/walking_half</i>	0.1759	0.0683	0.0247	0.0124
<i>fr3/walking_rpy</i>	0.1507	0.0681	0.0816	0.0470

(RPE) are commonly employed as evaluative metrics, with their root mean square error typically utilised to gauge accuracy. ATE quantifies the absolute discrepancy (in metres) between the actual and estimated positions across all frames, whereas RPE assesses the error in relative pose estimation (in radians). The standard deviation (std) is employed to quantify the dispersion of values within a dataset, signifying the mean distance between data points and the dataset's average. A larger standard deviation indicates a more dispersed distribution, directly reflecting the global trajectory algorithm's precision. RPE measures the variance in positional changes at identical timestamps, encompassing translation error (in metres per second) and rotation error (in degrees per second). The final error metric is derived by averaging the outcomes of multiple experiments conducted on the same dataset.

5.2.3 Implementation configuration

The neural network model is seamlessly integrated within the vSLAM framework. Utilising TensorRT, the neural network is converted into a TRT (fp16) model, with both the pre-processing and post-processing stages of the network parallelised and executed on the GPU. The entire algorithmic suite is developed in C++. This approach optimises the network's speed while maintaining accuracy.

In operational scenarios within substations, enhancing the algorithm's performance is paramount, with a particular emphasis on accelerating its execution speed. Hence, in comparison to our prior work [34], the methodology we propose achieves a two- to three-fold increase in speed, simultaneously ensuring enhanced precision and greater stability.

The proposed algorithm necessitates the input of an RGB image, functioning effectively provided the selected hardware is capable of supplying RGB images. For instance, when employing stereo vision, the left-eye image is utilised as input.

5.3. Experimental comparison

5.3.1 Comparison with ORB-SLAM3

To ascertain the efficacy of the proposed method, we undertook experimental evaluations across various datasets. Ensuring the reliability of our findings, we performed tests on our apparatus using both the original ORB-SLAM3 and our proposed method, facilitating a direct comparison of outcomes. For a more intuitive analysis of the experimental data, we intend to produce a comparative chart of the trajectories. The sequences *fr3/walking_xyz*, *fr3/walking_static*, *fr3/walking_half*, and *fr3/walking_rpy* from the TUM dataset, each exhibiting varying levels of dynamic activity, were selected for this comparative analysis. The outcomes of this comparison are delineated in Table III and illustrated in Fig. 6.

The experimental outcomes indicate that the proposed methodology yields results comparable to those of ORB-SLAM3 under static conditions. In scenarios marked by minimal object movement, the outlier optimisation mechanism utilised by ORB-SLAM3 proves effective in discarding dynamic points, with outcomes closely mirroring those of our proposed approach. However, in environments characterised by considerable object movement, ORB-SLAM3's optimisation methods and mechanisms encounter difficulties in adequately addressing the influence of dynamic objects. This

Table IV. The absolute trajectory error (ATE) is used as the standard for comparison. [m].

Sequence	RGB-D	Detect	KMOP	RGBD	Rts	DS-S	Dyna	Ours
fr3/walking_xyz	0.0874	0.0241	0.0190	0.0140	0.0194	0.0247	0.0150	0.0179
fr3/walking_static	0.0108	-	0.0320	0.0100	0.0111	0.0081	0.0060	0.0081
fr3/walking_half	0.0354	0.0514	0.1760	0.0280	0.0290	0.0303	0.0250	0.0247
fr3/walking_rpy	0.1608	0.2959	0.0490	0.0330	0.0371	0.4442	0.0350	0.0816

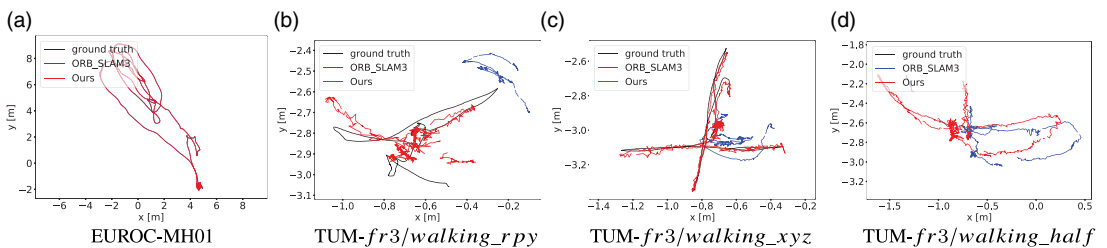


Figure 6. Comparison of trajectory estimation accuracy of the proposed method with ORB-SLAM3. (a) The EUROC datasets V101 were generated in a static environment. (b), (c) and (d) The TUM dataset includes fr3/walking_rpy, fr3/walking_xyz and fr3/walking_half, which were collected in a high-motion environment.

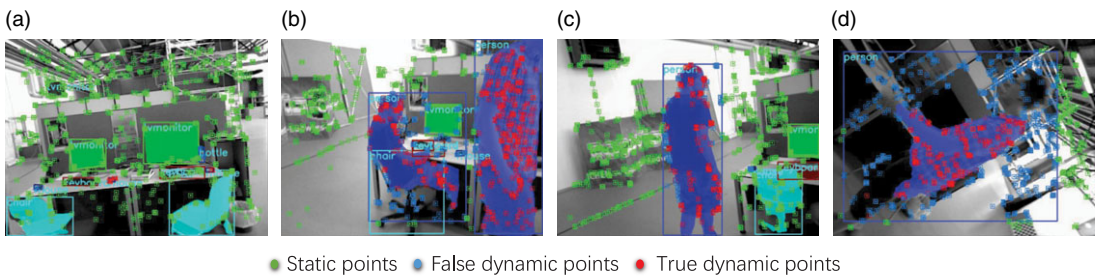


Figure 7. Interception of experimental results. (a). Results under static scenes. (b), (c) and (d). Experimental screenshots under different dynamic scenarios.

challenge becomes increasingly evident as the frequency of object movement escalates or in consistently high-motion scenes, leading to a notable decline in ORB-SLAM3’s performance. Conversely, our optimised method maintains commendable performance under these conditions. Throughout the experimentation phase, we garnered specific insights, illustrated in Fig. 7, which portrays the precision in eliminating dynamic objects.

5.3.2 Comparison with other vSLAM

Furthermore, the integration of deep learning into vSLAM is gaining traction, prompting a comparison of our method with several leading-edge vSLAM technologies, including Detect-SLAM [11], KMOPvSLAM [13], Rts-SLAM [14], RGBD-SLAM [15], DS-SLAM [17], DynaSLAM [25], and RGB-D SLAM [22]. The comparative analysis focuses on both accuracy and speed. While some dynamic SLAM systems exhibit rapid processing speeds, their accuracy leaves much to be desired. Others, such as DynaSLAM [25], demonstrate superior performance in dynamic environments but are constrained to offline execution, thus precluding real-time operation. The comparative outcomes, employing ATE as a metric, are presented in Table IV. Additionally, the results for RPE, encompassing

Table V. Comparison of root mean square error (RMSE) for translational drift in meters per second (m/s) and RMSE for rotational drift in degrees per second ($^{\circ}/s$).

Sequence	Translational RPE				Rotational RPE			
	KMOP	DS-S	Rts	Ours	KMOP	DS-S	Rts	Ours
<i>fr3/walking_xyz</i>	0.0260	0.0333	0.0234	0.0217	0.6890	0.8266	0.6368	0.5412
<i>fr3/walking_static</i>	0.0330	0.0102	0.0117	0.0971	0.6270	0.2690	0.2872	0.0234
<i>fr3/walking_half</i>	0.0700	0.0297	0.0423	0.0243	1.5950	0.8142	0.9650	0.7230
<i>fr3/walking_rpy</i>	0.0650	0.1503	0.0471	0.0597	1.1050	3.0042	1.0587	1.3122

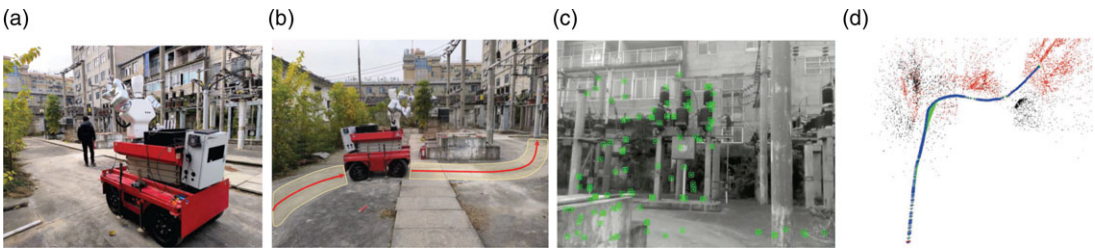


Figure 8. Experiments in different real-world scenarios. (a). Experimental environments with fast moving objects. (b). Realistic route of substation robot movement. (c). Screenshots of the experimental process. (d). Trajectory maps generated in mobile environments.

both translation and rotation, are summarised in Tables V, underscoring the advanced nature of the proposed method.

5.4. Real-world experiment

Real-world experiments were carried out using robotic equipment within substations, where substation robots are primarily deployed to support staff in routine maintenance and inspection tasks, operating amidst complex scenes populated with dynamic objects.

Initially, the efficacy of the proposed vSLAM method was assessed through the odometry technique. The integration of multi-sensor odometry fusion methods has gained widespread acceptance for enhancing mobile robot localisation and navigation due to its straightforward application, acceptable error margin, and cost-effectiveness [35, 36, 37]. In the context of substations, where privacy concerns preclude the use of GPS, our experiments relied on the fusion of IMU and encoder-based odometry as the benchmark for validating the proposed vSLAM method's effectiveness. Encoders fitted on both the front and rear steering mechanisms, coupled with IMU data from the bio-camera, ensure precise odometry. Moreover, the odometry fusion method delineated in prior research [36] was employed.

The field experiment within the substation is depicted in Fig. 8a, where the algorithm's capacity to identify dynamic objects and its efficacy in eliminating such objects were tested by an experimenter swiftly moving in front of the substation robot. Figure 8b illustrates the trajectory of the substation robot as it advances in the real-world scenario, with the procedural outcomes presented in Fig. 8c. As demonstrated in Fig. 8d, the generated map remains unimpacted by dynamic entities, underscoring the superiority of the proposed algorithm in practical applications.

Time efficiency serves as a critical metric for evaluating an algorithm's performance. To ascertain the temporal demands of each module within the proposed method, experiments were conducted to record the operational duration of each component under dynamic and static conditions. Table VI enumerates the average running times. In dynamic keyframe scenarios, the algorithm necessitates the amalgamation of detection and segmentation results.

Table VI. Time consumption of each part [ms]. The RMSE comparison uses the results of ORB-SLAM3 as the relative benchmark.

Methods	Segmentation	Fusion results	Tracking	Inference time per frame	Average RMSE ratio
ORB_SLAM3	0?>	0	31	31.0	1.0
Ours	2.4	0.3/ 0	31	33.7	0.5

Table VII. Comparison of computation time [ms].

Methods	Segmentation (Tracking)	Segmentation (Local mapping)	Geometry
DS-SLAM	0	76.0	48.5
DynaSLAM	0	885.1	587.6
Rts-SLAM	0	72.8	30.14
Ours	20.4	0	-

Within static keyframes, the algorithm obviates the need for fusion, thereby not incurring additional time expenditure. This observation underscores that the proposed method operates efficiently without compromising real-time performance. To further substantiate the algorithm's real-time capabilities, experiments were executed on the NVidia Jetson AGX Xavier, equipped with a 512-core Volta GPU, an 8-core ARM 64-bit CPU, and 16 GB of RAM. The results of these comparative studies on the temporal demands of various algorithms are presented in Table VII.

6. Conclusion and discussion

This study introduces a lightweight, multi-modal semantic SLAM framework designed to enhance mapping accuracy while diminishing the computational demands of the fusion process. Utilising an instance segmentation model, the framework adeptly identifies and eliminates dynamic objects within each frame, thereby ensuring the integrity of map generation. The foundation of this framework is the incorporation of state-of-the-art detection models and a segmentation head, predicated on the understanding that the amalgamation of neural networks with SLAM requires not the utmost segmentation precision but a balanced approach. Moreover, the adoption of lightweight strategies aims to reduce computational burdens while maintaining the efficacy of the neural network model employed. This approach further amalgamates the detection bounding boxes with instance segmentation outcomes, mitigating potential discrepancies in instance segmentation accuracy. The proposed method has demonstrated enhancements in both positional accuracy and operational speed across a variety of scenarios, affirming its innovative contributions.

Nonetheless, the proposed method is not without limitations. Currently, ORB feature points are delineated manually, bypassing the neural network's capability to extract and fortify feature points' robustness. Future endeavours will explore the substitution of the entire front-end process with neural network operations, leveraging the network's potential for feature point extraction. Additionally, efforts will be directed towards integrating the removal of dynamic feature points within the neural network's feature point extraction process, utilising initial semantic insights to dispense with dynamic feature points, thereby cultivating robustness in diverse and intricate environments.

Author contributions. Zhijun Li and Shaohu Li conceived the method, built the framework, designed the experiments, and conducted the theoretical analysis. Shaohu Li, Shaofeng Li, and Bixiang Guo designed and performed the experiments. Shaohu Li processed the data and performed analysis. Jason Gu carried out the discussion and refinement of the paper. Shangbing Gao and Feng Zhao performed the experiments. Yuwei Yang and Guoxin Li assisted in revising the paper. Lanfang Dong gave expertise on multi-modal fusion.

Financial support. This work was supported by the National Key Research and Development Program of China (Grant No. 2020YFB1313602) and the Anhui Provincial Natural Science Foundation, Anhui Energy-Internet Joint Program (No. 2008085UD01).

Competing interests. The authors declare no competing interests exist.

Ethical approval. None.

References

- [1] S. Lu, Y. Zhang and J. Su, "Mobile robot for power substation inspection: A survey," *IEEE/CAA J Automat Sin* **4**(4), 830–847 (2017).
- [2] G. Li, Z. Li, C.-Y. Su and T. Xu, "Active human-following control of an exoskeleton robot with body weight support," *IEEE Trans Cybernet* **53**(11), 7367–7379 (2023).
- [3] Z. Li, G. Li, X. Wu, Z. Kan, H. Su and Y. Liu, "Asymmetric cooperation control of dual-arm exoskeletons using human collaborative manipulation models," *IEEE Trans Cybernet* **52**(11), 12126–12139 (2022).
- [4] J. Li, M. Cong, D. Liu and Y. Du, "Enhanced task parameterized dynamic movement primitives by GMM to solve manipulation tasks," *Robot Intell Automa* **43**(2), 85–95 (2023).
- [5] H. Wang, C. Zhang, Y. Song, B. Pang and G. Zhang, "Three-dimensional reconstruction based on visual SLAM of mobile robot in search and rescue disaster scenarios," *Robotica* **38**(2), 350–373 (2020).
- [6] J. Cai, F. Yan, Y. Shi, M. Zhang and L. Guo, "Autonomous robot navigation based on a hierarchical cognitive model," *Robotica* **41**(2), 690–712 (2023).
- [7] C. Hao, L. Chengju and C. Qijun, "Self-localization in highly dynamic environments based on dual-channel unscented particle filter," *Robotica* **39**(7), 1216–1229 (2021).
- [8] S. Naudet-Collette, K. Melbouci, V. Gay-Bellile, O. Ait-Aider and M. Dhome, "Constrained RGBD-SLAM," *Robotica* **39**(2), 277–290 (2021).
- [9] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Trans Robot* **37**(6), 1874–1890 (2021).
- [10] H. Fei, Z. Wang, S. Tedeschi and A. Kennedy, "Boosting visual servoing performance through RGB-based methods," *Robot Intell Automat* **43**(4), 468–475 (2023).
- [11] F. Zhong, S. Wang, Z. Zhang, C. Chen and Y. Wang, "Detect-SLAM: Making Object Detection and SLAM Mutually Beneficial," *In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, (2018) pp. 1001–1010.
- [12] Q. Ul Islam, H. Ibrahim, P. K. Chin, K. Lim and M. Z. Abdullah, "FADM-SLAM: A fast and accurate dynamic intelligent motion SLAM for autonomous robot exploration involving movable objects," *Robot Intell Automat* **43**(3), 254–266 (2023).
- [13] Y. Liu and J. Miura, "KMOP-vSLAM: Dynamic Visual SLAM for RGB-D Cameras using K-means and OpenPose," *In: Proceedings of the IEEE/SICE International Symposium on System Integration (SII)*, (2021) pp. 415–420.
- [14] T. Ji, C. Wang and L. Xie, "Towards Real-time Semantic RGB-D SLAM in Dynamic Environments," *In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (2021) pp. 11175–11181.
- [15] W. Xie, P. X. Liu and M. Zheng, "Moving object segmentation and detection for robust RGBD-SLAM in dynamic environments," *IEEE Trans Instru Measure* **70**, 1–8 (2021).
- [16] L. Kenye and R. Kala, "Improving RGB-D SLAM in dynamic environments using semantic aided segmentation," *Robotica* **40**(6), 2065–2090 (2022).
- [17] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei and Q. Fei, "DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments," *In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (2018) pp. 1168–1174.
- [18] M. Henein, J. Zhang, R. Mahony and V. Ila, "Dynamic SLAM: The Need For Speed," *In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (2020) pp. 2123–2129.
- [19] J. Li, M. Cong, D. Liu and Y. Du, "Robot task programming in complex task scenarios based on spatio-temporal constraint calculus," *Robot Intell Automat* **43**(4), 476–488 (2023).
- [20] R. Miao, Q. Jia and F. Sun, "Long-term robot manipulation task planning with scene graph and semantic knowledge," *Robot Intell Automat* **43**(1), 12–22 (2023).
- [21] T. Zhang, H. Zhang, Y. Li, Y. Nakamura and L. Zhang, "FlowFusion: Dynamic Dense RGB-D SLAM Based on Optical Flow," *In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (2020) pp. 7322–7328.
- [22] W. Dai, Y. Zhang, P. Li, Z. Fang and S. Scherer, "RGB-D SLAM in dynamic environments using point correlations," *IEEE Trans Pattern Anal Mach Intell* **44**(1), 373–389 (2022).
- [23] J. Hu, H. Fang, Q. Yang and W. Zha, "MOD-SLAM: Visual SLAM with Moving Object Detection in Dynamic Environments," *In: Proceedings of the 40th Chinese Control Conference (CCC)*, (2021) pp. 4302–4307.
- [24] R. Bao, R. Komatsu, R. Miyagusuku, M. Chino, A. Yamashita and H. Asama, "Stereo camera visual SLAM with hierarchical masking and motion-state classification at outdoor construction sites containing large dynamic objects," *Adv Robotics* **35**(3), 228–241 (2021).

- [25] B. Bescos, J. M. FÁCil, J. Civera and J. Neira, “DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes,” *IEEE Robot Automat Lett* **3**(4), 4076–4083 (2018).
- [26] K. He, G. Gkioxari, P. Dollár and R. Girshick, “Mask R-CNN,” *In: Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, (2017) pp. 2980–2988.
- [27] V. Badrinarayanan, A. Kendall and R. Cipolla, “SegNet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Trans Pattern Anal Mach Intell* **39**(12), 2481–2495 (2017).
- [28] J. A. H. A. Wong, “Algorithm as 136: A k-means clustering algorithm,” *J R Stat Soc* **28**(1), 100–108 (1979).
- [29] A. Dumitriu, F. Tatui, F. Miron, R. T. Ionescu and R. Timofte, “Rip Current Segmentation: A Novel Benchmark and YOLOv8 Baseline Results,” *In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2022) pp. 1261–1271.
- [30] H. Li, J. Li, H. Wei, Z. Liu, Z. Zhan and Q. Ren, “Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles,” (2022). arXiv preprint, 2022 arXiv: 2206.02424.
- [31] M. Soudy, Y. Afify and N. Badr, “RepConv: A novel architecture for image scene classification on intel scenes dataset,” *Int J Intell Comp Infor Sci* **22**(2), 63–73 (2022).
- [32] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik and R. Siegwart, “The EuRoC micro aerial vehicle datasets,” *Int J Robot Res* **35**(10), 1157–1163 (2016).
- [33] J. Sturm, W. Burgard and D. Cremers, “Evaluating egomotion and structure-from-motion approaches using the TUM RGB-D benchmark,” *In: Proc. of the Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RJS International Conference on Intelligent Robot Systems (IROS)*, (2012) pp. 13.
- [34] S. Li, J. Gu and Y. Feng, “Visual SLAM with a Multi-Modal Semantic Framework for the Visually Impaired Navigation-Aided Device,” *In: Proceedings of the IEEE International Conference on Advanced Robotics and Mechatronics (ICARM)*, (2023) pp. 870–876.
- [35] B. Li, C. Zhang, C. Ye, W. Lin, X. Yu and L. Meng, “A Robust Odometry Algorithm for Intelligent Railway Vehicles Based on Data Fusion of Encoder and IMU,” *In: The 46th Annual Conference of the IEEE Industrial Electronics Society In IECON*, (2020) pp. 2749–2753.
- [36] V. Girbés-Juan, L. Armesto, D. Hernández-Ferrándiz, J. F. Dols and A. Sala, “Asynchronous sensor fusion of GPS, IMU and CAN-based odometry for heavy-duty vehicles,” *IEEE Trans Veh Technol* **70**(9), 8617–8626 (2021).
- [37] X. Zhang, T. Mononen, J. Mattila and M. M. Aref, “Mobile Robotic Spatial Odometry by Low-Cost IMUs,” *In: 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, (2018) pp. 1–6.