Research Article

INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS

International Journal of Advanced Robotic Systems
September–October 2024: 1–11

DOI: 10.1177/17298806241279782
journals.sagepub.com/home/arx
Sage

Dynamic human–object interaction detection for feature exclusion in visual simultaneous localization and mapping (SLAM)

Shival Indermun[1], Kristiaan Schreve[1], Thomas Weber[2] and Matthias Rätsch[2]

Abstract

Visual simultaneous localization and mapping (SLAM) remains a focal point in robotics research, particularly in the realm of mobile robots. Despite the existence of robust methods such as ORBSLAM3, their effectiveness is limited in dynamic scenarios. The influence of moving entities in these scenarios poses challenges to data association, leading to compromised pose estimation accuracy. This paper proposes a novel approach that utilizes spatial reasoning to reduce the influence of dynamic entities present in an environment. Our approach, known as human–object interaction detection, identifies the dynamic nature of an object by evaluating the intersecting area between the bounding boxes of a person and the object. We tested our approach by extending the ORBSLAM3 RGB-D SLAM algorithm. Consequently, all ORB features associated with dynamic objects are filtered out from the ORBSLAM3 tracking thread. To validate our approach, we conducted evaluations on highly dynamic sequences extracted from the TUM RGB-D dataset. Our results exhibited a significant performance enhancement over ORBSLAM3. Furthermore, in comparison to other state-of-the-art research, our results remained competitive, given the simplicity of our approach.

Keywords

Human–object interaction, visual SLAM, object detection, dynamic environments, ORBSLAM3

Date received: 16 January 2024; accepted: 6 August 2024

Topic: Vision Systems
Topic Editor: Sunita Bansal
Associate Editor: Ankita Rajput

Introduction

Simultaneous localization and mapping (SLAM) is regarded as the problem of navigating an unknown environment. The approach is founded on navigating an unknown environment while simultaneously self-localizing and generating a map.[1] In general, SLAM is used either as a means of providing detailed maps (or models) of the surrounding environment or producing accurate positions of robots within a given environment.[2]

A continuously growing topic of interest has been the concept of visual SLAM. As opposed to the conventional use of light detection and ranging (LiDAR) systems, which provide trajectory and distance measurements through data association, visual SLAM introduces the application of cameras to provide pose estimates. The

[1] Mechanical and Mechatronic Engineering, Stellenbosch University, Stellenbosch, Western Cape, South Africa
[2] Reutlingen Research Institute, Reutlingen University, Reutlingen, Baden-Württemberg, Germany

Corresponding author:
Shival Indermun, Mechanical and Mechatronic Engineering, Stellenbosch University, Joubert Street, Stellenbosch, 7602, Western Cape, South Africa.
Email: shivalindermun@sun.ac.za

three dominant approaches of visual SLAM are feature-based, direct, and red–green–blue-depth (RGB-D) SLAM. The feature-based approach was founded on tracking and mapping feature points, while direct methods considered the whole image as an input, specifically the photometric consistency between images, thus making it more computationally demanding.[3]

Davison et al.[4] developed monoSLAM, a feature-based method that used the extended Kalman filter to estimate camera motion and feature positions of an unknown environment. One of the key issues of this method was the scale of the environment, which was largely related to the extent of computation required. The parallel tracking and mapping method introduced the concept of using parallel threads for tracking and mapping, thus reducing the computational cost of the approach.[5] With the introduction of utilizing different threads for the different modules of the visual SLAM process, later approaches adopted the technique, such as ORBSLAM.[6] A common example of the direct method approach is LSD-SLAM,[7] whereby a three-dimensional (3D) environment is constructed as a pose-graph of keyframes related to semi-dense depth maps.

With the development of RGB-D cameras, improved visual SLAM approaches were introduced. These cameras produce both RGB and depth images. Common RGB-D cameras determine depth perception through the projection of infrared patterns, constraining their usage in indoor environments.[3] However, depth ranges are often limited to 5 m.[8] Examples of RGB-D SLAM include the work of Endres et al.,[9] who used the Kinect camera to simultaneously provide camera trajectory and generate dense 3D models of indoor environments, and Salas-Moreno et al.,[10] who developed SLAM++, a method that registered prior 3D objects to replace existing detected objects, leading to increased mapping efficiency.

Most of these methods are limited in two key areas: lack of interpretation of the environment and the inability to cater to dynamic entities. The robustness and precision of state estimation have largely been attributed to the base assumption of a static environment. Feature extraction is dependent on recognizing stable and unique visual elements to facilitate tracking. Dynamic objects introduce complications such as occlusion of other objects, changes in lighting conditions, motion blur, variations in scale, erroneous data associations, and deformations. Consequently, the presence of dynamic entities frequently disrupts this process, resulting in system failure.

Visual SLAM methods have seen rapid progress over the last decade in conjunction with the development of both computer vision and machine learning algorithms. Several approaches[11,12] make use of convolutional neural networks to detect and identify dynamic objects within a scene, leading to the removal of dynamic feature points. However, these approaches are limited to their predetermined definition of dynamic objects. Defining static and dynamic objects in a scene is largely attributed to the environment. However, a key assumption in our work is that objects that have the potential to be dynamic are normally due to human interaction. We recognize that many other factors can cause objects to behave dynamically (wind, animals, moving machinery, etc.), but we limit the scope of this work to human-induced dynamics.

The current research proposes an approach that extends ORBSLAM3[13] by filtering ORB features when extracted from environments consisting of human motion and object interaction. The approach utilizes the recently developed YOLOv8[14] object detection models to assist in determining a spatial relationship between humans and objects within an environment. The contributions of the current research include:

- The human–object interaction detection (HOID) method utilizes spatial reasoning in either the two-dimensional (2D) or 3D perspective to identify human–object interactions.
- An extended RGB-D SLAM algorithm, utilizing our HOID method, to identify and remove dynamic objects in an environment.

Related work

Dynamic and semantic SLAM approaches

To address dynamic environments in visual SLAM methods, various strategies incorporate visual methods that include both object detection and semantic segmentation. Bao Ai et al.[11] employed YOLOv4[15] with a dynamic object probability to mitigate the impact of dynamic entities. Similarly, Guan et al.[16] enhanced the ORB-SLAM3 framework by integrating YOLOv5[17] for object detection in dynamic indoor scenes.

Kaneko et al.[18] presented a monocular visual SLAM algorithm incorporating DeepLab v2[12] for semantic segmentation, selectively excluding feature points in outdoor dynamic areas to enhance stability. Xiao et al.[19] proposed Dynamic-SLAM, utilizing a custom solid state drive (SSD)[20] object detection model based on a convolutional neural network, coupled with a missed detection compensation algorithm. This method emphasizes the critical role of recall rate, particularly significant when employing object detection or segmentation methods in SLAM algorithms.

In another approach, Zhong et al.[21] proposed Detect-SLAM, which integrates object detection and SLAM to mutually enhance each other. This method implemented a deep neural network, SSD,[20] to detect both static and dynamic objects. The detection of dynamic objects, combined with a moving probability, was used to reduce their influence. An object map generated by the SLAM process was utilized as prior knowledge to enhance the object detector.

DS-SLAM[22] and DE-SLAM[23] are similar in their approach to enhancing robustness in dynamic environments. DS-SLAM

applies a semantic segmentation model (SegNet[24]) to filter features observed from dynamic objects, combining semantic segmentation with a moving consistency check (based on an optical flow technique) to omit feature points related to dynamic objects. DS-SLAM generates an octo-tree map[25] representing colored voxels with corresponding semantic labels. Similarly, DE-SLAM focuses on highly dynamic environments by employing a combination of semantic detection with MobileNet V2 SSD[26] and adaptive particle filtering for dynamic feature rejection, enhancing robustness against moving objects in real-time applications. Islam et al.[27] introduced MVS-SLAM, which utilizes enhanced multiview geometry to improve semantic RGB-D SLAM performance in dynamic scenarios. This approach leverages semantic segmentation to refine feature matching, thereby increasing the system's robustness in environments characterized by significant motion.

More recently, Li et al.[28] proposed YVG-SLAM, an advancement of the ORB-SLAM3 algorithm that integrates view geometry with the YOLOv5 algorithm to dynamically remove feature points without relying on a priori assumptions. This method effectively reduces the impact of dynamic elements by employing a feature recognition strategy that determines geometric consistency in detected bounding boxes. Similarly, Zhong et al.[29] introduced DynaTM-SLAM, a method that integrates visual and semantic information for SLAM in dynamic environments. They utilize YOLOv7[30] for object detection and apply template matching within a sliding window to efficiently filter dynamic feature points. Additionally, their approach includes building an online object database to maintain consistent data association for static objects, which are then used to optimize camera poses through bundle adjustment with semantic constraints.

Furthermore, Wang et al.[31] proposed VIS-SLAM, an innovative approach that integrates visual, inertial, and semantic information for enhanced SLAM performance. Their method employs a non-blocking model to extract semantic information and assigns prior motion probabilities to feature points based on object detections. A propagation model is also utilized to estimate motion probabilities for frames without semantic information. The integration of inertial measurement unit (IMU) data assists in robot localization by correcting errors inherent in visual data. Experimental results demonstrate significant improvements in localization accuracy, emphasizing the significance of sensor fusion methods.

Fang et al.,[32] presented a novel visual SLAM method that uses semantic segmentation and a knowledge graph to create a semantic descriptor. The knowledge graph represents the relationships between objects in the environment. The study was motivated by the application of robotics as an aid against cross-contamination that may occur during the COVID-19 pandemic.

The semantic descriptor was described as a $n \times n$ matrix consisting of scalar quantities. These quantities represent 80 different object categories that can be classified by the semantic segmentation model (Mask R-CNN[33]). The approach determined whether an object is dynamic based on its relationship defined in the knowledge graph but more specifically how close a human (moving object) is to the moveable object.

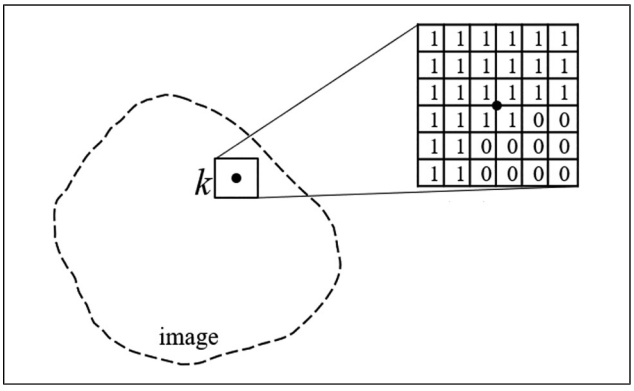

Figure 1. Semantic descriptor.

Consider Figure 1, in which a random key point k, consists of a human, representing a scalar quantity 1, and a moveable object (e.g. book), representing a scalar quantity 0. Both objects are close and thus the book can be considered dynamic. Therefore, the influence of the human and object (in terms of feature points) can be filtered out. The size of the descriptor and the close proximity (distance) of entities are defined in the study based on an experimental threshold.

Fang's descriptor introduced the concept of using masks around a key point to discern meaningful relations between objects and individuals. However, it was observed to be limited by the contouring issues generated by Mask R-CNN. To our knowledge, this is the only approach to consider spatial reasoning, which is in relation to key points. In contrast, our approach aims to define human–object interactions through the intersection between object-bounding boxes. Furthermore, our method incorporates depth information to ascertain the significance of a 3D perspective in Visual RGB-D SLAM.

Limitations

These methods effectively utilize neural networks through object recognition models to remove features associated with predefined dynamic objects. Although these approaches have enhanced the robustness of SLAM systems, their performance is highly sensitive to the specific detection models used. To improve the overall robustness of the SLAM system, methods utilizing detection models make use of accompanying geometry methods, such as optical flow techniques or probability checks. Our approach seeks to mitigate the dependency by being independent of any particular detection model. This design allows for greater flexibility and scalability, making our method

adaptable to various contexts and environments of the intended applications. Furthermore, our work places significant emphasis on the concept of human–object interaction. By focusing on the spatial relationships and interactions between humans and objects, our method can discern dynamic and potentially dynamic elements in the environment, potentially improving the accuracy and robustness of the SLAM system.

Method

The current section details our method in three subsections. First, the overview of the extended ORBSLAM3 approach is given. We then present the use of YOLOv8 and the definition of dynamic objects. Finally, we describe the HOID method in depth.

System overview

Our approach extends the use of ORBSLAM3, which was built on the preceding ORBSLAM2.[34] The algorithm supports both visual and visual–inertial modes on monocular, stereo, and RGB-D systems. In principle, our method will be able to extend various visual SLAM algorithms and is not limited to specific features. We aimed to improve the RGB-D system and make use of the depth information. The system architecture is presented in Figure 2.

The integration of the HOID method into the ORBSLAM3 framework involves specific modifications to the tracking thread to filter dynamic features. The following steps outline the detailed integration process:

1. **Feature extraction and initial processing:** The tracking thread first extracts ORB features from the current RGB frame and associates these features with depth information from the corresponding depth image.
2. **HOID filtering process:** After feature extraction, the tracking thread invokes the HOID method to analyze human–object interactions within the frame. The HOID method utilizes object detection to identify dynamic entities by analyzing the overlap between object-bounding boxes and human-bounding boxes in the RGB frame. If the interaction area between a human and an object exceeds a predefined threshold, the object is classified as dynamic. A more detailed explanation of the HOID method is provided in a subsequent section of this paper.
3. **Dynamic feature exclusion:** ORB features associated with dynamic objects are filtered out, ensuring that only static features remain for further processing. This filtering process helps exclude the dynamic features from subsequent SLAM computations, thereby enhancing the robustness of the system in dynamic environments.
4. **Reintegration into ORBSLAM3 architecture:** The static features obtained after the HOID filtering

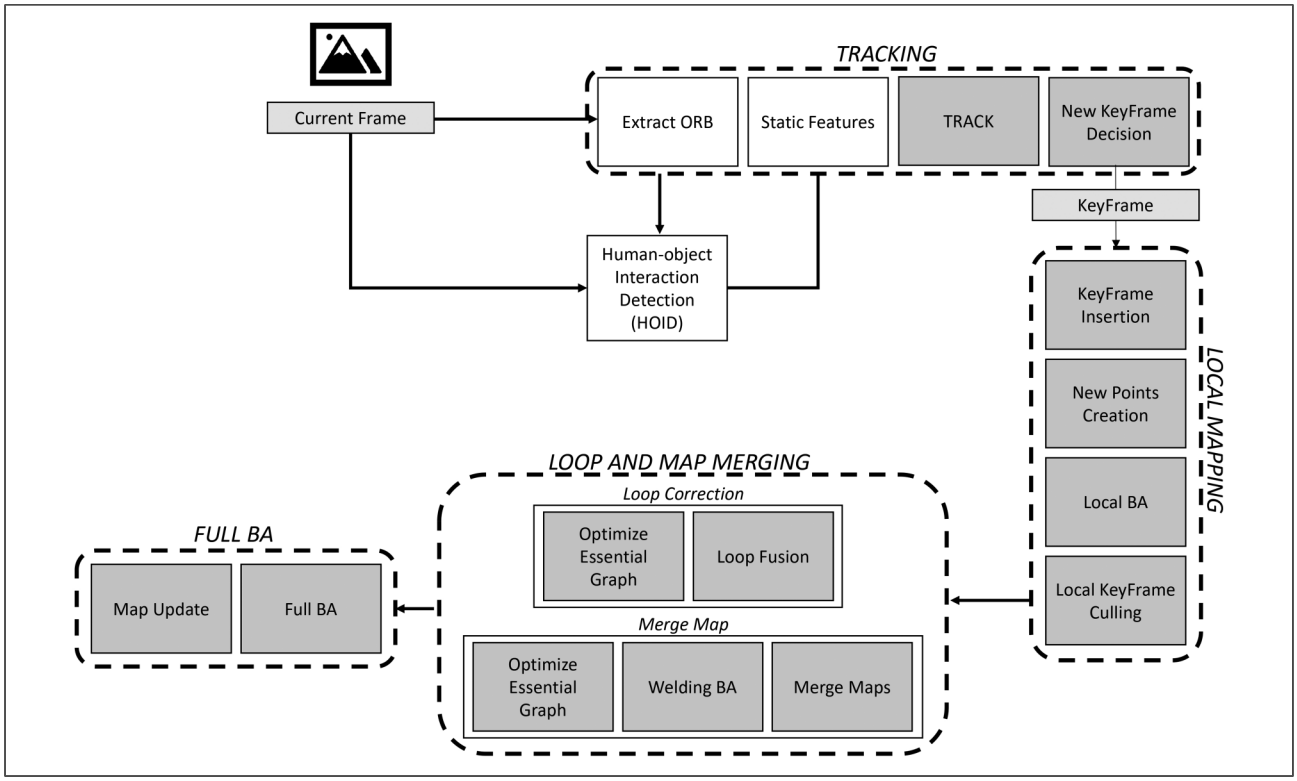

Figure 2. System overview.

process are reintegrated back into the original ORBSLAM3 architecture. These static features are then used for pose estimation and map updates, maintaining the integrity and accuracy of the original ORBSLAM3 system.

Object detection

Object detection is a vast field, encompassing various algorithms and datasets. There have been a number of object detection models that have been applied within the SLAM and robotics field. Some of the popular approaches include Faster R-CNN[35] and YOLO.[36] In addition to classifying and localizing objects using bounding boxes, segmentation models have gained favor for their ability to preserve object shapes through masking.

In the context of the work presented by Fang et al.,[32] Mask R-CNN exhibited issues along the contouring of the classified objects. This is not ideal as it may result in errors when determining whether objects are interacting with one another. Therefore, our implementation utilized the latest version of the YOLO models, YOLOv8 by ultralytics.[14]

YOLOv8 is not limited to detection but also includes models that incorporate segmentation, pose estimations, and tracking. Our method requires bounding box information and will therefore focus on detection. Table 1 represents the performance of pre-trained YOLOv8 models, on the COCO dataset. As the mean average precision (mAP) increases, so does the inference time. Furthermore, while the choice of model can impact object identification precision, our approach is versatile and the model can be changed to suit the desired performance.

Using object detection, it is possible to remove objects from a scene based on selecting classes of interest that have been defined as dynamic. Our approach only assumes humans as a dynamic class and that objects are dynamic solely due to human interaction. Given that the YOLOv8 models were pre-trained on the COCO dataset, the following objects were chosen to be potentially dynamic: *chair, cup, book, bottle, keyboard, laptop, mouse, person, and TV/monitor*. The selection of the dynamic objects was based on objects typically found in the TUM dataset,[37] on which our approach was evaluated. Selection criteria for dynamic objects will be context-dependent and tailored toward the specific environment being mapped.

Table 1. Performance of YOLOv8.[14]

Model	mAP
YOLOv8n	37.3
YOLOv8s	44.9
YOLOv8m	50.2
YOLOv8l	52.9
YOLOv8x	53.9

mAP: mean average precision.

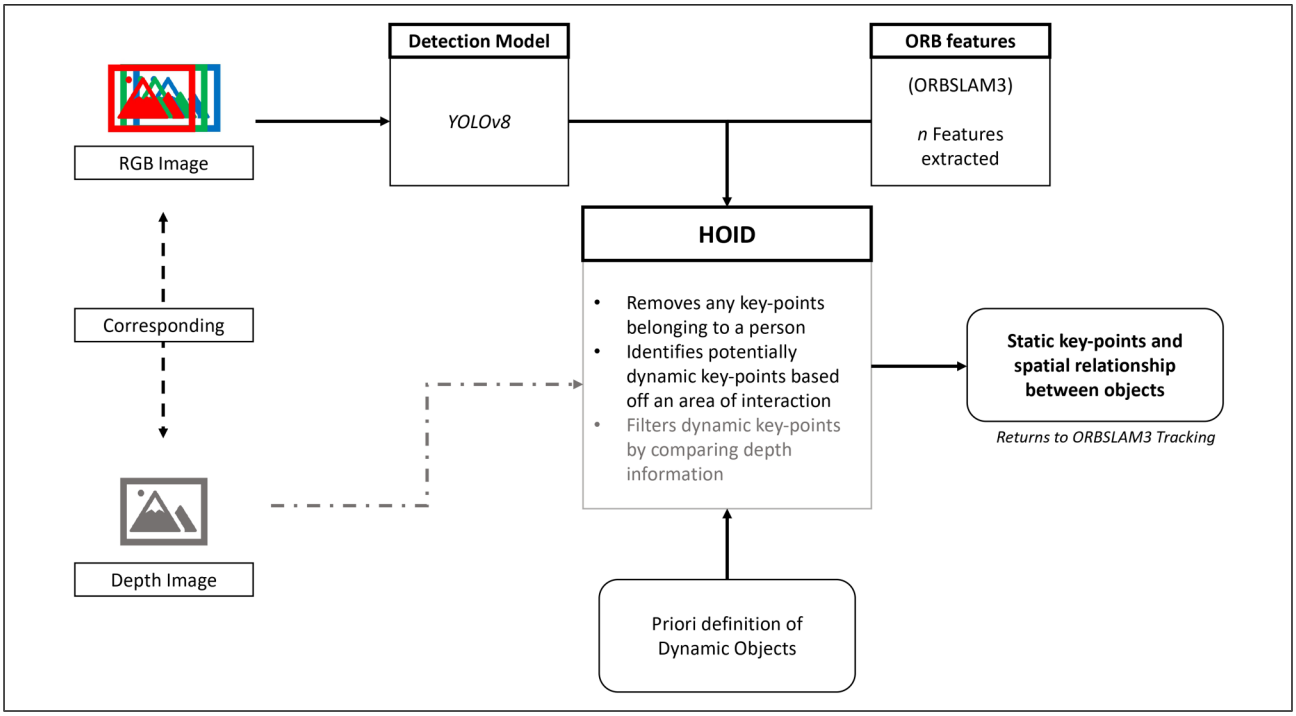

Figure 3. Human–object interaction detection.

Human–object interaction detection

This paper introduces a novel approach for deducing human–object interactions by leveraging spatial reasoning. Figure 3 provides an overview of the HOID method. We utilized two distinct versions of the method. The initial version relied solely on the RGB frame, the generated ORB features, and a list of potentially dynamic objects. Similar to Fang's descriptor, these relationships were detected from a 2D standpoint. In a subsequent iteration, we incorporated the depth image to refine the selection of object feature points. This was achieved by discerning relationships in the 3D space.

The premise behind the method is founded on the interacting area between the bounding box of a person and a potentially dynamic object. The interacting area pertains to the shared region between two bounding boxes and the extent to which they overlap. If the ratio between the overlapped area and the object area is greater than a certain threshold, referred to as the area of interaction, it is assumed that human–object interaction is detected. Therefore, determining the object to be dynamic.

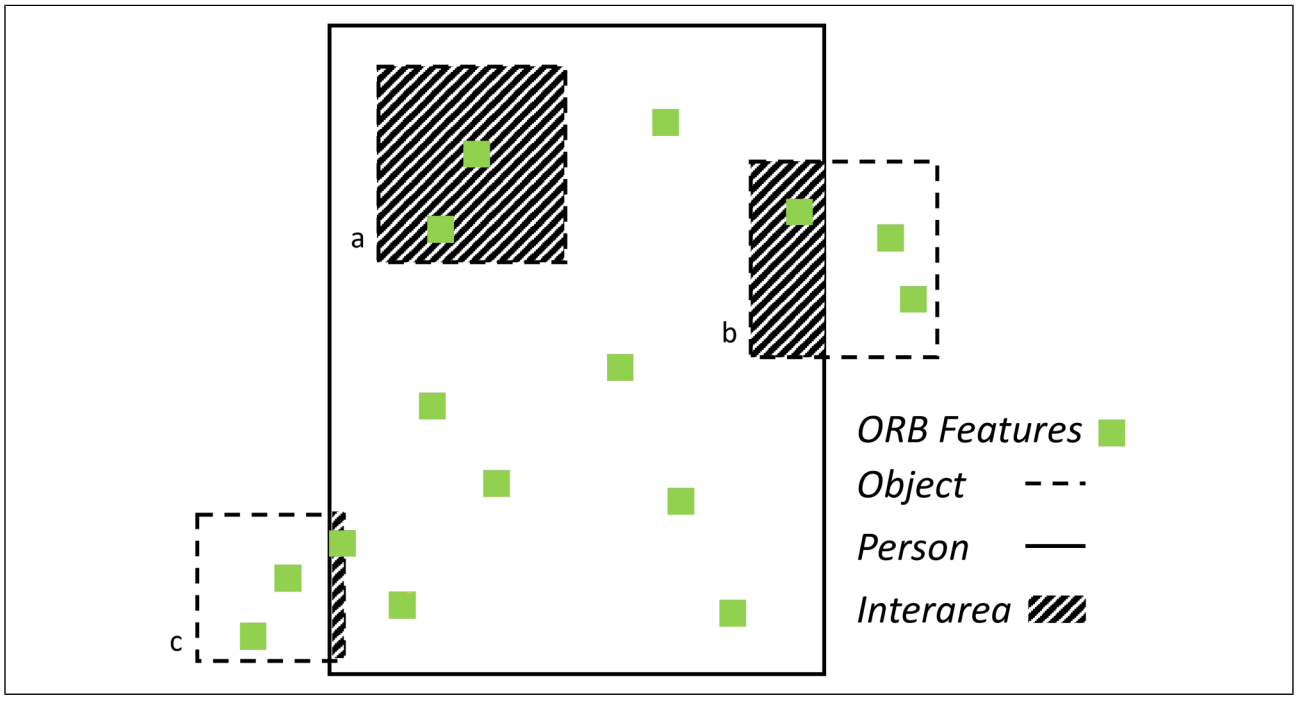

Figure 4. Interacting area scenarios.

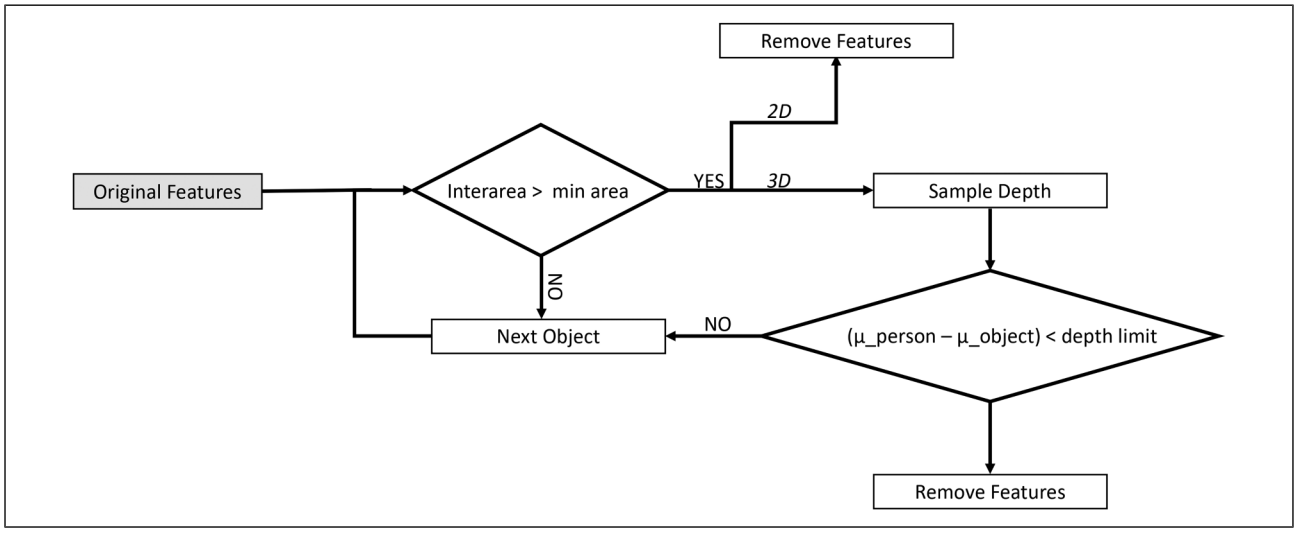

Figure 5. Feature removal process.

In Figure 4, three example object scenarios are presented. Object A is completely encompassed within a person's bounding box, classifying it as dynamic. Object B exhibits an area surpassing the designated minimum interaction area, hence it is also classified as dynamic. In contrast, Object C displays an area below the specified threshold for interaction, rendering it non-dynamic. Consequently, ORB features from bounding boxes (a) and (b) would be removed.

In order to potentially enhance the precision of the proposed SLAM method, we considered depth data. Once an object met the criteria for being considered dynamic in relation to the interaction area, a second filtering step was implemented. This involved comparing depth data using sample boxes positioned at the center of both the object and person bounding boxes. If the disparity between these sample boxes fell below a predefined depth threshold, it signified that the object was in close proximity to the person. Consequently, such an object was classified as dynamic, and its associated features were subsequently removed. A significant assumption of this approach is that the object is ideally positioned at the center of the corresponding bounding box. However, this assumption may not hold true, especially for narrower objects. A flow diagram representing the feature removal process for both the 2D and 3D methods is shown in Figure 5.

We display our method using a single RGB-D frame extracted from a sequence within the TUM dataset. In Figure 6, we present the observed detections and 1000 generated ORB features. Implementing the HOID method with a minimum interaction area set at 0.4 effectively eliminates dynamic features, as depicted in Figure 7(a). Introducing depth into the analysis, with a corresponding limit of 0.2 m, reveals the retained features, as illustrated in Figure 7(b). Both figures emphasize the impact of including depth in preserving features. However, the effectiveness of retaining or removing features is discussed in the next section.

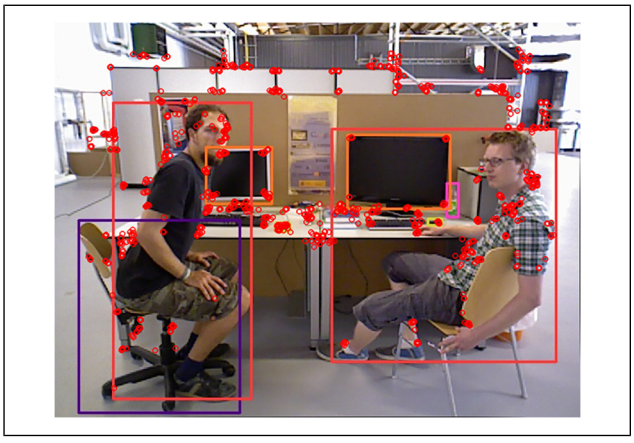

Figure 6. Sample frame with generated ORB features and YOLOv8 detections.

Experimental results and discussion

To assess the performance of our proposed RGB-D SLAM method, we utilized the TUM dataset. This dataset is a valuable resource containing both RGB-D images and corresponding ground-truth data, specifically designed for evaluating visual SLAM techniques. It encompasses a

Table 2. Trajectory errors [m] on the minimum area of interaction.

	Minimum area		
Sequence	0.2	0.3	0.4
f3-walking-static	0.0087	0.0078	0.0084
fr3-walking-xyz	0.0162	0.0154	0.0152
fr3-walking-halfsphere	0.0349	0.0315	0.0278

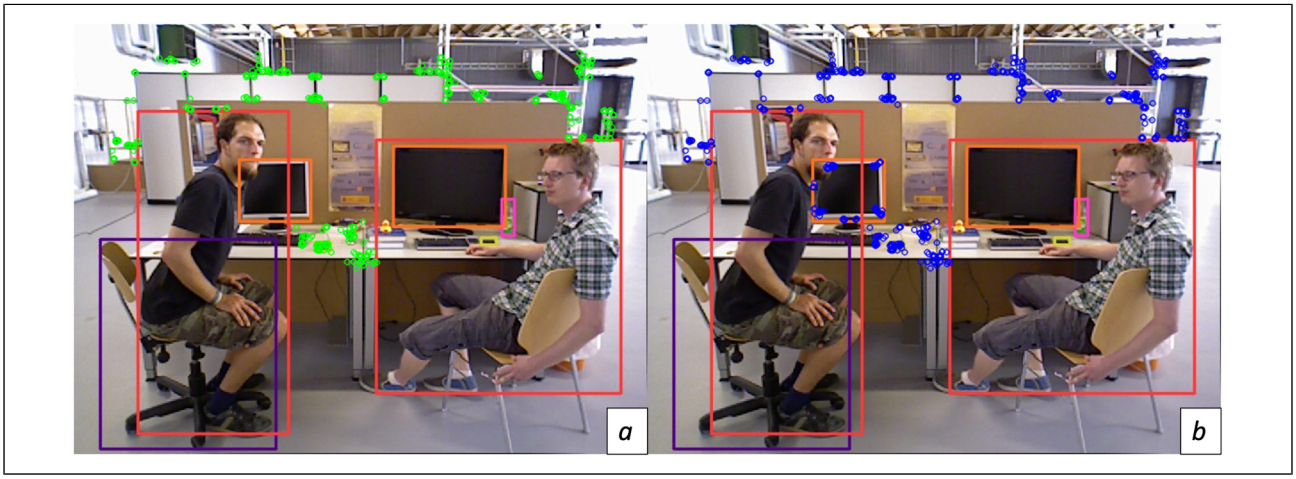

Figure 7. Sample frames from our approach.

Table 3. ATE represented in meters as the RMSE, mean and median values.

	ORBSLAM3			Ours		
Sequence	RMSE	Mean	Median	RMSE	Mean	Median
fr3-walking-static	0.4169	0.4014	0.4411	0.0084	0.0070	0.0013
fr3-walking-xyz	0.7580	0.6420	0.6250	0.0152	0.0134	0.0123
fr3-walking-halfsphere	0.2146	1940	0.1895	0.0278	0.0235	0.0208

ATE: absolute trajectory error; RMSE: root mean square error.

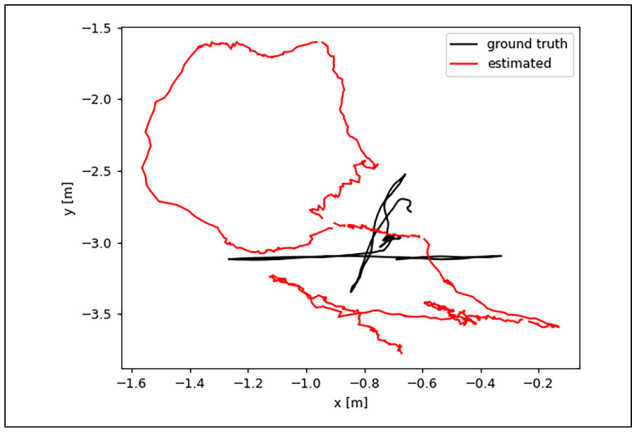

Figure 8. ORBSLAM3 trajectory from f3-walking-xyz sequence.

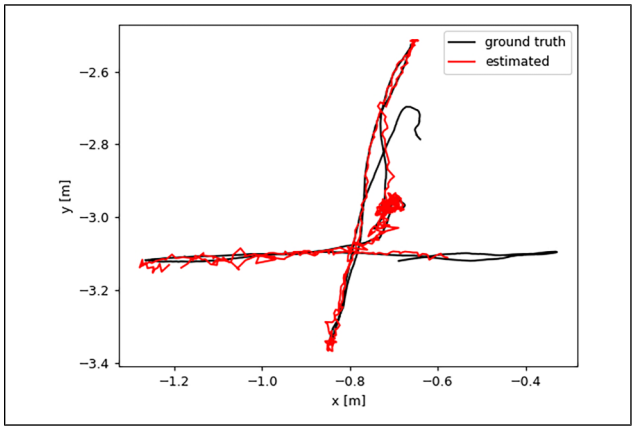

Figure 9. Our approach trajectory from f3-walking-xyz sequence.

diverse range of scenes, including static and dynamic scenarios.

For our research, we focused on three primary subsets within the dataset: f3-walking-static, f3-walking-xyz, and f3-walking-halfsphere. These subsets represent high-dynamic scenarios featuring two individuals walking in an office environment. This selection allows us to rigorously evaluate the robustness and effectiveness of our approach in real-world, highly dynamic settings. Each set consists of a sequence of images recorded at a frame rate of 30 fps with a resolution of 640×480. All three sequences underwent five iterations, with the median value chosen as the final result to ensure accuracy and consistency.

Prior to performing a comparison against ORBSLAM3 and the current research, it was necessary to determine the minimum area of interaction. The resulting root mean square error (RMSE) of the absolute trajectory errors (ATEs) are shown in Table 2. Throughout the experimental runs, the optimal range that demonstrated the lowest errors fell within 0.2 to 0.4. Specifically, an interacting area of 0.3 resulted in the lowest ATE for the f3-walking-static sequence, whereas an area of 0.4 yielded the lowest ATE for the f3-walking-xyz and f3-walking-halfsphere sequences. Consequently, the chosen minimum interaction area was set at 0.4.

The determination of the depth limit is inherently contextual, as it pertains to defining the proximity at which a human is deemed to be in alignment with an object. We have decided to set this limit at 0.2 m, a threshold we consider subjectively proximate enough for meaningful human–object interaction. This parameter directly correlates with the measurement of distance differences between objects.

Table 4. Computation analysis.

Algorithm	Method	Computation time (ms)
DS SLAM[22]	Moving consistency check	29.5
DynaSLAM[38]	Multiview geometry & Background inpainting	±400
DynaTMSLAM[29]	Template matching	4.33
Ours	HOID	12

HOID: human–object interaction detection.

Table 3 shows the performance comparison between ORBSLAM3 and our method. There is a significant improvement observed in our approach. Figures 8 and 9 represent the trajectories for ORBSLAM3 and our approach with respect to the f3-walking-xyz subset, respectively. In the context of our study, the observed gap in trajectory at the initial and final segments of the generated trajectory is attributed to the time-stamp associations between RGB and depth images. Given our emphasis on enhancing

Table 5. Absolute trajectory error (ATE) [m] comparison with current research.

Sequence	Yue et al.[22]	Bescos et al.[38]	Zhong et al.[21]	Fang et al.[32]	Guan et al.[16]	Zhong et al.[29]	Our approach
fr3-walking-static	0.0081	0.0068	–	0.0104	–	0.0068	0.0084
fr3-walking-xyz	0.0247	0.0156	0.0241	0.0164	0.0140	0.0149	0.0152
fr3-walking-halfsphere	0.0303	0.0301	0.0514	0.0923	0.0550	0.0291	0.0278

RGB-D SLAM, such occurrences are consistent. The subsets f3-walking-static and f3-walking-xyz demonstrated a significant improvement of 98%, while the f3-walking-halfsphere subset showed an 87% enhancement. Evidently, our method exhibits a greater performance than ORBSLAM3 in highly dynamic environments.

Our approach has been compared to current research methods that have aimed to improve the robustness and performance of visual SLAM within dynamic environments. The comparison can be seen in Table 5. Our results are competitive with the literature, achieving the lowest error in the f3-walking-halfsphere subset. Furthermore, our method stands out by eliminating the need for supplementary motion checks or optical flow, relying instead on spatial reasoning within a frame. It is worth noting that the method's performance is contingent upon the speed and efficiency of the object detection model. Therefore, the speed of our method is limited to the inference speed of the selected model. Ignoring the detection model, the process of filtering features and removing dynamic objects averaged 12 ms per frame. The computation speed is based on the following laptop specifications: 7th Generation Intel i7-7700HQ, 3.8 GHz processor, 4x Cores, 8x Threads with 16 GB RAM, and an NVIDIA Geforce GTX 1060 6 GB GDDR5 graphics card. All processing involved frames of 640×480 resolution and the programming was done in C++ on Ubuntu 20.04.6 LTS. Assessing and enhancing computational efficiency is contingent upon effective resource management and the specific hardware configuration utilized, which complicates direct comparisons with existing literature. To accurately evaluate and benchmark computational efficiency, it is imperative to assess resource utilization across diverse hardware configurations. Nonetheless, considering this complexity, Table 4 presents a comparison of time performance based on the primary method employed, excluding detection speeds. It is important to acknowledge that not all methods were designed for real-time performance. However, the study conducted by Zhong et al.[29] demonstrated the fastest computational time.

In the second iteration of our method, we explored the integration of associated depth images to assess the potential enhancement of 3D spatial reasoning on performance. This entailed a comparison of depth data, where sample boxes positioned at the center of both the object and person bounding boxes were compared. A sample box size of 20×20 was specifically chosen for objects related to the dataset. The selection of a sample size is contingent upon the objects within an environment. However, the limitation of this method stems from the assumption that most objects are not narrow and cover a substantial area within their bounding boxes. The use of segmentations for comparing depth data may prove more advantageous, and this avenue will be explored in future work.

Table 6. ATE RMSE for 2D and 3D method.

Sequence	HOID 2D	HOID 3D
fr3-walking-static	0.0084	0.0087
fr3-walking-xyz	0.0152	0.0153
fr3-walking-halfsphere	0.0278	0.0326

ATE: absolute trajectory error; RMSE: root mean square error; 2D: two-dimensional; 3D: three-dimensional; HOID: human–object interaction detection.

Table 6 presents a comparison between the 2D and 3D methods. Notably, the differences between the two approaches are minimal, with the most significant gap being 14.7%, favoring the 2D method in the f3-walking-halfsphere sequence. It appears that the inclusion of depth data does not necessarily improve the approach, as it retains features not aligned with the people within the scene. This suggests that the removal of features through a 2D perspective might yield more benefits. In a scene where a person moves across objects, their movement can negatively influence the features being tracked on an object. The HOID approach employs a minimum area of interaction, removing these influences shortly before and after a person has moved across it.

The comparison of both methods is constrained to the TUM dataset, and a more comprehensive assessment can be achieved in a dataset with increased instances of human–object interaction. Additionally, both methods face limitations concerning the space a human occupies within a given frame, as the corresponding bounding box removes a considerable number of feature points. Arguably, this might not be as large a drawback, given that scenes where humans dominate the frame would inherently pose challenges to tracking regardless.

The potential of the 3D method should become more beneficial in scenes illustrating clear interactions between humans and objects. Moreover, this approach holds promise for applications in semantic SLAM, where the extraction of detailed scene information is highly desirable. We intend to further explore these scenarios in future research.

Conclusion

This paper introduces a novel approach that leverages spatial reasoning to mitigate the influence of dynamic entities within an environment. Our method was tested by building upon the open-source visual SLAM algorithm, ORB-SLAM3. Referred to as HOID, our approach determines whether an object is dynamic by analyzing the shared intersection between the object and a person's bounding boxes. If the interacting area exceeded an experimentally defined threshold, the objects were classified as dynamic. Consequently, the associated features were eliminated and reintegrated into the tracking thread of the algorithm.

To evaluate the effectiveness of our method, we conducted tests on dynamic sequences from the TUM dataset. Our method significantly outperformed ORBSLAM3 in dynamic scenarios. Furthermore, our results were competitive with the literature. Considering the simplicity of our method, human–object spatial reasoning proved capable of effectively improving pose estimation.

Our findings highlight several broader implications for the field of robotics and autonomous systems. By effectively filtering out dynamic features, the HOID method enhances the accuracy and reliability of visual SLAM systems in dynamic environments, which is crucial for the deployment of autonomous robots in real-world scenarios where dynamic interactions are common. Furthermore, the ability to detect and exclude dynamic human–object interactions significantly improves the safety and efficiency of robots operating in close proximity to humans, with important implications for collaborative robots (cobots) in industrial and service applications. Furthermore, our research lays a foundation for further studies into dynamic SLAM, encouraging the development of more advanced algorithms that incorporate additional dynamic factors and improve computational efficiency.

However, while our study presents promising results, several limitations should be acknowledged. The evaluation was primarily conducted using the TUM RGB-D dataset, which, while widely recognized and extensively used, may not fully encompass the range of dynamic scenarios encountered in real-world environments. Our focus on human–object interactions as the primary source of dynamics excludes other potential dynamic factors such as moving machinery and environmental conditions.

A second iteration of our method incorporated the use of depth information. However, it was found that there was no significant change to warrant integrating depth perception. As a result, our findings suggest that retaining features of objects not aligned with a human-led to a higher level of error. The inclusion of depth may be beneficial for semantic SLAM applications, where extracting more contextual information of an environment is desirable. We aim to investigate this further.

Future research should aim to incorporate more diverse and highly dynamic datasets to understand the full potential and limitations of our approach in various real-world scenarios. This will help assess the scalability of the method in handling more complex dynamic environments. Additionally, we aim to extend the application of HOID to encompass both semantic and risk-aware navigation, to potentially improve the overall reliability and safety of autonomous systems. We plan to investigate this further and implement our approach in real-world scenarios involving increased human–object interaction to generate semantic-based maps.

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Shival Indermun https://orcid.org/0000-0002-5569-5036

References

1. Thrun S, Burgard W and Fox D. *Probabilistic robotics*. Cambridge, MA: The MIT Press, 2005.
2. Siciliano B and Khatib O. *Robotics and the handbook*. Cham: Springer International Publishing, 2016, pp.1–6.
3. Taketomi T, Uchiyama H and Ikeda S. Visual SLAM algorithms: A survey from 2010 to 2016. *IPSJ Trans Comput Vis Appl* 2017; 9: 16.
4. Davison AJ, Reid ID, Molton ND, et al. Monoslam: real-time single camera slam. *IEEE Trans Pattern Anal Mach Intell* 2007; 29: 1052–1067.
5. Klein G and Murray D. Parallel tracking and mapping for small ar workspaces. In: *Proceedings of the 2007 6th IEEE and ACM international symposium on mixed and augmented reality (ISMAR 2007)*, Nara, Japan, 13–16 November 2007, pp.255–234. Piscataway, NJ: IEEE.
6. Mur-Artal R, Montiel JM and Tardos JD. ORB-SLAM: A versatile and accurate monocular slam system. *IEEE Trans Robot* 2015; 31: 1147–1163.
7. Engel J, Schöps T and Cremers D. LSD-SLAM: large-scale direct monocular SLAM. In: *Proceedings of the 13th European conference on computer vision (ECCV 2014)*, Zurich, Switzerland, 6–12 September 2014, pp.834–849. Berlin, Germany: Springer.
8. Henry P, Krainin M, Herbst E, et al. RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In: *Proceedings of the international symposium on experimental robotics (ISER 2010)*, Delhi, India, 18–21 December 2010, pp.477–491. Berlin, Germany: Springer.
9. Endres F, Hess J, Engelhard N, et al. An evaluation of the RGB-D SLAM system. In: *2012 IEEE international conference on robotics and automation*, Saint Paul, MN, USA, 14–18 May 2012, pp.1691–1696. Piscataway, NJ: IEEE. DOI: 10.1109/ICRA.2012.6225199.

10. Salas-Moreno RF, Newcombe RA, Strasdat H, et al. SLAM++: Simultaneous localisation and mapping at the level of objects. In: *2013 IEEE conference on computer vision and pattern recognition*, Portland, OR, USA, 23–28 June 2013, pp.1352–1359. Piscataway, NJ: IEEE. DOI: 10.1109/CVPR.2013.178.
11. Bao Ai Y, Rui T, Yang XQ, et al. Visual SLAM in dynamic environments based on object detection. *Def Technol* 2020; 17: 1712–1721.
12. Chen LC, Papandreou G, Kokkinos I, et al. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans Pattern Anal Mach Intell* 2018; 40: 834–848.
13. Campos C, Elvira R, Rodríguez JJG, et al. ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM. *IEEE Trans Robot* 2021; 37
14. Jocher G, Chaurasia A and Qiu J. Yolo by ultralytics, 2023. https://github.com/ultralytics/ultralytics.
15. Bochkovskiy A, Wang CY and Liao HYM. YOLOv4: Optimal speed and accuracy of object detection, 2020. http://arxiv.org/abs/2004.10934.
16. Guan H, Qian C, Wu T, et al. A dynamic scene vision SLAM method incorporating object detection and object characterization. *Sustainability (Switzerland)* 2023; 15: 3048.
17. Jocher G. YOLOv5 by Ultralytics, 2020. https://github.com/ultralytics/yolov5. DOI: 10.5281/zenodo.3908559.
18. Kaneko M, Iwami K, Ogawa T, et al. Mask-SLAM: Robust feature-based monocular SLAM by masking using semantic segmentation. In: *2018 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW)*, Salt Lake City, UT, USA, 18–22 June 2018, pp.371–379. Piscataway, NJ: IEEE. DOI: 10.1109/CVPRW.2018.00063.
19. Xiao L, Wang J, Qiu X, et al. Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment. *Robot Auton Syst* 2019; 117: 1–16.
20. Liu W, Anguelov D, Erhan D, et al. Single shot multibox detector. In: Leibe B, Matas J, Sebe N, and Welling M (eds) *Computer vision—ECCV 2016. ECCV 2016. Lecture notes in computer science*. Cham: Springer, 2016, vol. 9905, pp.21–37.
21. Zhong F, Wang S, Zhang Z, et al. Detect-SLAM: Making object detection and SLAM mutually beneficial. In: *2018 IEEE winter conference on applications of computer vision (WACV)*, Lake Tahoe, NV, USA, 12–15 March 2018, pp.1001–1010. Piscataway, NJ: IEEE. DOI: 10.1109/WACV.2018.00115.
22. Yu C, Liu Z, Liu XJ, et al. DS-SLAM: A semantic visual SLAM towards dynamic environments. In: *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Madrid, Spain, 1–5 October 2018, pp.1168–1174. Piscataway, NJ: IEEE. DOI: 10.1109/IROS.2018.8593691.
23. Xing L, Liu Q, Zhang K, et al. DE-SLAM: SLAM for highly dynamic environments. *J Field Rob* 2022; 39: 1157–1173.
24. Badrinarayanan V, Kendall A and Cipolla R. SegNet: A deep convolutional encoder–decoder architecture for image segmentation. *IEEE Trans Pattern Anal Mach Intell* 2017; 39: 2481–2495.
25. Hornung A, Wurm KM, Bennewitz M, et al. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Auton Robots* 2013; 34: 189–206.
26. Sandler M, Howard A, Zhu M, et al. MobileNetV2: Inverted residuals and linear bottlenecks. In: *2018 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, Salt Lake City, UT, USA, 18–22 June 2018, pp.4510–4520. Piscataway, NJ: IEEE. DOI: 10.1109/CVPR.2018.00474.
27. Islam MJ, Tang H and Yang W. Mvs-slam: enhanced multiview geometry for improved semantic RGB-D slam in dynamic environments. *J Field Rob* 2023; 40: 234–256.
28. Li X, Wang H and Zhang L. Yvg-slam: dynamic feature removal slam algorithm without a priori assumptions. *IEEJ Trans Electr Electron Eng* 2024; 19: 123–135.
29. Zhong M, Hong C, Jia Z, et al. DYNATM-SLAM: Fast filtering of dynamic feature points and object-based localization in dynamic indoor environments. *Rob Auton Syst* 2024; 174: 104634.
30. Wang C, Bochkovskiy A and Liao H. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: *2023 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, Vancouver, BC, Canada, 18–22 June 2023, pp. 7464–7475. Piscataway, NJ: IEEE. DOI: 10.1109/CVPR52729.2023.00721.
31. Wang Y, Liu X, Zhao M, et al. Vis-slam a real-time Dynamic SLAM algorithm based on the fusion of visual, inertial, and semantic information. *ISPRS Int J Geoinf* 2024; 13: 163.
32. Fang B, Mei G, Yuan X, et al. Visual slam for robot navigation in healthcare facility. *Pattern Recognit* 2021; 113: 107822.
33. He K, Gkioxari G, Dollár P, et al. Mask-RCNN. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 22–29 October 2017, pp. 2980–2988. Piscataway, NJ: IEEE. DOI: 10.1109/ICCV.2017.322.
34. Mur-Artal R and Tardos J. Orb-slam2: An open-source slam system for monocular, stereo and RGB-D cameras. *IEEE Trans Robot* 2017; 33: 1255–1262.
35. Ren S, He K, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell* 2017; 39: 1137–1149.
36. Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection. In: *2016 IEEE conference on computer vision and pattern recognition (CVPR)*, Las Vegas, NV, USA, 26 June–1 July 2016, pp. 779–788. Piscataway, NJ: IEEE. DOI: 10.1109/CVPR.2016.91.
37. Sturm J, Engelhard N, Endres F, et al. A benchmark for the evaluation of RGB-D SLAM systems. In: *2012 IEEE/RSJ international conference on intelligent robots and systems*, Vilamoura-Algarve, Portugal, 7–12 October 2012, pp. 573–580. Piscataway, NJ: IEEE. DOI: 10.1109/IROS.2012.6385773.
38. Bescos B, Facil J, Civera J, et al. Dynaslam: Tracking, mapping and inpainting in dynamic scenes. *IEEE Robot Autom Lett* 2018; 3: 1.