Proceedings of CONF-MLA Workshop: Mastering the Art of GANs: Unleashing Creativity with Generative Adversarial Networks
DOI: 10.54254/2755-2721/80/2024CH0065

A Lightweight Deep-Learning Visual SLAM for Indoor Dynamic Environment Using Yolov10

Ziye Luo

Shien-Ming Wu School of Intelligent Engineering, South China University of Technology, Guangdong, China

wi202164010349@mail.scut.edu.cn

Abstract. Vision simultaneous localization and mapping (SLAM) technology has become a key research direction in the field of mobile robotics in recent years. However, the accuracy and stability of traditional vision SLAM technology greatly affected by the dynamic environment, and the mainstream dynamic feature point rejection method combining vision and semantic segmentation techniques is not applicable to edge-end devices with limited resources and high real-time requirements. This research suggests a visual SLAM algorithm based on the YOLOv10n lightweight target detection model and the GCNv2 feature point extraction model to accomplish real-time dynamic feature point rejection to address those problems. To compensate for the detection accuracy and stability issues of the YOLOv10n model while leveraging its real-time advantages, the algorithm also employs multi-target Kalman filtering with data association through the Hungarian algorithm, history window smoothing, and potential dynamic feature point recording methods to enhance robustness. The algorithm is validated on the TUM RGB-D Dataset, and the results demonstrate that the method can effectively reject the dynamic feature points in the dynamic environment, and it has a significant improvement in the accuracy and stability of the visual SLAM system.

Keywords: Visual SLAM, dynamic scenes, feature extraction, YOLOv10.

1. Introduction

With the rise of Robotics 4.0, there is an increasing demand for intelligence in smart devices such as mobile robots [1]. Compared with the mature SLAM solutions for static scenes, SLAM solutions for dynamic scenes with more universality and practicability are becoming a popular direction in related research fields. Laser SLAM based on LiDAR sensors is a traditional solution with high ranging accuracy, resistance to external interference and high cost, which is suitable for large and complex scenes [2]. In contrast, vision sensors have received extensive attention from researchers due to their rich collection of information, low cost, and low power consumption. In the last decade, many visual SLAM algorithms have appeared one after another. Such as ORB-SLAM [3], SVO [4], DSO [5] and other mainstream conventional vision SLAM methods have obtained more mature results in static environments. However, the performance of conventional vision algorithms is greatly impacted in scenarios including dynamic objects. Due to the fact that the motion of dynamic objects can have a serious impact on the process of localization and mapping, tracking failures may even happen [6]. For this problem, researchers have started to try to combine other methods to obtain a vision SLAM scheme

DOI: 10.54254/2755-2721/80/2024CH0065

with sufficiently high accuracy and robustness to complex environments. Combining the optical flow method to judge moving objects [7] is one of the representative schemes, however, this method has a strong condition of constant luminance in the front and back frames, which has limitations for scenes with frequent light changes. Other researchers have tried to combine visual and inertial data [8], and multi-sensor fusion to improve accuracy and robustness [9]. While these fusion methods enhance accuracy and robustness, the increased computational load significantly impacts real-time performance. Consequently, it is challenging to balance accuracy, robustness, and real-time performance simultaneously.

Combining deep learning methods with vision to solve SLAM problems in complex environments is a more emerging research direction. Currently there exists YOLO [10], SegNet [11], ESPNet [12], RangeNet++ [13] and other representative image semantic information extraction networks. A variety of visual SLAM methods combining semantic segmentation have appeared in recent years. Yu et al. proposed a visual SLAM for dynamic scenes based on ORB-SLAM2 framework and incorporates a lightweight SegNet semantic segmentation network, and uses an FPGA for acceleration to improve real-time performance [14, 15]. Bescos et al. proposed a model [16] based on ORB-SLAM2 framework combined with Mask R-CNN [17], but real-time is not considered and the mean tracking time of frames is around 1 second, which cannot be applied to real-time scenes. While the real-time performance is significantly impacted, the accuracy of the visual SLAM method in conjunction with the semantic segmentation model is improved significantly. Because of this problem, it is challenging to apply to edge scenarios where real-time performance is crucial and resources are scarce, such as mobile robots. How to guarantee the system's real-time performance while ensuring sufficient accuracy has become one of the research focuses of scholars in recent years.

The purpose of this work is to design a new visual SLAM method that enhances the robustness and real-time performance of SLAM algorithms in dynamic scenes. The method combines visual SLAM and deep learning by adding the GCNv2 feature point extraction model to the existing ORB-SLAM2 framework [18], and by adding a lightweight YOLOv10 target detection model [10] to detect dynamic objects. For the problem of insufficient stability of the target detection model, a combination of Kalman filter [19], history window smoothing and potential dynamic feature points recording to enhance the robustness of Yolov10n model. The goal is to ensure that the algorithms can operate in real-time with sufficient accuracy and stability in edge scenarios, for examples, mobile robots and low-power embedded devices.

This part introduces the background and motivation for this research. The second part details the methodology, including the integration of GCNv2 and YOLOv10n models, and the use of Kalman filtering, history window smoothing, and potential dynamic feature points recording. The third part describes the datasets, experimental process and evaluation criteria. The fourth part presents the results, analysis, and relative discussion. Finally, the paper's conclusion and likely future study directions are suggested in the last part.

2. Methodology

2.1. Algorithmic framework

Figure 1 show the entire process of the algorithm's framework. Replacing the original ORB feature extractor for feature point extraction with a GCNv2 lightweight neural network model based on the ORBSLAM2 framework to improve accuracy and robustness [18]. The RGB image is then fed into the YOLOv10 model for target detection to obtain image regions of dynamic objects. The feature points in the region are rejected and the descriptors of them are recorded, and then the target detection results are recorded into the history window. Finally, the Kalman filters are updated with the target detection results.

DOI: 10.54254/2755-2721/80/2024CH0065

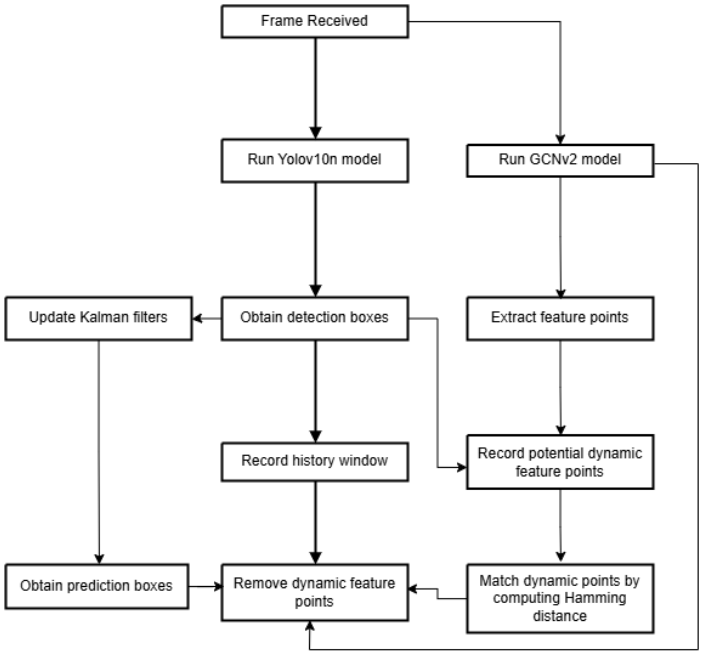

Figure 1. Overall flow of the algorithmic framework

Kalman filters calculate the predicted image region for target detection. Feature points within the predicted region are rejected to compensate the missed detection of dynamic objects in motion. The Hamming distance between previously recorded feature points and those currently outside the target detection region is calculated. This calculation helps match and reject missed dynamic feature points, improving the robustness of detecting dynamic objects in both stationary and motion states. Additionally, rejecting feature points within the target detection region recorded in the history window further enhances robustness for detecting dynamic objects in both states.

2.2. Feature point extraction

GCNv2 is a lightweight deep learning model, and it is used to replace the built-in ORB feature point extractor in the ORB-SLAM2 framework. The GCNv2 model has higher robustness compared to the ORB feature point extractor, which ensures that the feature points will not be lost when the lens is rotated too fast and improves the stability of visual SLAM [19]. In addition, the uniformity of feature points extracted by GCNv2 are higher compared to those of the ORB feature point extractor, which ensures that the SLAM system still has enough feature points for localization and mapping after dynamic object rejection and avoids tracking failures of the SLAM system when the dynamic object occupies too much screen area.

2.3. Dynamic feature point rejection

YOLO is a cutting-edge target detection method in recent years. YOLOv10 is based on YOLOv8 with innovative improvements by eliminating non-maximal suppression (NMS), optimizing the model architecture, and other strategies. Yolov10 increases inference speed and dramatically reduces the quantity of parameters over previous versions, while improving the accuracy to a certain extent, which makes it well suited for edge device-side algorithm deployment[10]. The YOLOv10n model with the best real-time performance and the fewest parameters is selected for this article.

During the experiment, it was found that YOLOv10n has poor stability and has intermittent and longtime failure problem for human target detection. For the static standing human body, the number of failed frames is 1~5 frames. For the static seated human body, there are roughly 1 to 8 unsuccessful frames. For the dynamic human bodies (either in motion or with the lens rotation causing them to move

DOI: 10.54254/2755-2721/80/2024CH0065

in the screen), the number of failed frames is generally more than 3 frames. In some extreme cases, whether static or dynamic, only a few frames can be detected during the complete process of the whole human body in the screen, and most of the frames are failure frames. To compensate for the lack of stability of YOLOv10n, this paper predicts the trajectories of dynamic objects in motion through the Kalman filters, and simultaneously, history window smoothing and dynamic feature points recording modules are added to improve the robustness of the task of dynamic feature point rejection.

Aiming at the problem that the target detection model is unable to track the trajectory stably in the case of dynamic human body, this paper employs the multi-objective Kalman filter combined with the Hungarian algorithm [20] to carry out the prediction of the target detection frame. The Kalman filters update based on the target detection results of the previous and current frames, and then predicts the detection region for the next frame. The predicted region will be regarded as the image region of the dynamic object together and continue to exist for a certain number of frames until the next Kalman filter update. For scenes with multiple targets, the single-target Kalman filter algorithm cannot work properly. In this paper, an independent Kalman filter is created for each detected target. Through the Hungarian algorithm, the matching degree of all the targets detected in the current frame with each existing target is calculated to match the target detection results when there are multiple targets in the frame. This allows for the accurate update of the corresponding Kalman filter for multiple targets, realizing the Kalman filtering algorithm for multiple targets.

To address the stability problem of the target detection model in both dynamic and static human body scenarios, which may only accurately detect the human body once every certain number of frames, this paper employs the history window smoothing and potential dynamic feature points recording method to reject the dynamic objects. The history window records the target detection records of a certain number of recent frames (set to 4 in this paper), and when there is no target detection result at the current moment, all the target detection records in the history window will be used as the target detection result for the dynamic feature point rejection, and at the same time, the history window will be filled with an empty record value. If there is a new target detection result at the current moment, the record of the nearest frame to the current moment in the history window will be used as the image region of the dynamic object together with the target detection result of the current frame for dynamic feature point rejection. The potential dynamic feature points recording method, on the other hand, records the descriptors of feature points within the center region of the target detection boxes (to avoid recording too many static feature points) and sets the frame counter (set to 5 frames in this paper) every time there is a new target detection result. Every feature point outside the target detection boxes will calculate the Hamming distance with all feature points whose frame counters are not zeroed. When the value falls below the setting threshold (set to 30 in this paper), then it is regarded as a missed dynamic point and will be rejected, and the corresponding records are deleted if the frame counters of the feature points being recorded are zeroed. Figure 2 shows the visualization of the algorithm presented in this paper.

3. Experimental process and evaluation criteria

3.1. Datasets

This paper uses the TUM RGB-D Dataset published by the Technical University of Munich [21] to validate the accuracy and robustness of the algorithms in this paper. This is a benchmark dataset for evaluating vision SLAM systems. The dataset is captured through Microsoft Kinect sensors, the accelerometer data of the Kinect sensors is captured, and the true pose of the sensors is measured using a frequency of 100Hz and color images and depth images with a resolution of 640x480 pixels are recorded at a frequency of 30Hz. 39 data sequences of dynamic objects, 3D object reconstruction, etc. with a range of indoor sceneries are included in the dataset. This paper focus on verifying the effect of rejection of dynamic objects in indoor dynamic scenes. Therefore, the walking_xyz and walking_rpy datasets under Dynamic Objects are chosen for algorithm verification.

DOI: 10.54254/2755-2721/80/2024CH0065

3.2. Evaluation criteria

In this paper, there are three indexes are selected to test the practical effect of different algorithms. Absolute trajectory error (ATE) represents the mean value of the deviation between the actual camera position and the position predicted by the algorithm. Relative trajectory error (RTE) represents the mean value of the deviation between the actual position change of the camera and the position change predicted by the algorithm in the same two timestamps. Mean tracking time evaluates the real-time performance. The results are compared with ORBSLAM2 and ORBSLAM3 [22] algorithms. The test results are compared horizontally to analyze the differences in real-time, robustness, and accuracy.

The experimental environment is as follows: the CPU model is Intel Core i7-9750H 2.6 GHz with 8 GB of RAM. a VMware virtual machine with the operating system Ubuntu 20.04.

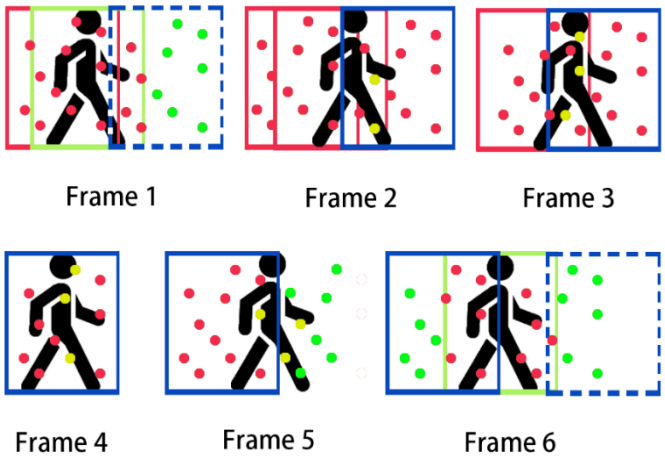

Figure 2. Schematic diagram of algorithm within this paper for a dynamic human body case, the green boxes are the target detection boxes, and the blue boxes are the Kalman filters prediction boxes, and the red boxes are the records in history window, and the red points are recorded and rejected, and the yellow points are missed and rejected

4. Experimental results and discussion

Figure 3 shows the dynamic objects rejection effect of the algorithm presented in this paper on the indoor dynamic scene dataset walking_rpy when the YOLOv10n model fails. From the figure, the algorithm presented in this paper can effectively identify and reject dynamic feature points. In the left figure, the lens rotates quickly, and the human body remains stationary. The YOLOv10n model decreases in stability and fails to detect the human body. The history window record (red box) and Kalman filter prediction (blue box) ensure that the dynamic points in the region where the human body is located are rejected. In the right figure, the lens rotates slowly and the human body remains stationary. The YOLOv10n model can stably detect a standing human body, but is unstable for a seated human body. The history window record (red box) ensures that dynamic points in the region where the human body is located (red points) are rejected. To some extent, this compensates for the accuracy issues when using a lightweight model.

Figure 4 shows the missed feature point rejection effect of the algorithm presented in this paper when the YOLOv10n model fails on the indoor dynamic scene dataset walking_rpy. As can be seen from the figure, regardless of whether the human body is in motion or not, the method of potential dynamic feature points recording adopted in this paper is effective when the human body in the screen remains relatively stable and the target detection fails, while it performs poorly when the human body moves and the lighting angle and other conditions change, so the potential dynamic feature points recording is mainly used to make up for the robustness of the target detection of the static human body, and is used in conjunction with the history window smoothing and the Kalman filter to further improve the robustness of dynamic feature point rejection.

DOI: 10.54254/2755-2721/80/2024CH0065

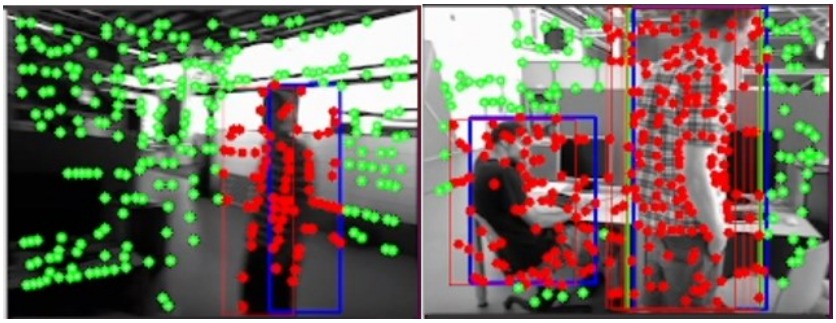

Figure 3. Dynamic point (red point) rejection effect of the combination of YOLOv10n model target detection (green box), Kalman filter prediction (blue box), and history window record (red box) in the presence of moving objects (left) and stationary objects (right) in the scenes

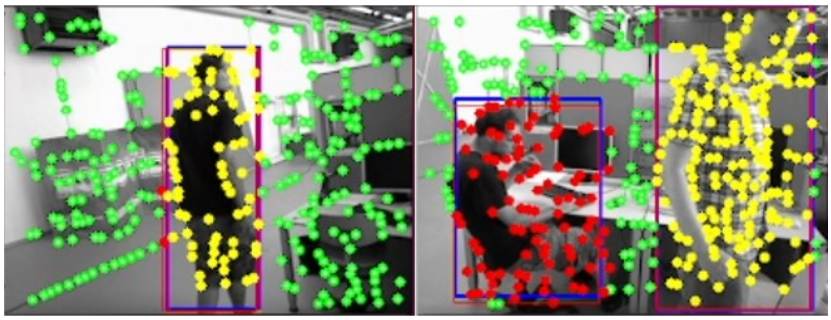

Figure 4. Dynamic point (red and yellow points) rejection effect of YOLOv10n model target detection (green box), Kalman filter prediction (blue box), history window record (red box), and missed dynamic points (yellow points) in the presence of moving objects (left) and stationary objects (right) in the scenes

Figure 5 compares the results of ATE of the algorithm presented in this paper with ORBSLAM2 and ORBSLAM3 on the indoor dynamic datasets walking_rpy and walking_xyz. Compared with ORBSLAM3, the ATE of the algorithm presented in this paper are reduced. The reduction of mean error is 59.28% and 56.03%, and the reduction of root mean square error is 59.42% and 56.14%, respectively, with respect to ORBSLAM3 on the two datasets, and the overall accuracy and robustness have been significantly improved.

By comparing the data in Figure 5, it can be obviously found the dataset walking_rpy has higher requirements for SLAM algorithms, and the difference in the localization effect is more significant. Probably because there are more images with the presence of dynamic human bodies compared to the dataset walking_xyz, so the dataset walking_rpy is used to compare the ATE images. Figure 6 compares the ATE trajectory images of the algorithm presented in this paper with the ORBSLAM3 on the indoor dynamic dataset walking_rpy. Through the comparison, it can be intuitively found that the proposed algorithm is more accurate and robust than ORBSLAM3, but there are still more localization jitter problems, the presumed reason is that in some scenes such as the presence of ambiguity in the dynamic objects in the state of motion, the target detection model fails for too long, and the dynamic objects lead to localization jitter repeatedly.

Figure 7 compares the results of the RTE of the proposed algorith with ORBSLAM2 and ORBSLAM3 on the indoor dynamic datasets walking_rpy and walking_xyz. Compared with ORBSLAM3, the RTE of the algorithm presented in this paper are reduced. The reduction of mean error is 14.94% and 13.41%, and the reduction of root mean square error is 12.00% and 11.54%, respectively, with respect to ORBSLAM3 on the two datasets, and the overall accuracy and robustness have been improved to some extent.

DOI: 10.54254/2755-2721/80/2024CH0065

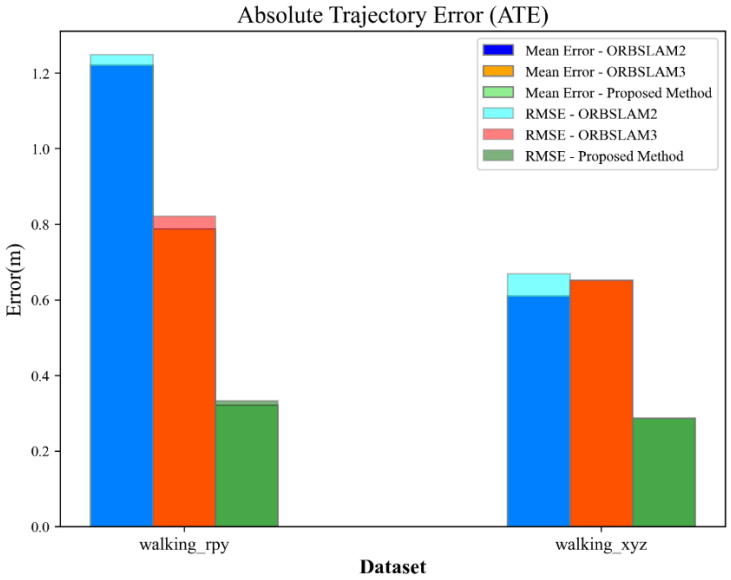

Figure 5. Comparison of ATE between the algorithm presented in this paper and other models

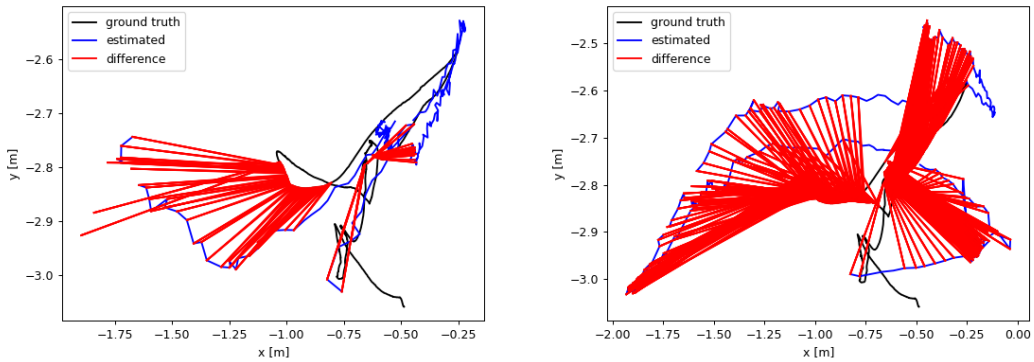

Figure 6. Comparison of ATE between the algorithm presented in this paper (left) and ORBSLAM3 (right) on walking_rpy dataset

The result of ORBSLAM2 has an unusually low value of RTE on the walking_xyz dataset, while the value of ATE is similar to that of ORBSLAM3. However, the distinction between the root mean squared error and the mean error of ATE is clearly larger than in other algorithms. This is likely due to dynamic objects obstructing with the camera's field of view at certain moments, which may result in an unusually low value of RTE. Meanwhile, because the small variations between adjacent frames, the calculated values of ATE are still large.

By comparing the data in Figure 7, it can be visualized that the difference between the walking_rpy dataset and the walking_xyz dataset has a smaller effect on the RTE of both ORBSLAM3 and the algorithm presented in this paper. In order to maintain consistency with Figure 6, the walking_rpy dataset is continued to be used to compare the RTE images of the algorithm presented in this paper with those of ORBSLAM3. Figure 8 compares the RTE trajectory images of the proposed algorithm and ORBSLAM3 on the indoor dynamic dataset walking_rpy. Through the comparison, it can be intuitively found that the algorithm presented in this paper is more accurate than ORBSLAM3.

Table 1 compares the mean tracking time data of the algorithm presented in this paper with ORBSLAM2 and ORBSLAM3 on the indoor dynamic datasets walking_rpy and walking_xyz. By comparison, it can be found that the algorithm presented in this paper has a worst performance on real-

DOI: 10.54254/2755-2721/80/2024CH0065

time, presumably due to the experimental platform selected in this paper through only virtual CPU for inference of deep learning models. The inference speed of the GCNv2 and Yolov10 deep learning models selected in this paper will be greatly affected, and therefore has a greater impact on real-time. The algorithm of this paper is mainly oriented to edge scenarios, and the mean tracking time will be probably decreased for the actual deployment to low-power embedded devices with independent GPU components (e.g., NVIDIA Jetson TX2).

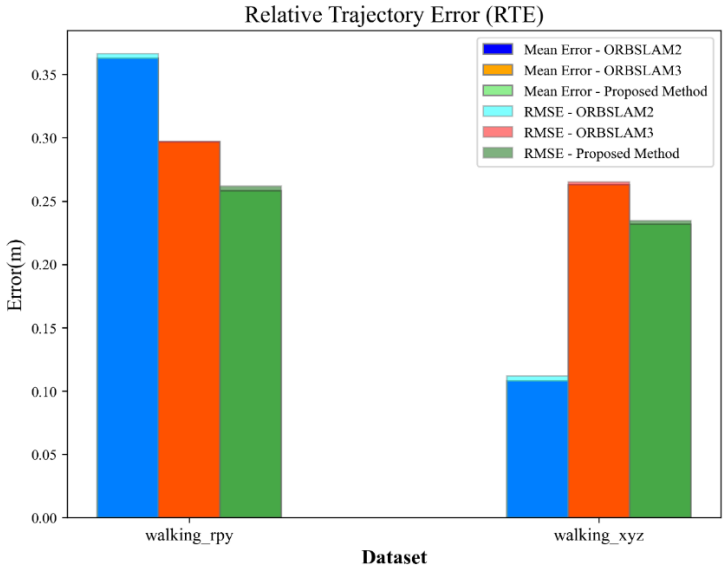

Figure 7. Comparison of RTE between the algorithm presented in this paper and other models

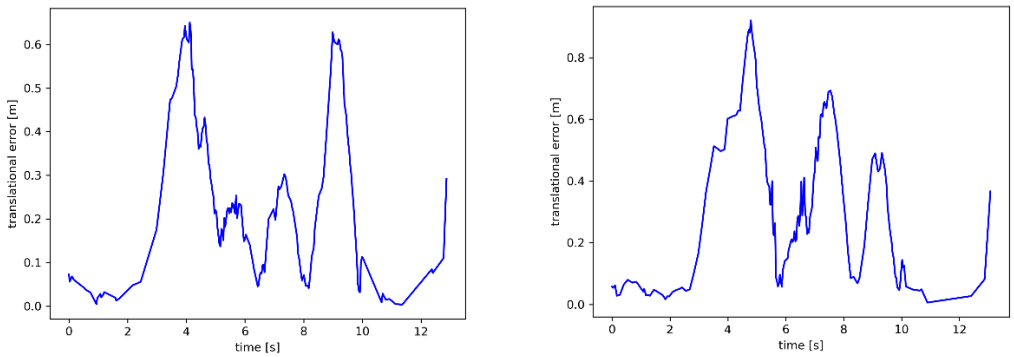

Figure 8. Comparison of RTE between the algorithm presented in this paper (left) and ORBSLAM3 (right) on walking_rpy dataset

The Yolov10n target detection model used in this paper has poor rotational invariance. In the dataset walking_rpy, when the lens is rotated around its roll axis, the target detection model will be completely invalid during the process. The performance of the target detection model will be slightly improved only when the rotation angle is 90 degrees for a short period of time (it can be successfully detected in some frames in the middle), but the accuracy and robustness are still quite different compared to those of the target detection model when it is not rotated. This will be improved in future research by retraining or selecting other target detection models.

DOI: 10.54254/2755-2721/80/2024CH0065

Table 1. Results of average tracking time

Dataset	ORBSLAM2/s	ORBSLAM3 /s	Proposed Method/s
walking_rpy	0.04132	0.02172	0.2121
walking_xyz	0.04374	0.02674	0.2276

For ensuring the stability of the Yolov10n target detection model, more static feature points are eliminated compared to the target detection only approach in this work. This paper attempts to utilize the pixel depth information of feature points within the target detection box to set the depth threshold for dynamic objects. The goal is to classify feature points that have significantly different pixel depths from human feature points (obtained by averaging the shallowest pixel depths from the central region of the target detection box) as static feature points. This aims to improve the accuracy. However, the frequent occurrence of outliers in the depth image data in the test dataset is likely to result in false retention of human feature points. Even after median filtering of the depth image data, this method was still ineffective, presumably because of fluctuating conditions such as light angle during dataset acquisition, and therefore the module for retaining static object feature points in combination with depth was not added in the end. This will be attempted to be addressed in future research by replacing the test dataset or performing a more complete data cleaning of the entire dataset.

5. Conclusion

In this paper, a visual SLAM method for dynamic objects rejection is proposed. This method performs feature point extraction and dynamic feature point rejection using the GCNv2 model and YOLOv10n model respectively. It combines the Kalman filter algorithm, history window smoothing, and potential dynamic feature points recording to ensure the robustness of the YOLOv10n target detection and dynamic feature point rejection process. The method is based on ORB-SLAM2 framework and experimentally validated on the TUM RGB-D Dataset. The experimental results demonstrate that this algorithm can significantly enhance the localization accuracy and robustness with less impact on real-time and can be applied to complex indoor scenes with multiple dynamic targets. Certainly there are some obvious shortcomings in the current methodology. Future research will focus on further addressing the lack of rotational invariance of the target detection model and static feature point retention by combining potential dynamic feature point records with depth data to further enhance the performance of visual SLAM.

References

[1] Huang P, Zeng L, Chen X, Luo K, Zhou Z, Yu S. *Edge robotics: edge-computing-accelerated multi-robot simultaneous localization and mapping.* arXiv preprint arXiv:2112.13222. 2022 [cited 2024 Aug 30]. Available from: http://arxiv.org/abs/2112.13222

[2] Hu Y, Xie F, Yang J, Zhao J, Mao Q, Zhao F, et al. *Efficient path planning algorithm based on laser SLAM and an optimized visibility graph for robots. Remote Sens.* 2024 Aug 10; 16(16):2938.

[3] Mur-Artal R, Montiel JMM, Tardos JD. *ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans Robot.* 2015 Oct; 31(5):1147–63.

[4] Forster C, Pizzoli M, Scaramuzza D. *SVO: fast semi-direct monocular visual odometry.* In: Proceedings of the *2014 IEEE International Conference on Robotics and Automation (ICRA).* Hong Kong, China: IEEE; 2014 [cited 2024 Aug 30]. p. 15–22. Available from: http://ieeexplore.ieee.org/document/6906584/

[5] Engel J, Koltun V, Cremers D. *Direct sparse odometry.* arXiv preprint arXiv:1607.02565. 2016 [cited 2024 Aug 30]. Available from: http://arxiv.org/abs/1607.02565

[6] Chen L, Ling Z, Gao Y, Sun R, Jin S. *A real-time semantic visual SLAM for dynamic environment based on deep learning and dynamic probabilistic propagation. Complex Intell Syst.* 2023 Oct; 9(5):5653–77.

DOI: 10.54254/2755-2721/80/2024CH0065

[7] Zhang T, Zhang H, Li Y, Nakamura Y, Zhang L. *FlowFusion: dynamic dense RGB-D SLAM based on optical flow.* arXiv preprint arXiv:2003.05102. 2020 [cited 2024 Aug 30]. Available from: http://arxiv.org/abs/2003.05102

[8] Song S, Lim H, Lee AJ, Myung H. *DynaVINS: a visual-inertial SLAM for dynamic environments. IEEE Robot Autom Lett.* 2022 Oct; 7(4):11523–30.

[9] Lin J, Zheng C, Xu W, Zhang F. *R2LIVE: a robust, real-time, LiDAR-inertial-visual tightly-coupled state estimator and mapping.* arXiv preprint arXiv:2102.12400. 2021 [cited 2024 Aug 30]. Available from: http://arxiv.org/abs/2102.12400

[10] Wang A, Chen H, Liu L, Chen K, Lin Z, Han J, et al. *YOLOv10: real-time end-to-end object detection.* arXiv preprint arXiv:2405.14458. 2024 [cited 2024 Aug 30]. Available from: http://arxiv.org/abs/2405.14458

[11] Badrinarayanan V, Kendall A, Cipolla R. *SegNet: a deep convolutional encoder-decoder architecture for image segmentation.* arXiv preprint arXiv:1511.00561. 2016 [cited 2024 Aug 30]. Available from: http://arxiv.org/abs/1511.00561.

[12] Mehta S, Rastegari M, Shapiro L, Hajishirzi H. *ESPNetv2: a lightweight, power efficient, and general purpose convolutional neural network.* arXiv preprint arXiv:1811.11431. 2019 [cited 2024 Aug 30]. Available from: http://arxiv.org/abs/1811.11431.

[13] Milioto A, Vizzo I, Behley J, Stachniss C. *RangeNet++: fast and accurate LiDAR semantic segmentation.* In: Proceedings of the *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* Macau, China: IEEE; 2019 [cited 2024 Aug 30]. p. 4213–20. Available from: https://ieeexplore.ieee.org/document/8967762.

[14] Yu C, Liu Z, Liu XJ, Xie F, Yang Y, Wei Q, et al. *DS-SLAM: a semantic visual SLAM towards dynamic environments.* In: Proceedings of the *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* Madrid: IEEE; 2018 [cited 2024 Aug 30]. p. 1168–74. Available from: https://ieeexplore.ieee.org/document/8593691.

[15] Mur-Artal R, Tardos JD. *ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. IEEE Trans Robot.* 2017 Oct; 33(5):1255–62.

[16] Bescos B, Fácil JM, Civera J, Neira J. *DynaSLAM: tracking, mapping and inpainting in dynamic scenes. IEEE Robot Autom Lett.* 2018 Oct; 3(4):4076–83.

[17] He K, Gkioxari G, Dollár P, Girshick R. *Mask R-CNN.* arXiv preprint arXiv:1703.06870. 2018 [cited 2024 Aug 30]. Available from: http://arxiv.org/abs/1703.06870.

[18] Tang J, Ericson L, Folkesson J, Jensfelt P. *GCNv2: efficient correspondence prediction for real-time SLAM.* arXiv preprint arXiv:1902.11046. 2019 [cited 2024 Aug 30]. Available from: http://arxiv.org/abs/1902.11046.

[19] Zelinsky A. *Learning OpenCV: computer vision with the OpenCV library (Bradski GR et al.; 2008). IEEE Robot Autom Mag.* 2009 Sep; 16(3):100–100.

[20] Nandashri D, Smitha P, East-West Institute of Technology. *An efficient tracking of multi-object visual motion using the Hungarian method. IJERT.* 2015 Apr 30; V4(04).

[21] Sturm J, Engelhard N, Endres F, Burgard W, Cremers D. *A benchmark for the evaluation of RGB-D SLAM systems.* In: Proceedings of the *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems.* Vilamoura-Algarve, Portugal: IEEE; 2012 [cited 2024 Aug 30]. p. 573–80. Available from: http://ieeexplore.ieee.org/document/6385773.

[22] Campos C, Elvira R, Rodríguez JJG, Montiel JMM, Tardós JD. *ORB-SLAM3: an accurate open-source library for visual, visual-inertial and multi-map SLAM. IEEE Trans Robot.* 2021 Dec; 37(6):1874–90.