Sensors and Materials, Vol. 36, No. 12 (2024) 5445–5458
MYU Tokyo

S & M 3878

Dynamic Point Cloud Removal Technology Based on Spatial and Temporal Information

Junyang Bian, He Huang, Junxing Yang,* Junxian Zhao, and Siqi Wang

School of Geomatics and Urban Spatial Information, Beijing University of Civil Engineering and Architecture, No. 15, Yongyuan Road, Huangcun Town, Daxing District, Beijing 102616, China

(Received April 18, 2024; accepted November 25, 2024)

Keywords: semantic mapping, dynamic point extraction, deep learning, multisensor fusion system

In the implementation of autonomous driving systems, accurate acquisition of the vehicle's location and orientation is crucial to providing a basis for path planning and obstacle avoidance. Although satellite navigation technology offers reliable positioning information, its signal is susceptible to interference in certain environments, such as areas with dense obstructions, which will affect the accuracy of vehicle localization and environmental mapping. To address the interference from dynamic points, we implement a method based on laser-vision multisensor fusion for identifying and extracting dynamic point clouds in environments where satellite signals are disrupted. We propose a dynamic point cloud extraction algorithm based on deep learning, utilizing semantic information to guide the network in extracting more precise information about the dynamic environment. We also construct a method for preliminary constraint of dynamic obstacles using global semantic information and design a multiframe point cloud processing approach with a sliding window mechanism, where residual data are accumulated and fed into the network as a composite model input. Experimental results demonstrate that our method significantly improves the positioning accuracy and map construction quality in dynamic environments, giving it considerable competitiveness compared with other advanced algorithms.

1. Introduction

Dynamic obstacles add significant complexity to simultaneous localization and mapping (SLAM) systems, impacting their performance and accuracy. When moving entities interfere with observations, it creates challenges, particularly since traditional SLAM algorithms often struggle to distinguish between dynamic and static elements. This confusion can undermine the stability of the maps generated. Moreover, dynamic objects can obscure important static features, which is especially problematic for SLAM algorithms that assume a static environment. This is a critical issue, particularly when using LiDAR sensors, as they can be easily disrupted by moving obstacles. To effectively identify and eliminate points associated with dynamic objects, specialized algorithms and mathematical approaches are needed. For instance, filtering

*Corresponding author: e-mail: yangjunxing@bucea.edu.cn
https://doi.org/10.18494/SAM5079

ISSN 0914-4935

algorithms can handle minor movements, like the gentle swaying of grass or fluttering leaves. However, for larger, more noticeable movements of objects in closer proximity, motion models and temporal consistency methods are more effective. Additionally, there is growing interest in using pretrained neural networks to recognize and remove dynamic objects, which is an exciting area of research. Most SLAM systems tend to mitigate the effects of dynamic objects by ignoring them, but this approach often fails in highly dynamic environments or when there are many obstacles moving at different speeds. Therefore, developing robust solutions to tackle these challenges is crucial for advancing SLAM technology.

With continual advancements in algorithms and the reduction in sensor costs, SLAM algorithms based on multisensor fusion have attracted widespread attention, demonstrating exceptional robustness and precision in various environments. Vision sensors exhibit unique advantages in semantic parsing, large-scale map construction, and target detection, while LiDAR is indispensable for precise measurements, is unrestrained by lighting conditions, and is capable of real-time measurement. Multisensor fusion systems enhance the accuracy of localization and mapping by combining the strengths of each sensor type and integrating them closely using optimized mathematical models. Visual odometry, divided into direct and feature-based methods, requires certain computational resources to process image features but performs well in dynamic, large-scale scenes. For example, Mono-SLAM[(1)] pioneered real-time monocular tracking and map construction in unknown scenes. The ORB-SLAM[(2)] series has improved the precision of localization and map construction under image rotation owing to feature point matching and loop closure detection. On the other hand, VINS-Mono[(3)] combines visual and IMU data to enhance system performance in rapid movement and dynamic environments. In the realm of LiDAR SLAM, algorithms such as LeGO-LOAM[(4)] have improved localization accuracy and real-time map construction through various methods. Algorithms such as LIO-SAM[(5)] further enhance the system's real-time capabilities and environmental adaptability through algorithmic optimization and multisensor data fusion. Notably, LVI-SAM[(6)] optimizes system performance in feature-scarce environments by integrating visual and laser data.

In recent years, the focus of research has shifted towards the effective removal of dynamic objects from the environment, which helps mitigate their impact on system performance and enhance the universality of SLAM technology. For smaller dynamic objects, such as raindrops or snowflakes, their impact can be considered as noise, typically conforming to Gaussian or median distributions. Recently, Kurup and Bos[(7)] have developed the DSOR filter, merging the advantages of DROR and SOR filters to improve the efficiency and speed of noise elimination. In 2021, Lim *et al.*[(8)] proposed the ERASOR algorithm, an innovative SLAM postprocessing method that removes dynamic point clouds by analyzing differences between laser frames. However, as a postprocessing approach, it cannot meet real-time requirements. Conversely, the DS-SLAM[(9)] system demonstrates effectiveness in practical applications by filtering dynamic objects through a combination of semantic information and motion consistency checks, although it may sometimes misjudge static environmental elements. Deep learning techniques, such as DynaSLAM[(10)] developed by Bescos *et al.*, leverage Mask R-CNN[(11)] and geometric methods to identify and remove dynamic objects, showcasing the potential of deep networks in analyzing dynamic objects. These advancements not only enhance the stability of SLAM systems in

dynamic environments but also support their generalization across wide application fields. Current research faces several challenges. First, multisensor fusion system architectures lack leveraging semantic information, leading to conflicts between real-time performance and multilevel map output. Second, in dynamic environments, the reliability of feature detection decreases owing to a lack of robust priors for loop closure detection, whereby the efficiency of this functionality is diminished. Furthermore, existing methods for removing dynamic objects do not effectively integrate temporal sequence information, and their use of semantic information is insufficient.[12] Additionally, the application and training of neural networks lack specificity, leading to suboptimal system optimization and difficulties in ensuring safe operation in dynamic environments.[13]

We investigate methods for feature identification and dynamic point cloud extraction in laser point clouds, for which we construct a deep-learning-based algorithm for dynamic point cloud recognition and extraction, as shown in Fig. 1. Unlike previous approaches that directly feed point cloud data into deep learning networks for dynamic point cloud analysis, this method employs semantic information as a range constraint to guide the deep learning network. It uses the three-dimensional spatial coordinates of the laser point cloud to project the three-dimensional point cloud onto a two-dimensional image coordinate system, obtaining a range image from that perspective.[14] By combining the generated residual images, a sliding window dynamic analysis range is constructed and input into the deep learning network. This achieves more precise dynamic point cloud extraction in dynamic environments, retains more valid environmental information, and enhances the accuracy of localization.

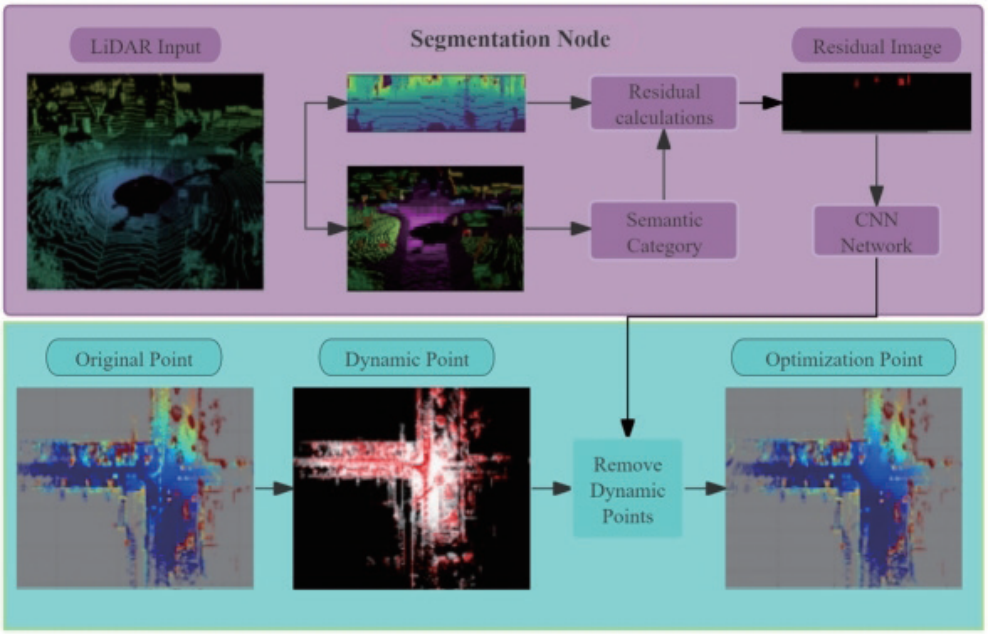


Fig. 1. (Color online) Dynamic point cloud extraction module framework.

2. Methods

2.1 Semantic segmentation

In this section, we primarily discuss methods for removing dynamic objects and the principles and processes involved in constructing semantic maps. In the part concerning matching and localization in dynamic environments, by converting each point cloud frame into a range image, residual and range images are introduced to guide the existing semantic segmentation networks, thereby enhancing the network's performance in detecting and eliminating dynamic objects.[15] In terms of semantic map construction, the cleaned maps obtained after the removal of dynamic objects in previous steps are used for semantic segmentation based on LiDAR point clouds. At this stage, the map retains only stationary objects, thereby producing multilevel maps that include global semantic maps and point cloud maps. To continue refining this content, the same level of refinement will be applied to the information provided subsequently.[16]

To achieve improved localization in dynamic environments and ensure the accuracy of feature point matching and pose estimation, we propose a deep-learning-based dynamic feature point extraction module based on LiDAR point cloud data.[17] Integrating this module into the point cloud preprocessing stage can effectively reduce the impact of dynamic obstacles on subsequent SLAM systems.[18] Initially, the network's region of interest is guided by semantic information from the point cloud data using a deep-learning-based semantic segmentation approach. Deep analysis is performed on the point clouds labeled as potential moving objects (such as vehicles and pedestrians) over several consecutive frames. The point cloud data are projected into range images in accordance with temporal information, and several frames of residual images are constructed within the sliding window frame range relative to the current frame. These are merged with range images to provide spatial and temporal information for the CNN[19,20] network, which further guide it in dynamic object segmentation. The road environment contains various semantic information, including vehicles, pedestrians, roads, and vegetation, with many elements in the collected point cloud data potentially interfering with localization and mapping. To effectively segregate these objects from the surrounding environment, we employ a deep-learning-based semantic segmentation model, which, after learning and being trained on the point cloud features of relevant objects, iteratively develops a model tailored for dynamic environments to complete the semantic segmentation tasks. For instance, upon detecting semantic labels such as pedestrians and vehicles, the point cloud preprocessing module automatically lists them as areas containing potential moving objects. The CNN network then primarily conducts dynamic analysis on the objects in these areas to determine whether the contained point clouds possess dynamic attributes. In contrast, static objects detected by the semantic segmentation model, such as road signs, trees, and roads, are ignored in subsequent dynamic analyses to conserve computational resources and enhance the overall efficiency of the dynamic point cloud extraction module,[21] thereby ensuring the real-time performance of the overall SLAM system. The construction method for a deep-learning-based semantic segmentation model requires the initial setup of training, validation, and test

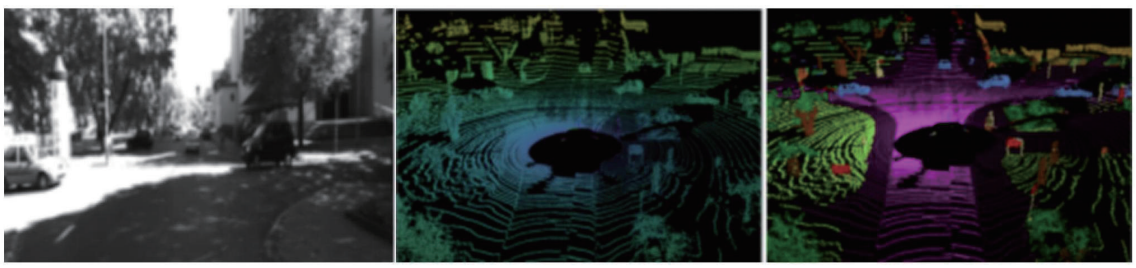
datasets. Within the training set, objects in the dataset are prelabeled to delineate the most common semantic categories in road scenarios, with semantic labels applied to the point clouds of corresponding areas. Subsequently, a deep learning model is constructed using a fully convolutional neural network. The model is then applied to the validation set to obtain experimental results, and its ability to robustly and accurately complete relevant semantic segmentation tasks is assessed. Finally, the model's generalization ability and segmentation performance are validated on the test set.

The SalsaNext model is employed for the training and prediction of the semantic segmentation model.[22] SalsaNext is a lightweight model that, on a 3070 model GPU, can achieve a speed of 10 Hz when the segmentation accuracy's mean intersection over union (mIoU) is 68.0, satisfying real-time requirements. The training data were obtained using images and semantic annotation files from the SemanticKITTI dataset. The segmentation results are shown in Fig. 2. The semantic labels are divided into 33 categories, including unlabeled, buildings, fences, pedestrians, lanes, roads, sidewalks, tree trunks, cars, walls, and traffic signs.

2.2 Range image

The semantic segmentation approach initially identifies regions potentially containing dynamic point clouds, since dynamic entities such as vehicles and pedestrians may remain static for various reasons, such as awaiting traffic signals or being parked. Consequently, leveraging semantic segmentation results as constraints for conducting dynamic analysis on point clouds of these areas via a CNN presents a more refined strategy. To augment the CNN's efficiency in isolating dynamic point clouds, it is crucial not only to utilize the semantic segmentation module for filtering areas of potential movement but also to integrate range and residual images derived from the point cloud data's four-dimensional attributes as additional constraints. These constructed range and residual images supply spatial and temporal data to the CNN, respectively. This dual-information framework further tightens the constraints on and directs the CNN's dynamic point cloud analysis, significantly enhancing the precision and robustness of the module.

In practical implementations, to minimize the system's susceptibility to dynamic points, it is necessary to supplement semantic segmentation with the introduction of range images for



(a) (b) (c)

Fig. 2. (Color online) Semantic segmentation of road scenes. (a) Real road, (b) original point cloud, and (c) segmented scene.

conveying spatial information to the network for the dynamic analysis of points. Compared with relying solely on point cloud data as input,[23] range images convert the spatial point clouds into an image format, facilitating easier comparison with subsequent states of the point cloud.[24] Coupled with ensuing temporal sequence residual images, this integration substantially enhances the identification of dynamic points. By employing the arbitrary LiDAR point coordinate p,

$$p = (x, y, z), \tag{1}$$

the point cloud data are initially transformed into spherical coordinates and then into image coordinates. Let w and h represent the width and height of the image, respectively. The image coordinates can be represented as follows.

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\left[1 - \arctan(y, x)\pi^{-1}\right] \bullet w \\ \left[1 - (\arcsin(zr^{-1}) + f_{up})f^{-1}\right] \bullet h \end{pmatrix} \tag{2}$$

In this setup, f encompasses f_{up} and f_{down}, representing the sensor's vertical scanning range mounted on the vehicle, from top to bottom. The variable $r = ||p_i||_2$ indicates the distance to each laser point. During the continuous scanning process, we accumulate a list of tuples concerning (u, v), which includes the three-dimensional spatial coordinates, along with the distance r and reflectivity e for each laser point. Following the outlined procedures, the data are initially projected onto spherical coordinates and ultimately translated into image coordinates. This transformation stores the known point cloud data within a range image, enabling the constructed image to be directly integrated into the network framework without necessitating modifications to the network architecture owing to data format, thus enhancing the system's generalizability.

2.3 Residual image

The concept of constructing residual images is inspired by optical flow techniques, and the spatial attributes of point cloud data are used to focus dynamic analysis on the variations across multiple consecutive frames from the same viewpoint.[25] Given the initial pose transformation calculated from the previous N frames, residual images are generated by employing existing sensor readings and computations. The range image is shown in Fig. 3. Once the range images have been constructed, generating residual images requires transforming subsequent image

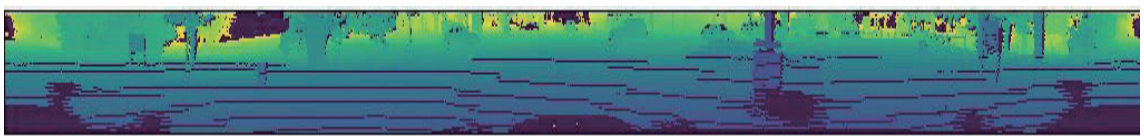


Fig. 3. (Color online) Range image after point cloud data conversion.

coordinates to align with the coordinate system of the current frame, which involves transformations and reprojections of point cloud data. We define the *j*-th scan within the sequence of continuous scans from the past 1 to *N* frames as S_j, where the scan contains *M* points represented in homogeneous coordinates.

$$S_j = \left\{ p_i \in R^3 \right\} \tag{3}$$

The transformation matrix obtained through the position and orientation estimation is denoted as

$$T_N^{N\text{-}1}, ..., T_1^0 \ , \tag{4}$$

Let *T* be a 4 × 4 transformation matrix. Applying this transformation matrix in the experiment enables the perspective of any frame after the current image to be transformed to that of the viewpoint of the current image. For instance, consider the transformation from the viewpoint of frame *k* to frame *j*.

$$\begin{aligned} S^{k\to l} &= \left\{ T_k^l p_i \mid p_i \in S_k \right\} \\ T_k^l &= \prod\nolimits_{j=k}^{l+1} T_j^{j\text{-}1} \end{aligned} \tag{5}$$

After reprojecting subsequent frames back to the current viewpoint, it is necessary to calculate the normalized difference for each pixel point to compare the changes in subsequent frames relative to the current frame. The normalized difference, denoted as $d_{k,i}^l$, is expressed as

$$d_{k,i}^l = \frac{\left| r_i - r_i^{k\to l} \right|}{r_i} \ . \tag{6}$$

Within this framework, r_i designates the distance value associated with pixel point p_i within the image coordinate system (u_i, z_j) of the current frame, and $r_i^{k\to l}$ signifies the variance in distance for the identical pixel point after transforming subsequent frames from *k* to *l*. This methodology undertakes the calculation of normalized differences solely for pixel points endowed with measurement values; for those pixels where measurements are indirectly present, the normalized difference is designated as 0. This is reflected as the nonreflective background portion within the residual image. Through this technique, the distance modifications exhibited by moving entities within the residual images become distinctly discernible. Aside from dynamic noise, the alterations in distance for other moving pixels escalate with an increase in reflectivity. However, this manifestation is not as evident for moving objects characterized by slower velocities or smaller dimensions, necessitating an adjustment in the number of residual images for detailed examination. The broader the frame range encompassed by the sliding

window employed to generate residual images, the more effective is the segmentation outcome for objects moving at a slower pace. The final residual images, when integrated with range images, are concatenated to form a new channel that encompasses both spatial and temporal sequence information. Each fused pixel (u_i, z_j) contains distinct information $(x_i, y_i, z_i, r_i, e_i, d^0_{1,i}, ..., d^0_{j,i}, ..., d^0_{N-1,i})$. $d^0_{j,i}$ represents the residual image calculated between the last j frames and the current frame. In practice, the system's sensitivity to dynamic objects can be enhanced by adjusting the number of image frames included within the sliding window. Figure 4 shows the calculated image of the binarized residuals. This optimized integration facilitates a more refined detection and analysis of dynamic entities within the observed environment.

3. Results and Discussion

3.1 Experimental setup and data acquisition

The experimental setup in this study employs the AGILE mobile platform, which is equipped with a nine-axis IMU for precise inertial navigation, four infrared obstacle avoidance sensors, and wheel motors as the propulsion components, achieving a maximum speed of 5 m/s and a load capacity of 50 kg. As shown in Fig. 5, the control system utilizes a high-performance industrial computer with an i7 CPU and 16 GB of memory, supporting the ROS operating system and ensuring robust computational power and stability. Additionally, the experimental platform is outfitted with HESAI's Pandar40 LiDAR, capable of 360-degree omnidirectional scanning, high-density point cloud output, and long-range distance measurement, facilitating efficient, accurate, and comprehensive environmental perception.

Fig. 4. (Color online) Binarized residual image obtained after computing.

Fig. 5. (Color online) Platform for experiments.

On the software front, the experimental platform operates on Ubuntu 18.04, integrated with the ROS robotic operating system, providing reliable software support and a rich library of algorithms for the experiments. To validate the dynamic point removal method proposed in this paper, the publicly available KITTI dataset is used. Given that the KITTI dataset contains relatively few scenarios with dynamic objects, in this study, we specifically extract segments containing dynamic objects from the KITTI odometry sequences 03, 08, and 09 for experimentation. Additionally, experiments are conducted using a proprietary campus dataset recorded with the self-constructed experimental platform to further validate the proposed approach.

3.2 Evaluation of dynamic point removal performance

We integrated with the LVI-SAM algorithm's preprocessing and utilized the SalsaNext neural network for semantic segmentation, followed by the channeling of the processed range images and residual images into a CNN for dynamic point analysis. The results of experiments demonstrate that the model employing eight residual images excels in detecting slower-moving objects. Conversely, the model with a single residual image showcases heightened sensitivity in identifying fast-moving entities. Hence, dynamic point analysis is executed using a combined multiresidual model ranging from one to eight residual images. It was observed that when the number of combined residuals exceeds eight, there is a notable decline in both the method's efficiency and its effectiveness in detecting dynamic points.

Figure 6 shows sequential schematic illustrations of local dynamic points in the KITTI dataset sequences 03, 07, and 08. Panel (a) shows the effects of removal utilizing the more recent algorithm, DGCNN. It is observable that rapidly moving vehicles cannot be effectively eliminated by this approach. Conversely, our method incorporates both temporal and spatial information, enabling the efficient identification and removal of fast-moving objects.

Experiments conducted on the KITTI dataset demonstrate that the proposed algorithm, based on temporal consistency, surpasses the mainstream approaches based on DGCNN in removing dynamic point clouds. This advantage enables better mapping results in dynamic environments. The performance of dynamic point cloud identification is evaluated using precision and recall

(a) (b)

Fig. 6. (Color online) Motion object feature point rejection: (a) 8-residual model and (b) 1-residual model.

rates, with higher $F1$ scores indicating superior recognition capabilities. As illustrated in Tables 1 and 2 and Fig. 7, our method achieves significant improvements in recognition accuracy at the expense of a slight reduction in speed. The experiment encompasses a comparison with various mainstream neural networks on the KITTI dataset.

In addition to validating the effectiveness on public datasets, we also collected data within a campus setting. From Figs. 8 and 9, it is evident that the constructed map experienced slight distortions due to the dynamic environment created by multiple pedestrians moving near the vehicle's trajectory, leaving ghost images of dynamic object movements on the ground. After

Table 1
Dynamic point removal performances of various methods.

Method	PR (%)	RR (%)	$F1$	Time (ms)
Pointnet++	88.912	94.763	0.917	107
PointCNN	90.309	90.692	0.905	96
DGCNN	90.565	93.895	0.922	83
Ours	94.565	98.282	0.964	103

Table 2
Trajectory errors of different algorithms.

	ALOAM	Lego-LOAM	LVI-SAM	Dynamic-SLAM	Ours
max	11.6687	127.01442	5.225659	4.748566	2.991671
mean	2.22051	21.360402	0.771923	0.715856	0.557303
median	1.561343	14.668778	0.483162	0.396458	0.467972
min	0.419578	3.645072	0.257591	0.213685	0.030178
rmse	3.166035	30.425564	1.210426	1.169874	0.851084
sse	1774.208792	178662.984494	8.653313	8.541686	7.028927
std	2.256793	21.666752	0.932345	0.897425	0.503529

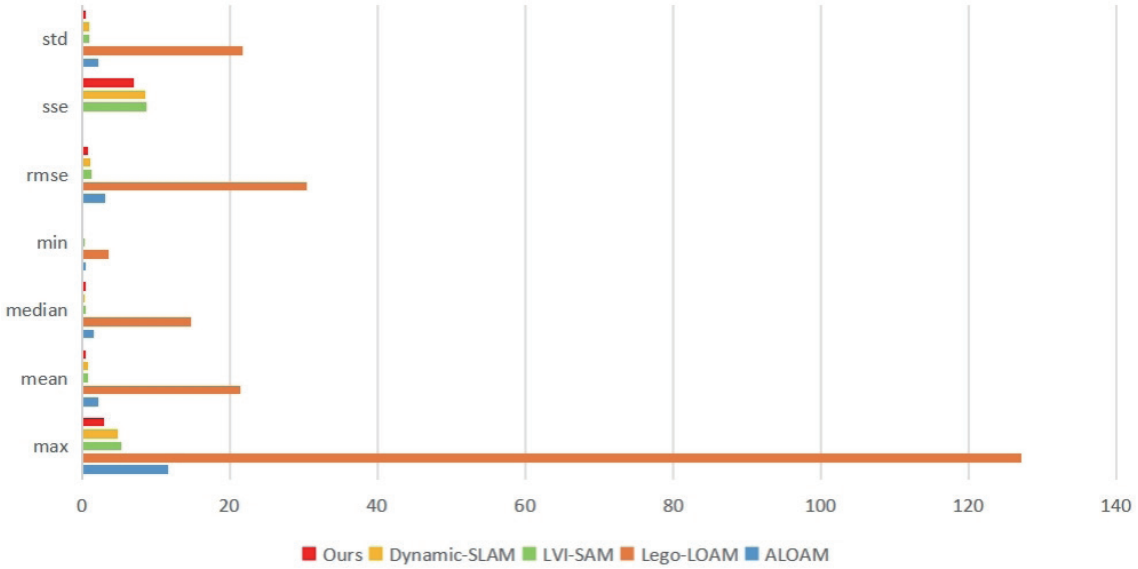
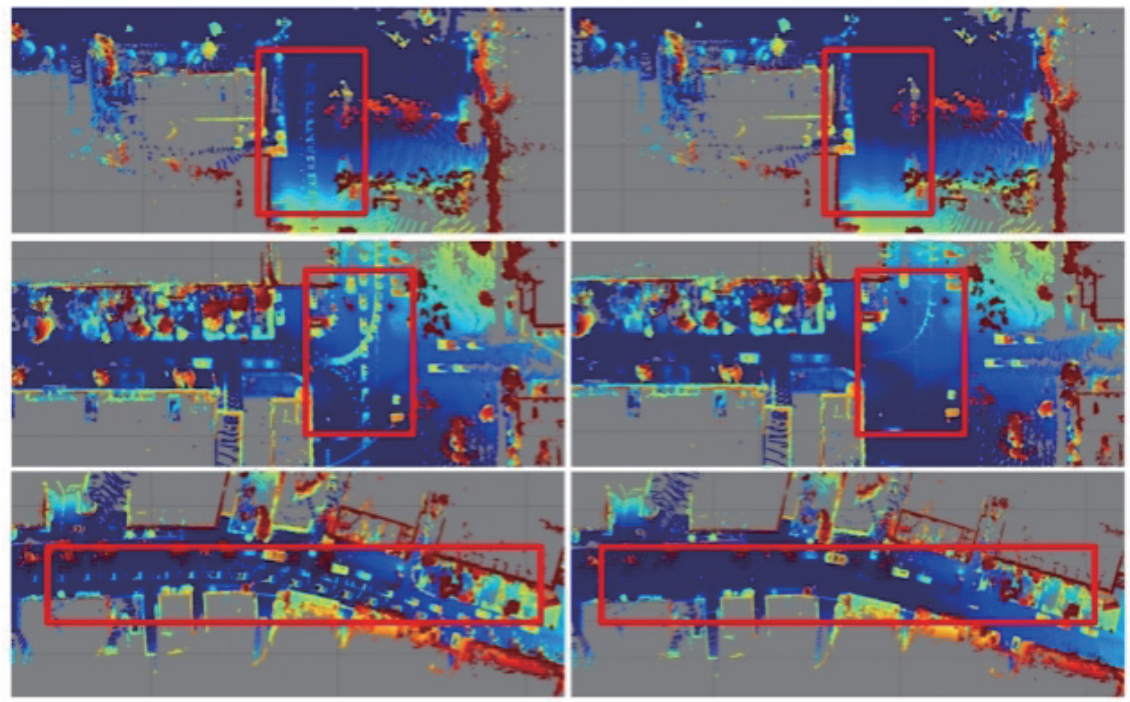
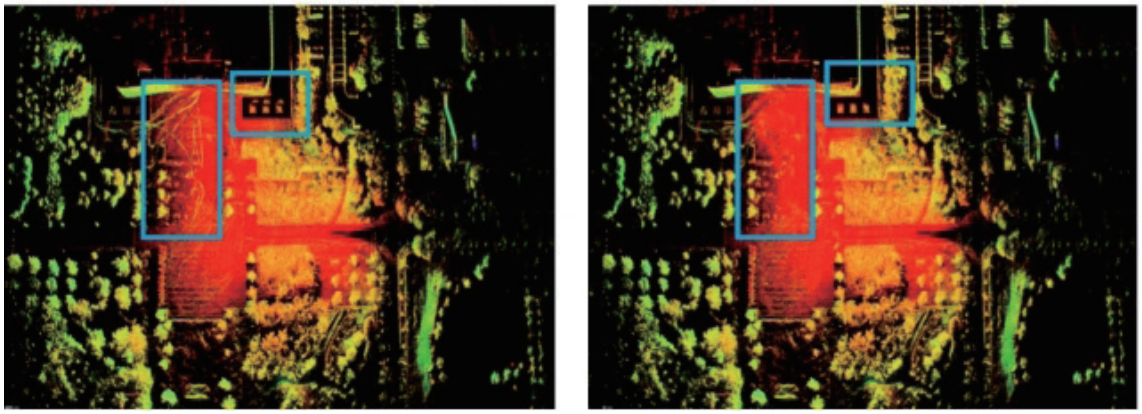


Fig. 7. (Color online) Absolute pose errors of various methods.



(a) (b)

Fig. 8. (Color online) Results of dynamic point removal in Kitti dataset. (a) Ghosting artifacts are caused by local dynamic points in the sequences from 03, 07, and 08. (b) Results after their removal.



(a) (b)

Fig. 9. (Color online) Mapping with dynamic points removed. (a) Original mapping results. (b) Outcomes after applying our method to eliminate ghosting artifacts caused by bicycles and pedestrians.

integrating the method developed in this study, distortions in the pond and structural deformations no longer occur, and the ghost images on the map also disappear as dynamic points are no longer included in the map construction process.

In addition, we have conducted comparative analyses of various algorithms based on the mapping trajectories in dynamic environments, utilizing the evo tool for data analysis of experimental trajectories.

Comparative analysis from the trajectory deviation and pose estimation graphics in Fig. 9 indicates that the method developed in this study aligns more closely with the actual pose data than do other algorithms without dynamic object removal. Our method significantly outperforms others as it incorporates spatial and temporal information through range and residual images, enhancing the precision of dynamic object extraction without substantially increasing computational resources, thus improving the accuracy of the mapping process. The ALOAM algorithm, owing to its lower robustness to interference from dynamic objects, exhibits significant errors during the solving process, resulting in substantial deviations in both the movement trajectory and the platform pose estimation. Although other algorithms demonstrate some robustness in dynamic scenarios, from the error comparison chart, we can observe more intuitively that our method reduces the root mean square error by approximately 27% and the maximum error by about 37% compared with mainstream algorithms. The performance improvements are particularly notable in two sets of data from the campus dataset, sufficiently demonstrating the method's effectiveness and real-time capabilities in dynamic environments.

4. Conclusions

In this study, we tackled the challenge of vehicle localization in dynamic environments where satellite navigation is not an option. We proposed a novel localization method that leverages spatiotemporal information. Our method was rigorously tested using both public datasets and data we collected ourselves, and the results showcased its real-world applicability. Initially, we explored techniques for extracting dynamic point clouds in these complex environments. We specifically examined the obstacles associated with mapping in dynamic settings using SLAM technology and how cumulative errors can affect mapping outcomes. To address these issues, we introduced a technique that converts point cloud data into range and residual images, which could then be analyzed for dynamic point clouds using deep learning networks. To further improve the effectiveness of our method, we developed a sliding-window approach for processing multiple frames of point cloud data. This technique accumulates residual data from one to eight frames to create a composite model for input into the network while also replacing data that is older than seven frames. Our results showed a significant enhancement in the robustness and accuracy of existing algorithms for eliminating dynamic points. Additionally, this research yielded a reliable solution for mapping in dynamic environments relevant to autonomous navigation. The proposed method not only provided new insights into managing dynamics in spatial contexts but also highlighted the importance of integrating semantic understanding with traditional data processing techniques.

References

1 J. Davison, I. D. Reid, N. D. Molton, and O. Stasse: IEEE Trans. Pattern Anal. Mach. Intell. **29** (2007) 1052. https://doi.org/10.1109/TPAMI.2007.1049
2 R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós: IEEE Trans. Robot. **31** (2015) 1147. https://doi.org/10.1109/TRO.2015.2463671
3 T. Qin, P. Li, and S. Shen: IEEE Trans. Robot. **34** (2018) 1004. https://doi.org/10.1109/TRO.2018.2853729
4 T. Shan and B. Englot: 2018 IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS, Madrid, Spain) (2018) p. 4758. https://doi.org/10.1109/IROS.2018.8594299
5 T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus: 2020 IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS, Las Vegas, NV, USA) (2020) p. 5135. https://doi.org/10.1109/IROS45743.2020.9341176
6 T. Shan, B. Englot, C. Ratti, and D. Rus: 2021 IEEE Int. Conf. Robot. Autom. (ICRA, Xi'an, China) (2021) p. 5692. https://doi.org/10.1109/ICRA48506.2021.9561996
7 A. M. Kurup and J. P. Bos: arXiv Preprint arXiv:2109.07078 (2021). https://doi.org/10.48550/arXiv.2109.07078
8 H. Lim, S. Hwang, and H. Myung: IEEE Robot. Autom. Lett. **6** (2021) 2272. https://doi.org/10.1109/LRA.2021.3061363
9 C. Yu, Z. Liu, X. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei: 2018 IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS, Madrid, Spain) (2018) pp. 1168–1174. https://doi.org/10.1109/IROS.2018.8593691
10 B. Bescos, J. M. Fácil, J. Civera, and J. Neira: IEEE Robot. Autom. Lett. **3** (2018) 4076. https://doi.org/10.1109/LRA.2018.2860039
11 K. He, G. Gkioxari, P. Dollár, and R. Girshick: 2017 IEEE Int. Conf. Comput. Vision (ICCV, Venice, Italy) (2017) pp. 2980–2988. https://doi.org/10.1109/ICCV.2017.322
12 F. Zhong, S. Wang, Z. Zhang, C. Chen, and Y. Wang: 2018 IEEE Winter Conf. Appl. Comput. Vision (WACV, Lake Tahoe, NV, USA) (2018) p. 1001. https://doi.org/10.1109/WACV.2018.00115
13 T. Naseer, M. Ruhnke, C. Stachniss, L. Spinello, and W. Burgard: 2015 IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS, Hamburg, Germany) (2015) p. 2529. https://doi.org/10.1109/IROS.2015.7353721
14 L. Xiao, J. Wang, X. Qiu, Z. Rong, and X. Zou: Robot. Auton. Syst. **117** (2019) 1. https://doi.org/10.1016/j.robot.2019.03.012
15 G. Bresson, Z. Alsayed, L. Yu, and S. Glaser: IEEE Trans. Intell. Veh. **2** (2017) 194. https://doi.org/10.1109/TIV.2017.2749181
16 G. Bresson, T. Féraud, R. Aufrère, P. Checchin, and R. Chapuis: IEEE Trans. Intell. Transp. Syst. **16** (2015) 1827. https://doi.org/10.1109/TITS.2014.2376780
17 J. Xie, F. Nashashibi, M. Parent, and O. G. Favrot: 2010 11th Int. Conf. Control Autom. Robot. & Vision (ICARCV, Singapore) (2010) p. 1397. https://doi.org/10.1109/ICARCV.2010.5707329
18 G. A. Bekey: IEEE Robot. & Autom. Mag. **15** (2008) 110. https://doi.org/10.1109/MRA.2008.928399
19 P. Bender, J. Ziegler, and C. Stiller: 2014 IEEE Intell. Veh. Symp. Proc. (Dearborn, MI, USA) (2014) p. 420. https://doi.org/10.1109/IVS.2014.6856487
20 R. Wang, Z. Li, J. Cao, T. Chen, and L. Wang: 2019 Int. Joint Conf. Neural Networks (IJCNN, Budapest, Hungary) (2019) pp. 1–6. https://doi.org/10.1109/IJCNN.2019.8852406
21 L. Chen, Z. Ling, Y. Gao, R. Sun, and S. Jin: Complex Intell. Syst. **9** (2023) 5653. https://doi.org/10.1007/s40747-023-01031-5
22 T. Cortinhal, G. Tzelepis, and E. E. Aksoy: Adv. Vis. Comput. (ISVC 2020, San Diego, CA, USA) (2020) pp. 207–222. https://doi.org/10.1007/978-3-030-64559-5_16
23 F. Li, C. Fu, D. Sun, J. Li, and J. Wang: Big Data Res. **36** (2024) 100463. https://doi.org/10.1016/j.bdr.2024.100463
24 G. Yang, S. Mentasti, M. Bersani, Y. Wang, F. Braghin, and F. Cheli: 2020 AEIT Int. Conf. Electr. Electron. Technol. Automot. (AEIT AUTOMOTIVE, Turin, Italy) (2020) pp. 1–6. https://doi.org/10.23919/AEITAUTOMOTIVE50086.2020.9307387
25 X. Liu, C. R. Qi, and L. J. Guibas: 2019 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR, Long Beach, CA, USA) (2019) pp. 529–537. https://doi.org/10.1109/CVPR.2019.00062

About the Authors

Junyang Bian received his B.S. degree from Anhui University of Civil Engineering and Architecture, China, in 2022. He is now studying for his M.S. degree at Beijing University of Civil Engineering and Architecture. His research interests include point cloud semantic segmentation and LiDAR SLAM.

He Huang received his B.S. degree from Wuhan University, China, in 2000 and his M.S. and Ph.D. degrees from Sungkyunkwan University, South Korea, in 2004 and 2010, respectively. Since 2010, he has been a lecturer and associate professor at the Beijing University of Civil Engineering and Architecture, China. His research interests are in autonomous driving, high-precision navigation maps, and visual navigation and positioning. (huanghe@bucea.edu.cn)

Junxing Yang graduated from the School of Surveying and Mapping of Wuhan University, China, in December 2021 with a Ph.D. degree in engineering. He is mainly engaged in research in 3D reconstruction, computer vision, autonomous driving, image stitching, and other related fields. (yangjunxing@bucea.edu.cn)

Junxian Zhao received his B.S. degree from Beijing University of Civil Engineering and Architecture, China, in 2021. He is now studying for his M.S. degree at the same university. His research interests are in semantic SLAM and indoor positioning technology.

Siqi Wang received her B.S. degree from Beijing University of Civil Engineering and Architecture, China, in 2022. She is now studying for her M.S. degree at the same university. Her research interests are in semantic SLAM and indoor positioning technology.