Enhancing environmental modeling and 3D point cloud construction for unmanned vehicles using laser SLAM and stereo vision with Convolutional Neural Networks

Muwang Cao[1], Jiao Jiang[2,4,*], Haoran Zhang[3]

[1]College of Mechanical and Vehicle Engineering, Chongqing, China
[2]College of Engineering Science and Technology, Shanghai Ocean University, Shanghai, China
[3]School of Aeronautics and Astronautics, Sichuan University, Chengdu, China

[4]jj581@exeter.ac.uk
*corresponding author

Abstract. This research presents an innovative SLAM algorithm that integrates Convolutional Neural Networks (CNNs) with LIDAR and stereo vision to significantly enhance the accuracy of environmental modeling and the construction of dense 3D point cloud maps in complex and dynamic surroundings. By processing pre-recorded video data and employing advanced image segmentation techniques, this study achieves a deep fusion of visual and geometric data, resulting in highly detailed and precise 3D representations of the environment. The experimental results demonstrate that this approach effectively detects and excludes dynamic objects, thereby significantly improving the overall quality, robustness, and reliability of the constructed maps. This work represents a substantial advancement in SLAM technology, particularly in its capability to model and capture intricate environmental details under varying conditions, making it a powerful tool for precise environmental mapping.

Keywords: Convolutional Neural Networks (CNNs), LIDAR, Stereo vision, Image segmentation, 3D point cloud maps.

1. Introduction

In recent years, the rapid advancement of artificial intelligence and robotics has markedly accelerated the development of unmanned vehicles, making them pivotal in sectors such as automated logistics, intelligent security, and autonomous navigation. A cornerstone technology that enables these vehicles to operate autonomously is Simultaneous Localization and Mapping (SLAM). SLAM empowers vehicles to ascertain their real-time location based on sensor data while concurrently constructing a map of their environment. However, as environmental complexity increases, traditional SLAM systems encounter significant challenges, particularly in dynamic settings where objects can move unpredictably. Xie (2022) systematically analyzed the limitations of traditional visual localization methods in complex environments, emphasizing the need for advanced techniques like CNNs to enhance robustness [1]. The grid-based potential field approach, as proposed by Jung and Kim [2], has demonstrated effectiveness in addressing the local minimum issue by integrating repulsive fields between adjacent obstacles. This approach provides a solid foundation for further advancements, especially in the integration of sensor data to enhance environmental modeling.

The integration of deep learning techniques, particularly Convolutional Neural Networks (CNNs), into SLAM systems has opened new pathways to addressing these challenges. CNNs are especially proficient at extracting and integrating features from images, thereby enhancing the accuracy of SLAM systems in dynamic settings. Moreover, Tang et al.'s [3] advancements in efficient correspondence prediction techniques have played a crucial role in enhancing the robustness and real-time capabilities of SLAM systems in dynamic environments.

Despite the progress made in intelligent vehicles currently available on the market, there are still some limitations. Solutions that rely on either a single camera or a single LIDAR sensor offer unique advantages, yet each also presents specific challenges. LIDAR, known for its detailed environmental data capture, performs well in navigation and obstacle avoidance but can exhibit inconsistent point cloud density under certain conditions, particularly in sparse scenes where it may fail to detect all objects. Conversely, stereo cameras can capture color and texture details, but their effectiveness can be constrained by factors such as limited field of view and range, leading to potential blind spots in the areas monitored by the unmanned vehicle.

Recent developments in deep learning have had a profound impact on point cloud processing and semantic segmentation, both critical for environmental modeling and the construction of dense 3D point cloud maps. For instance, Li et al. introduced PointCNN, a convolutional neural network specifically designed for processing point cloud data. This innovation has significantly improved the ability to generate dense point cloud maps [4]. Building on this foundation, significant advancements in deep learning frameworks for 3D classification and segmentation of point clouds have been made, introducing innovative approaches to learning features directly from raw point cloud data. These advancements have set new standards in accurate environmental modeling. Subsequent research, including the work by Charles et al., has further refined these architectures to address more complex classification and segmentation tasks, reinforcing their role as critical tools in 3D point cloud processing [5].

In the realm of action recognition, Demisse et al. proposed a novel pose encoding method that substantially enhances the robustness of skeleton-based action recognition in noisy environments, demonstrating the versatility of deep learning across different applications [6]. Additionally, Chen et al. made significant strides in semantic image segmentation with the development of DeepLab. This model integrates deep convolutional networks, atrous convolution, and fully connected Conditional Random Fields (CRFs), greatly improving the accuracy of object segmentation within images, which in turn advances the precision of environmental modeling [7,8].

To address the challenges mentioned above, this paper presents a SLAM algorithm that integrates laser-assisted stereo vision. This algorithm tackles issues such as the low detection accuracy and limited range often encountered in stereo vision methods, as well as the degradation caused by insufficient geometric information in purely laser-based approaches. The proposed approach leverages the strengths of visual sensors, which excel at capturing color and texture details, and combines them with structural features derived from laser scanning to achieve more accurate and robust pose estimation. In this method, CNNs are utilized to process images and extract key features efficiently, which are critical for real-time environmental perception and scene comprehension. By merging the visual information obtained from CNNs with the geometric data provided by LIDAR, the system can construct high-precision 3D point cloud maps, significantly enhancing the visual perception and operational accuracy of unmanned vehicles in complex environments.

Future studies will explore the integration of additional sensor data types to further enhance the adaptability and robustness of SLAM systems in dynamic environments. Moreover, improving the real-time performance and computational efficiency of these algorithms will be crucial for enabling reliable autonomous navigation across a broader spectrum of applications. It is anticipated that the continuous advancement of these technologies will make substantial contributions to the progress of intelligent logistics, service robotics, and other fields reliant on automation.

2. Overall Structure of the Stereo Vision SLAM Unmanned Vehicle

This research developed a stereo vision SLAM unmanned vehicle by integrating stereo vision with LIDAR SLAM technology, aimed at enhancing autonomous navigation capabilities in complex environments. The system's hardware and software architectures were carefully crafted to ensure efficient performance in dynamic and unpredictable conditions.

2.1. Control System

The control core of the unmanned vehicle is built around the Raspberry Pi 4B, chosen for its computational prowess and versatility. Equipped with a robust ARM Cortex-A72 CPU, the Raspberry Pi 4B efficiently handles complex SLAM computations, particularly in real-time visual data processing. Ubuntu 20.04 serves as the operating system, and the control system was developed using the Robot Operating System (ROS) framework. ROS offers a comprehensive suite of tools and libraries that facilitate the development and integration of diverse robotic applications, especially in SLAM, path planning, and autonomous navigation.

The ORB-SLAM algorithm, recognized for its effectiveness in SLAM applications, was integrated into the vehicle's control system with the assistance of ROS, enabling real-time map generation and localization. To overcome the challenges associated with traditional potential field methods in dynamic environments, our approach utilizes the grid-based potential field method developed by Jung and Kim [2]. Furthermore, integrating techniques such as Direct Sparse Odometry, as discussed by Engel et al. [9], enhances the robustness and accuracy of our SLAM implementation.

2.2. Data Collection

The data collection system of the unmanned vehicle comprises stereo cameras and LIDAR, working in unison to deliver comprehensive environmental perception. High-resolution USB stereo cameras capture stereo vision data, which is utilized to derive depth information, providing accurate visual input for the SLAM system. These stereo cameras obtain stereo information from images and generate dense depth maps through stereo matching algorithms, laying the groundwork for environmental modeling.

The N10P LIDAR further enhances the accuracy of environmental modeling by providing high-precision point cloud data. LIDAR is particularly effective in environmental scanning, capturing geometric features with greater detail than visual data alone. By combining stereo cameras with LIDAR, the system maximizes their respective strengths, achieving precise 3D environmental modeling.

During data collection, the RGB-D data from the stereo cameras and the point cloud data from the LIDAR are processed by the Raspberry Pi and fed into the SLAM system in real-time. The SLAM algorithm on the Raspberry Pi processes this sensor data to create a high-precision environmental map and continually updates the vehicle's position information, enabling smooth navigation through dynamic environments.

2.3. Appearance

The exterior design of the unmanned vehicle was carefully considered to ensure its adaptability for both indoor and outdoor use. The vehicle features a compact structure, with stereo cameras, LIDAR, and other sensors strategically arranged to provide optimal working views for each sensor without causing interference. The design not only fulfills functional requirements but also enhances the overall aesthetics and practicality of the system. As illustrated in Figure 1, the vehicle's layout is compact and demonstrates excellent environmental adaptability, ensuring the stable operation of the system.

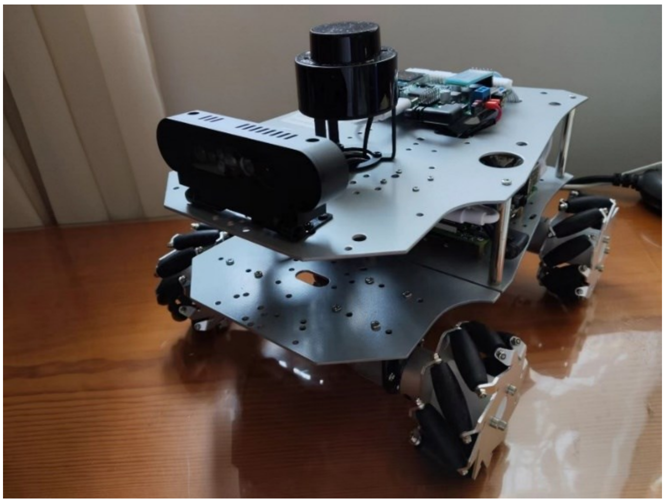


Figure 1. Exterior Design of the Stereo Vision Autonomous Vehicle

3. Camera Calibration

Before utilizing the cameras, it is essential to perform camera calibration to correct for image distortion caused by lens aberrations. The primary goal of camera calibration is to establish a mapping relationship from the 3D world to the 2D image plane, enabling accurate determination of the 3D geometric position of any point on an object's surface and its corresponding point in the image during object reconstruction and recognition. This process involves calculating both the intrinsic and extrinsic parameters of the camera.

Before utilizing the cameras, it is essential to perform camera calibration to correct for image distortion caused by lens aberrations. This fundamental process involves calculating both the intrinsic and extrinsic parameters of the camera to ensure accurate mapping from the 3D world to the 2D image plane. Consistent with standard procedures in binocular vision systems, accurate calibration is essential for enhancing environmental perception and navigation precision [10].

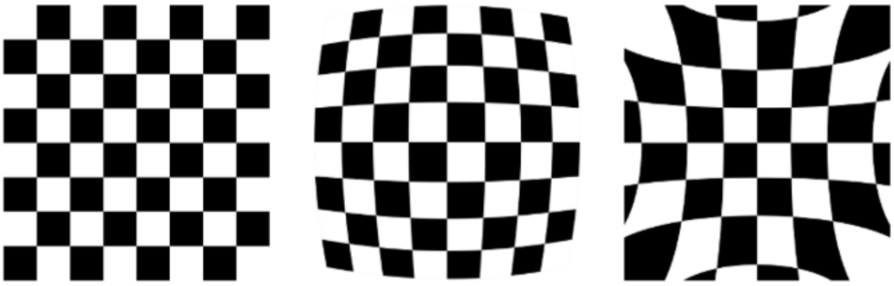


Figure 2. (From left to right) Normal, Barrel Distortion, Pincushion Distortion

Figure 2 illustrates various types of lens distortions, including normal, barrel distortion, and pincushion distortion. Intrinsic parameters include the focal length (fx, fy) and principal point coordinates (cx, cy), which define the projection relationship from 3D space to the 2D image plane. Correcting these distortions ensures that straight objects are represented as straight lines in the image, minimizing geometric errors caused by lens distortion.

Extrinsic parameters describe the rotation and translation relationships between the camera's coordinate system and the world coordinate system. By solving the rotation matrix (R) and translation vector (T), points in the 3D world coordinate system can be transformed into the camera coordinate system and then projected onto the image plane using the intrinsic parameters.

To obtain accurate depth images, this study employed a stereo matching algorithm. This algorithm utilizes images captured simultaneously by two cameras, analyzing the disparity between corresponding points in the left and right images to calculate the depth information of objects in the scene. The key steps in stereo matching include image rectification and disparity calculation:

Image Rectification: By correcting image distortion, the optical axes of the left and right cameras are aligned parallel, ensuring that the projection of the same object in both images appears on the same horizontal line. This step utilized the image calibration package in ROS, with a (9×7) checkerboard pattern for calibration. Figure 3 shows the image before calibration, where the obvious distortion of the checkerboard indicates the presence of aberration. Figure 4 shows the image after calibration, where the distortion has been corrected.

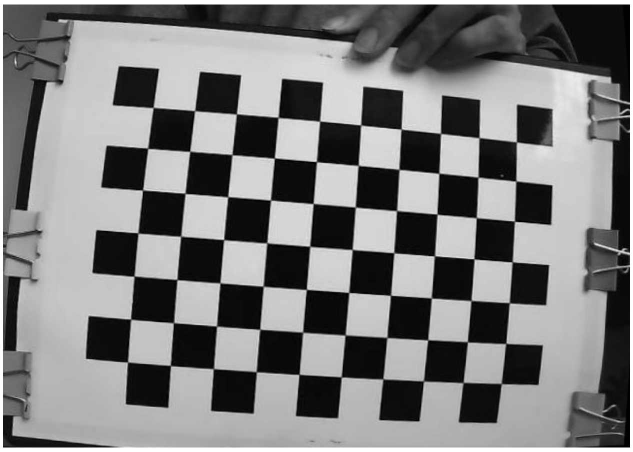


Figure 3. Stereo Camera Image Before Calibration

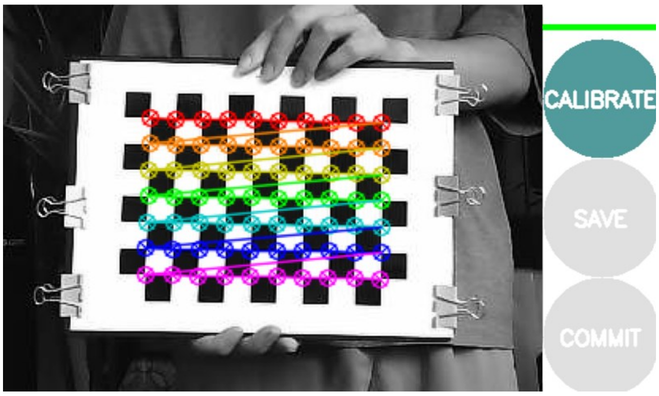


Figure 4. Stereo Camera Image After Calibration

Disparity Calculation: Following image rectification, the pixels in the left and right images are matched using epipolar constraint methods to calculate the disparity, which is then used to deduce depth. Disparity minimization is achieved through an energy cost function, and the resulting depth maps provide essential data for 3D reconstruction.

This camera calibration process lays the groundwork for the stereo SLAM system, ensuring the accuracy and stability of subsequent image processing and feature extraction.

4. Fundamental Concepts of Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) represent a foundational architecture in deep learning, particularly well-suited for handling image and video data. The primary strength of CNNs lies in their ability to automatically learn and extract essential features from images without requiring manual feature engineering. This ability makes CNNs highly effective for tasks such as image classification, object detection, and semantic segmentation.

4.1. Basic Structure of CNNs

CNNs are composed of several key components:

Convolutional Layer: This layer applies convolutional kernels (or filters) to the input data, capturing local features from images. These kernels are trained to recognize edges, corners, textures, and other significant image features.

Activation Function: The Rectified Linear Unit (ReLU) is commonly used as the activation function. ReLU introduces non-linearity into the model, allowing it to approximate more complex functions.

Pooling Layer: This layer down-samples the feature maps produced by the convolutional layers, reducing the computational burden and memory requirements while improving the model's robustness to variations. Common pooling methods include Max Pooling and Average Pooling.

Fully Connected Layer: In the later stages of the network, fully connected layers are employed to integrate the high-level features extracted by the convolutional and pooling layers, typically for classification or regression tasks.

Figure 5 provides a visual representation of a Convolutional Neural Network (CNN) including pooling operations. It details how the various components—convolutional layers, activation functions, pooling layers, and fully connected layers—are organized to effectively analyze and classify image data.

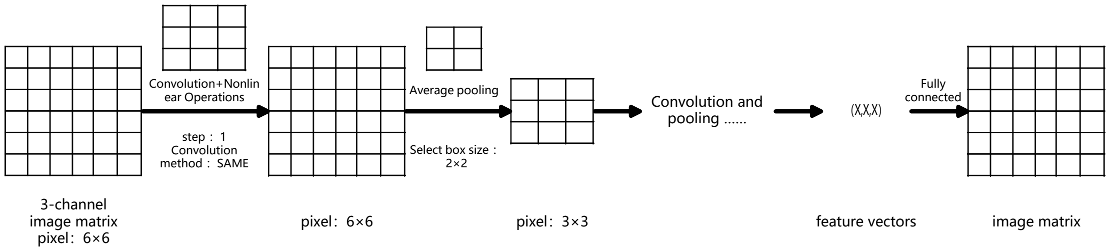


Figure 5. Convolutional Neural Network with Pooling

4.2. Application of CNNs in DeepLabv3+

The DeepLabv3+ architecture builds on the robust feature extraction capabilities of traditional CNNs and tailors them to the needs of modern semantic segmentation tasks. By integrating atrous (dilated) convolution and an encoder-decoder structure, the model effectively balances segmentation accuracy and computational efficiency.

5. Image Segmentation Based on DeepLabv3+

5.1. Network Architecture and Methodology

This study employs the DeepLabv3+ architecture for the image segmentation component, a sophisticated Convolutional Neural Network (CNN) designed to enhance segmentation accuracy and efficiency. DeepLabv3+ extends traditional CNNs by integrating atrous (dilated) convolution techniques with an encoder-decoder structure. This architecture is particularly adept at managing multi-scale features and refining feature maps to more effectively capture image details. The structure of the algorithm is shown in Figure 6:

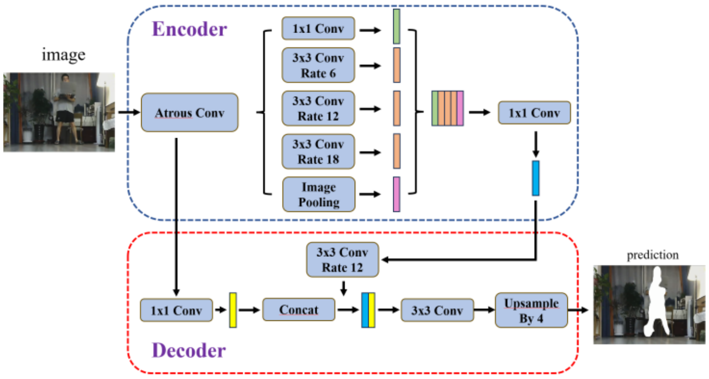


Figure 6. Structure of the DeepLabv3+ Algorithm

Encoder: The encoder utilizes ResNet or Xception as the backbone network for feature extraction and incorporates the Atrous Spatial Pyramid Pooling (ASPP) module. ASPP employs convolutional kernels with varying dilation rates to capture features at different scales, allowing the model to gather rich contextual information from the image.

Decoder: The decoder processes high-level and low-level feature maps received from the encoder. Using 1x1 convolutions for dimensionality reduction followed by upsampling, the decoder generates high-resolution segmentation maps that align with the original input image size. This ensures that fine details in the image are preserved, which is crucial for accurate 3D reconstruction in SLAM applications.

5.2. Training Data

To evaluate the model's performance in dynamic scenarios, this study designed a specific indoor experimental setup. An adult repeatedly moved within the camera’s range on the mobile robot, simulating the presence of dynamic objects in real-world environments. This setup was used to assess the model's ability to detect and exclude dynamic objects.

The training dataset used in this experiment was based on the methodology presented by Costea et al. in their CVPR 2017 paper titled "Fast Boosting based Detection using Scale Invariant Multimodal Multiresolution Filtered Features." This dataset includes object detection tasks in various dynamic scenes, covering multiple target categories such as pedestrians and vehicles. The experimental results demonstrated that this method achieved an average precision (AP) of 83.79% in "simple" scenarios, 70.76% in "medium" difficulty scenarios, and 64.81% in "difficult" scenarios. These outcomes indicate that the method is effective in detecting dynamic objects like pedestrians(see Figure 7 for a detailed view of precision and recall changes in pedestrian detection tasks).

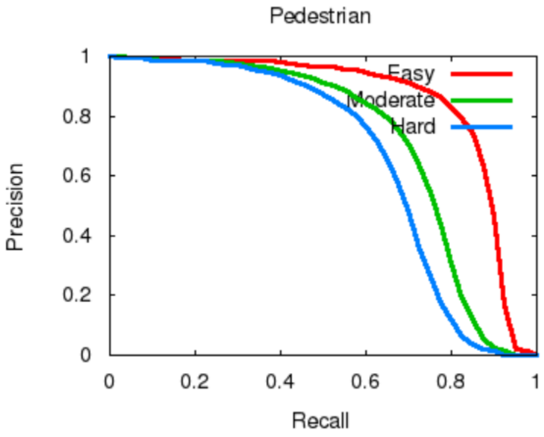


Figure 7. Precision and Recall Changes in Pedestrian Detection Tasks

Using this dataset for model training, along with precise 2D and 3D bounding box information, the model's detection capabilities in complex dynamic scenarios were enhanced through boosting techniques and multi-resolution feature extraction. A rigorous annotation process ensured the reliability of the data, making it suitable for both academic research and engineering applications. The model successfully identified and excluded dynamically moving pedestrian targets in indoor experiments, significantly improving the mobile robot's navigation capabilities in dynamic environments.

5.3. Image Segmentation

In this study, the code was run within an Ubuntu 20.04 environment, utilizing Python 3.8 as the programming language and leveraging PyTorch 1.9.0 for deep learning model development and training. To enhance the efficiency of processing large datasets and model training, the system was equipped with an NVIDIA GeForce RTX 3090 GPU, utilizing CUDA 11.1 for parallel computation. The development process was supported by Visual Studio Code (VS Code) as the integrated development environment, while Jupyter Notebook was employed for documenting experiments and visualizing results, allowing for real-time analysis and adjustments of experimental data. Dependency management was handled via a requirements.txt file, ensuring the reproducibility of the project environment.

Following the completion of model training using the KITTI dataset, a 1-minute 55-second video was captured using a stereo camera mounted on a mobile robot. This video was specifically designed to assess the performance of the trained model in a dynamic environment. The stereo camera's depth information was crucial for accurately segmenting and identifying moving objects, such as pedestrians, in the recorded scenes. This validation step was essential in demonstrating the model's effectiveness in real-world dynamic scenarios, ensuring that the mobile robot could navigate efficiently by detecting and excluding dynamically moving obstacles.

Table 1 presents a comparison of images before and after segmentation at various timestamps in the video. The table includes the time of capture, the original images before segmentation, and the segmented images after applying the model. This comparison highlights the effectiveness of the model in improving the clarity of moving objects and distinguishing them from the background.

Table 1. Comparison of images before and after segmentation.

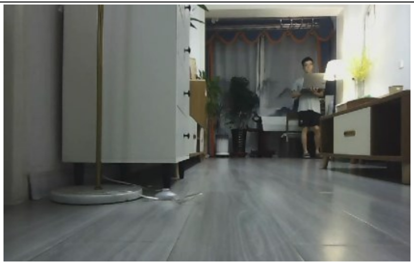
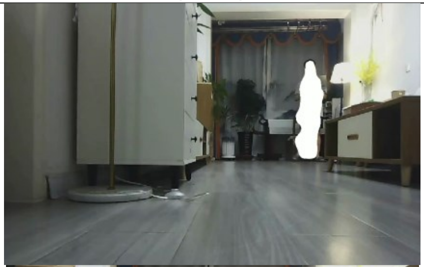
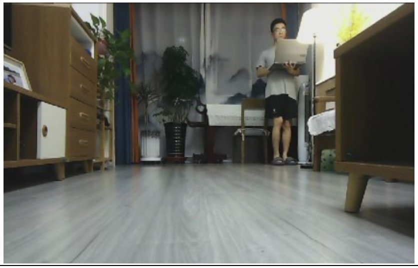
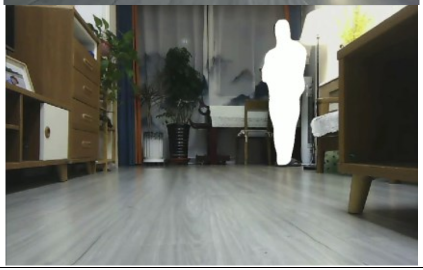
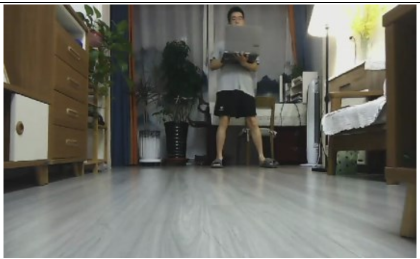
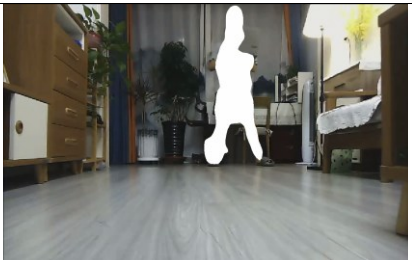
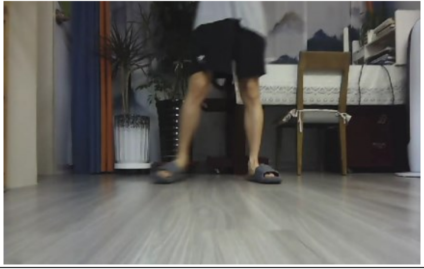
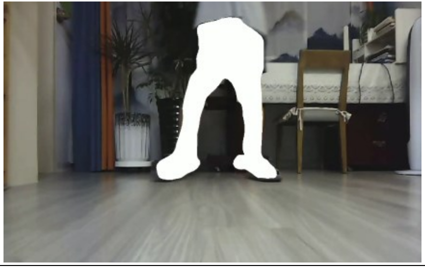
Time(MM)	Before image segmentat	After image segmentation
00:35		
01:17		

Table 1. (continued).

01:24		
01:33		

6. 3D Point Cloud Map Construction

6.1. Data Collection and Hardware Control

While the system's hardware architecture is designed to support real-time data collection, this study opted to use pre-recorded RGB-D video data for constructing dense 3D point cloud maps. The video data, captured by a stereo camera, documents the entire movement of the mobile robot within an indoor setting. Using pre-recorded data ensures consistent inputs across various experiments, allowing for rigorous validation of the SLAM system's performance.

The RGB-D data from the pre-recorded video is processed by the SLAM system to produce high-precision environmental maps. By simulating real-time data collection via ROS, the system processes this data to construct detailed and accurate point cloud maps, which are critical for precise navigation in dynamic environments.

6.2. Image Segmentation and Point Cloud Data Fusion

The pre-recorded RGB-D video data is input into the RTAB-Map node, where image segmentation techniques are applied to extract key features, such as pedestrians, from the video. These features are then combined with the geometric information provided by LIDAR to create an enhanced 3D point cloud map. Although the data is not captured in real-time, this method ensures that the segmentation and point cloud generation processes are both consistent and efficient, contributing to the overall accuracy of the SLAM system.

6.3. TF Transformation and Coordinate System Alignment

A critical step in constructing dense 3D point cloud maps is ensuring that all data is aligned within a common coordinate system. The RGB-D data and geometric information extracted from LIDAR must be unified under the same coordinate system for effective processing. The ROS TF transformation node manages this alignment, ensuring that all data inputs are correctly aligned for map construction.

6.4. Dense Point Cloud Map Generation

Once the data is aligned, the RTAB-Map node processes the fused RGB-D data as if it were real-time sensor input. This method generates dense point cloud maps that accurately capture both the geometric features and texture information of the environment. The resulting maps are robust and precise, as demonstrated in Figures 8 and 9, which compare the point cloud map to the actual environment.

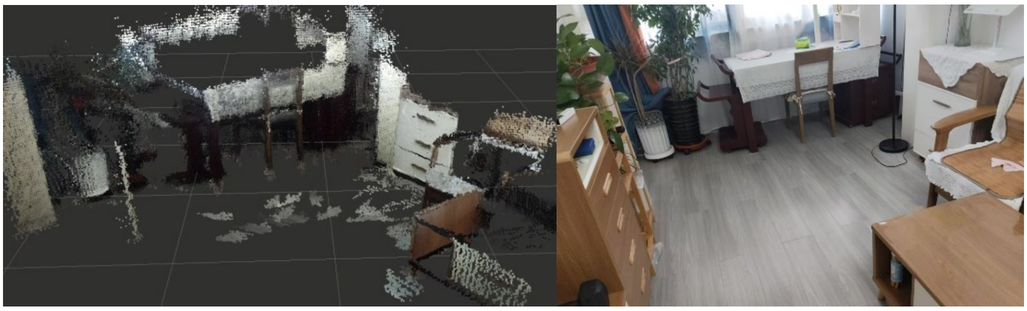


Figure 8. Comparison between the Point Cloud Map and Reality (1)

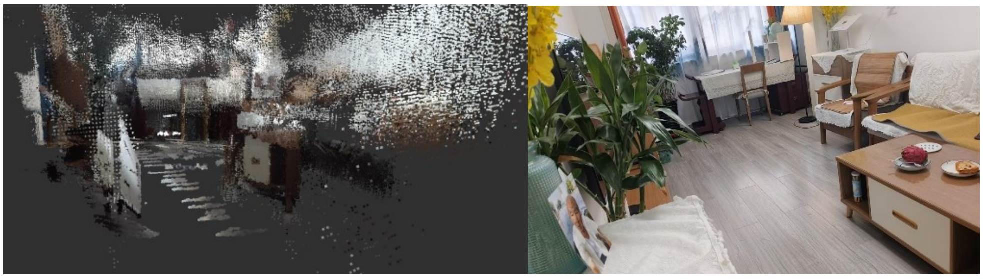


Figure 9. Comparison between the Point Cloud Map and Reality (2)

Through trajectory optimization and loop closure detection techniques, the generated map remains highly robust and accurate even when using pre-recorded video.

7. Conclusion

This research presents an innovative SLAM framework that merges deep learning techniques, particularly CNNs, with laser-assisted stereo vision, effectively tackling the complexities of navigating dynamic environments. Through the precise segmentation of pedestrians and other moving entities in video data, and the seamless integration of this segmented data into the SLAM system, the proposed approach shows a marked enhancement over conventional SLAM methodologies. Experimental findings confirm the method's efficacy in filtering out dynamically moving objects, which facilitates the creation of reliable and accurate 3D point cloud maps.

Looking ahead, future studies will focus on refining the algorithm's real-time capabilities and enhancing its robustness in environments with even greater dynamism. Moreover, integrating additional sensor modalities, such as thermal imaging and ultrasonic sensors, may further refine the spatial resolution and accuracy of the produced maps. These developments are anticipated to expand the scope of SLAM technology in sectors like smart logistics, service robotics, and other areas where autonomous navigation is crucial.

References

[1] Xie W. Research on Visual Localization Technology of Mobile Robots Based on Convolutional Neural Networks [D]. Anhui: Anhui University of Science and Technology, 2022.

[2] Jung, J. H. and Kim, D. H. (2020). Local Path Planning of a Mobile Robot Using a Novel Grid-Based Potential Method. *IEEE Transactions on Robotics* 20(1):26–34.

[3] Tang, J., Ericson, L., Folkesson, J., et al. (2019). GCNv2: Efficient Correspondence Prediction for Real-Time SLAM. *IEEE Robotics and Automation Letters* 4(2):1234–1241. DOI: 10.1109/LRA.2019.2927954.

[4] Li Y, Bu R, Sun M, et al. PointCNN: convolution on x-transformed points. *IEEE Trans actions on Pattern Analysis and Machine Intelligence* 41(8):1881–1893, 2019. DOI: 1 0.1109/TPAMI.2018.2864187.

[5] R. Charles, H. Su, M. Kaichun and L. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017 pp. 77-85.doi: 10.1109/CVPR.2017.16

[6] Demisse G G, Papadopoulos K, Aouada D, and Ottersten B. Pose encoding for robust skeleton-based action recognition. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Salt Lake City, UT, USA, 2018, pp. 301–3016. DOI: 10.1109/CVPRW.2018.00056.

[7] Chen L C, Papandreou G, Kokkinos I, Murphy K, and Yuille A L. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully conne cted CRFs. IEEE Trans. Pattern Anal. Mach. Intell. 40(4): 834–848, 2018. DOI: 10.1 109/TPAMI.2017.2699184.

[8] Chen LC, Zhu Y, Papandreou G, Schroff F, and Adam H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In: Ferrari V, Hebert M, Sminchisescu C, and Weiss Y (eds), *Computer Vision – ECCV 2018*, Lecture Notes in Computer Science, vol 11211. Springer, Cham, 2018.

[9] Engel, J., Koltun, V., and Cremers, D. (2018). Direct Sparse Odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40(3):611–625.

[10] Yuan W. Research on Binocular Vision Localization and Navigation of Unmanned Delivery Vehicles [D]. Guangdong: Guangdong University of Technology, 2019.