Efficient Object-Level Semantic Mapping with RGB-D Cameras

Benchun Zhou[1*], Furmans Kai[1], Yunxiang Fu[1] and Hao Pang[1]

[1]Institute for Material Handling and Logistics, Karlsruhe Institute of Technogy, Karlsruhe, 76131, Germany.

*Corresponding author(s). E-mail(s): benchun.zhou@kit.edu;

Abstract

To autonomously navigate in real-world environments, mobile robots require a dense map to guarantee safety, such as a 3D occupancy map. However, this map lacks semantic information for scene understanding. On the other hand, semantic objects can be introduced to the map with the help of deep neural networks, but they may suffer from critical run-time issues due to heavy processing components. In this paper, we present an efficient semantic mapping system to incrementally build a voxel-based map with individual objects. Firstly, a frame-wise object segmentation scheme is adopted to segment 3D objects from RGB-D images. Then, a new object association strategy with geometry and semantic descriptor is proposed to track and update object information, Finally, these objects are integrated into a CPU-based voxel mapping approach to incrementally build a global object-level volumetric map. Experiments on publicly available indoor datasets show that the proposed system achieves a good semantic mapping performance. Besides, our method outperforms other object-level mapping algorithms in terms of segmentation results and computational efficiency. Furthermore, the system is evaluated within a logistical robotic platform to demonstrate the use case in real-world applications.

Keywords: object segmentation, object-level mapping, efficient, application

1 Introduction

Semantic mapping aims to estimate the geometry of an environment and simultaneously attach a semantic label to the elements that are reconstructed in the map. With the aid of RGB-D cameras, mobile robots can perceive the surrounding space and map the world with essential information. Specifically, for autonomous navigation, the robotic vision system is able to discover, track and reconstruct object elements that appear in the environment. Recent research work has shown an impressive performance on object detection tasks from single RGB images, such as predicting object-wise bounding boxes (Redmon and Farhadi, 2017) or generating pixel-wise object masks (He et al, 2017). Although object detection in 2D images can be regarded as well studied, 3D object perception remains a difficult challenge due to different object appearances, poses, and viewpoints. Besides, real-time performance is another issue that should be considered when transferring to real-world applications.

On the other hand, robots require an up-to-date map to design a flexible and collision-free path when they navigate in the environment. Some researchers explore mapping systems by representing the environment with explicitly free space (Oleynikova et al, 2017), the reconstructed map can be directly used for navigation purposes.

On this basis, semantic information could be introduced to the existing map to create a meaningful map and benefit intelligent navigation (Grinvald et al, 2019).

Although various approaches have been proposed for semantic mapping systems with RGB-D cameras, they frequently encounter significant real-time limitations due to the computational demands of their processing components, such as pixel-wise object segmentation and point cloud processing (Nakajima and Saito, 2018). Current GPU technology can accelerate the computation process, but it is not always available and practical for some industrial projects due to cost, compatibility, and space constraints.

In this paper, we present an efficient semantic mapping system to incrementally build an object-level map with a localized RGB-D camera. For each frame, the use of an unsupervised geometric object detection algorithm provides object classes and corresponding 2D bounding boxes within the image. Next, a 3D object segmentation algorithm is adopted to obtain object points, remove outliers and generate a cuboid representation. Then, an object association considering a variety of geometric and semantic descriptors is proposed to match the detected objects in the current frame to existing mapped objects, where the detected objects will be either merged or introduced to the map. Finally, the object instances will be integrated into a voxel-based mapping system to incrementally reconstruct a global object-level volumetric map. Experiments on publicly available indoor datasets show that the proposed system achieves a good semantic mapping performance while reducing the computational cost. Furthermore, we also evaluate the system in an intra-logistics to demonstrate its effectiveness. In a nutshell, the contributions can be summarized as follows:

- An efficient segmentation method to extract object instances from RGB-D images with points and geometric representation.
- An object association strategy based on geometric and semantic descriptors to incrementally update object instances across multiple frames.
- An object-level semantic mapping system, where the object instances are integrated into the 3D volumetric map.
- Experiments on a public open-source dataset and real-world robotic platform to evaluate the performance of the proposed system.

2 Related Work

The related work is structured into object segmentation and semantic mapping. Firstly, we review the most important works that focus on 3D object segmentation. Then, previous methods of object-level semantic mapping are discussed.

2.1 Object Detection and Segmentation

Objects are high-level elements of the environment that can benefit semantic mapping and robotic navigation. Driven by deep learning methods, a number of research work has shown significant results on the task of 3D object segmentation.

One common solution is to detect object masks or bounding boxes from the RGB image and then extract 3D points from the corresponding depth pair. Mask R-CNN (He et al, 2017) is a well-known 2D object detector that can generate pixel-wise semantic masks of each instance, receiving state-of-the-art results on COCO instance-level semantic segmentation tasks. On this basis, some label fusion methods (Mascaro et al, 2022) can be used to associate object instances in 3D space. One of the major limitations of these pixel-wise segmentation methods is that they have a huge time complexity. In contrast, the YOLO network (Redmon and Farhadi, 2017) can generate 2D bounding boxes from RGB images with a comparative accuracy but a high inference speed. Thanks to some outlier exclusion and multi-view optimization algorithms (Nakajima and Saito, 2018), 3D objects can also be leveraged.

Besides, 3D object instances can also be predicted directly from RGB-D images or point clouds by neural network architectures. Focusing on the indoor environment, Qi et al (2019) proposed VoteNet to generate high-quality object proposals by learning to vote object centroid directly from point clouds and aggregate votes through their features and local geometry. On this basis, ImVoteNet (Qi et al, 2020) designed a joint 2D-3D voting scheme to leverage both geometric and semantic cues in 2D images, achieving state-of-art 3D object detection performance

on indoor datasets. These deep learning methods can achieve significant object detection accuracy. However, they also have two disadvantages. One is the requirement of large amounts of training data for robust and accurate detection, and the other is the high computational demand, where these methods are time-consuming and cannot be applied to mobile robotic platforms.

Object association is another important component in multi-view optimization, which can be explained to match the same objects across multiple frames. Bowman et al (2017) formulated a probabilistic data association for semantic SLAM, which estimated the distribution and maximize the expected measurement log-likelihood over the previously computed distribution. This method tightly coupled inertial, geometric, and semantic observation into a single optimization framework. Specifically, for object association, Li et al (2020a) proposed to use a set of geometric and semantic information for the association, such as object translation, dimension, inliers, etc. Lin et al (2021) represented the object map as a semantic graph with topological information and proposed an efficient graph matching for robust data association. In a map with several objects of the same class, data association becomes crucial.

In this paper, we aim to efficiently detect 3D object instances from RGB-D sequences and incrementally update the object information. So, we utilize a fast object detector to recognize objects with 2D bounding boxes and extract the corresponding 3D points to segment 3D objects from every frame. Moreover, an object association strategy using geometric and semantic descriptors is presented to match the frame-detected objects to the existing objects in the global map, which can track and update object properties across multiple frames.

2.2 Object-Level Semantic Mapping

In 3D reconstruction, mapping refers to the process of converting sensor information into a representation of the environment. Here we discuss four maps with different functions. An occupancy map (Grisetti et al, 2007) is well-studied and widely used for planar navigation in an indoor environment, it decomposes the space into a fixed-size grid and each cell of the grid indicates whether this area is occupied or not. A feature-based map (Zhou et al, 2022) represents the environment as different landmarks, including feature points, geometry planes, semantic objects, etc. This map provides an easy way to update, thus being useful for scene understanding and localization. A point cloud map (McCormac et al, 2017) is a dense map that reconstructs the world with hundreds of points, which shows the detail of the environment and is intuitive for visualization. Finally, a voxel-based map models the environment with cubic volumes of equal size and discretizes the mapped area with explicitly free space. This map is mainly used for 3D navigation. In the following, we focus on the voxel-based map and review some representative methods that introduce object information to create an object-level map.

A voxel-based map is a 3D map representation that uses voxels, or cubic volumes, to store environmental information. Each voxel can be either occupied or free, corresponding to its occupancy probability. Octomap (Hornung et al, 2013) is a typical 3D voxel-based map that uses a tree-based representation to efficiently process and update large-scale 3D information. It performs a probabilistic occupancy estimation, allowing for uncertainty and sensor noise. On the basis, Liu et al (2019) extended the octree to include object information and create an object-aware semantic map for the indoor scene. The main drawback of octomap is that the maximum size of the map must be known a priori and cannot be dynamically changed.

A Truncated Signed Distance Field (TSDF) map is another type of voxel map, it represents the environment using only signed distance information rather than occupancy probability. Each voxel saves the signed distance to the closest surface points, indicating the points are whether outside or inside the surface. This information is valuable for surface reconstruction and collision detection. Voxblox (Oleynikova et al, 2017) is one of the CPU-based mapping systems that can densely reconstruct volumetric TSDF maps in unexplored environments, providing valuable free and occupancy space information to guarantee navigation safety. Adding semantics to these maps, Rosinol et al (2020) presented Kimera-Semantics to create a semantic map, where they

annotated semantics with 2D pixel-wise segmentation and built a global 3D mesh using voxel-based (TSDF) approach.

Previous work has addressed semantic mapping of the whole environment, while other object-oriented approaches focus on identifying and reconstructing individual objects. Pham et al (2019) designed a higher-order conditional random field (CRF) to infer optimal segmentation labels and employed an efficient super-voxel clustering method for object segmentation in 3D indoor scenes. Grinvald et al (2019) presented a combined geometric-semantic scheme to incrementally build a volumetric object-centric map, which retrieves both recognized scene objects as well as previously unobserved elements. Inspired by this work, Li et al (2020b) represented object-instance as a Gaussian mixture model considering the projection relationship between voxel and pixel. Mascaro et al (2022) emphasized the data association module and employed a label diffusion scheme to regularize the final instance segmentation. The above methods can achieve high performance in 3D object segmentation accuracy while building a semantic map. One of the common issues is that these approaches rely on on 2D semantic masks, which require high computational costs and can not run on a CPU-only robotic platform.

Instead of performing pixel-wise segmentation masks with a heavy neural network, we seek a geometric object segmentation method to refine 3D objects from RGB-D sequences. In addition, we employ a CPU-based volumetric mapping system and enrich the reconstructed map with high-level object information, which can benefit not only scene understanding but also robotic navigation.

3 Method

The proposed object-level semantic mapping pipeline is illustrated in Fig. 1, which takes RGB-D sequences as input and incrementally builds a volumetric map enriched with object instances. To achieve these functions, the RGB-D sequences are initially processed by a camera pose tracking framework (Section 3.1). Then, an object instance segmentation method is employed to detect and extract semantic 3D objects from a single frame (Section 3.2). After that, these frame-based detected objects are matched to globally mapped objects via an object association strategy, which uses geometric and semantic descriptors to track and update objects across multiple frames (Section 3.3). Finally, the associated objects are incorporated into a TSDF volumetric mapping framework to generate an object-level dense map (Section 3.4).

3.1 Camera Pose Tracking

The proposed mapping system requires camera poses when robots move in an unknown scene. The movements can be estimated by wheel encoder, feature tracking (Mur-Artal and Tardós, 2017), and laser scan matching (Grisetti et al, 2007), depending on which sensor is available. Many SLAM (Simultaneous Localization and Mapping) systems emphasize how to accurately estimate camera poses with RGB-D sequences, such as (Zhou et al, 2022). Since this is not the main part of this paper, we assume the localization tasks are solved and focus on the mapping process.

3.2 Object Instance Segmentation

In the paper, objects are represented as 3D cuboids with geometric and semantic features, including class names, associated 3D points, cuboid parameters, etc. The mathematic description for 3D cuboids consists of 9 DoF (Degrees of Freedom) parameters: 3 DoF translation $\mathbf{T} = (t_x, t_y, t_z)^\top$, 3 DoF rotation $\mathbf{R} = (\theta_x, \theta_y, \theta_z)^\top$, and 3 DoF dimension $\mathbf{D} = (d_x, d_y, d_z)^\top$. The cuboid coordinate frame is located at the cuboid center, aligned with the main axes.

The object segmentation process can be divided into three steps: Firstly, given an RGB-D frame with 2D bounding boxes, the 3D points inside each of the bounding boxes can be obtained. Since they contain outliers, the next step is to adopt an outlier exclusion algorithm to cluster object points. Finally, we compute object parameters and draw a cuboid to cover all points. Fig. 2 describes the whole process.

1) Point Cloud Extraction. For each RGB-D frame, we first choose YOLO (Redmon and Farhadi, 2017) network to detect the object classes together with 2D bounding boxes, which is comparatively accurate and requires less computation cost. Next, we aim to detect the ground. To do so, we convert each RGB-D frame into a 3D

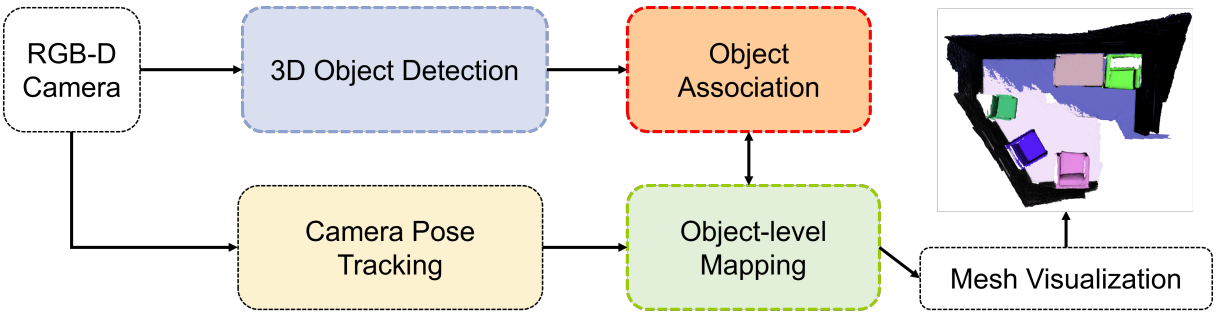


Fig. 1 Overview of the proposed semantic mapping system. We take RGB-D sequences as input to build a volumetric object-oriented map. The input is processed by camera pose tracking and object instance segmentation processes to get camera localization and object information, then, an object association strategy is adopted to update objects across multiple frames, finally, all objects together with other information are integrated into a 3D volumetric map.

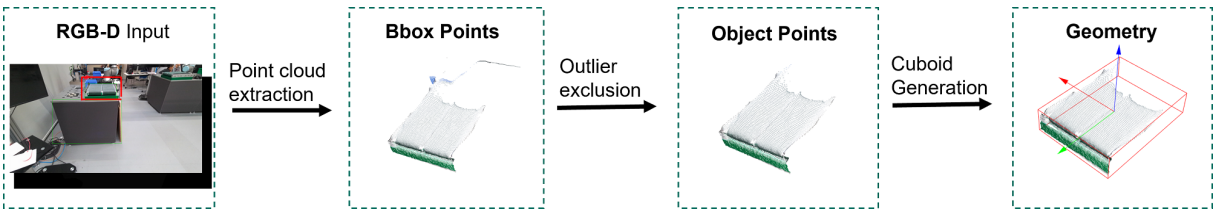
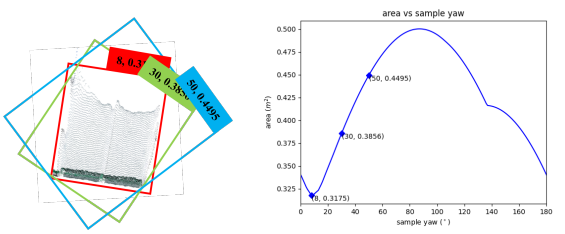


Fig. 2 Object segmentation process. Taking RGB-D images as input, we detect object bounding boxes and extract the points inside the bounding boxes, then, we utilized a robust outlier exclusion algorithm to remove points that do not belong to the objects, finally, the object is represented as a geometric cuboid.



(a) Heading angle sampling. (b) Areas under different angles.

Fig. 3 Object geometry computation. (a) We perform a dense sample on the heading angle from 0° to 180°. (b) The area of oriented bounding boxes under different heading angles.

point cloud and apply a multiple plane estimation method (Trevor et al, 2013) to detect all planes in the frame. the ground's normal should be parallel to the world coordinate and the relative distance should be zero. After obtaining the object bounding box in the image, we can retrieve the corresponding point cloud within the box, filtering out any ground points.

2) Outlier Exclusion. Due to the presence of sensor noise and detection errors, there are many discrete outliers in the preliminary object model. In this case, a statistical outlier removal step (Rusu and Cousins, 2011) is performed to increase the accuracy of the point cloud. Additionally, a Euclidean cluster extraction (Rusu and Cousins, 2011) is implemented to separate points into several groups and cluster object points. To ensure reliability, only one cluster is selected, which should encompass at least 70% of all points.

3) Cuboid Generation. After obtaining object points, the cuboid model with 9 DoF parameters can be computed. To achieve this goal, we first transform all point clouds into the world coordinate. Building on the assumption that real-world objects are parallel to the ground, object roll θ_x and pitch θ_y are 0, The translation t_z and the height d_z should be calculated as the mean and difference of the maximum and minimum value of the z axis among all points. Finally, the remaining $\{t_x, t_y, d_x, d_y, \theta_z\}$ can be determined by projecting all object points on the 2D X-Y plane. As illustrated in Fig. 3, we perform a discrete sampling of rectangle rotation from 0° to 180° and draw oriented rectangles to cover all projected points. For all sampled oriented rectangles, we observe that the smallest rectangle matches the object points best, whose parameters (the position, dimension,

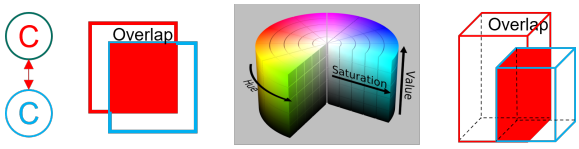


Fig. 4 Proposed data association strategy for semantic mapping. From left to right are: class name; 2D IoU; HSV; and 3D IoU.

and yaw angle $\{t_x, t_y, d_x, d_y, \theta_z\}$) can be adopted to represent the 3D objects.

3.3 Object Association

Since the frame-wise segmentation processes the current image independently of each frame, an object association module is proposed to determine correspondences among multiple frames and incrementally update the object in the global map.

For each frame detected object A and global mapped object B, many geometric and semantic descriptors are extracted to achieve an accurate matching. Firstly, we check their class difference $C(A, B)$ and frame closeness $\eta(A, B)$, which are defined as:

$$C(A, B) = |Class(A) - Class(B)| \quad (1)$$

$$\eta(A, B) = |Frame\_ID(A) - Frame\_ID(B)| \quad (2)$$

where the $Class$ and $Frame\_ID$ are represented as numbers and $|.|$ defines the difference.

In image space, we are able to obtain 2D bounding box and color information. So, we first calculate the 2D IoU of two objects. Then, we also compare their color similarities using HSV (Hue Saturation Value) vectors. To achieve this, we crop the image within the 2D bounding box and calculate the color vectors $HSV(A)$ and $HSV(B)$. These vectors can be utilized to generate a correlation coefficient, which indicates the degree of appearance similarity between the two objects.

$$IoU_{2D} = \frac{BBOX_{2D}(A) \cap BBOX_{2D}(B)}{BBOX_{2D}(A) \cup BBOX_{2D}(B)} \quad (3)$$

$$Correlation = \frac{HSV(A) \times HSV(B)}{\sqrt{HSV(A)^2 + HSV(B)^2}} \quad (4)$$

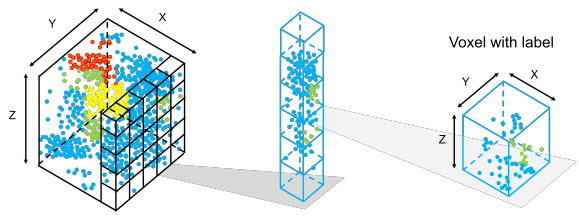


Fig. 5 Proposed voxel-based mapping process. We convert the labelled 3D point cloud (blue, red, and yellow) into a voxel-based map, where each voxel is extended to store the object label, and this information will be incrementally updated by counting object labels inside the voxel (blue).

Besides, we also check their spatial relationship by calculating the 3D IoU as follows:

$$IoU_{3D} = \frac{BBOX_{3D}(A) \cap BBOX_{3D}(B)}{BBOX_{3D}(A) \cup BBOX_{3D}(B)} \quad (5)$$

To sum up, two objects are considered as an associated pair if they satisfy the following constraints as $C(A, B) = 0$, $\eta(A, B) < 10$, $IoU_{2D} > 0.5$, $Correlation > 0.3$, $IoU_{3D} > 0.5$. It is worth noting that, one object might be recognized as different classes due to similar appearance, such as "$sofa$" and "bed". In this case, if they are quite close to each other ($IoU_{3D} > 0.8$), we think they are also associated and should be merged.

If there are no existing mapped objects that are associated with frame-detected objects, we will add this object to the global map with the following parameters: $< Frame\_ID$, $Mapped\_ID$, $Class$, $bbox$, $\mathbf{T}$, $\mathbf{R}(\theta)$, $\mathbf{D}$, HSV, $3D\ points, ... >$. In contrast, when one mapped object is matched to a frame-detected object, it should be merged and updated. In this case, the semantic information ($Frame\_ID$, $Class$, $bbox$, and HSV) will be replaced by new information, while the 3D points are accumulated and the 9 DoF parameters of the 3D cuboid will be computed again as described in Section 3.2.

3.4 Object-Level Mapping

The object-level map is expected to not only contain object information to better understand the scenario, but also provide valuable information for navigation. In our case, we have opted for Voxblox (Oleynikova et al, 2017) foundational framework for mapping the environment using TSDF information. This decision stems from the fact that each voxel incorporates signed distance data,

which directly enhances navigation safety. Additionally, Voxblox is a real-time solution designed for CPU usage, making it particularly well-suited for integration with robotic platforms in real-world scenarios. Many voxel mapping systems use raycasting (Curless and Levoy, 1996) to project an RGB-D image onto the voxel grid, which casts a ray from the camera optical center to the center of each observed point, and updates all voxels from the center to truncation distance behind the points. However, Voxblox uses a grouped raycasting approach to accelerate the mapping process. For each point, they group it with all points that map the same voxel, take the mean color and distance across the grouped points, and perform raycasting only once. This leads to a similar reconstruction result while enabling a real-time solution on the CPU platform.

We augment the mapping system with object information. For each RGB-D frame, the associated object points are labelled with object $Class$ and $Mapped\_ID$ while other points are labelled with "0", indicating the so-called "background" information. Following the approach introduced in (Grinvald et al, 2019), each voxel in the TSDF grid is also extended to store the object $Class$ and $Mapped\_ID$, and this information will be incrementally updated by counting object labels inside the voxel, as shown in Fig. 5. When a new segmented object is integrated into the global map, the corresponding voxel v will update its count ψ for different label li as :

$$\psi(v, li) \leftarrow \psi(v, li) + 1. \tag{6}$$

After all objects have been integrated, the voxel label $L(v)$ will be updated by the object label that has the maximum count as:

$$L(v) = \underset{i}{\operatorname{argmax}} \psi(v, li), \tag{7}$$

During the mapping process, our object detection and association modules ensure consistency across different frames to provide stable object labels.

4 Experiment

4.1 Dataset

The performance of the proposed system is evaluated on indoor environments from SceneNN dataset (Hua et al, 2016), which features RGB-D scans of different indoor scenes, including offices, bedrooms, and kitchens. Besides, this dataset also provides the ground truth with object instances and is commonly used in other research work to compare the reconstruction of object-level mapping approaches.

Fig. 6 SceneNN dataset. This dataset provides several RGB-D sequences of different indoor scenes with object instances. The figure comes from the official website: https://hkust-vgd.github.io/scenenn/.

To run the experiments, a ThinkPad laptop (i7-8565U 1.80 GHz CPU, 16 GB RAM, no GPU, Ubuntu 18.04) is used. Due to hardware constraints, we choose YOLOv2 (Redmon and Farhadi, 2017) as our object detector with general pre-trained weights downloaded from the official website. The mapping framework, as well as mesh visualization tools, comes from (Grinvald et al, 2019). All components, including object detection, association, and mapping are implemented in C++. To show the efficiency, we also transfer our system to a GPU platform, the runtime performance is also reported.

Furthermore, we demonstrate the applicability of our method on a real robotic platform with RGB-D sensors, where the mobile robot is driven to map an unknown intra-logistics environment.

4.2 Evaluation on the SceneNN Dataset

1) Qualitative Results. We first visualize one example of our object-level mapping results on sequence 011 of the SceneNN dataset. As shown in Fig. 7, the left side shows the point model of a table and a chair from a single frame, while the right side shows the whole point cloud map with objects. The corresponding voxel models of objects and TSDF map are also displayed.

More mapping results can be found in Fig. 8, where the first row is the input RGB image with the 2D bounding box, and the second row shows the object points with cuboid representation, Although the objects are partially observed, they will be associated and incrementally updated with multi-view optimization. Finally, the point cloud maps and volumetric maps augmented with individual objects are shown in the third and last rows.

2) Quantitative Results. During the mapping process, the TSDF map has a voxel size of $2cm$. It is difficult to quantify the difference between the reconstructed map and the ground truth. Instead, a possible way to evaluate the accuracy of the map is by comparing the object positions within it. Following the evaluation procedure introduced by (Grinvald et al, 2019), our object mapping algorithm is assessed on 10 indoor sequences from SceneNN dataset (Hua et al, 2016), and we consider 9 object categories (i.e., *bed*, *chair*, $sofa$, *table*, *books*, $refrigerator$, *television*, *toilet*, and *bag*) for comparison with other research work. For each sequence, the per-class Average Precision (AP) score is computed using the 3D Intersection over Union (IoU) threshold of 0.5 over segmented objects and ground truth. The mean Average Precision (mAP) of each sequence is directly calculated by averaging the per-class AP scores.

Table 1 and Table 2 show the 3D IoU and mAP results of all sequences. Besides, Table 3 illustrates the comparison of mAP with other object-level mapping systems (Oleynikova et al, 2017; Grinvald et al, 2019; Mascaro et al, 2022), where the data is cited from the original paper. While other methods use pixel-wise masks for object segmentation, our object-wise detection and cluster method can also achieve good segmentation accuracy. These tables demonstrate that the proposed approach outperforms other baselines on 6 of the 10 evaluated sequences, and triggers a significant increase in the achieved 3D segmentation accuracy. However, it is worth pointing out that the reported mAP values are computed over a smaller set of classes.

Focusing on the specific categories, we notice that the segmentation performance varies in different objects. $sofas$ and *chairs* achieve a better segmentation performance due to their size and multiple observations. Our object association strategy, which utilizes both geometric and semantic descriptors, provides robust object tracking and benefits for object refinement. For small objects like *books*, our method cannot distinguish them from the cluttered background because there are not enough points remaining after the exclusion of outliers. Besides, we observe that object segmentation with bounding boxes might lead to over-segmentation or overlap problems, especially for *chairs* that are under the *table*, as shown in the last column of Fig. 8. In this case, we first segment the *table* as it has a convex shape with surface planes and is easy to estimate from the environment, then, inside the *chair* bounding box, we can remove the points that belong to *table* and other outliers to prevent wrong detection.

3) Runtime Performance. Another advantage of our proposed system is the runtime performance, which is analysed and compared with other state-of-the-art systems. Table 4 shows the evaluation of the execution times of the individual model of the proposed pipeline averaged over 10 evaluated sequences in SceneNN (Hua et al, 2016) dataset. The object detection module running by YOLOv2(Redmon and Farhadi, 2017) takes 725ms to detect 2D bounding boxes in each frame, which is the most time-consuming process, it can be accelerated when transferred to a GPU platform (NVIDIA RTX 3050TI GPU). When compared to other systems, as shown in Table 5, our system achieves a speed of 1 Hz in CPU and 10 Hz in GPU. Li et al (2020b) also uses YOLOv2 as the 2D object detector and reaches a comparative speed, while other methods employ Mask R-CNN (He et al, 2017) to generate pixel-wise segmentation and show a slower performance. We substantially reduced the computational time in two fields, one is to use a fast and stable object detector, followed by a robust outlier exclusion method to segment 3D objects from RGB-D images. The other is to exploit a voxel-based mapping and updating scheme that can run in real-time on the CPU. Most importantly, our system can be extended to a real robotic platform without GPU requirements, which provides a convenient and cheap solution for real-world applications.

Table 1 Evaluation of 3D object detection accuracy (3D IoU) of 10 sequences from the SceneNN dataset.

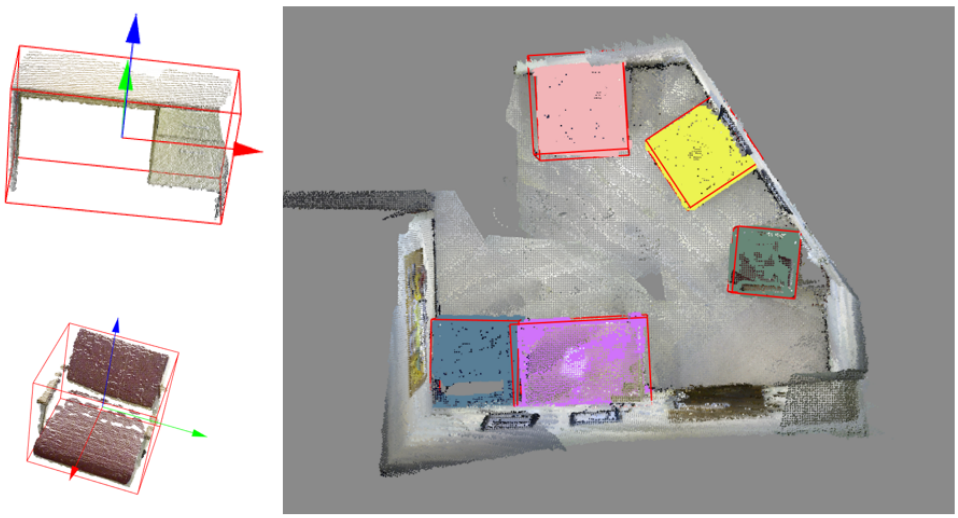
Sequence ID	Bed	Chair	Sofa	Table	Books	Refrigerator	Television	Toilet	Bag	Avg.(Ours)
011	-	70.2	70.2	86.3	-	-	-	-	-	78
016	51.3	-	71.9	0	-	-	-	-	-	41.4
030	-	57.6	80.4	85.7	0	-	-	-	-	57.4
061	-	74.5	62.7	95.1	-	-	-	-	-	77.4
078	-	45.9	-	0	0	67.8	-	-	-	13.9
086	-	58.2	-	0	0	-	-	-	53.8	56.0
096	63.1	60.9	-	0	0	-	32.3	-	0	31.3
206	-	56.5	23.3	65.5	-	-	-	-	29.6	43.7
223	-	63.7	-	69.2	-	-	-	-	-	66.5
255	-	-	-	-	-	55.8	-	-	-	55.8

Table 2 Evaluation of 3D object detection accuracy (mAP) on 10 sequences from the SceneNN dataset. The result from (Grinvald et al, 2019) is listed in the table for comparison.

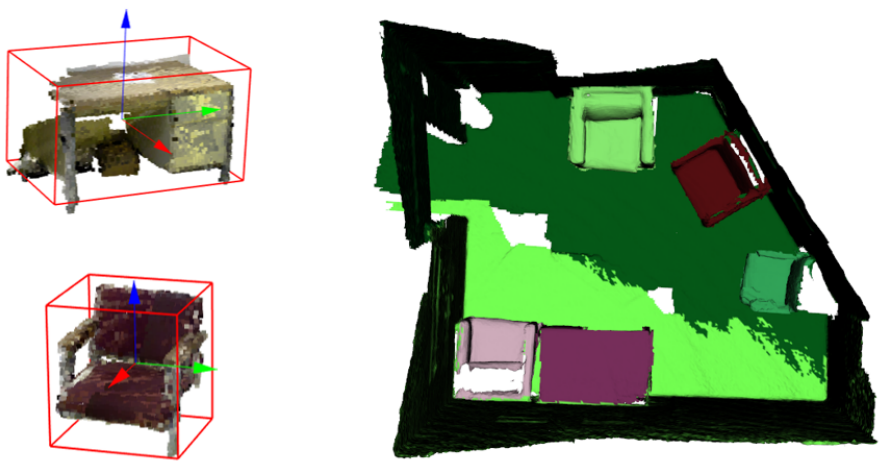
Sequence ID	Bed	Chair	Sofa	Table	Books	Refrigerator	Television	Toilet	Bag	Avg.(Ours)	Avg.(Voxblox++)
011	-	**100**	**100**	**100**	-	-	-	-	-	**100**	75.0
016	**100**	-	**100**	0	-	-	-	-	-	**66.7**	33.3
030	-	**72**	**100**	**66.7**	0	-	-	-	-	**59.7**	56.1
061	-	62.5	100	33.3	-	-	-	-	-	65.3	**66.7**
078	-	50	-	0	0	100	-	-	-	37.5	**45.2**
086	-	**75**	-	0	0	-	-	-	50	**31.3**	20.0
096	**100**	**100**	-	0	0	-	0	-	0	**33.3**	29.2
206	-	41	0	40	-	-	-	-	0	20.3	**79.6**
223	-	**100**	-	**50**	-	-	-	-	-	**75**	43.8
255	-	-	-	-	-	**100**	-	-	-	**100**	75.0

Table 3 Comparison of the 3D object segmentation accuracy (mAP) among different system. The results of (Pham et al, 2019), (Grinvald et al, 2019), and (Li et al, 2020b) come from the original research work.

Method	011	016	030	061	078	086	096	206	223	225	Average
Pham et al (2019)	52.1	34.2	56.8	59.1	34.9	35.0	16.5	41.7	40.9	48.6	43.0
Grinvald et al (2019)	75.0	33.3	56.1	**66.7**	45.2	20.0	29.2	**79.6**	43.6	75.0	54.4
Li et al (2020b)	78.6	25.0	58.6	46.6	**69.8**	**47.2**	26.6	78.0	45.8	75.0	55.1
Ours	**100**	**66.7**	**59.7**	65.3	37.5	31.3	**33.3**	20.3	**75**	**100**	**58.9**



(a) Point cloud map with objects.



(b) voxel-based map with objects.

Fig. 7 Mapping results on 011 sequence of the SceneNN dataset. The left shows the objects model, while the whole map is on the right. It is worth noting that we build voxel-based maps, but visualize them as mesh maps.

Table 4 Average execution time of each processing module in our proposed system. Note that we create a separate thread for the object mapping module, which does not affect the frame rate of the whole system.

Module	Runtime-CPU (mSec)	Runtime-GPU(mSec)
Object Detection	725	32
Object Segmentation	17.75	15
Object Association	5.22	5
Object Mapping	299	50
Total	1046.97	102

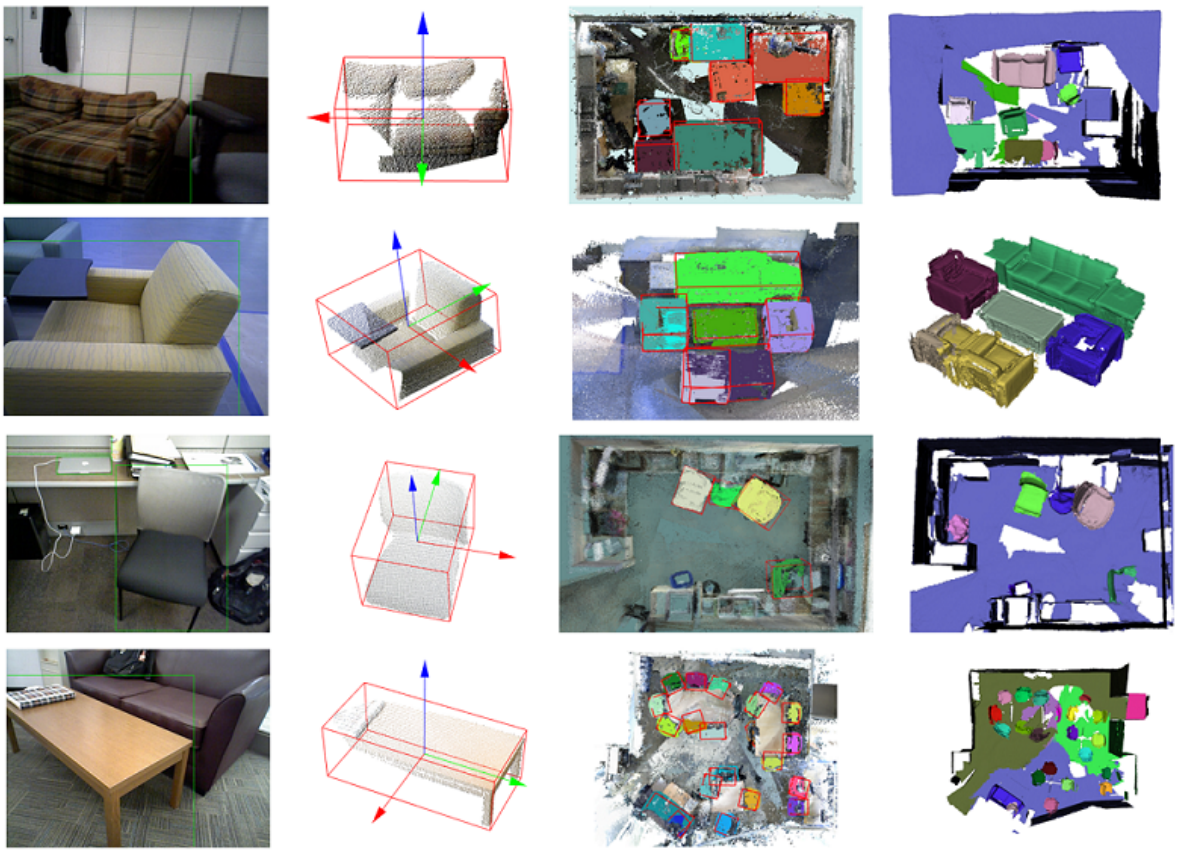


Fig. 8 Examples of detection and mapping results on SceneNN dataset. The first row is the input RGB image with the 2D bounding box, the second row shows the object points with cuboid representation, and the last two rows present the point cloud maps and volumetric maps augmented with individual objects

Table 5 Comparison of average execution time among different object mapping systems.

Method	Platform	Frequency	Map	FPS
Pham et al (2019)	Nvidia Quadro M2200	every frame	objected-oriented	1 Hz
Grinvald et al (2019)	NVIDIA Titan X	every frame	objected-oriented	1 Hz
Li et al (2020b)	Nvidia GeForce GTX1080 Ti	every frame	objected-oriented	10.8 Hz
Ours-GPU	Nvidia GeForce GTX3050 Ti	every frame	objected-oriented	10 Hz
Ours-CPU	None	every frame	objected-oriented	1 Hz

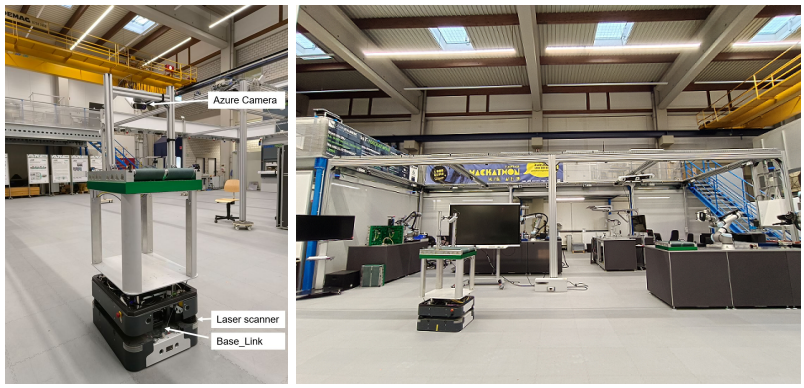
4.3 Evaluation on a robotic platform

We also evaluate our system within the AgiProbot project [1] (Agile PROduction system using mobile, learning roBOTs with multi-sensors for uncertain product specifications), which is an intra-logistics system for an agile remanufacturing product system and built at Karlsruhe Institute of Technology (KIT) (Klein et al, 2021). In this project, mobile robots are tasked to transport and transfer items among different workstations. Specifically, this requires the mobile robots to not only recognize and localize *conveyors* that are installed on the workstation, but also perform global navigation to the workstation. In this case, our object-level semantic mapping system provides a suitable solution to build a dense map with semantic objects.

The vehicle is shown in Fig. 9, where a 2D SICK laser scanner [2] and a Microsoft Azure

[1] http://agiprobot.de/

[2] https://www.alliedelec.com/product/sick/s30b-3011gb/70872466/



(a) The robotic platform. (b) The AgiProbot environment.

Fig. 9 The robotic platform in AgiProbot project.

Kinect camera [3] are mounted. The SICK laser scanner is installed horizontally at a height of $0.3m$ and in the northwest corner of the vehicle. It can measure maximal distance as $30m$ with 270° scanning angle and publish the scan data up to 30 Hz. The Microsoft Azure Kinect camera is mounted at a height of 1.4 m on the top of the vehicle center, facing downwards at 45°. It captures RGB and depth images in 720P resolution (1280x720) with $90° \times 60°$ field of view (FOV) and publishes images up to 15 Hz. Calibration[4] between the laser scanner and the camera is solved by (Zhang and Pless, 2004).

To generate an object-level map, we use the joystick to drive the mobile robot around the whole environment ($6 * 12m$) and record laser scan data and RGB-D images. The laser scan data are used for the camera (robot) pose tracking, while the RGB-D sequences are fed to our object-level mapping framework to build a semantic map. Since no GPU is available on board, the frame rate is set as 1 Hz, any frames that exceed the processing abilities of the system are discarded and not used to reconstruct the object-level map of the scene.

Camera Pose Tracking. The dependency of robot localization is solved by ICP registration (Censi, 2008) on laser scan data, which achieves a good robot pose estimation. On this basis, a point cloud map is generated in Fig. 10. Compared to the real environment, the point cloud map has a detailed description, which identifies that the camera pose tracking module functions well. We can also manually annotate the object from the point cloud map as ground truth to evaluate object detection results.

Table 6 Evaluation of object detection accuracy on the logistic environment

Object ID	IoU_{3D}	$E_{trans}(m)$	$E_{rot}(°)$
1	0.7446	0.058	3.4
2	0.7140	0.060	0.9
3	0.8061	0.029	1.6
4	0.9056	0.035	1.0
Average	0.7925	0.045	1.7

Object Detection. For object detection, 100 images containing *conveyors* are captured and manually annotated to train the YOLO neural network (Redmon and Farhadi, 2017), after that, the pre-trained weights are used to detect *conveyor* with bounding boxes in every frame. The results of object accuracy are shown in Table 6, where the object IoU is an average of 0.79, and the translation error and rotation error are $0.045m$ and 1.7° respectively. Two important factors may influence the results. One is sensor noise. We observe that when the robot is rotating, the 2D object detections and point cloud measurements are inaccurate and will deteriorate the segmentation result. We propose to remove these bad detections. The other is the object updating strategy. We simply accumulate object points and compute object geometry, it is hard to remove outliers. A better idea is to calculate and update the probabilities of all corresponding 3D points.

[3] https://learn.microsoft.com/en-us/azure/Kinect-dk/hardware-specification

[4] https://github.com/MegviiRobot/CamLaserCalibraTool

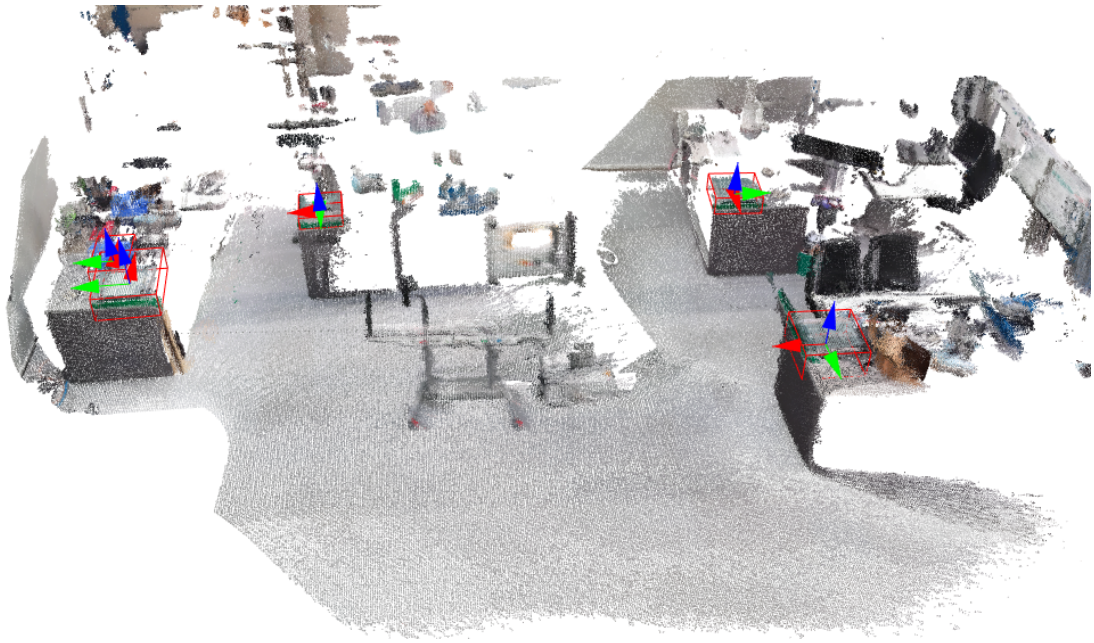


Fig. 10 The point cloud map in the logistics environment. We track robot poses with laser scan data and generate a point cloud map with the RGB-D camera, where the ground truth of objects is labelled.

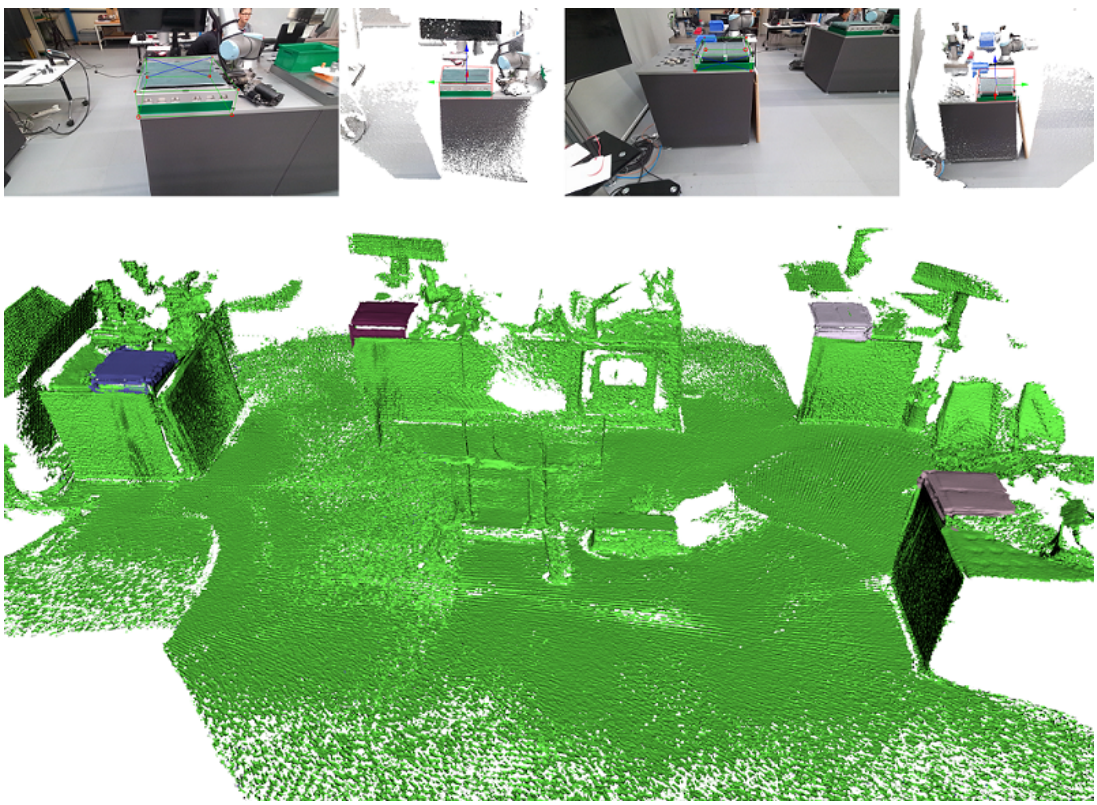


Fig. 11 Mapping results on the logistics environment. The reconstructed map is at a voxel size of $2cm$, and the conveyor objects are assigned different colors while the background points are green.

Object Mapping. The results for the reconstructed map with a voxel resolution of $2cm$ are shown in Figure 11, where the *conveyors* are marked with different color while the background is green. As we can observe, our proposed method can integrate incoming RGB-D images into the map volume, providing a comprehensive representation of the surface geometry for individual objects. This volumetric map contains object information to better understand the scene, and the additional free space information is relevant to safe planning for autonomous navigation. Although the system operates at only 1 Hz with CPU, it validates the online framework and shows its benefit for real-world applications. We also provide a demo video [5] to illustrate the whole process of incrementally reconstructing the semantic object-level map of the scene.

5 Conclusion

In this paper, we present an efficient object-level semantic mapping system, which takes RGB-D sequences as input to build a volumetric object-oriented map. Firstly, the RGB-D sequences are processed with an object detection module to segment object points from a single frame. Then, a new data association strategy with geometric and semantic descriptors is designed to track and update object information. Finally, the partial segmentation information is incrementally fused into a global map and results in an object-level volumetric map, which can be further used for scene understanding and autonomous navigation.

Experiments on publicly available indoor datasets and logistics environments show that our system has a comparative performance on 3D object segmentation while avoiding high computational costs. By employing an efficient object detector and an efficient voxel mapping framework, our system can be extended to a CPU-only robotic platform for real-world application.

A future research direction involves investigating the simultaneous localization and mapping system with object information with loop closure to improve map accuracy. Besides, it is valuable to explore other object detection algorithms such as feature matching to further reduce the computation time.

[5] https://github.com/benchun123/object-level-mapping

Acknowledgments. The authors would like to thank Institute for Material Handling and Logistics (IFL, KIT) for the support for the hardware and test environment.

Declarations

No potential conflict of interest was reported by the author(s)

Funding

This work was supported by China Scholarship Council (CSC) Foundation under Grant 201906020168.

Notes on contributor(s)

Benchun Zhou is a Ph.D. student at the Institute for Material Handling and Logistics, Karlsruhe Institute of Technology, Germany. He received his B.S. degree from School of Automation, Chongqing University, China in 2016 and his M.E. degree from the School of Automation Science and Electrical Engineering, Beihang University, China in 2019. His research interests include visual SLAM, computer vision, and field robotics.

Furmans Kai is the Director of the Institute for Material Handling and Logistics. He is also a professor of Mechanical Engineering at Karlsruhe Institute of Technology.

Yunxiang Fu received his B.S. degree from the School of Mechanical Engineering, China University of Mining and Technology, China in 2019 and his M.S. degree from the School of Mechatronics and Information Technology, Karlsruhe Institute of Technology, Germany in 2023. His research interests include SLAM, computer vision, and robotics.

Hao Pang is an employee at the Institute for Material Handling and Logistics, Karlsruhe Institute of Technology, Germany. He received his B.S. degree from Wuhan University of Technology (China) and his M.S. degree from the Karlsruhe Institute of Technology (Germany). His research interests include semantic mapping and navigation.

References

Bowman SL, Atanasov N, Daniilidis K, et al (2017) Probabilistic data association for semantic slam. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 1722–1729

Censi A (2008) An icp variant using a point-to-line metric. In: 2008 IEEE International Conference on Robotics and Automation, Ieee, pp 19–25

Curless B, Levoy M (1996) A volumetric method for building complex models from range images. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, pp 303–312

Grinvald M, Furrer F, Novkovic T, et al (2019) Volumetric instance-aware semantic mapping and 3d object discovery. IEEE Robotics and Automation Letters 4(3):3037–3044

Grisetti G, Stachniss C, Burgard W (2007) Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE Transactions on Robotics 23(1):34–46

He K, Gkioxari G, Dollár P, et al (2017) Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2961–2969

Hornung A, Wurm KM, Bennewitz M, et al (2013) Octomap: An efficient probabilistic 3d mapping framework based on octrees. Autonomous Robots 34:189–206

Hua BS, Pham QH, Nguyen DT, et al (2016) Scenenn: A scene meshes dataset with annotations. In: 2016 Fourth International Conference on 3D Vision (3DV), Ieee, pp 92–101

Klein JF, Wurster M, Stricker N, et al (2021) Towards ontology-based autonomous intralogistics for agile remanufacturing production systems. In: 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, pp 01–07

Li J, Koreitem K, Meger D, et al (2020a) View-invariant loop closure with oriented semantic landmarks. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 7943–7949

Li W, Gu J, Chen B, et al (2020b) Incremental instance-oriented 3d semantic mapping via rgb-d cameras for unknown indoor scene. Discrete Dynamics in Nature and Society 2020

Lin S, Wang J, Xu M, et al (2021) Topology aware object-level semantic mapping towards more robust loop closure. IEEE Robotics and Automation Letters 6(4):7041–7048

Liu K, Fan Z, Liu M, et al (2019) Object-aware semantic mapping of indoor scenes using octomap. In: 2019 Chinese Control Conference (CCC), IEEE, pp 8671–8676

Mascaro R, Teixeira L, Chli M (2022) Volumetric instance-level semantic mapping via multi-view 2d-to-3d label diffusion. IEEE Robotics and Automation Letters 7(2):3531–3538

McCormac J, Handa A, Davison A, et al (2017) Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 4628–4635

Mur-Artal R, Tardós JD (2017) Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics 33(5):1255–1262

Nakajima Y, Saito H (2018) Efficient object-oriented semantic mapping with object detector. IEEE Access 7:3206–3213

Oleynikova H, Taylor Z, Fehr M, et al (2017) Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 1366–1373

Pham QH, Hua BS, Nguyen T, et al (2019) Real-time progressive 3d semantic segmentation for indoor scenes. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, pp 1089–1098

Qi CR, Litany O, He K, et al (2019) Deep hough voting for 3d object detection in point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 9277–9286

Qi CR, Chen X, Litany O, et al (2020) Imvotenet: Boosting 3d object detection in point clouds with image votes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 4404–4413

Redmon J, Farhadi A (2017) Yolo9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 7263–7271

Rosinol A, Abate M, Chang Y, et al (2020) Kimera: an open-source library for real-time metric-semantic localization and mapping. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 1689–1696

Rusu RB, Cousins S (2011) 3d is here: Point cloud library (pcl). In: 2011 IEEE International Conference on Robotics and Automation, IEEE, pp 1–4

Trevor AJ, Gedikli S, Rusu RB, et al (2013) Efficient organized point cloud segmentation with connected components. Semantic Perception Mapping and Exploration (SPME) pp pp–1

Zhang Q, Pless R (2004) Extrinsic calibration of a camera and laser range finder (improves camera calibration). In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), IEEE, pp 2301–2306

Zhou B, Gilles M, Meng Y (2022) Structure slam with points, planes and objects. Advanced Robotics 36(20):1060–1075