sensors

MDPI

Article

Online Scene Semantic Understanding Based on Sparsely Correlated Network for AR

Qianqian Wang, Junhao Song, Chenxi Du and Chen Wang *

The School of Computer and Artificial Intelligence, Beijing Technology and Business University,
Beijing 102488, China; chuy0603@163.com (Q.W.); sjhao12@163.com (J.S.); dcx270806@163.com (C.D.)
* Correspondence: wangc@btbu.edu.cn

Abstract: Real-world understanding serves as a medium that bridges the information world and the physical world, enabling the realization of virtual–real mapping and interaction. However, scene understanding based solely on 2D images faces problems such as a lack of geometric information and limited robustness against occlusion. The depth sensor brings new opportunities, but there are still challenges in fusing depth with geometric and semantic priors. To address these concerns, our method considers the repeatability of video stream data and the sparsity of newly generated data. We introduce a sparsely correlated network architecture (SCN) designed explicitly for online RGBD instance segmentation. Additionally, we leverage the power of object-level RGB-D SLAM systems, thereby transcending the limitations of conventional approaches that solely emphasize geometry or semantics. We establish correlation over time and leverage this correlation to develop rules and generate sparse data. We thoroughly evaluate the system's performance on the NYU Depth V2 and ScanNet V2 datasets, demonstrating that incorporating frame-to-frame correlation leads to significantly improved accuracy and consistency in instance segmentation compared to existing state-of-the-art alternatives. Moreover, using sparse data reduces data complexity while ensuring the real-time requirement of 18 fps. Furthermore, by utilizing prior knowledge of object layout understanding, we showcase a promising application of augmented reality, showcasing its potential and practicality.

Keywords: real scene analysis; instance segmentation; RGBD SLAM; augmented reality

check for updates

Citation: Wang, Q.; Song, J.; Du, C.; Wang, C. Online Scene Semantic Understanding Based on Sparsely Correlated Network for AR. *Sensors* **2024**, *24*, 4756. https://doi.org/10.3390/s24144756

Academic Editor: Paolo Gastaldo

Received: 18 June 2024
Revised: 9 July 2024
Accepted: 18 July 2024
Published: 22 July 2024

1. Introduction

The deep integration of technologies such as computer simulation and augmented reality with manufacturing has accelerated the development of intelligence and precision in the consumer electronics manufacturing industry, leading to a surge in digital twin technology. The core of intelligent manufacturing is to achieve interaction and integration between the digital world and the physical world [1,2]. It demands a precise perception of real-world geometric and semantic information, enabling the extension of new functionalities to physical entities through data parsing, virtual simulation, and virtual–real interaction. This integration empowers intelligent manufacturing to operate with enhanced efficiency and effectiveness. Therefore, building a real 3D scene semantic model is the primary task in establishing a high-density and real-time connection between the information and physical worlds. First and foremost, it is essential to achieve the reconstruction of large-scale and dense geometry to simulate the real world. Specifically, a comprehensive scene-understanding capability to discern individual objects becomes crucial for facilitating reality–virtual interaction. The most prevalent approach involves utilizing RGB sensors to perceive the 3D world through image sequences and processing the data stream into a digitized representation. This process entails reconstructing a dense geometric model, identifying objects of interest within a scene, and categorizing these objects into specific categories. Although RGB images are commonly used for analyzing scene content, their ability to provide comprehensive and accurate analyses is limited by the absence of stereo information. To overcome this limitation, there has been significant attention towards indoor scene understanding based on RGBD images [3,4]. The comprehensive analysis must understand the geometric structure and the object layout relationship.

Several simultaneous localization and mapping (SLAM) methods have been proposed to leverage depth sensors, like Microsoft Kinect for camera motion estimation and dense geometric model reconstruction. Methods such as KinectFusion [5], BundleFusion [6], and OcclusionFusion [7] have achieved real-time performance. Significant progress has been made in 2D semantic analysis tasks like object detection [8] and instance segmentation [9], contributing to developing a knowledge model. However, these remarkable achievements have primarily focused on geometric or semantic understanding separately. In other words, the reconstructed geometric models lack semantic information, and the semantic labels are limited to individual RGBD frames.

To reconstruct a 3D scene model with geometric and object-level semantic information, we propose a novel approach combining instance segmentation with SLAM, resulting in an object-level RGBD SLAM system. The current semantic-level SLAM systems primarily generate sparse maps. The main improvements have been focused on three aspects: utilizing semantic information for real-time performance [7], enhancing the accuracy of camera localization [9], and ensuring data stability during the reconstruction process in large-scale scenes [10]. However, the main challenge lies in designing a real-time instance segmentation network that can be seamlessly integrated with the SLAM algorithm. In particular, existing semantic instance segmentation methods often fall short when applied independently to single images without contextual information. Since consecutive frames in a video stream typically exhibit small camera motion amplitudes resulting in highly similar images, it is natural to leverage this similarity to relate previous segmentation results to the current frame. Additionally, the distribution of different objects within a scene tends to be sparse. Inspired by the characteristics of sparse convolution, we investigate an enhanced sparse convolution technique for efficient instance labeling.

Our work introduces a real-time instance-level RGBD SLAM system based on sparse convolution. Our system captures a sequence of RGBD input frames, such as those obtained from a Kinect sensor. It performs the following tasks: computing 6-degrees of freedom (6DoF) rigid poses, labeling object instances, and reconstructing 3D semantic instance models. To incorporate contextual relevance, we propose a deep network architecture called the Sparsely Correlated Network (SCN) for labeling object instances in each RGBD frame. Firstly, we employ a series of 2D convolutions to extract texture and geometric features from RGBD images. Additionally, we leverage the projection correspondence to establish feature associations across time. For data spaces that lack contextual relevance, we record their coordinates following a predefined rulebook, resulting in sparse data representation. Next, we employ spatially sparse convolution to extract features and use a region proposal network to infer the bounding box locations and semantic labels for object instances. Finally, we fuse the consecutive instance segmentations into a consistent instance map.

To sum up, our contributions are the following:

- We introduce a novel neural network architecture called SCN, which utilizes sparse convolution for online instance segmentation. Combining the instance segmentation network with an RGB-D SLAM system achieves scene understanding at the object level.
- We recognize projective associations between contexts and use hash tables to organize sparse data without association. We also design a sparse convolution architecture to process data without context relevance, which satisfies the real-time requirement.
- We devise a sparse rule for sparse convolution and develop a grouping strategy for dense data by leveraging spatial topological structures and temporal correlation properties.

2. Related Work

Dense RGB-D SLAM.

The Visual SLAM has a long-standing history of tracking camera motion and reconstructing sparse model maps. The emergence of consumer-grade depth cameras has advanced research in this area, enabling dense and real-time scene mapping. Recent methods have demonstrated accurate mapping of indoor environments, which in turn has addressed the major requirements in fields such as industrial production and intelligent manufacturing. These advancements have paved the way for further developments and applications in these domains. KinectFusion [5] harnesses the advantages of the truncated signed distance function (TSDF) and regularized voxel grid representation to achieve real-time camera tracking and robust scene mapping. To overcome the limitations of space representation, subsequent studies [6,11] extended these principles to large-scale environments by utilizing data structures such as hash tables. Furthermore, an alternative representation based on surfels [12,13] has been proposed to approximate sparse point clouds, where attributes such as normal and radius encode associations between points and surfaces, thereby reducing data redundancy and optimizing data space utilization.

Instance Segmentation.

In recent years, convolutional neural network architectures have significantly advanced object detection [14] and instance segmentation [15]. These advancements have greatly improved the accuracy and efficiency of these tasks, pushing the boundaries of computer vision research.

Early instance segmentation methods mainly concentrated on single RGB images and can be classified into two categories: two-stage methods [16] and one-stage methods [17]. The two-stage methods generally achieve higher segmentation accuracy, while the one-stage methods pursue real-time performance. However, RGB-based instance segmentation methods often suffer from severe error segmentation in low-contrast scenes, as the texture information struggles to reflect the geometric topology between object instances. With the advent of consumer-grade depth sensors, integrating depth information with color and texture has been demonstrated to improve semantic segmentation accuracy [18,19]. In addition, instance segmentation approaches targeting 3D models can provide more comprehensive global information in practical applications. Voxel-based methods represent voxel data in a format suitable for neural network processing. For instance, VoxNet [20] applies convolutions and pooling operations in three dimensions to capture spatial features within 3D data. Voxelization [21] transforms 3D voxel data into a fixed-size voxel grid, making it compatible with input to neural networks. Inspired by PointNet [22], point clouds have gained significant attention due to their advantage of low model representation complexity [23,24]. RandLA-Net [23] proposed a local feature aggregation module to capture the relative positional relationships between points in a local neighborhood, enhancing the efficiency of point cloud recognition by establishing associations between central points and neighboring points. On the other hand, KPConv [24] introduces a variable point convolution that calculates the kernel transformation matrix using an arbitrary number of kernel points, thereby extracting local compelling features in the spatial domain of the point cloud. These point-based methods leverage the spatial information inherent in the point cloud data to perform instance segmentation tasks.

From the perspective of candidate box generation, instance segmentation methods can be categorized into anchor-based detection and anchor-free grouping methods. Anchor-based detection utilizes a predefined set of anchors to generate a proposal and perform classification and pixel-level segmentation for each candidate box. Faster R-CNN [25] proposes a region proposal network (RPN) to generate candidate boxes, speeding up the region proposal process by sharing convolutional layers. Integrating feature extraction, candidate box selection, classification, and bounding box regression into a single end-to-end network effectively enhances detection accuracy and efficiency. He et al. [26] extend Faster R-CNN with a segmentation branch, forming Mask R-CNN, which generates masks for pixels within each candidate box, enabling simultaneous object detection and instance segmentation, thus enhancing segmentation accuracy and precision. Sparse R-CNN [27] replaces the dense anchors in the RPN network with a fixed set of candidate regions, enabling a transition from dense to sparse representation. In contrast to anchor-based detection, anchor-free methods do not rely on predefined anchors but directly generate bounding boxes and masks from the image. Jiang et al. [28] propose a bottom-up 3D instance segmentation framework based on point cloud data, assigning points in the point cloud to different object instances or semantic categories using adaptive neighborhood clustering and global optimization, effectively handling 3D instance segmentation tasks. Zhong et al. [29] introduce a hierarchical point grouping algorithm called MaskGroup. They also present a novel MaskScoreNet to generate binary point masks for all grouped instances, eliminating noisy points. Additionally, MaskScoreNet predicts confidence scores for the final instances to refine the segmentation results further.

Semantic RGBD SLAM.

From the semantics perspective, previously proposed semantic mapping systems can be categorized into two main approaches: the dense labeling and the object-level, as shown in Table 1. SemanticFusion [30] combines the VGG16 network architecture to achieve pixel-level semantic segmentation. Additionally, it updates the class probability distribution using a Bayesian model, enhancing the accuracy and reliability of the segmentation results. DA-RNN [31] introduces data-associated RNNs to establish spatiotemporal associations, achieving more accurate and consistent results through global feature preservation. However, these approaches primarily address scene-level semantic labeling and do not explicitly address the localization and distinction of individual objects within the scene. SLAM++ [32] is a real-time object-level incremental SLAM system. This method constructs an object model library and continuously detects the actual object in the model library to feed into a pose graph. Tateno et al.'s approach [33] is used for rigid body instance segmentation but requires a prior model. These previous methods focus on object-level analysis but limit the categories of objects to the 3D object database. MaskFusion [34], Fusion++ [35], and MID-Fusion [36] have taken advantage of the advantages of using instance-level semantic segmentation and building object-oriented map representation. MaskFusion [34] goes beyond the traditional system of outputting only geometric maps and integrates multiple tasks such as identifying, partitioning, and assigning semantic classes to different moving objects. However, the quantitative detection accuracy is not outstanding because it mainly concerns the camera trajectory accuracy. PanopticFusion [37] achieves comprehensive 3D geometrical and semantic volumetric mapping, enabling the discrimination of individual objects based on Mask R-CNN [26]. Additionally, it applies a fully connected conditional random field (CRF) model to regularize the map by panoptic labels. This regularization step enhances the consistency and coherence of the resulting map, further improving the accuracy of object discrimination and semantic understanding. However, the time efficiency cannot meet the real-time requirement because it takes 235 ms in the object detection network stage. By integrating a two-stream object detection network with a SLAM system, Li et al. [38] achieved dense construction and specific objects while maintaining a sparse global map. This approach successfully improves memory requirements and computational complexity. However, global sparsity mapping limits its potential applications in digital twins. In contrast to the studies mentioned above, our method mainly considers the time requirement of the system, recognizes the context data association relation of the video stream data, and reconstructs a 3D semantic model with global consistency.

Table 1. Semantic mapping methods comparison.

Method	Online	TSDF Volume	Surfels	Large-Scale	Dense Labeling	Object-Level
SemanticFusion [30]	√		√	√	√	
DA-RNN [31]	√	√			√	
SLAM++ [32]	√					√
Tateno et al. [33]	√	√				√
MaskFusion [34]	√		√			√
Fusion++ [35]	√	√				√
MID-Fusion [36]	√	√			√	√
PanopticFusion [37]	√	√		√	√	√
Li et al. [38]	√		√	√		

3. Method

The system framework of our approach is illustrated in Figure 1. Building upon the RGBD reconstruction method [15], we incorporate an instance segmentation module into our pipeline. Our approach begins by estimating the camera motion and computing the 6DOF pose (Section 3.1). Subsequently, we input the current RGBD frame into SCN networks to obtain pixel-wise object instance labels, fused to generate accurate segmentation results (Section 3.2). To ensure robust tracking, the instance labels are carefully referenced against the volumetric map at that particular moment. Furthermore, we integrate the probability distributions of class labels into the map, incorporating depth measurements for improved accuracy (Section 3.3).

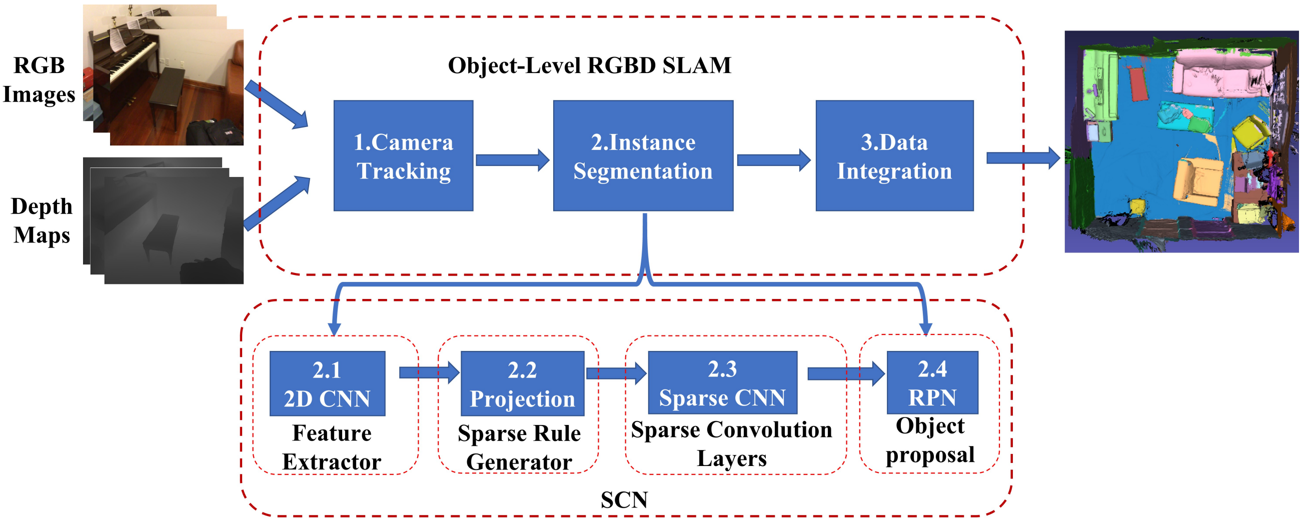


Figure 1. Overview of our object-level RGBD SLAM based on SCN. In three steps, the input RGBD data stream is used to achieve 3D semantic model reconstruction. Step 1: frame-by-frame camera localization is performed. Step 2: camera positions and image data are fed into the SCN network to obtain per-pixel object semantic information. Step 3: the current-time semantics are transformed into global semantics.

3.1. Camera Tracking

To obtain the corresponding vertex map at time t, we transform the raw depth image $\boldsymbol{D}_t$ using the camera intrinsic parameters $\boldsymbol{K}$, resulting in $\boldsymbol{V}_t(\boldsymbol{u}) = \boldsymbol{D}_t\boldsymbol{K}^{-1} \in \mathbb{R}^3$. Additionally, we calculate a normal map $\boldsymbol{N}_t$ using the method proposed in [7]. To estimate the six-degrees of freedom camera pose $\boldsymbol{T} = [\mathrm{R}|\mathrm{t}]$, which includes rotation matrix $R \in \mathbb{SO}_3$ and translation vector t $\in \mathbb{R}^3$, we utilize an iterative closest point algorithm similar to the one described in [39]. This algorithm iteratively refines the pose estimation by aligning the observed 3D points with the reconstructed model.

3.2. Instance Segmentation

After estimating the camera's motion, we need to predict pixel-wise object instance labels. Our network architecture is also shown in Figure 1. It consists of four primary components: a feature extractor (Section 3.2.1) for transforming raw RGB and depth images into a more meaningful and representative format, sparse rule generator (Section 3.2.2) for associating data with geometric consistency (i.e., data that the camera have already observed) and recording newly observed data, sparse convolution layers (Section 3.2.3) for learning the feature description of newly generated data, and object proposal (Section 3.2.4) like object detection for identifying the object instance and predicting semantic class. In the following, we will outline the critical components of our architecture design.

3.2.1. Feature Extractor

For the initial frame at time $t = 0$, the detection accuracy plays a critical role in the subsequent online learning process. To achieve this, we utilize the Mask R-CNN [26] to generate pixel-level segmentation results, including the object category $L_1(u) \in \boldsymbol{\mathcal{L}}$; the object instance ID $Z_1(u) \in \boldsymbol{\mathcal{Z}}$; the surrounding bounding box $B_1 = \{< \triangle x_1^1, \triangle y_1^1, w_1^1, h_1^1 >, ..., < \triangle x_{num_o}^1, \triangle y_{num_o}^1, w_{num_o}^1, h_{num_o}^1 >\}$ representing the coordinates of each bounding box, respectively; and the feature map f_1^{rgbd}, obtained by fusing the RGB and depth modalities resulting in a comprehensive representation that encompasses both visual and geometrical information. In the context of our method, $\boldsymbol{\mathcal{L}} = \{l_1, ..., l_{num_l}\}$ represents a collection of all possible object types in the indoor scene, where l_1 to l_{num_l} denote the individual object types. The num_l represents the total number of object species. The $\boldsymbol{\mathcal{Z}} = \{1, ..., num_o\}$ represents a set of object IDs, ranging from 1 to num_o, where num_o corresponds to the total number of objects. The $u = (x, y)$ denotes the pixel location. Overall, Mask R-CNN enables us to obtain accurate object segmentation results, including object categories, instance IDs, bounding boxes, and fused feature mappings. These results are the foundation for subsequent online learning and further analysis of our method.

We employ the ResNet architecture for the subsequent frames to generate initial 2D feature maps. In Figure 2, the 2D network architecture separately takes the $H \times W$ RGB and depth image as inputs. We extract multi-modality features: f_n^r $(H \times W \times c)$ from the RGB encoder and f_n^d $(H \times W \times c)$ from the depth encoder. These feature maps capture meaningful representations of the input data. To fuse these features, we use a concatenation layer, which combines the feature maps into a single input $\hat{f}_n^{rgbd}$ $(H \times W \times c_{in})$ for the sparse convolution layers. This input retains the spatial dimensions $(H \times W)$ but has a different number of channels (c_{in}) determined by the concatenation of the RGB and depth features.

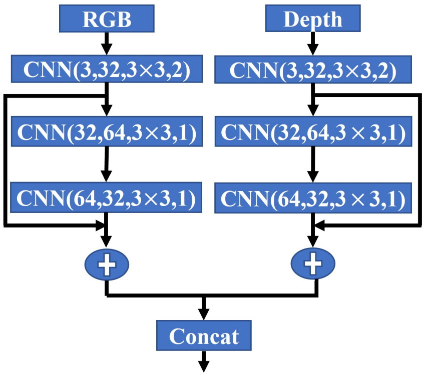


Figure 2. ResNet blocks for 2D feature learning.

3.2.2. Sparse Rule Generator

The composite features need to correlate the feature information in the previous part through the projection and organize sparse data for unrelated data. Minimal camera motion at adjacent moments leads to high similarity in the image, as shown in Figure 3. At times t and $t+1$, the new data collected by the camera only account for about 15 percent of the entire image and are distributed discreetly. Therefore, it is unnecessary to calculate the image feature description repeatedly for data with geometric consistency. We project the current coordinates back to the previous time to establish the temporal connection relationship. For each pixel position $\boldsymbol{u}_t$ and the corresponding vertex map $\boldsymbol{V}_t(\boldsymbol{u})$, we calculate the world coordinates $W_t(\boldsymbol{V}_t(\boldsymbol{u}))$ based on the current camera pose. Next, we aim to find the projection location $\boldsymbol{u}^*_{t-1}$ at the previous time step, i.e., $W_{t-1}(\boldsymbol{V}_{t-1}(\boldsymbol{u}^*)) \approx W_t(\boldsymbol{V}_t(\boldsymbol{u}))$. We can associate the feature between the two consecutive frames by establishing this projection relation.

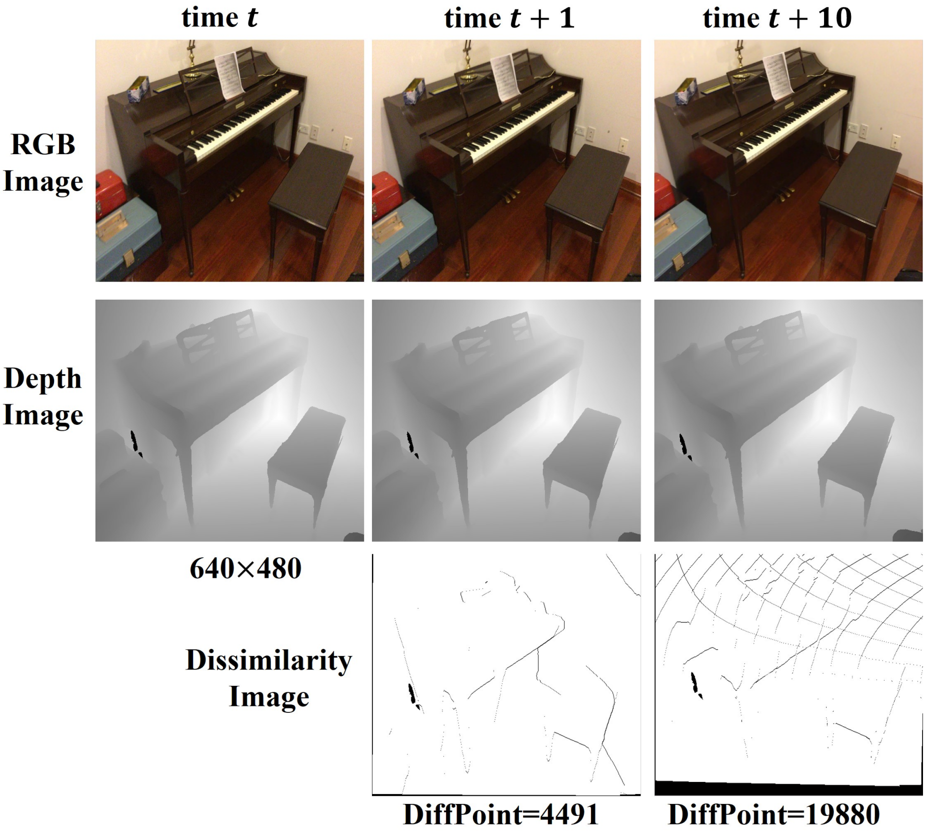
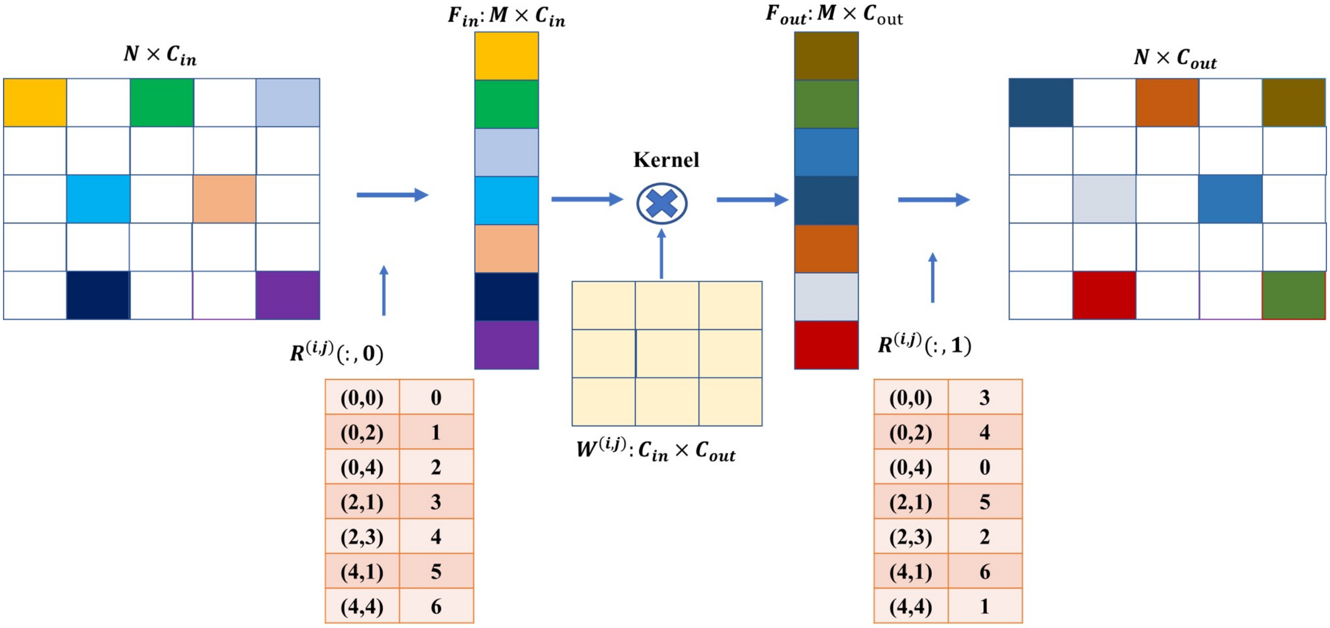


Figure 3. Image differences at different time intervals.

Complete convolution operations are not appropriate when dealing with data that lack inherent correlation due to their discrete distribution. We can employ sparse convolution, as described in [40], for deep feature learning. This approach requires designing rules to organize discrete data as sparse, as depicted in Figure 4. The size of the input data is denoted as $N \times C_{in}$, where $N = h_{in} \times w_{in}$ represents the field size and C_{in} denotes the number of feature map channels. The filter kernel size is given by $(C_{in}, C_{out}, k \times k, s)$, where k denotes the size of the convolution kernel and s represents the stride. We apply this filter to the input data results with an output data size of $C_{out} \times h_{out} \times w_{out}$, with $h_{out} = (h_{in} - k)/s$ and $w_{out} = (w_{in} - k)/s$.

We utilize a hash table structure to organize the input data as sparse data. This hash table is responsible for recording the positions of the sparse data during the convolution process. More specifically, the input data F_{in} have a size of $M \times c_{in}$ and only retain information about the sparse data. Each row vector in F_{in} represents the feature vector of M sparse pixels across all c_{in} channels. Furthermore, we introduce another hash table called R, which records the location map of the sparse pixels in the raw input data as $\{(x, y), \text{value}\}$. Here, (x, y) indicates the pixel position and value is the row index in F_{in}. The maximum size of the hash table R is $h_{in} \times w_{in}$, which typically occurs during the initial object detection phase when there are the most sparse pixels in the data. We multiply the weight matrix parameter W with F_{in}, and the output feature map F_{out} has a size of $M \times c_{out}$. The weight matrix W has dimensions $c_{in} \times f \times f \times c_{out}$, and $W^{(i,j)}$ has a size of $c_{in} \times c_{out}$, where (i, j) represents the offset to the center of the convolution kernel. For a specific offset (i, j), the input feature matrix can be fetched using $R^{(i,j)}(:, 0)$. After multiplying it by the corresponding weight parameters $W^{(i,j)}$, the output data are scattered and constructed as the output feature matrix according to the output pixel index in $R^{(i,j)}(:, 1)$. In summary, we can effectively address the challenges associated with data lacking inherent correlation by incorporating sparse convolution techniques and designing rules to organize discrete data.


$N \times C_{in}$
$F_{in}: M \times C_{in}$
Kernel
$F_{out}: M \times C_{\text{out}}$
$N \times C_{out}$
$R^{(i,j)}(:, 0)$
$W^{(i,j)}: C_{in} \times C_{out}$
$R^{(i,j)}(:, 1)$

(0,0)	0
(0,2)	1
(0,4)	2
(2,1)	3
(2,3)	4
(4,1)	5
(4,4)	6

(0,0)	3
(0,2)	4
(0,4)	0
(2,1)	5
(2,3)	2
(4,1)	6
(4,4)	1

Figure 4. Specific rules for sparse data convolution.

3.2.3. Sparse Convolution Layers

Following the rules mentioned above, the sparse convolution layers leverage the fused input to learn and extract relevant features. These layers operate sparsely, focusing on specific regions of interest rather than processing the entire input space. This approach allows the network to effectively capture and encode the most informative aspects of the data while minimizing computational overhead. Building upon these sparse convolution rules, we construct the pyramidal feature hierarchy network named Sparse Correlated Network (SCN), as shown in Figure 5.

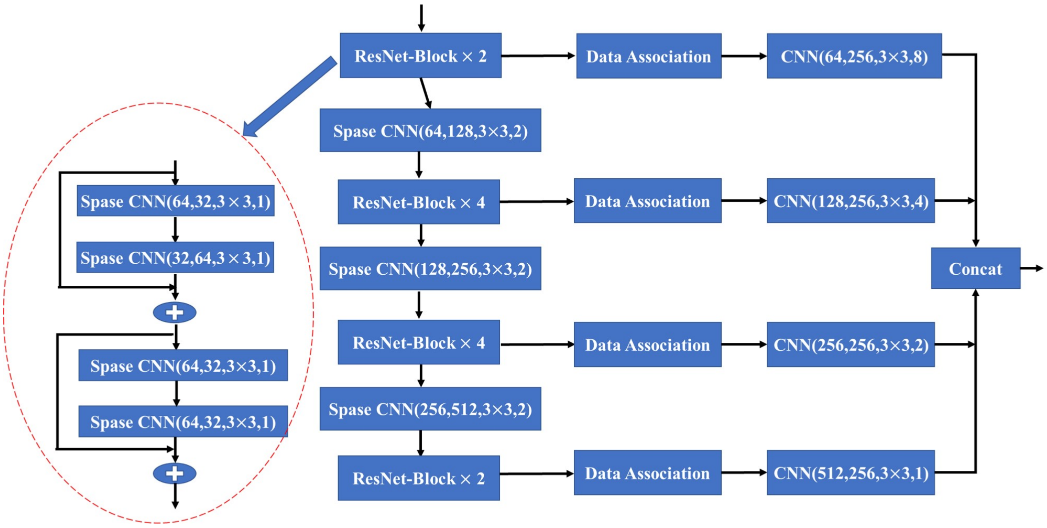


Figure 5. The detailed architecture of Sparse Correlated Network.

We establish connections between the four pyramidal levels to produce high-level feature maps, which use convolutional layers with stride sizes of (8, 8), (4, 4), or (2, 2) to capture multi-scale feature details. Combining the associated features from the previous frame with the current sparse features after performing a series of sparse convolutions is crucial to ensure a complete feature map. Specifically, we partition the dense feature maps into two datasets, denoted as $\boldsymbol{O}$ and $\boldsymbol{N}$. Let us assume that $u_n = (x_n,\ y_n)$ represents the pixel position index in the n_{th} frame's depth image D_n. Using the camera extrinsic $\boldsymbol{T}_n$, we can obtain the coordinates $W_n(u_n)$ at the world coordinate system for the point u_n. If in the $(n-1)_{th}$ frame's depth image there exists a point $W_{n-1}(u_{n-1})$ such that it is equal to $W_n(u_n)$, then we update the $\boldsymbol{O}$ set as follows: $\boldsymbol{O} = \boldsymbol{O} \cap f_{n-1}^{rgbd}(u_{n-1})$. Otherwise, we update the $\boldsymbol{N}$ set: $\boldsymbol{N} = \boldsymbol{N} \cap \hat{f}_n^{rgbd}(u_n)$. N is the SCN input and accomplishes local feature extraction by employing sparse convolution. These extracted features are subsequently processed through the data association module, creating a dense feature map. Ultimately, the feature map is outputted using a concatenation layer and represented as f_n^{rgbd}.

3.2.4. Object Proposal

To optimize computational efficiency, we introduce several steps in our approach. Firstly, we transform the center coordinates $(\triangle x_i^{n-1}, \triangle y_i^{n-1})$ of each bounding box in the set B_{n-1} to the world coordinate system and project them onto the corresponding positions $(\triangle x_i^n, \triangle y_i^n)$ in the n_{th} frame. Compared to the anchor mechanism with no prior knowledge, we initialize the bounding box set as $B_n^* = \{b_1, ..., b_i, ..., \}$, where each b_i is represented as $b_i = < \triangle x_i^n, \triangle y_i^n, w_i^{n-1}, h_i^{n-1} >$. Next, we compute the Intersection over Union (IoU) between each b_i in B_n^* and each a_j in the anchor set $\mathcal{A}$. This calculation is performed using the formula $IoU(b_i, a_j) = |b_i \cap a_j| / |b_i \cup a_j|$. If the resulting IoU value exceeds a predefined threshold θ_U, denoting significant overlap, we remove the corresponding anchor a_j from the anchor set $\mathcal{A}$. Finally, we employ a bounding box regressor to obtain comprehensive object instance information that refines the remaining anchors in $\mathcal{A}$. This regression process helps refine the localization and characteristics of the detected objects. By incorporating these optimization techniques, our method effectively reduces the computational burden and improves the overall efficiency of the system while accurately identifying object instances.

3.3. Data Integration

Semantic reconstruction involves continuously updating and fusing depth image data, known as incremental modeling [39]. Each captured depth image frame is fused into a globally consistent 3D voxel-based model according to the camera pose. The TSDF measures the distance between each voxel grid in the model and the model representation, ensuring geometric consistency constraints. Additionally, semantic elements such as class probabilities and instance bounding boxes are incorporated into the voxel grid to ensure semantic consistency. Through these steps, a 3D model is generated that satisfies both geometric and semantic consistency.

4. Evaluation

4.1. Dataset

The proposed methods are validated using the ScanNet v2 dataset [41]. This large-scale RGB-D dataset is used for indoor scene understanding and reconstruction. It primarily consists of diverse indoor scenes such as offices, homes, stores, and schools. The dataset comprises 1201 training scenes and 312 open test scenes. For each scene, it provides camera poses, semantic labels, instance labels, and surface normals. For testing purposes, we also utilize the NYU Depth v2 dataset [42], which includes 1449 RGBD images covering 40 common semantic types.

4.2. Network Details

We utilize the ResNet architecture to process the 640 × 480 RGB image and depth map separately. These inputs are passed through the network, resulting in two features of dimensions 320 × 240 × 32. The features are then stacked, resulting in a 64-channel feature. For optimization, we employ the ADAM optimizer with a momentum of 0.9, a learning rate of 0.0002, a batch size of 5, and a sequence length of 10. Our network trains on an NVIDIA GeForce RTX 3090 GPU, taking approximately two days to complete 1800 epochs.

4.3. Quantitative and Qualitative Results

To evaluate the effectiveness of SCN, we measure its segmentation accuracy using the mean Average Precision (mAP) of bounding boxes on two datasets: NYU Depth v2 [42] and ScanNet v2 [41]. The performance of SCN on the NYU Depth v2 is presented in Table 2. Our method demonstrates superior performance, primarily due to Mask R-CNN's inability to consider the correlation between detection results from consecutive frames in the scene. This confirms the validity of our bounding box generation strategy proposed in the object proposal module, which initializes the $t = n$ frame using the bounding box from the $t = n - 1$ frame.

Table 2. Comparison of semantic segmentation results between SCN and Mask R-CNN on NYU depth v2 dataset.

Method	Input	Baseline	mAP (%)
Mask R-CNN [26]	RGB + Depth	VGG	56.9
Mask R-CNN [26]	RGB + Depth	ResNet-101	62.5
SCN	RGB + Depth	VGG	63.4
SCN	RGB + Depth	ResNet-101	71.2

To evaluate the influence of inter-frame correlations on segmentation accuracy, we select two sets of data with different perspectives of the same scene, as seen from Figure 6. Both Mask R-CNN and SCN accurately segment the chair enclosed by the yellow box in view 1. However, there is a discrepancy in the segmentation results of these two methods in view 2. Mask R-CNN makes it difficult to retain the compelling features extracted in the segmentation process from the previous perspective, resulting in incorrect segmentation, which verifies the necessity of SCN to improve the accuracy of instance segmentation.

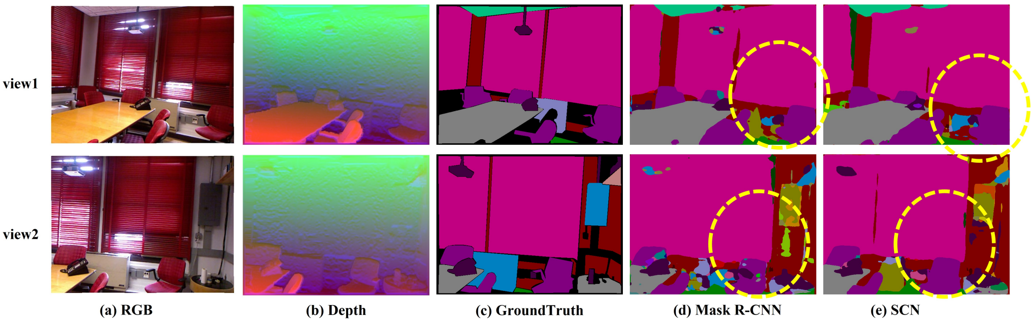


Figure 6. Comparison of semantic segmentation accuracy in NYU depth dataset.

We have also applied our SCN method to the ScanNet v2 dataset [41], yielding instance segmentation results. Figure 7 demonstrates the semantic segmentation outcomes of an individual frame image, illustrating the improved accuracy achieved by our method compared to Mask R-CNN [26]. Our results exhibit enhanced correctness in semantic segmentation. Moreover, to establish the global effectiveness of our approach, we have fused the segmentation results from various scenarios into semantic and instance-level 3D models, as depicted in Figure 8. This fusion serves as evidence of the semantic consistency attained by our approach.

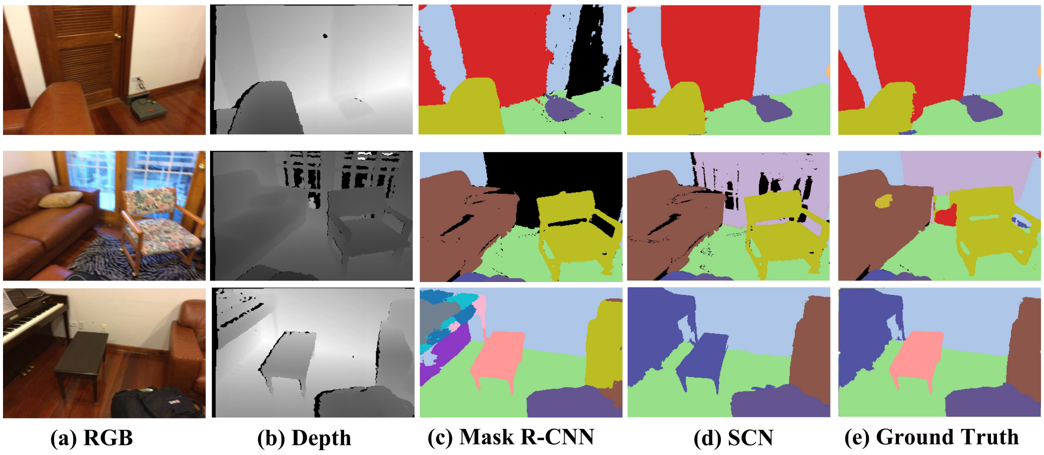


Figure 7. Comparison of semantic segmentation accuracy in ScanNet dataset.

For quantitative evaluation, we compare our results with per-voxel class accuracies, following the evaluation metrics used in previous works such as Mask R-CNN [26], PanopticFusion [37], Sparse R-CNN [27], PointGroup [28], and Mask-Group [29]. Table 3 numerically compares the mean average precision with the IoU threshold of 0.5 over 19 classes of our method against other deep learning approaches. For instance segmentation, PanopticFusion utilizes Mask R-CNN as the baseline network, Sparse R-CNN adopts sparse convolution techniques, and PointGroup and Mask Group employ point cloud grouping. These methods share similarities in design with SCN. The results highlight that SCN has an advantage in segmenting objects that frequently repeat across multiple frames, such as beds and desks. However, due to the low repeatability of small objects in adjacent frames, the proposed approach for associating features between consecutive frames may not perform optimally, resulting in suboptimal accuracy. Furthermore, Figure 9 demonstrates the object detection process SCN performs in a real indoor scene composed of 4650 depth image sequences.

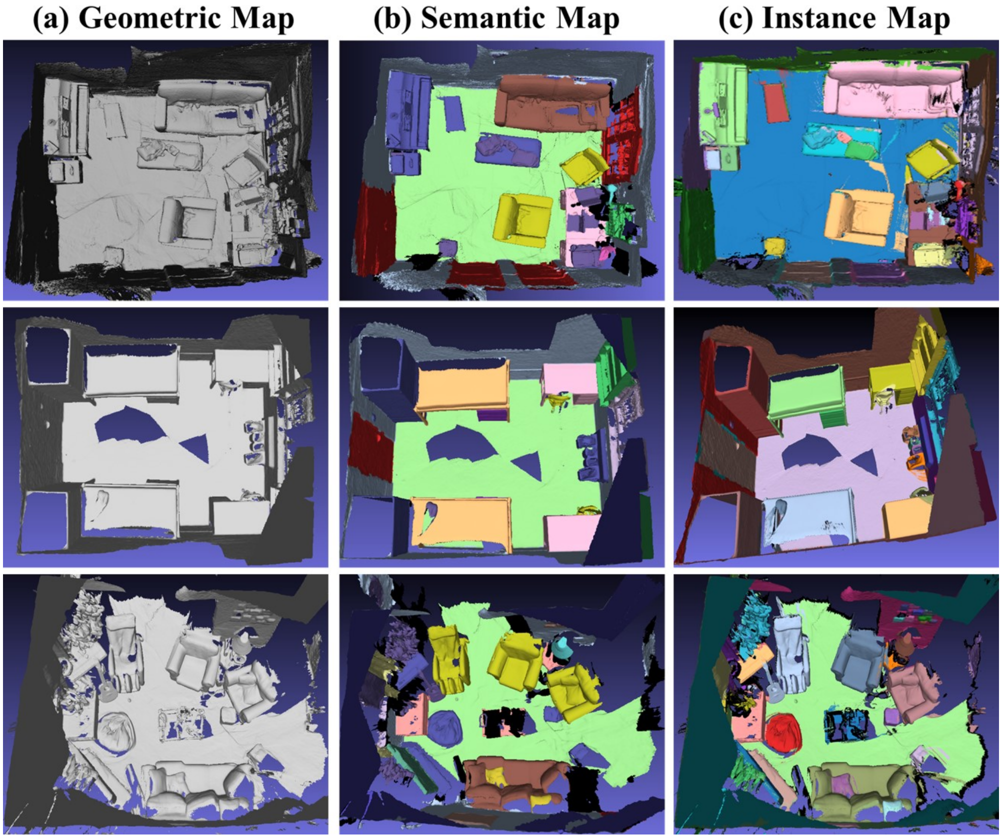


Figure 8. Instance Segmentation results obtained with our network. The integrated geometric model, semantic map, and instance map are displayed from left to right.

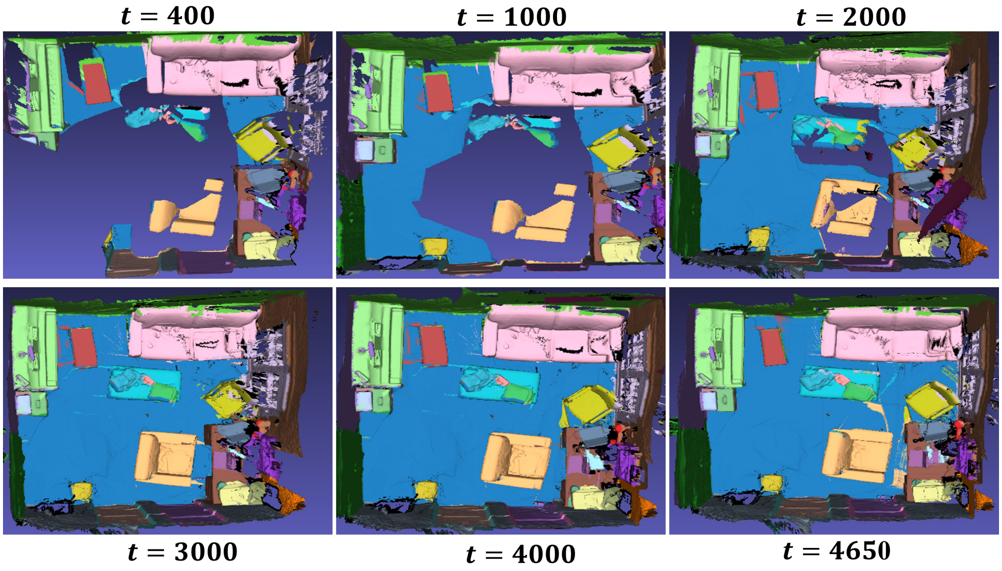


Figure 9. Online 3D semantic model generation based on camera motion.

Table 3. Comparison of the mAP_{50} of our method against other referential approaches on ScanNet v2 dataset.

Category	**Mask R-CNN** [26]	**PanopticFusion** [37]	**Sparse R-CNN** [27]	**PointGroup** [28]	**Mask-Group** [29]	**SCN**
bathtub	0.333	0.667	**1.000**	**1.000**	**1.000**	**1.000**
bed	0.002	0.712	0.538	0.765	0.822	**0.841**
bookshelf	0.000	0.595	0.282	0.624	**0.764**	0.752
cabinet	0.053	0.259	0.468	0.505	0.616	**0.620**
chair	0.002	0.550	0.790	0.797	**0.815**	0.772
counter	0.002	0.000	**0.173**	0.116	0.139	0.163
curtain	0.021	0.613	0.345	**0.696**	0.694	0.685
desk	0.000	0.175	0.429	0.384	0.597	**0.629**
door	0.045	0.250	0.413	0.441	0.459	**0.462**
otherfurniture	0.024	0.434	0.484	0.559	**0.566**	0.495
picture	0.238	0.437	0.176	0.476	**0.599**	0.424
refrigerator	0.065	0.411	0.595	0.596	0.600	**0.637**
shower curtain	0.000	**0.857**	0.591	1.000	0.516	0.480
sink	0.014	0.485	0.522	0.666	**0.715**	0.692
sofa	0.107	0.591	0.668	0.756	0.819	**0.820**
table	0.020	0.267	0.476	0.556	0.635	**0.661**
toilet	0.110	0.944	0.986	0.997	**1.000**	0.960
window	0.006	0.359	0.327	0.513	0.603	**0.721**
mAP	0.058	0.478	0.515	0.636	**0.664**	0.656

4.4. Run-Time Analysis

Utilizing the parallel computing capabilities of CUDA 10.0 architecture, our RGBD SLAM system achieves an impressive processing time of approximately 55 ms per frame, which includes 5 ms for camera tracking, 30 ms for instance segmentation, and 20 ms for data integration. In general, our indoor scene analysis system achieved a performance of 18 frames per second (fps), allowing for interactive rates of real-time processing. Table 4 compares the running time for object detection using SCN against the depth image sequence in the ScanNet dataset. Our SCN method follows the two-stage detection idea of Mask R-CNN but establishes spatiotemporal correlation within scenes, resulting in shorter running time and higher detection accuracy. In comparison, PanopticFusion directly utilizes Mask R-CNN for object detection in multi-frame depth images and then employs a CRF model for global consistency of object detection results. Compared to PanopticFusion, our method achieves improved detection accuracy and time performance.

Table 4. Time comparison.

Method	**FPS**	**Time ms**	**mAP%**
Mask R-CNN [26]	8.6	116.3	5.8
PanopticFusion [37]	4.3	232.5	47.8
SCN	18.0	55.6	65.6

4.5. AR Application

By leveraging existing semantic information, we can develop immersive augmented reality applications. Our approach utilizes the spatial distribution of detected objects within the scene, allowing us to seamlessly integrate virtual objects, such as robots and dinosaurs, into the real environment. As shown in Figure 10, this fusion process ensures that virtual objects are rendered realistically, seamlessly blending with the surrounding real scene.

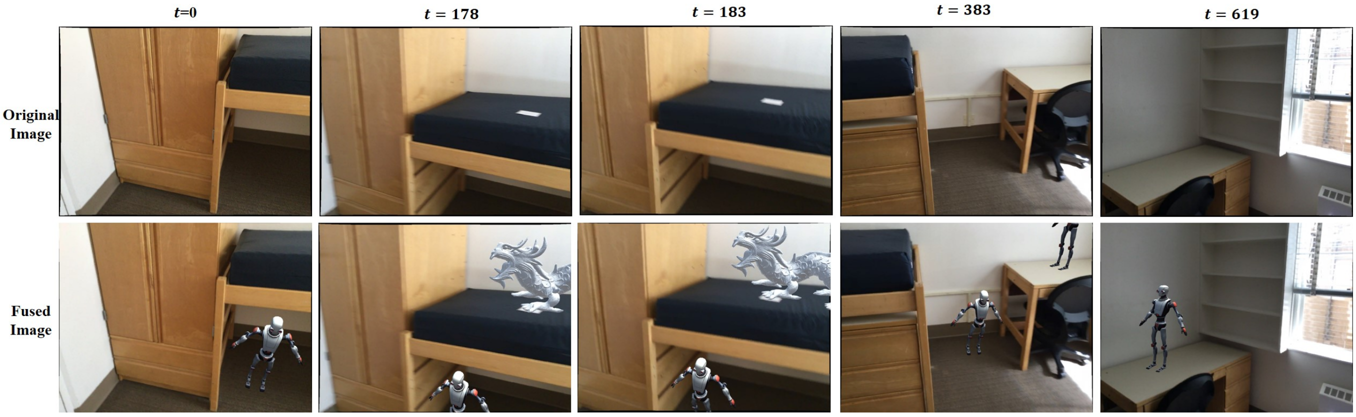


Figure 10. AR applications that combine 3D instance mapping with virtual objects.

Furthermore, our system enables the replacement of semantically known objects in the scene, such as desks and chairs, with corresponding semantic 3D models. Additionally, we can enrich the scene presentation by adding semantically associated objects, such as computers, to enhance the virtual reality experience further. Figure 11 demonstrates this augmentation and enrichment of the scene. By incorporating these advancements, our method can provide users with a highly realistic and interactive experience where virtual and real elements coexist seamlessly. The combination of precise spatial alignment and semantically meaningful object placement creates a truly immersive and captivating augmented reality environment and introduces a crucial interface for interactive intelligent manufacturing. This interface aims to redefine the production process. By seamlessly integrating virtual operation commands with real-world industrial production floors, we provide essential technological support for the digital twin platform in intelligent manufacturing, propelling the manufacturing industry into a completely new paradigm.

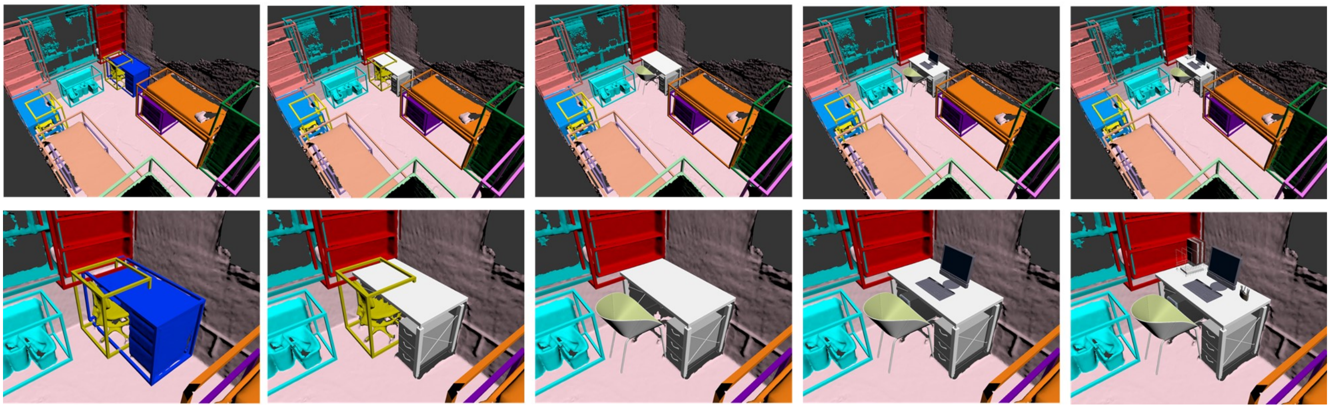


Figure 11. Virtual–real substitution based on semantic association rules.

5. Conclusions

We present a new instance segmentation network called SCN, which incorporates an object-level RGBD SLAM system. Our approach recognizes the significance of context correlation in establishing relationships and distinguishing individual objects. In the initial frame, we employ Mask R-CNN to generate the instance segmentation result and establish the mapping rule by leveraging inter-frame correlation during subsequent camera movement. We adopt various feature generation methods to accommodate different data attributes and effectively reduce computational complexity by integrating sparse convolution. Experimental results demonstrate that our network outperforms or achieves comparable performance to state-of-the-art methods regarding instance segmentation. Furthermore, our network meets the required time performance criteria. In future work, we will research dynamic industrial scenarios to enable intelligent analysis and facilitate virtual reality interactions within large-scale complex scenes.

Author Contributions: Conceptualization, C.D. and C.W.; methodology, C.W.; software, Q.W. and J.S; validation, Q.W., J.S. and C.D.; formal analysis, C.W.; investigation, C.W.; resources, C.W.; data curation, Q.W.; writing—original draft preparation, Q.W. and J.S.; writing—review and editing, C.W.; visualization, C.D.; supervision, C.W.; project administration, C.W.; funding acquisition, C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science and Technology Major Project (No. 2022ZD0119502), the National Natural Science Foundation of China (No. 62201018, No. 62201017), and the Beijing Natural Science Foundation and Haidian Original Innovation Joint Fund (No. L222052).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, M.; Ye, Z.-M.; Shi, J.-C.; Yang, Y.-L. Scene-Context-Aware Indoor Object Selection and Movement in VR. In Proceedings of the 2021 IEEE Virtual Reality and 3D User Interfaces (VR), Lisboa, Portugal, 27 March–1 April 2021; pp. 235–244. [CrossRef]
2. Zhang, A.; Zhao, Y.; Wang, S.; Wei, J. An improved augmented-reality method of inserting virtual objects into the scene with transparent objects. In Proceedings of the 2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Christchurch, New Zealand, 12–16 March 2022; pp. 38–46. [CrossRef]
3. Seichter, D.; Köhler, M.; Lewandowski, B.; Wengefeld, T.; Gross, H.-M. Efficient RGB-D Semantic Segmentation for Indoor Scene Analysis. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 13525–13531. [CrossRef]
4. Zhou, W.; Yang, E.; Lei, J.; Wan, J.; Yu, L. PGDENet: Progressive Guided Fusion and Depth Enhancement Network for RGB-D Indoor Scene Parsing. *IEEE Trans. Multimed.* **2023**, *25*, 3483–3494. [CrossRef]
5. Newcombe, R.A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A.J.; Fitzgibbon, A. KinectFusion: Real-time dense surface mapping and tracking. In Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality, Basel, Switzerland, 26–29 October 2011; pp. 127–136. [CrossRef]
6. Dai, A.; Nießner, M.; Zollhöfer, M.; Izadi, S.; Theobalt, C. Bundlefusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration. *ACM Trans. Graph. (TOG)* **2017**, *36*, 1. [CrossRef]
7. Lin, W.; Zheng, C.; Yong, J.-H.; Xu, F. OcclusionFusion: Occlusion-aware Motion Estimation for Real-time Dynamic 3D Reconstruction. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–22 June 2022; pp. 1726–1735. [CrossRef]
8. Fan, Y.; Zhang, Q.; Tang, Y.; Liu, S.; Han, H. Blitz-SLAM: A semantic SLAM in dynamic environments. *Pattern Recognit.* **2022**, *121*, 108225. ISSN 0031-3203. [CrossRef]
9. Wu, W.; Guo, L.; Gao, H.; You, Z.; Liu, Y.; Chen, Z. YOLO-SLAM: A semantic SLAM system towards dynamic environment with geometric constraint. *Neural Comput. Appl.* **2022**, *34*, 6011–6026. [CrossRef]
10. Dai, A.; Diller, C.; Niessner, M. SG-NN: Sparse Generative Neural Networks for Self-Supervised Scene Completion of RGB-D Scans. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 846–855. [CrossRef]
11. Nießner, M.; Zollhöfer, M.; Izadi, S.; Stamminger, M. Real-time 3D reconstruction at scale using voxel hashing. *ACM Trans. Graph. (TOG)* **2013**, *32*, 1–11. [CrossRef]
12. Whelan, T.; Leutenegger, S.; Salas-Moreno, R.F.; Glocker, B.; Davison, A.J. ElasticFusion: Dense SLAM without a Pose Graph. In *Robotics: Science and Systems*; 2015. Available online: https://roboticsproceedings.org/rss11/p01.pdf (accessed on 17 June 2024).
13. Fontán, A.; Civera, J.; Triebel, R. Information-Driven Direct RGB-D Odometry. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 4928–4936. [CrossRef]
14. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *Proc. IEEE* **2023**, *111*, 257–276. [CrossRef]

15. Gu, W.; Bai, S.; Kong, L. A review on 2D instance segmentation based on deep neural networks. *Image Vis. Comput.* **2022**, *120*, 104401. ISSN 0262-8856. [CrossRef]
16. Fathi, A.; Wojna, Z.; Rathod, V.; Wang, P.; Song, H.O.; Guadarrama, S.; Murphy, K.P. Semantic instance segmentation via deep metric learning. *arXiv* **2017**, arXiv:1703.10277.
17. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. YOLACT: Real-Time Instance Segmentation. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9156–9165. [CrossRef]
18. Hazirbas, C.; Ma, L.; Domokos, C.; Cremers, D. *FuseNet: Incorporating Depth into Semantic Segmentation via Fusion-Based CNN Architecture*; Springer: Cham, Switherland, 2016.
19. Zhou, W.; Yuan, J.; Lei, J.; Luo, T. TSNet: Three-Stream Self-Attention Network for RGB-D Indoor Semantic Segmentation. *IEEE Intell. Syst.* **2021**, *36*, 73–78. [CrossRef]
20. Shen, X.; Stamos, I. Frustum VoxNet for 3D object detection from RGB-D or Depth images. In Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass, CO, USA, 1–5 March 2020; pp. 1687–1695. [CrossRef]
21. Charles, R.Q.; Su, H.; Kaichun, M.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 77–85. [CrossRef]
22. Han, X.; Dong, Z.; Yang, B. A point-based deep learning network for semantic segmentation of MLS point clouds. *ISPRS J. Photogramm. Remote Sens.* **2021**, *175*, 199–214. ISSN 0924-2716. [CrossRef]
23. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Markham, A. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11105–11114. [CrossRef]
24. Thomas, H.; Qi, C.R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; Guibas, L. KPConv: Flexible and Deformable Convolution for Point Clouds. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6410–6419. [CrossRef]
25. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
26. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988. [CrossRef]
27. Sun, P.; Zhang, R.; Jiang, Y.; Kong, T.; Xu, C.; Zhan, W.; Luo, P. Sparse R-CNN: End-to-End Object Detection with Learnable Proposals. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 14449–14458. [CrossRef]
28. Jiang, L.; Zhao, H.; Shi, S.; Liu, S.; Fu, C.-W.; Jia, J. PointGroup: Dual-Set Point Grouping for 3D Instance Segmentation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 4866–4875. [CrossRef]
29. Zhong, M.; Chen, X.; Chen, X.; Zeng, G.; Wang, Y. Maskgroup: Hierarchical point grouping and masking for 3d instance segmentation. In Proceedings of the 2022 IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, 18–22 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–6.
30. McCormac, J.; Handa, A.; Davison, A.; Leutenegger, S. SemanticFusion: Dense 3D semantic mapping with convolutional neural networks. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 4628–4635. [CrossRef]
31. Xiang, Y.; Fox, D. DA-RNN: Semantic mapping with data associated recurrent neural networks. *arXiv* **2017**, arXiv:1703.03098.
32. Salas-Moreno, R.F.; Newcombe, R.A.; Strasdat, H.; Kelly, P.H.J.; Davison, A.J. SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1352–1359. [CrossRef]
33. Tateno, K.; Tombari, F.; Navab, N. When 2.5D is not enough: Simultaneous reconstruction, segmentation and recognition on dense SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 2295–2302. [CrossRef]
34. Runz, M.; Buffier, M.; Agapito, L. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In Proceedings of the 2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Munich, Germany, 16–20 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 10–20.
35. McCormac, J.; Clark, R.; Bloesch, M.; Davison, A.; Leutenegger, S. Fusion++: Volumetric object-level slam. In Proceedings of the 2018 International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 32–41.
36. Xu, B.; Li, W.; Tzoumanikas, D.; Bloesch, M.; Davison, A.; Leutenegger, S. MID-Fusion: Octree-based Object-Level Multi-Instance Dynamic SLAM. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 5231–5237. [CrossRef]
37. Narita, G.; Seno, T.; Ishikawa, T.; Kaji, Y. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; IEEE: Piscataway, NJ, USA, 2018; pp. 4205–4212.

38. Li, G.; Fan, H.; Jiang, G.; Jiang, D.; Liu, Y.; Tao, B.; Yun, J. RGBD-SLAM Based on Object Detection with Two-Stream YOLOv4-MobileNetv3 in Autonomous Driving. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 2847–2857. [CrossRef]
39. Wang, C.; Oi, Y.; Yang, S. Recurrent R-CNN: Online Instance Mapping with context correlation. In Proceedings of the 2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), Atlanta, GA, USA, 21–26 July 2020; pp. 776–777. [CrossRef]
40. Li, X.; Guivant, J.; Kwok, N.; Xu, Y.; Li, R.; Wu, H. Three-dimensional backbone network for 3D object detection in traffic scenes. *arXiv* **2019**, arXiv:1901.08373.
41. Dai, A.; Chang, X.; Savva, M.; Halber, M.; Funkhouser, T.; Nießner, M. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2432–2443. [CrossRef]
42. Silberman, N.; Hoiem, D.; Kohli, P.; Fergus, R. Indoor segmentation and support inference from rgbd images. *ECCV* **2012**, *7576*, 746–760.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.