Volumetric Instance-Level Semantic Mapping Via Multi-View 2D-to-3D Label Diffusion

Author(s):
Mascaro, Ruben; Teixeira, Lucas; Chli, Margarita

Publication date:
2022-04

Permanent link:
https://doi.org/10.3929/ethz-b-000527844

Originally published in:
IEEE Robotics and Automation Letters 7(2), https://doi.org/10.1109/LRA.2022.3146502

Volumetric Instance-Level Semantic Mapping via Multi-View 2D-to-3D Label Diffusion

Ruben Mascaro, Lucas Teixeira, and Margarita Chli

***Abstract*—Robots operating in real-world settings often need to plan interactions with surrounding scene elements and therefore, it is crucial for them to understand their workspace at the level of individual objects. In this spirit, this work presents a novel approach to progressively build instance-level, dense 3D maps from color and depth cues acquired by either a moving RGB-D sensor or a camera-LiDAR setup, whose pose is being tracked. The proposed framework processes each input RGB image with a semantic instance segmentation neural network and uses depth information to extract a set of per-frame, semantically labeled 3D instance segments, which then get matched to object instances already identified in previous views. Following integration of these newly detected instance segments in a global volumetric map, an efficient label diffusion scheme that considers multi-view instance predictions together with the reconstructed scene geometry is used to refine 3D segmentation boundaries. Experiments on indoor benchmarking RGB-D sequences show that the proposed system achieves state-of-the-art performance in terms of 3D segmentation accuracy, while reducing the computational processing cost required at each frame. Furthermore, the applicability of the system to challenging domains outside the traditional office scenes is demonstrated by testing it on a robotic excavator equipped with a calibrated camera-LiDAR setup, with the goal of segmenting individual boulders in a highly cluttered construction scenario.**

***Index Terms*—Object detection, segmentation and categorization, RGB-D perception, mapping.**

I. Introduction

OVER the past few years, research in robotics has experienced a remarkable boost, leading to an increase in the use of mobile robots for a wide variety of tasks, such as exploration, data gathering and object manipulation. For these platforms, accurately estimating the 3D geometry of their surroundings is key to forming a backbone of spatial awareness and enabling safe navigation in previously unknown environments. Performing somewhat more intelligent tasks, however, often requires a deeper understanding of the workspace in which they operate. Particularly, in cases where interacting with certain scene elements is required (e.g. for manipulation), robotic perception systems must be able to recognize and map the objects of interest at the level of individual instances, distinguishing them from the background and other entities being mapped.

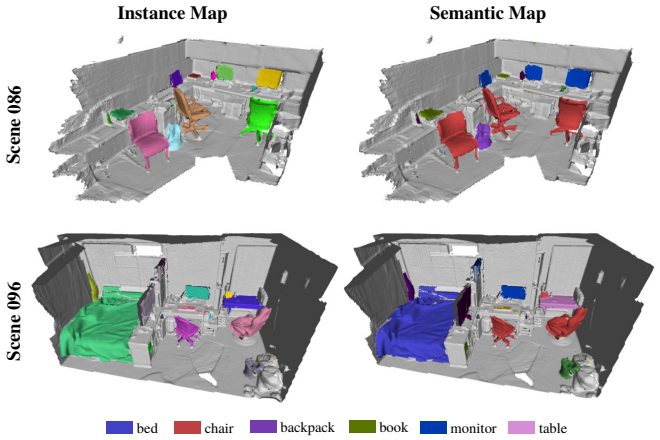


Fig. 1. Instance- and semantic-level reconstruction of two scenes from the SceneNN dataset [7] produced by the proposed incremental mapping framework. The 3D map describing the scanned scene geometry is augmented with information about the location, the shape and the semantic category of individual object instances detected in multiple fused views. Different colors in the instance maps represent different instances, while the colors associated to the recognized object categories in the semantic maps are indicated in the legend. Accompanying video is available at https://youtu.be/unkA3ZSf7wA.

With continuous progress in the development of 3D sensing technologies, such as LiDARs and depth cameras, robots can nowadays instantaneously acquire precise spatial measurements that, when fused across multiple viewpoints, usually lead to detailed 3D reconstructed scene models. Applying object recognition techniques on raw 3D data alone, however, is quite challenging, as only partial information about the scene geometry can be acquired from each view. RGB cameras, on the other hand, provide color- and texture-rich information that can be more naturally leveraged by computer vision algorithms, which recently showed to achieve unprecedented performance on image-based recognition tasks such as object detection and semantic instance segmentation [1], [2]. Combining both sensing modalities, mainly by means of RGB-D cameras so far, a number of works have successfully managed to densely reconstruct the 3D structure of an observed scene, while acquiring high-level understanding about the spatial relationships and layouts of objects in it [3], [4], [5], [6].

Following this line of research, in this work we present an approach to progressively build volumetric 3D maps during online scanning of unknown environments, while simultaneously locating and segmenting semantically meaningful objects in them. Designed for deployment on both RGB-D and camera-LiDAR sensor setups, the proposed framework processes each incoming RGB image with a semantic instance

Manuscript received: September, 9, 2021; Revised November, 8, 2021; Accepted January, 10, 2022.

This paper was recommended for publication by Editor M. Vincze upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Amazon Research Awards program and the Swiss National Science Foundation (SNF) through the National Centre of Competence in Research on Digital Fabrication (NCCR DFAB).

The authors are with the Vision for Robotics Lab, Department of Mechanical and Process Engineering, ETH Zurich, Zurich 8092, Switzerland. {rmascaro, pilucas, chlim}@ethz.ch

Digital Object Identifier (DOI): see top of this page.

segmentation neural network and uses the predicted 2D object masks, together with the depth information provided by the sensor-suite, to produce a set of per-frame, semantically-annotated 3D segments. A data association strategy subsequently matches the segments discovered in the current frame to the previously detected and mapped object instances. Finally, the newly associated segments get integrated into the global map volume, where an efficient label diffusion scheme that leverages both multi-view instance predictions and spatial context within the reconstructed scene geometry is used to enhance the resulting 3D segmentation. As shown in Figure 1, the maps produced by the proposed framework describe the scene geometry with high fidelity and provide awareness of object instances such as chairs, tables, etc., opening up exciting prospects for advanced robotic navigation and manipulation capabilities.

In a nutshell, this work presents the following contributions:

- A map-based data association strategy based on a 3D Intersection over Union (IoU) score for reliable tracking of instance-level predictions across multiple frames.
- A novel and efficient map regularization approach based on label diffusion from multiple views, especially suitable for online operation.
- A thorough evaluation of the proposed mapping pipeline on a publicly available dataset featuring RGB-D scans of real-world indoor scenes, achieving an average 20% increase in 3D segmentation accuracy when compared to the state of the art.
- The demonstration of the system using RGB-LiDAR data streams acquired by a robotic excavator operating in an outdoor, highly cluttered construction scenario.

II. Related Work

Semantic mapping is commonly referred to as the process of simultaneously estimating the 3D geometry of a scene and attaching a semantic label (e.g. object categories) to the entities being mapped. Methods in this field are typically divided into dense labeling approaches and object-oriented approaches, both explained below.

Dense labeling approaches aim at assigning a class label or a probability distribution of class labels to each point, surfel or voxel in the reconstructed 3D map. One of the most representative frameworks following this paradigm is SemanticFusion [8], which uses a Convolutional Neural Network (CNN) to infer per-pixel class probability distributions and aggregates them onto surfel-based 3D reconstructed surfaces using a Bayesian update scheme. Despite allowing for higher-level scene understanding, this family of methods lacks the ability to distinguish individual instances belonging to the same category and, as a result, information about the number, geometry and relative placement of individual objects in the scanned scene is limited. To address this issue, Nakajima *et al.* [9] propose to incrementally segment the scene using geometric cues from depth, while Pham *et al.* [10] employ a purely semantic segmentation strategy and then cluster the semantically annotated scene into individual instances. Without instance-level semantic information, however, geometry-based approaches tend to over-segment articulated scene elements, thus failing to model individual object instances accurately. More recent approaches [11], [12] explore the novel *panoptic* segmentation paradigm [13] and are able to densely predict class labels of a background region, while individually segmenting and recognizing arbitrary foreground objects. Although these methods achieve unprecedented results in terms of holistic scene understanding, they tend to be more computationally intensive and might still be harder to deploy in unconventional environments, where limited training data is available.

Methods based on the object-oriented approach, on the other hand, focus on identifying and reconstructing a set of objects of interest, while typically ignoring the semantics of the rest of the observed scene. Early works on online object-oriented mapping [14], [15] leveraged 3D model databases, requiring the shapes of the objects in the scene to be exactly the same as the pre-learnt models and therefore being inapplicable to real environments, where objects with geometric variations are frequently encountered. More recently, several methods using CNN-based architectures for detecting objects in RGB images have been reported. Sünderhauf *et al.* [16] and Nakajima *et al.* [17], for example, combine unsupervised geometric segmentation schemes with learning-based 2D bounding-box object detectors to identify and merge geometric 3D segments being recognized as part of the same instance. Fusion++ [3], MaskFusion [4] and MID-Fusion [5] leverage Mask R-CNN [1], which predicts semantically annotated 2D segmentation masks for the objects recognized in the input images, to progressively build individual 3D object models. Voxblox++ [6] combines Mask R-CNN with an incremental geometric scene segmentation approach [18] in order to produce a complete instance-aware semantic mapping framework, aiming to retrieve the pose and shape of both recognized objects and newly discovered, previously unobserved object-like instances. Finally, Wang *et al.* [19] and Li *et al.* [20] use the Mask R-CNN predictions to extract 3D instance segments that are further refined using geometric segmentation or a Gaussian Mixture Model (GMM), respectively, before getting integrated into a global map.

Our instance-level mapping pipeline is inspired by the family of the most recent RGB-D-based object-oriented approaches and also takes as input frame-wise 2D instance segmentation masks predicted with a neural network. However, instead of performing geometric refinement of the segmentation masks predicted at each frame independently [4], [6], [19], [20], we take the raw neural network predictions without further processing and seek to regularize the instance-level segmentation in the background by leveraging spatial context within the 3D map. To this end, we extend our previous work on label diffusion for offline semantic scene segmentation [21] and reformulate its core graphical model, making it less memory consuming, more efficient and able to work online as the map is being built and new objects get detected. Besides reducing the computational processing required at each input frame, this design choice allows the system not to depend on dense depth maps for segmentation enhancement and makes it suitable for LiDAR input as well, as demonstrated in our experiments.

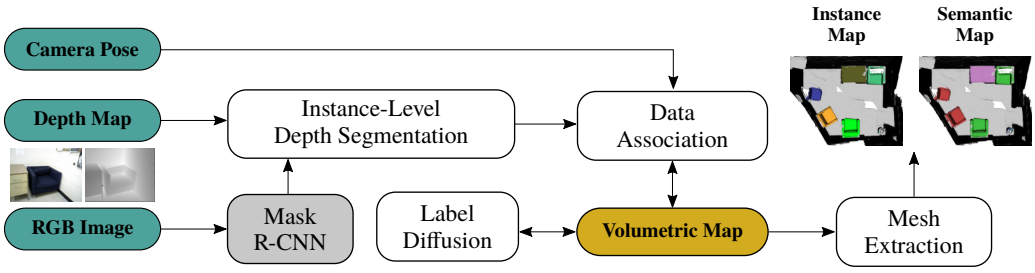


Fig. 2. Overview of the proposed approach to online instance-level mapping. Every input RGB frame is processed by Mask R-CNN [1] to predict 2D instance-level segmentation masks, which are used to segment the corresponding depth map into semantically-meaningful segments. These are then matched to previously detected and mapped object instances. Finally, the associated segments are integrated into a volumetric map, where an efficient label diffusion strategy exploits geometric context to regularize the 3D instance-level segmentation. While Mask R-CNN is used here, the proposed pipeline could work with any other semantic instance segmentation framework.

III. Method

The proposed instance-level semantic mapping pipeline, which is illustrated in Figure 2, takes as input a sequence of synchronized RGB image and depth map pairs and progressively builds a volumetric map enriched with instance-level information inferred from all the fused views (Section III-A). To this end, each incoming RGB frame is initially processed by a semantic instance segmentation framework (Section III-B) and the predicted 2D instance masks are used to extract semantically labeled 3D segments from the corresponding depth map (Section III-C). These are then matched to the object instances predicted in previous frames using a map-based data association strategy (Section III-D). Finally, the newly associated segments are integrated into the map volume (Section III-E), where a label diffusion scheme exploits spatial context to periodically regularize the resulting 3D instance-level semantic segmentation (Section III-F).

A. Volumetric Map Representation

Our approach to instance-level semantic mapping builds on top of the Voxblox TSDF-based volumetric map representation [22], which we augment with instance-level segmentation information gathered from multiple views. During the course of a mapping session, our system maintains a set $\mathcal{L}$ of unique instance labels that is tracked across frames and grows as new objects get detected. We distinguish between a predefined label $l_0 \in \mathcal{L}$, which we associate with the so-called "background" instance o_0, and the remaining $l_n \in \mathcal{L}$ with $n > 0$, each of them corresponding to an instance o_n from the set $\mathcal{O}$ of (partially) reconstructed objects in the map. Each of these instance labels $l_n \in \mathcal{L}$ is in turn associated with a semantic class $c_m \in \mathcal{C}$, with $\mathcal{C}$ being the set of object categories that the system is able to recognize, through a mapping $C(l_n) = c_m$. Instance-level segmentation information is thus stored in the map by assigning each voxel v_i an instance label $l_j \in \mathcal{L}$, $j \geq 0$, through a mapping $L(v_i) = l_j$ that determines the instance o_j it belongs to. An object instance $o_n \in \mathcal{O}$ is therefore defined as the set of voxels v_i that map to instance label l_n, which at the same time is associated with semantic class $C(l_n)$. Similarly, the background instance o_0 is referred to as the set of map voxels with label l_0, for which no semantic category is available.

B. Frame-wise 2D Semantic Instance Segmentation

The proposed mapping pipeline requires each input RGB frame to be fed into a neural network capable of detecting, segmenting and classifying individual objects present in a given image. In particular, our system assumes each of the K object instances detected in an input frame to be described by an index $k \in \{1, 2, \ldots, K\}$, a 2D binary mask M_k and a semantic class label $c_k \in \mathcal{C}$. Although our pipeline is not necessarily tied to any particular semantic instance segmentation framework, in the scope of this work we use Mask R-CNN [1] to process the RGB image stream, as it is common practice among similar methods [3], [6], [19], [20].

C. Instance-Level Depth Segmentation

For each RGB image, our system also takes as input a corresponding depth map which can either be provided by an RGB-D sensor or generated from raw 3D LiDAR data when using a calibrated camera-LiDAR setup. In the latter case, LiDAR scans acquired between consecutive input RGB images are accumulated in the map reference frame using the known sensor suite's pose and the resulting 3D point cloud is projected to the current camera view, providing depth information for a sparse set of image coordinates.

After a new RGB image has been processed by Mask R-CNN, the predicted instance segmentation masks M_k are used to segment the corresponding depth map. The masked image coordinates for which depth information is available are then back-projected into the global map frame using the known camera pose, resulting in a set $\mathcal{S}$ of globally referenced 3D segments s_k, each of them being labeled with its predicted semantic class $c_k \in \mathcal{C}$. Finally, all vertices in the depth map for which no object mask is predicted are grouped to form an additional 3D segment $s_0 \notin \mathcal{S}$, to which we assign a predefined "background" class label $c_0 \notin \mathcal{C}$.

D. Data Association

The goal of the data association module is to determine correspondences between instance segments extracted at each frame and object instances already stored in the global map. This is a required step as Mask R-CNN does not necessarily output consistent instance indices k for the same objects across

multiple frames, thus preventing direct integration of raw instance labels into the map.

Our approach to instance label tracking across frames is mostly inspired by [6] in the sense that label association takes place in 3D, thus removing the need for projecting the global map onto the current camera frame, as proposed by other similar works [3], [11], [23]. The novelty introduced by our method, however, lies in the fact that correspondences between predicted instance segments $s_k \in \mathcal{S}$ and object instances $o_n \in \mathcal{O}$ are determined according to a pairwise 3D Intersection over Union score, $\text{IoU}(s_k, o_n)$, which we compute as follows:

$$\text{IoU}(s_k, o_n) = \frac{\Pi(s_k, o_n)}{|s_k| + \Pi(s_0, o_n) + \sum_{k' \neq k} \Pi(s_{k'}, o_n)} \ . \quad (1)$$

Here, $|s_k|$ is the total number of 3D points in segment s_k and $\Pi(s_k, o_n)$ is defined as the number of 3D points in segment s_k that correspond to a voxel v_i which belongs to instance o_n, i.e. $L(v_i) = l_n$. For each object instance $o_n \in \mathcal{O}_v$, being $\mathcal{O}_v$ the subset of instances present in the current camera view, the index $\hat{k}_n$ of the best fitting segment in $\mathcal{S}$ is found as:

$$\hat{k}_n = \underset{k}{\text{argmax}} \, \text{IoU}(s_k, o_n) \ . \quad (2)$$

If the IoU score between instance o_n and its best fitting segment $s_{\hat{k}_n}$ exceeds a threshold θ, which we empirically set to 0.4, a correspondence between o_n and $s_{\hat{k}_n}$ is established by assigning $s_{\hat{k}_n}$ the instance label l_n associated with o_n, i.e. $L(s_{\hat{k}_n}) = l_n$. Finally, all segments $s_k \in \mathcal{S}$ for which no corresponding instance $o_n \in \mathcal{O}_v$ was found are assigned a new instance label l_{new}. The background segment s_0, on the other hand, is directly assigned the predefined background instance label l_0, as it does not correspond to any instance prediction.

It is important to point out that the proposed formulation prevents any predicted instance segment $s_k \in S$ from being associated with the background instance o_0 in order not to discard valuable segmentation information from the current frame. Furthermore, it disallows matching multiple predicted instance segments $s_k \in S$ to the same object instance $o_n \in \mathcal{O}_v$, thus making the system able to fix under-segmentation errors over time [6], [12]. In our case, the utilization of an IoU score instead of an absolute [6] or a normalized [3], [23] overlap metric, better helps avoid under-segmentation of foreground objects and prevents matching spurious detections to actual instances in the map.

E. Map Integration

After having been assigned a unique instance label, the 3D segments extracted from the current frame are integrated into the global volumetric map. In order not to lose segmentation information gathered from previous views, we opt for storing at each voxel v_i in our TSDF volume all the instance labels $l_j \in \mathcal{L}$ that have ever been assigned to it, together with their respective counts $\Psi_{v_i}(l_j)$. This information will then be used by our map regularization scheme to periodically update each voxel's most probable instance label, $L(v_i)$. Semantic information, on the other hand, is encoded alongside the map by storing the pairwise count between each object instance label $l_n \in \mathcal{L}$ and each semantic class $c_m \in \mathcal{C}$, which we denote as $\Phi(l_n, c_m)$.

This way, when a new segment s_k is integrated into the global map, the corresponding voxels v_i update their internal count associated with the segment's tracked instance label given by the mapping $L(s_k)$ as follows:

$$\Psi_{v_i}(L(s_k)) \leftarrow \Psi_{v_i}(L(s_k)) + 1 \ . \quad (3)$$

Additionally, if the segment does not correspond to the background instance, i.e. $L(s_k) \neq l_o$, the pairwise count between instance label $L(s_k)$ and the semantic class c_k associated with s_k is also incremented:

$$\Phi(L(s_k), c_k) \leftarrow \Phi(L(s_k), c_k) + 1 \ . \quad (4)$$

As soon as all segments extracted from the current frame have been integrated in the map, the semantic class assigned to each object instance label l_n is updated as:

$$C(l_n) = c_{\hat{m}} \ , \quad \text{with} \quad \hat{m} = \underset{m}{\text{argmax}} \, \Phi(l_n, c_m) \ . \quad (5)$$

F. Multi-View 2D-to-3D Label Diffusion

Aiming to make the system more robust towards potentially inaccurate frame-wise instance segmentation masks, we explore the benefits of using a label diffusion scheme that, besides the multi-view instance label predictions, also uses spatial context within the map to regularize the final instance segmentation. To this end, we reformulate our label diffusion approach for semantic segmentation of 3D point clouds using multiple views [21] and adapt it to work within an online instance-level mapping pipeline.

The proposed method relies on an efficient graphical structure that is used to propagate labels from a set of labeled nodes to a set of unlabeled nodes according to a series of edges defined among them. In contrast to our previous work, where semantically labeled 2D pixels are used as source nodes, here we define a set $\mathcal{U}$ of N_l virtual source nodes, with N_l being the current number of unique instance labels in the map, such that each virtual node $u_q \in \mathcal{U}$, $q \in \{1, 2, \ldots, N_l\}$, maps to instance label $l_{q-1} \in \mathcal{L}$ (note that the background label, l_0, is also considered for label diffusion). The subset $\mathcal{V}$ of the N_v map voxels belonging to the reconstructed surface act as nodes to which instance labels must be propagated.

To guide the label diffusion process, we first generate edges between the set $\mathcal{U}$ of virtual nodes and the set $\mathcal{V}$ of surface voxels, forming a subgraph $\boldsymbol{G}^{\mathcal{U}\to\mathcal{V}}$ that can be represented by a $N_v \times N_l$ adjacency matrix of the form:

$$\boldsymbol{G}^{\mathcal{U}\to\mathcal{V}}_{iq} = \lambda \cdot \Psi_{v_i}(l_{q-1}) \ . \quad (6)$$

It is worth noting that the edge between voxel v_i and virtual node u_q is weighted by the number of times instance label l_{q-1}, i.e. the label associated with u_q, has been assigned to v_i. This subgraph, therefore, encodes the instance label information being propagated from all gathered views to the 3D map, with the hyperparameter λ controlling the influence that frame-wise instance label predictions have on the final 3D segmentation.

We additionally encode relationships among surface voxels by constructing a nearest neighbor subgraph $\boldsymbol{G}^{\mathcal{V}\to\mathcal{V}}$ that reflects the underlying map geometry and whose $N_v \times N_v$ adjacency matrix is defined as:

$$\boldsymbol{G}^{\mathcal{V}\to\mathcal{V}}_{ii'} = \begin{cases} \omega(v_i, v_{i'}) & \text{if } v_{i'} \in \text{KNN}(v_i) \\ 0 & \text{otherwise} \end{cases} , \quad (7)$$

where $\text{KNN}(v_i)$ denotes the set of v_i's K neighboring surface voxels and

$$\omega(v_i, v_{i'}) = \exp\left(-\frac{\left\| \boldsymbol{x}_{v_i} - \boldsymbol{x}_{v_{i'}} \right\|_2^2}{2\sigma_d^2} - \frac{\left\| \boldsymbol{n}_{v_i} - \boldsymbol{n}_{v_{i'}} \right\|_2^2}{2\sigma_s^2} \right) . \quad (8)$$

In our formulation, $\boldsymbol{x}_{v_i}$ and $\boldsymbol{n}_{v_i}$ represent the 3D global coordinate and the surface normal vector associated with voxel v_i, while σ_d and σ_s are bandwidth hyperparameters for the Gaussian edge potential ω.

This way, the full graph for label diffusion is defined as:

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}^{\mathcal{V}\to\mathcal{V}} & \boldsymbol{G}^{\mathcal{U}\to\mathcal{V}} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} , \quad (9)$$

where $\boldsymbol{I}$ is the $N_l \times N_l$ identity matrix. It is worth pointing out that, compared to our previous approach [21], this novel formulation allows us to represent the label diffusion graph in a considerably more compact way.

Lastly, we define a label matrix $\boldsymbol{Z}$, where each element $\boldsymbol{Z}_{iq}$ represents the likelihood of node i being assigned instance label l_{q-1}. In our specific case, $\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{Z}^{\mathcal{V}} & \boldsymbol{I} \end{bmatrix}^T$, with $\boldsymbol{Z}^{\mathcal{V}}$ being a $N_v \times N_l$ matrix that can be arbitrarily initialized and $\boldsymbol{I}$ the $N_l \times N_l$ identity matrix. Label diffusion is then applied by iteratively performing the following computation:

$$\boldsymbol{Z} \leftarrow \boldsymbol{P} \cdot \boldsymbol{Z} , \quad \text{with} \quad \boldsymbol{P}_{ij} = \frac{\boldsymbol{G}_{ij}}{\sum_k \boldsymbol{G}_{ik}} . \quad (10)$$

Since the probability transition matrix $\boldsymbol{P}$ is row-normalized and guarantees the labels of the source nodes to remain unchanged by the multiplication, the algorithm is proven to converge according to [24].

In practice, as the 3D map and the set of unique instance labels $\mathcal{L}$ are constantly being updated during a mapping session, we run label diffusion periodically in a separate thread. Each time the algorithm is launched, both the label diffusion graph $\boldsymbol{G}$ and the label matrix $\boldsymbol{Z}$ are updated in order to incorporate the voxels and persistent instance labels that have been recently added to the map. Furthermore, to ensure faster convergence, elements in $\boldsymbol{Z}^{\mathcal{V}}$ are initialized to the value they reached at the end of the previous label diffusion step. If a virtual node u_q is considered for the first time due to the incorporation of a new instance label l_{q-1} in the map, all elements of the newly added column q in $\boldsymbol{Z}^{\mathcal{V}}$ are initialized to 0.

At the end of each run, likelihood values in $\boldsymbol{Z}^{\mathcal{V}}$ are used to update the instance label assigned to each voxel v_i according to:

$$L(v_i) = l_{\hat{j}} , \quad \text{with} \quad \hat{j} = \left(\underset{q}{\text{argmax}}\, \boldsymbol{Z}^{\mathcal{V}}_{iq} \right) - 1 . \quad (11)$$

IV. Experiments

The performance of the proposed framework is assessed in terms of the achieved instance-level 3D segmentation accuracy by running a series of experiments on the benchmarking SceneNN dataset [7], which features RGB-D scans of real indoor scenes and is commonly used in the literature to evaluate instance-level mapping approaches comparable to ours [6], [19], [20]. Furthermore, we demonstrate the applicability of the proposed system to other challenging domains and sensor configurations by testing it on sequences containing visual and LiDAR data acquired by a robotic excavator operating in highly cluttered construction scenarios.

All the experiments presented in this section are executed on a Lenovo laptop with an Intel Xeon E-2176M CPU and a Nvidia Quadro P200 GPU with 4 GB of memory. The mapping front-end, which comprises the depth segmentation, data association and map integration modules, is implemented in C++ and runs on the CPU, while the label diffusion component is developed in Python and runs on the GPU. The core map structure, as well as the mesh extraction and visualization tools, are adapted from the Voxblox++ [6] implementation[1] (note, however, that the two pipelines are completely different and that these specific components do not have any influence on the achieved 3D segmentation accuracy). Due to hardware constraints, frame-wise semantic instance segmentation is precomputed using a custom version of the publicly available Matterport's Mask R-CNN implementation[2] that is wrapped around a ROS interface in order to emulate real-time operation. It is worth pointing out, however, that Mask R-CNN is used in this work to allow for a direct and fair comparison against previous works [6], [19], [20], and that more lightweight instance segmentation frameworks, e.g. [2], could also be used in order to run both instance segmentation and label diffusion on small GPUs.

A. Evaluation on the SceneNN Dataset

The proposed instance-level mapping approach is evaluated on 10 indoor sequences from the SceneNN dataset [7] against three directly comparable frameworks from the state of the art [6], [19], [20]. In all these sequences, we use a Mask R-CNN model trained on the Microsoft COCO dataset [25] (like the aforementioned methods) and set the map voxel size to 2 cm. For label diffusion, the following parameters are used: $K = 24$, $\lambda = 10^{-4}$, $\sigma_d = 0.05$, and $\sigma_s = 0.15$.

Following the evaluation procedure introduced by [6], we consider 9 object categories (i.e. *bed*, *chair*, *sofa*, *table*, *books*, *refrigerator*, *television*, *toilet* and *bag*) from the Microsoft COCO object classes and, for each scene, compute the per-class Average Precision (AP) score using an Intersection over Union (IoU) threshold of 0.5 over the predicted 3D segmentation masks. For brevity, here we directly report the mean Average Precision (mAP) obtained at each of the 10 evaluated sequences, which is computed by averaging the per-class AP scores over the 9 considered categories.

[1] https://github.com/ethz-asl/voxblox-plusplus
[2] https://github.com/matterport/Mask_RCNN

Voxblox++ [6] **Ours w/o Diffusion** **Ours**

Fig. 3. Instance-level segmentation results obtained on Sequence 011 of the SceneNN dataset [7] with the two evaluated versions of our approach and the public implementation of Voxblox++ [6]. Compared to Voxblox++, our method is less prone to over-segmentation of single object instances (red circle). In addition, the ablation study shows that label diffusion helps our method achieve better 3D segmentation boundaries (blue circle). Note that different colors represent different recognized object instances in the scene.

TABLE I

EVALUATION OF 3D INSTANCE-LEVEL SEGMENTATION ACCURACY ON 10 SEQUENCES OF THE SCENENN DATASET. FOR EACH SCENE, PER-CLASS AP SCORES ARE EVALUATED USING AN IOU THRESHOLD OF 0.5 AND THEN AVERAGED OVER THE 9 CONSIDERED CLASSES, RESULTING IN THE REPORTED MAP VALUES. THE PROPOSED APPROACH IMPROVES OVER THE STATE OF THE ART IN 6 OF THE 10 EVALUATED SEQUENCES. IN ADDITION, THE ABLATION STUDY ON THE LABEL DIFFUSION MODULE DEMONSTRATES THAT IT CONTRIBUTES SIGNIFICANTLY TO INCREASE THE ACHIEVED 3D SEGMENTATION ACCURACY. BEST RESULTS ON EACH SCENE ARE HIGHLIGHTED IN BOLD, WHILE SECOND BEST SCORES ARE UNDERLINED.

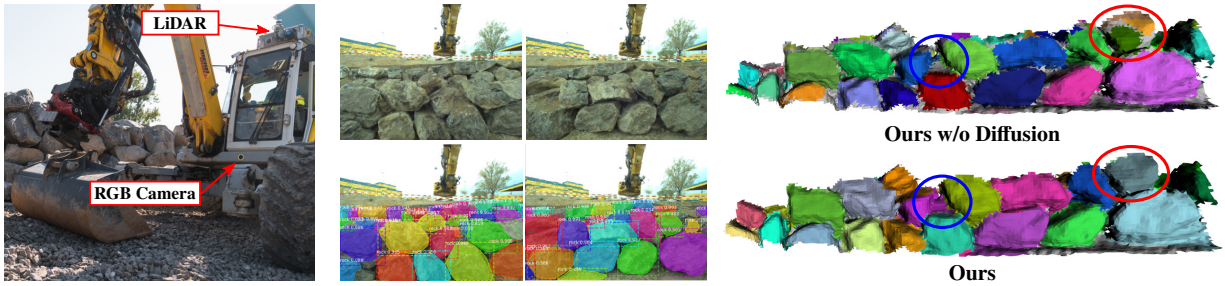
Method	011	016	030	061	078	086	096	206	223	225	Average
Voxblox++ [6]	75.0	33.3	56.1	**66.7**	45.2	20.0	29.2	**79.6**	43.8	75.0	52.4
Wang et al. [19]	62.2	43.0	60.7	36.3	49.3	45.8	32.7	46.0	**46.6**	56.4	47.9
Li et al. [20]	78.6	25.0	58.6	46.6	**69.8**	47.2	26.7	78.0	45.8	75.0	55.1
Ours w/o Diffusion	66.7	25.0	67.0	50.0	25.0	25.0	37.2	65.3	12.5	**100**	47.3
Ours	**100**	**75.0**	**72.5**	50.0	50.0	**50.0**	**51.3**	74.1	45.8	**100**	**66.8**

Besides drawing a comparison against previous methods in the literature, we also perform an ablation study where we analyze the influence of the proposed diffusion-based map regularization scheme on the final 3D segmentation accuracy. To this end, we disable the label diffusion module and run a baseline version of our system, which we call "Ours w/o Diffusion", where each voxel gets assigned the instance label with the highest count every time a new set of predicted instance segments is integrated in the map. Results obtained in our experiments, together with the performance scores reported by the three state-of-the-art frameworks mentioned above, are summarized in Table I.

Firstly, our ablation study demonstrates that the proposed label diffusion scheme for map regularization triggers a significant increase in the achieved 3D segmentation accuracy. Although, without label diffusion, our approach could theoretically reach 100% accuracy if used with ground-truth segmentation masks, the system's performance drops dramatically when taking as input predictions from Mask R-CNN. This is caused by the fact that 2D segmentation masks predicted with Mask R-CNN tend to be noisy and usually do not respect object boundaries very well. While 3D fusion of predictions from multiple views via voting strategies generally helps eliminate some inconsistencies, the resulting 3D segmentation still suffers from objects being partially segmented or background regions being labeled as parts of an instance, as shown in Figure 3. By additionally considering geometric context, our proposed label diffusion scheme effectively corrects for these mislabelled regions, achieving a considerably more accurate 3D segmentation in most of the evaluated scenes.

Results obtained in our experiments also indicate that the proposed approach outperforms the state of the art in 6 out of the 10 evaluated sequences, while achieving the second best score in other 3 of them. Previous methods we compare against focus on refining the 2D segmentation masks predicted by Mask R-CNN before integrating frame-wise instance-level segmentation information in the 3D map. To this end, they leverage geometric segmentation methods that tend to respect scene boundaries well and, therefore, are able to achieve better performance than our baseline method without label diffusion, which directly integrates the raw predicted masks in the map. However, by employing data association strategies simply based on overlap [6], [20] or proximity [19] and label fusion methods not taking context into account, these methods are not very robust towards inconsistent instance predictions across frames. We observed that this might lead to problems such as over-segmentation of individual instances, as shown in Figure 3. Our full system, despite relying on considerably less accurate segmentation masks being projected into 3D, ensures that only reasonably consistent instance predictions get associated and leverages spatial context to progressively regularize the resulting 3D segmentation, leading to an overall 20% more robust performance.

Analyzing the specific sequences where our method does not achieve the best accuracy, we noticed that performance tends to drop when dealing with very cluttered scenes containing partially occluded or small objects (e.g. books). This is mainly caused by Mask R-CNN often failing to detect these objects or predicting highly inaccurate segmentation masks. In these cases, our data association algorithm is sometimes not



(a) Camera-LiDAR sensor setup on the robotic excavator HEAP

(b) Example RGB input (top) and Mask R-CNN predictions (bottom)

(c) Instance-level 3D reconstructions

Fig. 4. Evaluation of the proposed system on a dataset collected in a construction site using the calibrated camera-LiDAR setup visible in (a) onboard the robotic excavator HEAP. In (b), two example input RGB images are shown together with the predictions provided by our custom Mask R-CNN model trained for the task of boulder detection. Finally, the instance-level 3D reconstructions obtained with the two evaluated versions of our method are visible in (c). With label diffusion enabled, the proposed system is able to capture 3D boundaries more accurately (blue circle) and eliminate over-segmentation errors (red circle). Note that different colors represent different instances.

able to find enough consistent matches and label diffusion ends up assigning the background label to voxels that are actually part of an object, leading to under-segmentation errors. We envision that leveraging geometry for frame-wise instance segmentation and adaptively setting the number of neighbors in our label diffusion graph could be interesting directions to further improve the system.

Table II shows the runtimes of the individual modules of the mapping pipeline averaged over the 10 evaluated sequences. With our aforementioned hardware settings, the system runs at approximately 2.3 Hz, restricted by the frame-rate at which Mask R-CNN can process the 640×480 input images. To place these numbers in context, we also report the average timings we measured while running the public implementation of Voxblox++ on the same 10 sequences. It is worth noting that both depth segmentation and data association are considerably faster in our framework. That is because Voxblox++ uses a geometric approach to segment each depth frame into multiple convex segments (some of which then get assigned an instance label) and aims at tracking these across frames. Our approach, on the contrary, directly extracts segments from depth based on the raw Mask R-CNN predictions and runs data association at the level of instances, which is generally less expensive as the number of detected instances in each frame is significantly lower than the number of geometric segments extracted by Voxblox++.

TABLE II

EXECUTION TIMES OF EACH MODULE IN THE PROPOSED INSTANCE-LEVEL MAPPING FRAMEWORK, AVERAGED OVER THE EVALUATED 10 SEQUENCES FROM SCENENN, AND COMPARED WITH MEASURED TIMINGS OF THE CORRESPONDING COMPONENTS IN VOXBLOX++ [6]. MASK R-CNN AND LABEL DIFFUSION RUN ON THE GPU, WHILE THE REMAINING STAGES ARE EXECUTED ON THE CPU. NOTE: *LABEL DIFFUSION RUNS ON ITS OWN THREAD, NOT AFFECTING THE FRAME-RATE AT WHICH THE SYSTEM CAN OPERATE.

Component	Frequency	Voxblox++ [6]	Ours
Mask R-CNN	Every frame	435	435
Depth segmentation	Every frame	677	13
Data association	Every frame	109	25
Map integration	Every frame	234	222
Label diffusion*	Every 5 sec.	-	695
Frame-rate		∼1.4 Hz	∼2.3 Hz

B. Evaluation on a LiDAR-based Mapping Application

The proposed approach is further evaluated on real-life data streams acquired with a robotic excavator operating in a construction site. The goal here is to demonstrate that the proposed method is not only restricted to RGB-D sensing applications, as it does not require dense depth cues to perform refinement of the 2D instance segmentation masks, in contrast to previous approaches [4], [6].

The robotic system used to record the evaluation datasets is the HEAP (Hydraulic Excavator for an Autonomous Purpose), a highly customized Menzi Muck M545 walking excavator developed for autonomous applications and advanced teleoperation [26]. In particular, we collect the data provided by a Ximea MC031CG-SY camera placed at the cabin's base and a Velodyne Puck LiDAR sensor mounted on the front edge of the cabin's roof, as shown in Figure 4(a), capturing a wall-like structure in the vicinity of the robot. Global cabin localization is achieved by a Leica iCON iXE3 with two GNSS antennas and a receiver that obtains Real-Time Kinematic (RTK) corrections for improved accuracy, while machine orientation is tracked with Inertial Measurement Units (IMUs) installed both in the cabin and on the chassis.

The target is to reconstruct and segment boulders in highly cluttered scenarios, which is a key step in order to perform advanced manipulation tasks (e.g. grasping from boulder stockpiles) as well as construction monitoring and planning [27]. To this end, we use a custom Mask R-CNN model that we trained for the task of boulder detection on a manually annotated dataset containing 960 images sampled from videos that were recorded with a handheld RGB camera while walking around different stockpiles. Since, at large distances, LiDAR measurements might become too sparse to allow for dense mapping and image-based instance segmentation networks might fail to recognize objects, we set a maximum range of 7.5 meters from the camera for mapping.

As ground truth is not available for these experiments, qualitative results for the reconstructed object-centric maps using our system with and without the label diffusion module are shown in Figure 4. Again, it can be observed that, with label diffusion, our method is able to capture actual 3D instance boundaries more accurately, while eliminating some over-segmentation errors. The resulting reconstructions densely describe the observed surface geometry and provide information about the shape and the pose of the individual boulders within the robot's workspace, which is key for further interaction planning.

V. Conclusion

In this paper, we presented an approach to volumetric instance-level semantic mapping using color and depth cues from localized sensors. The method incrementally fuses information about individual objects detected in multiple views, building a global 3D map of the observed scene that is augmented with the location, shape and semantic category of those recognized objects. In contrast to previous work relying on geometrically-refined 2D segmentation masks being projected to the 3D map, our method introduces an efficient map regularization strategy based on label diffusion that can handle less accurate instance predictions as input. This removes any dependency of the proposed approach on dense depth maps and enables its application to different sensing modalities than RGB-D, such as combinations of cameras and LiDARs. Results obtained by running the framework on a publicly available RGB-D indoor scene dataset validate the online nature of the approach and show its ability to achieve state-of-the-art performance in 3D segmentation accuracy. Furthermore, we show the applicability of the framework to the novel and challenging task of reconstructing and segmenting boulders from visual-LiDAR data in a highly cluttered construction scenario.

Future research directions involve investigating strategies to keep the complexity of the map regularization scheme bounded when dealing with larger scenes. In addition, we plan to integrate the presented framework on the robotic excavator HEAP in order to perform complex manipulation tasks in large-scale construction settings.

References

[1] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[2] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[3] J. McCormac, R. Clark, M. Bloesch, A. J. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level SLAM," in *International Conference on 3D Vision (3DV)*, 2018.

[4] M. Rünz, M. Buffier, and L. Agapito, "MaskFusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2018.

[5] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "MID-Fusion: Octree-based object-level multi-instance dynamic SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[6] M. Grinvald, F. Furrer, T. Novkovic, J. J. Chung, C. Cadena, R. Siegwart, and J. Nieto, "Volumetric instance-aware semantic mapping and 3D object discovery," *IEEE Robotics and Automation Letters (RA-L)*, vol. 4, no. 3, pp. 3037–3044, 2019.

[7] B. S. Hua, Q. H. Pham, D. T. Nguyen, M. K. Tran, L. F. Yu, and S. K. Yeung, "SceneNN: A scene meshes dataset with annotations," in *International Conference on 3D Vision (3DV)*, 2016.

[8] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "SemanticFusion: Dense 3D semantic mapping with convolutional neural networks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[9] Y. Nakajima, K. Tateno, F. Tombari, and H. Saito, "Fast and accurate semantic mapping through geometric-based incremental segmentation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[10] Q. H. Pham, B. S. Hua, D. T. Nguyen, and S. K. Yeung, "Real-time progressive 3D semantic segmentation for indoor scenes," in *IEEE Winter Conference on Applications of Computer Vision*, 2019.

[11] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji, "PanopticFusion: Online volumetric semantic mapping at the level of stuff and things," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[12] D.-C. Hoang, A. J. Lilienthal, and T. Stoyanov, "Panoptic 3D mapping and object pose estimation using adaptively weighted semantic information," *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, no. 2, pp. 1962–1969, 2020.

[13] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[14] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "SLAM++: Simultaneous localisation and mapping at the level of objects," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[15] K. Tateno, F. Tombari, and N. Navab, "When 2.5D is not enough: Simultaneous reconstruction, segmentation and recognition on dense SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[16] N. Sünderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid, "Meaningful maps with object-oriented semantic mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[17] Y. Nakajima and H. Saito, "Efficient object-oriented semantic mapping with object detector," *IEEE Access*, vol. 7, pp. 3206–3213, 2019.

[18] F. Furrer, T. Novkovic, M. Fehr, A. Gawel, M. Grinvald, T. Sattler, R. Siegwart, and J. Nieto, "Incremental object database: Building 3D models from multiple partial observations," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[19] L. Wang, R. Li, J. Sun, X. Liu, L. Zhao, H. S. Seah, C. K. Quah, and B. Tandianus, "Multi-view fusion-based 3D object detection for robot indoor scene perception," *Sensors*, vol. 19, no. 19, 2019, Art. ID 4092.

[20] W. Li, J. Gu, B. Chen, and J. Han, "Incremental instance-oriented 3D semantic mapping via RGB-D cameras for unknown indoor scene," *Discrete Dynamics in Nature and Society*, vol. 2020, 2020, Art. ID 2528954.

[21] R. Mascaro, L. Teixeira, and M. Chli, "Diffuser: Multi-view 2D-to-3D label diffusion for semantic scene segmentation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[22] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3D Euclidean Signed Distance Fields for on-board MAV planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[23] D. C. Hoang, A. J. Lilienthal, and T. Stoyanov, "Object-RPE: Dense 3D reconstruction and pose estimation with convolutional neural networks," *Robotics and Autonomous Systems*, vol. 133, 2020, Art. ID 103632.

[24] X. Zhu, "Semi-supervised learning with graphs," Ph.D. dissertation, Carnegie Mellon University, 2005.

[25] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision (ECCV)*, 2014.

[26] D. Jud, P. Leemann, S. Kerscher, and M. Hutter, "Autonomous free-form trenching using a walking excavator," *IEEE Robotics and Automation Letters (RA-L)*, vol. 4, no. 4, pp. 3208–3215, 2019.

[27] R. L. Johns, M. Wermelinger, R. Mascaro, D. Jud, F. Gramazio, M. Kohler, M. Chli, and M. Hutter, "Autonomous dry stone: On-site planning and assembly of stone walls with a robotic excavator," *Construction Robotics*, vol. 4, pp. 127–140, 2020.