Leveraging Foundation Models for Labeling Custom Object Masks in LiDAR Point Cloud Sequences

Alexander Prutsch[1], Mathias Schustereder[1] and Horst Possegger[1]

***Abstract*— 3D segmentation plays a crucial role in the perception stack of autonomous systems, such as mobile robots, by enabling fine-grained understanding of their surroundings. State-of-the-art approaches rely on deep learning and typically require large-scale annotated datasets, the creation of which is both costly and labor-intensive. Recent advances in vision foundation models, such as Segment Anything 2 (SAM 2), demonstrate strong generalization capabilities in segmenting objects across diverse video data. In this work, we present a novel pipeline for generating high-quality pseudo-labels for 3D point cloud segmentation with minimal human supervision. We propose a custom projection to transfer LiDAR point clouds to an 2D image proxy representation using range and reflectivity data. As a result, sequential LiDAR scans can be effectively treated as video input, which allows us to leverage SAM 2 for fast and efficient LiDAR mask generation. Our method produces accurate labels across a variety of object types and enables the training of 3D segmentation models solely on these semi-automatically generated annotations. Our approach significantly lowers the barrier to applying 3D segmentation in custom domains, especially for object categories not covered in existing public datasets.**

I. INTRODUCTION

Autonomous systems such as self-driving cars and autonomous mobile robots (AMRs) rely on robust perception systems to interpret and understand their environment. Computer vision tasks like object detection and semantic segmentation are essential for enabling such capabilities. As input to these tasks, autonomous vehicles employ a variety of sensors, including RGB cameras, time-of-flight sensors, and 2D/3D LiDAR scanners. Among these, 3D sensors offer the advantage of detailed spatial awareness, which is essential for accurate scene understanding. Recent progress in robot perception has been largely driven by deep learning. Neural network-based methods are state-of-the-art for tasks like semantic segmentation or object detection on both 2D and 3D modalities. However, developing models for *custom use-cases* often requires training or fine-tuning on domain-specific labeled datasets. For segmentation tasks in particular, where pixel- or point-level masks are needed, generating ground-truth annotations is time-consuming, resource-intensive, and often impractical — especially when targeting uncommon or custom object categories.

The public availability of large-scale computer vision datasets, combined with recent advances in computational power, has enabled the development of vision foundation models (VFMs) [1], [2], [3], [4], [5], [6]. These models are deep neural networks trained on diverse and extensive datasets, allowing them to generalize effectively across a wide range of domains. VFMs have been proposed for tasks like image classification [7], [8] monocular depth estimation [8], [9], semantic segmentation of images [2], [8], open-set object detection [6], or object tracking [4], [10]. VFMs can be adapted to specific data domains or tasks either through fine-tuning or by leveraging promptable architectures that enable task execution without retraining. Their broad training data enables VFMs to exhibit strong zero-shot capabilities, making them especially well-suited for applications involving custom data and rare object categories.

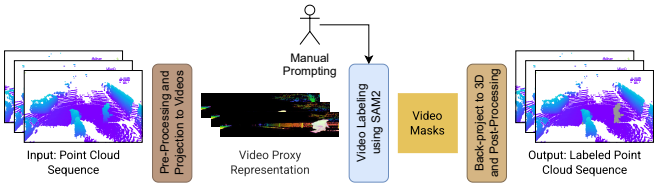


Fig. 1: Our labeling pipeline enables efficient annotation of custom object masks in 3D point cloud sequences. We first project the point clouds into a panoramic video using a custom range- and signal-based projection. Segment Anything 2 [4] is then used to generate object masks in the video data with only sparse human-provided prompts. The 2D segmentation masks are subsequently back-projected into 3D and refined through post-processing.

The strong generalization capability of VFMs makes them an effective method for generating high-fidelity annotations, such as segmentation masks, by leveraging promptable designs and minimal human supervision, *i.e.,* point prompts. In this work, we focus on the generation of segmentation masks for custom objects in 3D LiDAR sequences. A representative use case is an autonomous mobile robot that needs to detect and segment custom object classes directly from LiDAR data by identifying the corresponding object points. This task is typically addressed using deep learning-based 3D semantic segmentation models, which require annotated LiDAR scans (i.e., labeled point clouds) for training. However, producing such annotations manually is a time-consuming and labor-intensive process, especially when dealing with uncommon or application-specific objects, *e.g.,* a specific vehicle type within a warehouse or special machines in a factory.

We propose a novel semi-automatic labeling pipeline to enable fast and efficient generation of complex 3D segmentation masks on LiDAR sequence data. As illustrated in Figure 1, our pipeline leverages a novel proxy representation, where point cloud sequences are projected to panoramic videos using both range and reflectivity data.

[1] Affiliated with the Institute of Visual Computing, Graz University of Technology. Corresponding author: alexander.prutsch@tugraz.at

Our projection effectively preserves the shape of objects which are essential for executing object segmentation. To conduct the object segmentation we then employ Segment Anything 2 (SAM 2) [4], a vision foundation model designed for segmenting arbitrary objects in videos. Human annotators solely have to provide sparse point prompts on the objects of interest. Following, SAM 2 provides object masks and propagates them across subsequent frames further reducing the required annotation effort. Its strong emphasis on edges enables SAM 2 to segment objects from our proposed proxy representation without requiring a data adapter or fine-tuning. After generating object masks from the video data, we project them back into 3D space to obtain segmentation masks. As an example, our pseudo-labels can be used in downstream tasks like 3D segmentation models, such as those proposed by [11], [12].

Our labeling engine enables the annotation of custom point cloud data with minimal human intervention. Ideally, a single point prompt per object is sufficient to segment it throughout a recording sequence. Projecting to 2D panoramic videos simplifies prompt selection for human annotators compared to direct interaction with 3D data. As a result, our system enables fast and cost-efficient utilization of custom data for complex 3D perception tasks. Our method operates without training or fine-tuning, and the annotation stage requires no parameter tuning. Only the preprocessing and post-processing steps involve dataset-specific parameters – related to the sensor model and relevant annotation range. This makes our approach easy to deploy and broadly applicable across diverse scenarios.

Our main contributions include

- We propose a novel approach to annotate complex 3D segmentation masks using only sparse human supervision by modeling point cloud sequences in a video proxy representation.
- We highlight that our proxy representation effectively allows to represent 3D data as compact and easy to interpret 2D data while preserving necessary features for generating segmentation mask.
- We showcase our pipeline on public LiDAR data and conduct a proof of concept study for using our pipeline in a custom autonomous mobile robots setting.

II. RELATED WORK

In this work, we focus on generating segmentation mask annotations for 3D point cloud sequences, targeting both semantic and instance segmentation tasks. The core objective is to assign a categorical label to each point of interest within a 3D scene. For semantic segmentation, these labels correspond to object classes (*e.g.,* car, pedestrian, building), whereas instance segmentation distinguishes between individual object instances of the same class. 3D semantic segmentation plays a crucial role in the perception systems of autonomous vehicles. It enables the interpretation and understanding of complex, dynamic environments by providing detailed spatial and semantic context. It supports key tasks such as obstacle detection and scene understanding, which are essential for autonomous operation of mobile robots in real-world settings.

A. Weakly Supervised 3D Perception

Obtaining dense annotations for training state-of-the-art 3D perception models is a challenging and expensive task. Hence, previous research has focused on training models using weak supervision. For example, ScribbleKITTI [13] introduces sparse *scribble* annotations for 3D segmentation on the SemanticKITTI dataset [14]. Their dataset contains annotations for only 8.06% of all points. They propose a sophisticated multistep pipeline to train a Cylinder3D [11] network using these sparse annotations. Their work demonstrates that such minimal annotations can still yield highly accurate predictions on the SemanticKITTI dataset. Recently, Viswanath *et al.* [15] explored the use of Segment Anything [2] on projected LiDAR data as an efficient method for label generation. They adopt a basic projection approach using only calibrated intensity data, without incorporating range information or specifically tailored transfer functions. Additionally, unlike our approach, their work operates on single frames, neglecting the possibility of mask propagation across sequential data. Furthermore, they do not incorporate any preprocessing or 3D label refinement steps, both of which are essential to achieve accurate point cloud masks, especially for custom object classes.

The use of 2D bounding boxes as weak supervision for 3D object detection and automatic labeling of bounding boxes is also an active area of research [16], [17]. Annotating 2D bounding boxes is significantly easier compared to 3D annotations. By utilizing the camera-to-LiDAR calibration, it is possible to project them as search windows onto the LiDAR point cloud to guide object detection. In contrast, our approach operates without requiring an additional camera, which is an important advantage in robotics applications, where the number of sensors is often limited due to cost, power, and resource constraints. In many real-world robotics settings, camera supervision may also be infeasible or undesired due to privacy concerns or restricted environments. Furthermore, our method directly generates complex object masks rather than just bounding boxes. Relying on cameras can also introduce limitations, as their field of view may only cover a fraction of the LiDAR range, resulting in incomplete or biased supervision.

B. Efficient 3D Segmentation via 2D Projections

Projecting 3D LiDAR data into 2D image-like representations is a well-established strategy for achieving efficient 3D segmentation [18], [19], [20]. The primary advantage of this approach is that computationally intensive operations, such as 3D convolutions, can be replaced by simpler 2D operations, which are less resource-demanding. To preserve 3D information in the 2D space, depth values can be encoded using color values to create a range view. Furthermore, projecting additional data such as intensity values or other ambient information, generates pseudo-images that captures the shapes and structures of 3D objects.

C. Vision Foundation Models

For many years, the standard approach in deep learning has been to train models for particular tasks using domain-specific data. However, for custom use cases, collecting large amounts of annotated data for training is often challenging and costly. Due to limited data availability of labeled data and the high computational costs of training deep learning models from scratch, it has become common practice to leverage pretrained models and finetune them on custom data. For instance, in image classification, a model pretrained on a large dataset like ImageNet [21] can be adapted by modifying and finetuning its classification head to suit custom data.

Although fine-tuning significantly reduces the resources and data required for training, it still requires careful adaptation to specific domains and can be time-consuming. In recent years, vision foundation models (VFMs) [1], [2], [3], [4], [5], [6] have emerged. These models are trained on vast, diverse datasets, which enables them to generalize well across a wide range of tasks. VFMs tend to learn general-purpose features that make them suitable for multiple downstream tasks, exhibiting great zero-shot capabilities even on custom domains. Promptable architectures enable vision foundation models to be applied to custom data without the need for fine-tuning. Prompts are simple cues that guide the model to perform more complex tasks. For example, in image segmentation, the goal is to generate high-fidelity object masks. To segment custom objects in domain-specific data, a vision foundation model can be prompted with simple hints, such as single points or bounding boxes, to produce the desired segmentation results [2].

Segment Anything (SAM) [2] is a VFM for image segmentation, supporting various types of prompts, including points, and bounding boxes. It is trained on a custom large-scale dataset (named SA-1B) through a multi-stage training pipeline. SAM operates in a class-independent manner and demonstrates a strong zero-shot performance. Additionally, SAM is capable of generating segmentation masks with varying levels of granularity, such as whole objects or individual parts of an object. Segment Anything 2 (SAM 2) [4] extends SAM to handle spatiotemporal data by leveraging information from neighboring frames to generate masklets that span across an image sequence. For each object prompted in a given frame, SAM 2 propagates the mask trough a video using a streaming memory architecture.

D. VFMs for 3D Point Cloud Segmentation

Following the success of vision foundation models on image and video data, there has been active research [22], [23], [24], [25] on applying these models to 3D point cloud segmentation. In contrast to our work, these studies focus solely on single point clouds rather than continuous point cloud sequences. For instance, [22], [23] take multiple views of scenes as input, segment them using SAM [22] and then fuse the labels in 3D space. On the other hand, [24], [25] operate directly on the 3D data and also support prompting within the 3D space. Generally, selecting points or bounding boxes as prompts in 3D is a more challenging task for human annotators compared to our approach, which only requires 2D image-based prompts.

III. SEMI-AUTOMATIC LiDAR SEQUENCE LABELING

We propose a novel labeling pipeline that enables fast and efficient generation of annotation masks for custom objects in 3D point cloud sequences, requiring only sparse supervision from human annotators. To achieve this, we initially transform the point cloud sequence into a novel video-based proxy representation. This representation effectively compresses sequential 3D point cloud data into a compact video format while preserving essential object shapes and structural information, which are crucial for accurate segmentation. This transformation also allows us to leverage a vision foundation model for video segmentation (SAM 2 [4]) to perform object segmentation on point cloud data. In addition to enabling automated segmentation, the proxy representation also improves interpretability for human annotators. By following a simple click-to-annotate principle, annotators place point prompts on objects in individual video frames. SAM 2 then generates masks for the selected frames and subsequently propagates the masks across all frames in which the object appears. Our pipeline consists of three stages: (1) projection of point cloud sequences to our video proxy representation, (2) video annotation, and (3) re-projection of annotated data back to 3D. We also include data pre-processing prior to our data projection, as well as, mask post-processing in 3D to ensure high-quality annotations. Figure 2 shows the workflow of our pipeline: the top row illustrates the projection to the proxy representation, while the bottom row outlines the video annotation workflow and re-projection into 3D masks.

A. Projection of Point Cloud Sequences to Video Proxy Representation

To project 3D point cloud data from classical rotating LiDAR scanners into 2D panoramic images, the image coordinates $\text{image}_{\text{x,y}}$ for a given point $\mathbf{p}$ with 3D coordinates $p_{\text{x,y,z}}$ can generally be computed as follows [20]:

$$\text{range}_p = \sqrt{p_x^2 + p_y^2 + p_z^2} \tag{1}$$

$$\theta_p = \arctan\left(\frac{p_x}{p_y}\right) \tag{2}$$

$$\phi_p = \arcsin\left(\frac{p_z}{\text{range}_p}\right) \tag{3}$$

$$\text{image}_x = \left\lfloor \frac{\theta_p - h_{\text{fov,0}}}{h_{\text{fov,1}} - h_{\text{fov,0}}} \cdot h_{\text{res}} \right\rfloor \tag{4}$$

$$\text{image}_y = \left\lfloor \frac{\phi_p - v_{\text{fov,0}}}{v_{\text{fov,1}} - v_{\text{fov,0}}} \cdot v_{\text{res}} \right\rfloor. \tag{5}$$

h_{res} and v_{res} denote the horizontal and vertical resolution of the output image. h_{fov} and v_{fov} are the field-of-view of the LiDAR scanner. This process can be simplified by setting the vertical image resolution equal to the number of LiDAR scan lines and the horizontal resolution to the number of angular

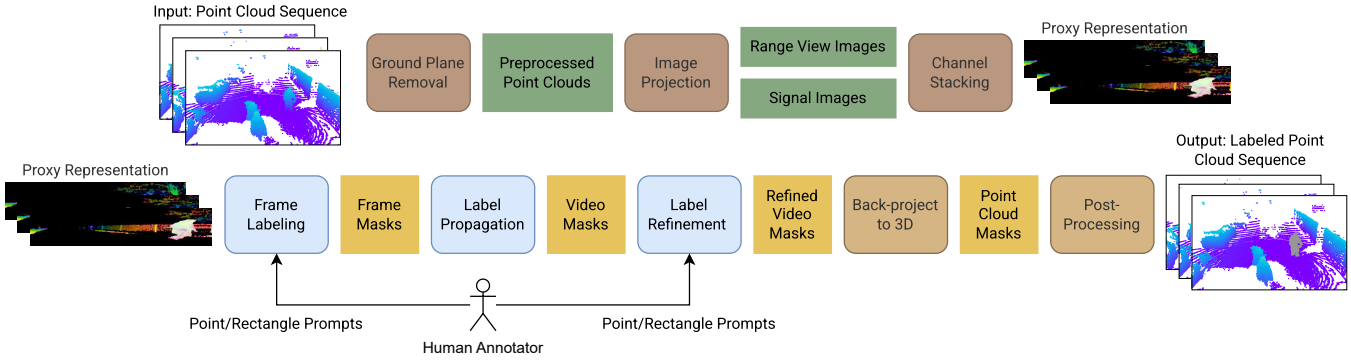


Fig. 2: Overview of our labeling pipeline which takes point cloud sequences as input and provides 3D masks as output. First, the point clouds are converted to video proxy representation using a custom a transformation for range and signal data. The videos are then labeled using SAM 2 [4], taking advantage of its label propagation feature across multiple frames. The video masks are then projected back to 3D, followed by post-processing to refine the masks.

sampling steps of the LiDAR. In this configuration, each LiDAR measurement corresponds to exactly one pixel in the image representation, resulting in a dense and unambiguous projection without interpolation.

Range view representations [26], [18], [27], [28] are commonly used to preserve depth information in lower-dimensional image data. We propose a custom range view transformation that enhances local contrast, providing better visual distinction between different objects. As a preprocessing step, we clip each depth value d to a dataset-specific maximum value d_{max}. To increase local contrast and encode higher-frequency details, we apply a sinusoidal transformation to the normalized depth values. We use two color channels to improve the overall contrast and ensure a more balanced representation of different intensity levels across the image. For a given depth value d_i, the modulated color values c_i^1 and c_i^2 are computed as follows:

$$d_i' = \min(d_i, d_{\text{max}}) \tag{6}$$

$$d_{i,\text{norm}} = \frac{d_i'}{\max(d_1', d_2', \ldots, d_n')} \tag{7}$$

$$d_i^* = 2\pi \cdot p \cdot d_{i,\text{norm}} \tag{8}$$

$$\mathrm{c}_i^1 = \frac{\sin(d_i^*) + 1}{2} \tag{9}$$

$$\mathrm{c}_i^2 = \frac{\cos(d_i^*) + 1}{2}. \tag{10}$$

Compared to linear mapping, sinusoidal transformation increases contrast between neighboring depth values, which is particularly beneficial to provide a clear visualization of object boundaries. The hyperparameter p controls the number of sinusoidal cycles, and thus determines the spatial frequency encoded in the resulting image.

Modern 3D LiDAR scanners commonly also provide ambient images that capture reflectivity information. We incorporate the ambient data as an additional input modality by stacking it alongside the encoded depth channels. To transform the signal strength data to image intensity value range we use a logarithmic transfer function. LiDAR signal images are characterized by a large number of low-intensity values and a few high-value outliers from highly reflective objects (see Figure 4 for details). Applying a logarithmic transfer function increases contrast in the low-value range while compressing the high-value outliers. This increased contrast improves the visual distinction between different objects. For a given signal image value a_i, the transformed color value c_i^3 is computed as follows:

$$\mathrm{c}_i^3 = \frac{\log(a_i + 1)}{\log(\max(a_1, a_2, \ldots, a_n) + 1)} \cdot 255. \tag{11}$$

Our proxy representation images are constructed by channel-wise stacking of the processed depth and ambient information, *i.e.,* $(c1, c2, 3)$ gives a color tuple. We apply channel-wise median filtering to the projected images to suppress noise in both the depth and reflectivity channels, producing clean and consistent inputs. Our image formulation process results in images with surreal colors and textures; however, it is specifically designed to emphasize depth discontinuities and variations in material properties. As a result, our proxy representation images offer a clear visual depiction of object shapes, which is essential for generating accurate segmentation masks (see Section IV for examples).

To further improve our proxy representation, we remove points belonging to the ground plane as a preprocessing step. This is particularly important in range data, where distinguishing object points from the ground near their base is often difficult. Explicit ground removal reduces ambiguity in these regions and leads to more precise segmentation masks. Depending on the data domain, ground plane removal can be done using simple approaches like height thresholding or RANSAC-based [30] plane estimation, more sophisticated methods like Patchwork [31] or Patchwork++ [32], or even deep-learning based methods [33].

B. Video Annotation: Prompting and Label Propagation

We use a custom graphical user interface (GUI) to interact with the Segment Anything 2 (SAM 2) [4] model for segmenting objects in our data. Our annotation framework supports positive and negative point prompts as well as positive bounding box prompts. After all objects in a frame are annotated with initial prompts, we perform a SAM 2 [4] inference pass to generate segmentation masks. Subsequently,

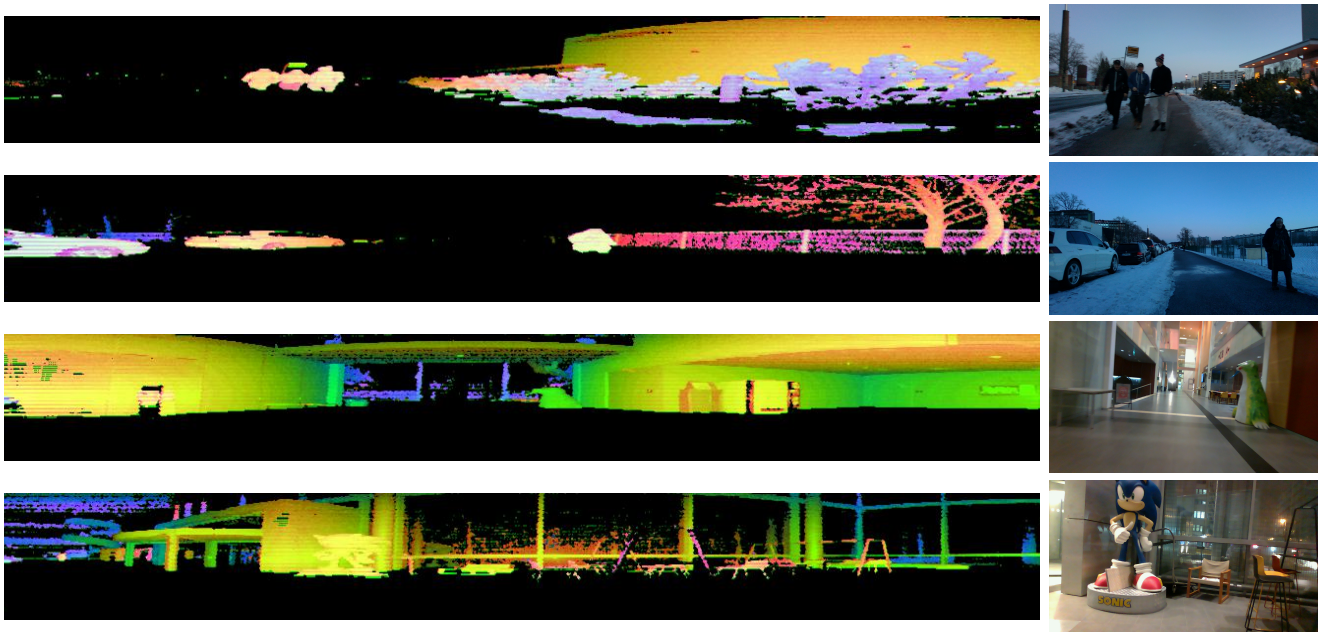


Fig. 3: Example point clouds from the TIERS dataset [29] projected to our proxy representation. The panoramic images offer a clear visual distinction between different objects, which is essential for generating accurate annotation masks. For easier interpretation, we also include corresponding camera images (field-of-view covers only a part of the LiDAR scan). We crop the images to better focus on the relevant regions. Best viewed in color and zoomed in on a digital device.

the masks can be refined using additional prompts until they meet the desired quality. Our tool allows annotators to provide prompts for multiple objects within the same frame. It also supports the addition of categorical information per mask, which makes it suitable for both semantic and instance segmentation. This initial prompting step is repeated for each frame where a new object first appears in the sequence. Once at least one mask is available for all objects in the current sequence, SAM 2 propagates them across the entire sequence. The frame rate of the point cloud sequence plays a crucial role in this process, as it directly affects both the quality of mask propagation and the overall efficiency of the labeling pipeline. If the frame rate is low, SAM 2 cannot reliably propagate object masks between consecutive frames due to larger object displacements. Higher frame rates might inflate dataset size with limited benefit for downstream tasks. In practice, we found frame rates between 5 and 10 Hz well suitable for our pipeline. Finally, the propagated masks can be refined further through targeted prompting to correct any inaccuracies introduced during propagation.

C. Re-Projection to 3D and Post-Processing

We can directly back-project the 2D masks to the corresponding points in 3D space, by storing the point-to-pixel correspondences during the projection step (see Sec. III-A), To further enhance label quality, we apply cluster-based post-processing in 3D. If a 2D mask slightly exceeds object boundaries, it may result in false positive labels for background points that lie behind the object. This phenomenon visually resembles a shadow-like outline cast by the LiDAR rays. To remove them, we discard small, isolated labeled regions. Specifically, we apply DBSCAN [34] clustering to the 3D mask points. The clustering thresholds, including the distance threshold parameter ϵ and the minimum cluster size, must be selected based on object sizes and point cloud density. Furthermore, we employ k-nearest neighbors [35] to complete small gaps in the segmentation masks.

IV. Labeling Pipeline Demonstration

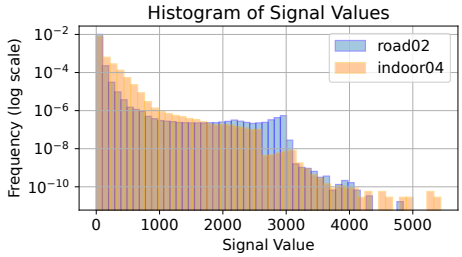
We showcase the workflow of our labeling pipeline using publicly available LiDAR data from the TIERS SLAM dataset [29]. The dataset features diverse indoor and outdoor scenarios and includes LiDAR data recorded by different sensor; for this demonstration we choose to use the data recorded with an *Ouster OS0*[1], a rotating 3D LiDAR scanner featuring 128 beams and a 90-degree vertical field of view.

A. Proxy Representation

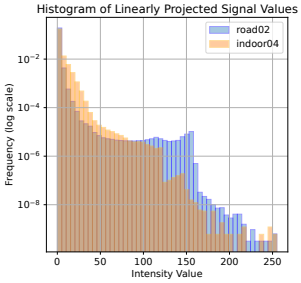
Figure 3 shows examples of frames taken from the TIERS dataset [29] represented in our custom proxy representation. It provides a clear depiction of individual objects preserving the object contours. Compared to the camera images, the shapes in LiDAR scans appear distorted due to differences in horizontal and vertical angular resolution.

Figure 4 shows the distribution of signal values for one indoor and one outdoor sequence from the dataset. The histograms in the bottom row compare a simple linear transformation of signal values to the image value range with our proposed logarithmic projection. It demonstrates that our custom transfer function makes better use of the image value range, yielding improved contrast across a wide

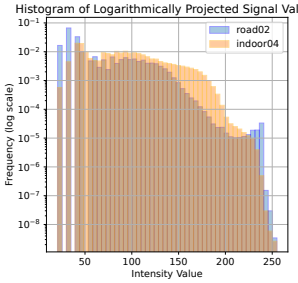
[1] https://ouster.com/products/hardware/os0-lidar-sensor



(a) Histogram for *Ouster OS0* LiDAR signal values for two sequences (one indoor and one outdoor scene) from the TIERS [29] dataset.



(b) Signal values projected to image intensity range using a linear projection.



(c) Signal values projected to image intensity range using a proposed logarithmic projection.

Fig. 4: Comparison of projecting raw LiDAR signal values to pixel intensity range using a standard linear transformation to our proposed logarithmic transfer function.

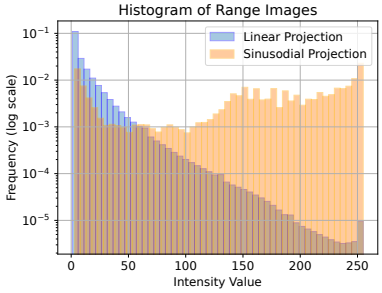


Fig. 5: Comparison of linear projection and our sinusoidal projection for mapping LiDAR range to image intensity.

range of signal values. This simplifies the annotator's task by making point prompt selection easier. Figure 5 shows the histograms for range images from the same sequences. We compare a standard (linear) range image representation with our sinusoidal projection. Again, our approach leads to significantly better utilization of the image intensity range, resulting in improved contrast for depth values. This is also supported by analyzing the standard deviation and the edges strength of the range images. In both cases our custom representation achieves values an order of magnitude higher, indicating substantially better image contrast, which is essential for accurate segmentation using SAM 2 [4].

B. Labeling Process and Post-Processing

We demonstrate the workflow for annotating objects using our interactive segmentation GUI in Figure 6. As an example, we annotate an uncommon object (a penguin statue) in a subset of 50 frames from the *indoor04* sequence in the TIERS [29] dataset. With only two user provided point prompts, we are able to generate highly accurate 3D segmentation masks on all frames. Initially, we place a point prompt on the main part of the statue, which results in a well-fitted mask, although a small part of the beak is missing. Adding an additional prompt, the beak gets included in the refined mask. After propagation across frames, the masks continue to fit the object well, both in 2D and 3D.

Figure 7 highlights the benefits of our 3D label post-processing. The colored points represent the initial 2D mask projected to 3D. During annotation, the 2D mask slightly exceeded the object boundaries, resulting in points behind the object being included in the 3D mask (turquoise). We apply DBSCAN clustering to filter out these small clusters of false positive points. The final segmentation mask (red points) is significantly more accurate after removing false positives through clustering.

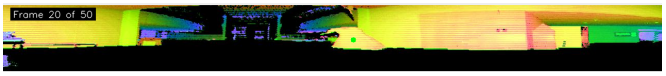
V. Real-World Application on a Custom Dataset

A. Dataset

To evaluate our labeling pipeline under real-world conditions, we conduct a proof of concept study by annotating a custom autonomous mobile robot dataset from the intralogistics domain. Special object types are very common in warehouses and production facilities. Due to the limited availability of public datasets in this domain, annotating custom data is often necessary. The dataset was recorded using an Ouster OS0 LiDAR, the same sensor model used in the example data from the TIERS dataset [29] (Sec. IV). Data was recorded at 10 frames per second, yielding a total of 28 hours of recordings, separated into 30-second sequences. The dataset captures an autonomous vehicle operating in a production facility and warehouse environment. The primary task is to detect various types of intralogistics vehicles, including forklifts, automated guided vehicles (AGVs), and order pickers. We manually filtered sequences that contain such vehicles, resulting in approximately 770 sequences – corresponding to around 231,000 frames/6.5 hours of data. For this proof-of-concept study, we randomly selected a subset of 250 sequences to demonstrate the capabilities of our labeling pipeline. To reduce redundancy and maintain a manageable data volume, we further subsampled the sequences to 5 frames per second, yielding a total of 37,500 raw frames in our proof-of-concept segmentation dataset.

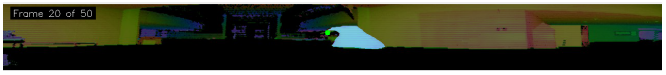
B. Labeling Process

Overall, it took approximately 15 work hours to label our dataset, which is significantly faster than fully manual segmentation. The generated masks are highly accurate across all vehicle types. In many cases, a single positive point prompt is sufficient to generate a precise mask. Additional prompts allow for refinement of fine-grained details: positive prompts can be used to recover small object parts that may be missing from the initial mask, while negative point prompts help shrink the mask to better align with the true object boundaries, for example, by excluding structural cut-outs. A typical case where we required negative prompts is when

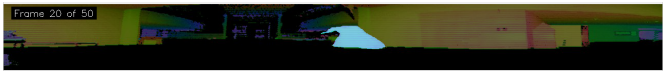


(a) Input frame taken from the TIERS [29] with a user defined point prompt (green dot) at the center of the penguin sculpture.

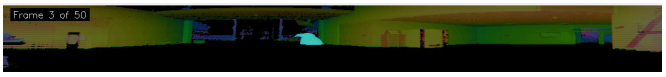
(b) Initial segmentation mask output yielded by SAM 2. The mask fits the body well, but a part of the beak is missing.



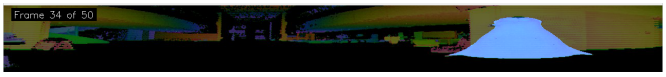
(c) Another point prompt is added to the tip of the beak.



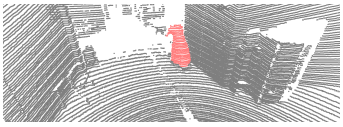
(d) Refined segmentation mask output yielded by SAM 2.



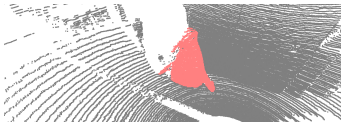
(e) Propagation of mask to an earlier sequence frame.



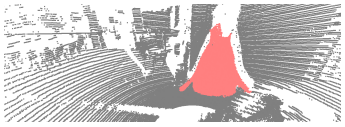
(f) Propagation of mask to a future sequence frame.



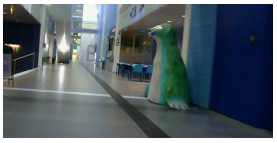
(g) Segmentation mask from (e) back-projected to the 3D points.



(h) Segmentation mask from (d) back-projected to the 3D points.



(i) Segmentation mask from (f) back-projected to the 3D points.



(j) Camera image showing the annotated object.

Fig. 6: Labeling of a custom, uncommon object (penguin statue) in the *indoor04* sequence from the TIERS [29] dataset. The top row shows the input frame with the user-generated point prompt, and the initial SAM 2 [4] mask prediction. The second and third rows illustrate the mask refinement and propagation process. The final row presents the post-processed masks back-projected to 3D, alongside a camera image of the statue. Best viewed in color and zoomed in on digital device.

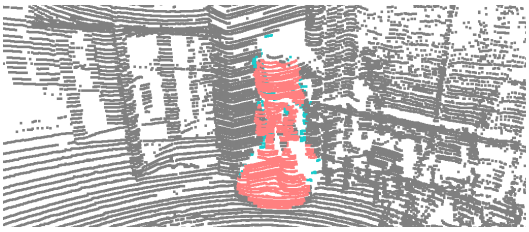


Fig. 7: All colored points belong to the initial 2D mask. Since the 2D SAM2 [4] mask slightly exceeds the comic figure statue in image space, small clusters of points on nearby walls are also included. We apply DBSCAN clustering to remove these outliers; the red points represent the final object mask after post-processing.

gaps – such as those between the bars of a forklift mast – are incorrectly filled in by the initial segmentation mask and must be removed. Some protruding vehicle parts, such as sensors mounted on top of a vehicle, are not always included in the object masks. These parts are rather small and have different material properties, making them hard to detect in our proxy representation, as their appearance strongly differs from that of main vehicle parts. However, these parts can be included in the final mask by applying explicit prompts, *i.e.,* an additional click, targeted at them. Also the label propagation works well on our custom data, and the masks are assigned to the correct objects. In some cases, when a vehicle drives in front of a rack, the object label does not stick with the moving vehicle but rather stays at the same image region. To mitigate this issue, the propagated labels are refined with additional negative and positive prompts.

After labeling, we obtained a dataset containing 18,000 non-empty frame masks. Visual inspection of the labeled masks in 3D confirms high precision, with recall mainly dependent on human supervision. As expected, masks for distant objects are slightly noisier and more prone to errors compared to those for objects closer to the ego-vehicle. In general, the quality of the labels correlates with the number of input prompts—additional prompts lead to more accurate segmentation. If the quality standards are not met for a given sequence, redoing the labeling process for that sequence with more input prompts from human supervision is generally sufficient to achieve satisfactory results.

C. Use of Pseudo-Labels in Downstream Task

To demonstrate the use of our labels in a downstream task, we train a SphereFormer [12] point cloud segmentation model on our data. We split the annotated frames into a train and validation set by randomly selecting 20 percent of the recorded sequences for validation. Compared to a random frame-based split, this method reduces information leakage between both sets. Additionally, we use 600 professionally annotated frames as a test set. We also created pseudo-labels for these frames, which achieve an Intersection over Union (IoU) of 0.925 compared to the fully manual labels, highlighting the high accuracy of our approach.

We use the default SphereFormer model architecture and only adapt optimization hyperparameters to our unbalanced two-class segmentation problem (vehicles and background). In addition, we adjust the point cloud range parameters to align with the properties of our dataset. To mitigate overfitting on our relatively small dataset, we train the model for only 10 epochs.

The SphereFormer model trained on our pseudo-labels achieves an IoU of 0.91 on the test set for vehicles closer than 5 meters, and 0.76 for vehicles in the 5 to 15-meter range. This demonstrates strong segmentation performance

and is further supported by manual inspection of the results. We observe that the model reliably predicts highly accurate masks for vehicles. However, we observe some false positives in the model output, where rack elements are incorrectly predicted as vehicles. These errors are mainly caused by label propagation failures in the training data, where human-prompted vehicle labels were mistakenly transferred to rack structures. This occurs because rack masts sometimes exhibit similar geometric structures to vehicle masts. Additionally, smaller object parts are occasionally missing from the predicted vehicle masks. Overall, these results are consistent with expectations for a baseline model trained on pseudo-labels and confirm the effectiveness of our approach for downstream segmentation tasks. Leveraging more sophisticated training strategies, such as weak supervision techniques like [13], presents a promising opportunity to further improve model training.

VI. CONCLUSIONS

We propose a new fast and cost efficient pipeline to generate 3D segmentation masks for point cloud data with minimal human intervention. By projecting 3D point cloud sequence data to a suitable video proxy representation, we can utilize the capabilities of vision foundation models for generating highly accurate 3D masks. We showcased the proposed pipeline on the public TIERS dataset [29] and successfully applied our approach on a custom dataset in the intralogistics domain. The resulting annotations are sufficiently accurate for practical use, requiring only minimal human supervision and incurring a fraction of the cost of professional annotation. These results confirm that our approach is well suited for efficiently annotating data in custom use-case scenarios.

REFERENCES

[1] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, *et al.*, "Simple Open-Vocabulary Object Detection," in *ECCV*, 2022.

[2] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick, "Segment Anything," in *CVPR*, 2023.

[3] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. Ni, and H.-Y. Shum, "DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection," in *ICLR*, 2023.

[4] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Raedle, C. Rolland, and L. Gustafson, "SAM 2: Segment Anything in Images and Videos," in *ICLR*, 2025.

[5] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "YOLO-World: Real-Time Open-Vocabulary Object Detection," in *CVPR*, 2024.

[6] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, *et al.*, "Grounding DINO: Marrying DINO with Grounded Pre-training for Open-Set Object Detection," in *ECCV*, 2024.

[7] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning Transferable Visual Models From Natural Language Supervision," in *ICML*, 2021.

[8] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "DINOv2: Learning Robust Visual Features without Supervision," *Trans. on Machine Learning Research*, 2023.

[9] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data," in *CVPR*, 2024.

[10] J. Videnovic, A. Lukezic, and M. Kristan, "A Distractor-Aware Memory for Visual Object Tracking with SAM2," in *CVPR*, 2025.

[11] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and Asymmetrical 3D Convolution Networks for LiDAR Segmentation," in *CVPR*, 2021.

[12] X. Lai, Y. Chen, F. Lu, J. Liu, and J. Jia, "Spherical Transformer for LiDAR-Based 3D Recognition," in *CVPR*, 2023.

[13] O. Unal, D. Dai, and L. Van Gool, "Scribble-Supervised LiDAR Semantic Segmentation," in *CVPR*, 2022.

[14] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *ICCV*, 2019.

[15] K. Viswanath, P. Jiang, and S. Saripalli, "Reflectivity Is All You Need!: Advancing LiDAR Semantic Segmentation," *arXiv preprint*, 2024.

[16] C. Liu, X. Qian, B. Huang, X. Qi, E. Lam, S.-C. Tan, and N. Wong, "Multimodal Transformer for Automatic 3D Annotation and Object Detection," in *ECCV*, 2022.

[17] H. Paat, Q. Lian, W. Yao, and T. Zhang, "MEDL-U: Uncertainty-aware 3D Automatic Annotation based on Evidential Deep Learning," in *ICRA*, 2024.

[18] B. Wu, A. Wan, X. Yue, and K. Keutzer, "Squeezeseg: Convolutional Neural Nets With Recurrent CRF for Real-Time Road-Object Segmentation From 3D LiDAR Point Cloud," in *ICRA*, 2018.

[19] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and Accurate LiDAR Semantic Segmentation," in *IROS*, 2019.

[20] L. Kong, Y. Liu, R. Chen, Y. Ma, X. Zhu, Y. Li, Y. Hou, Y. Qiao, and Z. Liu, "Rethinking Range View Representation for LiDAR Segmentation," in *CVPR*, 2023.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.

[22] J. Cen, Z. Zhou, J. Fang, W. Shen, L. Xie, D. Jiang, X. Zhang, Q. Tian, *et al.*, "Segment Anything in 3D With Nerfs," in *NeurIPS*, 2023.

[23] M. Xu, X. Yin, L. Qiu, Y. Liu, X. Tong, and X. Han, "SAMPro3D: Locating SAM Prompts in 3D for Zero-Shot Scene Segmentation," *arXiv preprint*, 2023.

[24] Y. Zhou, J. Gu, T. Y. Chiang, F. Xiang, and H. Su, "Point-SAM: Promptable 3D Segmentation Model for Point Clouds," *arXiv preprint*, 2024.

[25] Z. Guo, R. Zhang, X. Zhu, C. Tong, P. Gao, C. Li, and P.-A. Heng, "SAM2POINT: Segment Any 3D as Videos in Zero-Shot and Promptable Manners," *arXiv preprint*, 2024.

[26] Y. Wang, T. Shi, P. Yun, L. Tai, and M. Liu, "PointSeg: Real-Time Semantic Segmentation Based on 3D LiDAR Point Cloud," *arXiv preprint*, 2018.

[27] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer, "SqueezeSegV2: Improved Model Structure and Unsupervised Domain Adaptation for Road-Object Segmentation from a LiDAR Point Cloud," in *ICRA*, 2019.

[28] T.-H. Chen and T. S. Chang, "RangeSeg: Range-Aware Real Time Segmentation of 3D LiDAR Point Clouds," *IEEE Trans. on Intelligent Vehicles*, 2021.

[29] L. Qingqing, Y. Xianjia, J. P. Queralta, and T. Westerlund, "Multi-Modal LiDAR Dataset for Benchmarking General-Purpose Localization and Mapping Algorithms," in *IROS*, 2022.

[30] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, 1981.

[31] H. Lim, M. Oh, and H. Myung, "Patchwork: Concentric Zone-based Region-wise Ground Segmentation with Ground Likelihood Estimation Using a 3D LiDAR Sensor," *IEEE Robotics and Automation Letters*, 2021.

[32] S. Lee, H. Lim, and H. Myung, "Patchwork++: Fast and Robust Ground Segmentation Solving Partial Under-Segmentation Using 3D Point Cloud," in *IROS*, 2022.

[33] X. Wu, L. Jiang, P.-S. Wang, Z. Liu, X. Liu, Y. Qiao, W. Ouyang, T. He, and H. Zhao, "Point Transformer V3: Simpler Faster Stronger," in *CVPR*, 2024.

[34] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases With Noise," in *KDD*, 1996.

[35] T. Cover and P. Hart, "Nearest Neighbor Pattern Classification," *IEEE Trans. on Information Theory*, 1967.