2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Remove, then Revert: Static Point cloud Map Construction using Multiresolution Range Images

Giseop Kim[1] and Ayoung Kim[1*]

***Abstract*— We present a novel static point cloud map construction algorithm, called *Removert*, for use within dynamic urban environments. Leaving only static points and excluding dynamic objects is a critical problem in various robust robot missions in changing outdoors, and the procedure commonly contains comparing a query to the noisy map that has dynamic points. In doing so, however, the estimated discrepancies between a query scan and the noisy map tend to possess errors due to imperfect pose estimation, which degrades the static map quality. To tackle the problem, we propose a multiresolution range image-based false prediction reverting algorithm. We first conservatively retain definite static points and iteratively recover more uncertain static points by enlarging the query-to-map association window size, which implicitly compensates the LiDAR motion or registration errors. We validate our method on the KITTI dataset using SemanticKITTI as ground truth, and show our method qualitatively competes or outperforms the human-labeled data (SemanticKITTI) in ambiguous regions.**

I. Introduction

Recent advances in 3D light detection and ranging (LiDAR) mapping [1, 2] have been reported leveraging LiDAR odometry [3, 4], place recognition [5, 6], and simultaneous localization and mapping (SLAM) [7, 8]. The main SLAM strategy relies on static objects while rejecting dynamic objects as outliers to avoid confusion and achieve robustness. This research focus is also found in localization against a prior map [9] and in map maintenance and updates.

The existence of dynamic objects (e.g., moving cars and pedestrians) diversifies the structural appearance of the real-world space; hence, discrimination between dynamic objects and reliable static objects is not straightforward. This difficulty yields misclassified 3D points lingering in a constructed map (e.g., top of Fig. 1). These false static points in the map may deteriorate the localization [9, 10, 11] and map maintenance performance. Overcoming this challenge, two types of approaches are feasible. First, on the query sensor side, structural-variance-robust place representation [10] and multiple experience-based [12] methods have been reported, particularly targeting the localization problem. Another approach aims at building a clean prior map removing potentially error pruning dynamic objects from the map. This paper shares the same philosophy as the latter and focuses on building a 3D point cloud map containing only the static components in the environment. Our solution is to determine

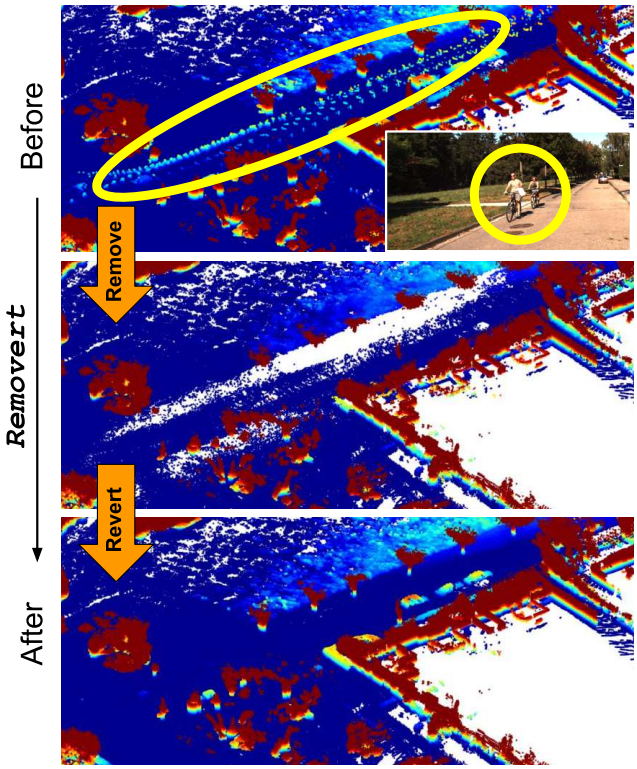

Fig. 1: A before-and-after of the proposed method for the KITTI dataset [13] sequence `08` around frame 84. Our method can remove dynamic points representing urban moving objects that have arbitrary shapes or locations without any assumptions while preserving static points.

a static-only meta-representation of a place that can support various applications on the query side.

Static map construction and dynamic object removal are chicken-and-egg problems due to their interwoven nature. Given a perfectly static map, discerning dynamic points is straightforward; if we perfectly distinguish dynamic objects, then removing them from incoming sensor data readily constructs a clean map. However, in reality, none of them is known perfectly. Existing studies [14, 15] have focused on an iterative state (i.e., static or dynamic) update of map points by sequentially fusing multiple measurements and checking visibility between a query scan and a map. This visibility-based approach needs to find an association between a single point and a map point. However, because the estimated pose of LiDAR motion contains an error, this association may be inaccurate and the possibility of deleting the wrong points exists. Nevertheless, existing methods assume correct poses

[1]G. Kim and A. Kim are with the Department of Civil and Environmental Engineering, KAIST, Daejeon, S. Korea [paulgkim, ayoungk]@kaist.ac.kr

This work was supported by [Localization in changing city] project funded by Naver Labs Corporation and by the National Research Foundation of Korea (NRF) grant (NRF-2019K2A9A1A06070173).

or correct registrations are given [15] or use a fixed size association rule (e.g., within 1°) [14].

In this paper, we propose two mechanisms to solve the false dynamic point removal problem caused by the association errors resulting from the ambiguity of LiDAR motions. The **first** mechanism is a *removing-and-reverting* algorithm. We first conservatively retain static points (see the middle of Fig. 1) and revert falsely removed points. This procedure processes a batch of measurements and, unlike sequential Bayesian updates of [14], focuses on the postprocessing role because our main purpose is to construct a certain length of static map and to ensure that dynamic points are strongly erased even if some false negatives occur (i.e., actual static points are wrongly erased). Therefore, our algorithm first removes points (except for solidly static points) in batches, and then reverses them with multiple confidence levels (details in Fig. 4 and Fig. 9). **Second**, we provide a mechanism to handle reverting confidences using multiresolution range images. This has an effect of the implicit association between a query and a map point within multiple sizes of search windows, thus a user can implicitly specify an acceptable level of LiDAR motion ambiguity and adjust the trade-off (e.g., between precision and recall) of the predicted static map's accuracy.

Because of the combination of the aforementioned two approaches, we call our method "Remove-then-revert (*Removert*)". Our contributions are threefold:

- Range image-based map point discrepancy calculation. (§III-B and §III-C)
- A novel remove-then-revert mechanism to construct and enhance a static map (§III-D).
- Multiresolution range image-based static map evolution to deal with LiDAR motion ambiguities (§III-D).

II. Related Work

A. Dynamic object removal

As mentioned earlier in §I, static map construction is closely related to dynamic removal. Between online and offline approaches, we exclude online in this paper because we are mainly interested in producing a high-quality static map without concerning the processing speed.

Conventional approaches. The remote-sensing community has widely investigated dynamic point removal as building up a pure environment's structures is important to the construction or understanding of the environment. In general, however, it requires high-cost, dense terrestrial laser scanning (TLS) point cloud data with accurately aligned poses to apply in time-consuming voxel ray casting-based methods [16, 17, 18].

Visibility-based approaches. To alleviate the computational burden, visibility-based methods have been proposed [19, 15, 14]. This type of method associates a query point (or a cluster) and a map point within an almost same narrow field of view (FOV) (e.g., cone-shaped [14]), then checks which is located further away and concludes the occluded points at further sites should be static.

Point cloud segmentation-based approaches. Related topics also include segmentation-based approaches. Point cloud segmentation tools have been developed [20, 21]. With correct segment points with dynamic labels, constructing a map is straightforward by excluding them [22, 23]. However, segmentation-based approaches currently rely heavily on the supervised labels and are vulnerable to human error or unknown classes [24] without considering the scan-to-map relationship. This semantic label of a point needs to be fused as a prior with the visibility-based method.

Solving motion ambiguities. To detect a discrepancy between a query and a map, we need to register the query scan correctly to the map. However, perfect registration is difficult to sustain in the real outdoor environment. Pomerleau et al. [14] modeled registration uncertainties using normal, incidence angle, and an overlap ratio from a conic aperture. However, the model could be incorrect and disturb the intended weights in case of a misregistration of scan. Yoon et al. [25] proposed to undistort a LiDAR scan for dynamic object detection, whereas our method can handle the registration error without the undistortion via implicit associations using multiscaled range images (Fig. 3 and §III-D).

B. 3D change detection and map update

The key concept of 3D change detection is basically similar to dynamic object detection; only the level of dynamicity varies (e.g., dynamic object, nonstationary static object, and stationary static object in [26]). Change detection is followed by map update [15, 14] to construct the static map. It is important that the proposed static map construction method needs to be preceded toward long-term map update.

III. Methodology

Unlike existing studies, we focus on building a static map in two phases rather than elaborately segmenting dynamic points for each scan. Toward this objective, we examine batch data of a certain length as in [27] to enhance the quality of the static map.

A. Problem definition

Setup and notations. Given a point cloud map constructed using a set of raw LiDAR scans, we aim to remove dynamic points within the map. The proposed method works as post-processing and can generally be used to detect pointwise long-term (permanent) changes. In doing so, we mainly consider two different coordinates, global unified map coordinate M and a local sensor query's frame Q. We use $\mathcal{P}^Q$ for a single scan of the query session and $\mathcal{P}^M$ for a submap within the global frame.

We assume the associated SE(3) pose $\mathbf{T}_k^Q$ for a $\mathcal{P}_k^Q$ (k is the index of a frame) is known using SLAM such as [3, 5, 8], but with some level of estimation error. Then, we select a query point cloud $\mathcal{P}_k^Q$ from the query set $\left\{\mathcal{P}_k^Q\right\}$ and compare it against $\mathcal{P}^M$ to remove for dynamic objects from map points.

During this comparison, we divide the target map into two mutually exclusive subsets: a set of static points $\mathcal{P}^{SM}$ and a

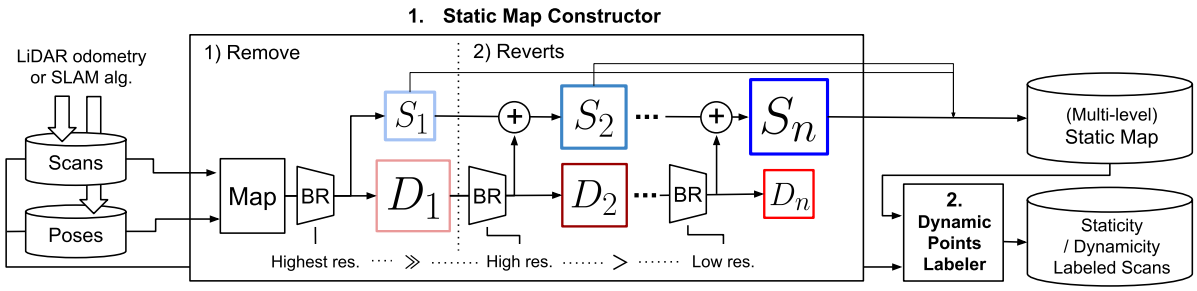

Fig. 2: The pipeline of the proposed static map construction method. The capital letters S_k and D_k represent a point cloud mainly composed of static and dynamic points, at kth iteration. The size of the box enveloping S_k or D_k represents the size of each point cloud (i.e., the number of points), and the color of the box represents its true prediction ratio within a point cloud; the proposed *Removert* algorithm iteratively enhances a central static map by applying fine to coarse resolutions of range images. A BR module is the abbreviation of BATCHREMOVAL (see §III-C and Algorithm. 1). As our by-product, we can self-label pointwise dynamicity and automatically parse dynamic objects for each scan using the constructed global static map (examples in the right bottom of Fig. 5).

set of dynamic points $\mathcal{P}^{DM}$. Formally, the aforementioned problem is expressed as

$$\mathcal{P}^{M} \quad = \quad \mathcal{P}^{SM} \cup \mathcal{P}^{DM} \quad , \tag{1}$$

while $\mathcal{P}^{SM} \cap \mathcal{P}^{DM} = \varnothing$.

Initial segregation leverages conventional segmentation methods to estimate $\hat{\mathcal{P}}^{SM}$ and $\hat{\mathcal{P}}^{DM}$ which inevitably contain status prediction errors. In this paper, we refer to the static status as *positive* (P) and the dynamic status as *negative* (N). Then the estimates $\hat{\mathcal{P}}^{SM}$ and $\hat{\mathcal{P}}^{DM}$ are expressed as

$$\hat{\mathcal{P}}^{SM} = \text{TP} \cup \text{FP} \quad \text{and} \quad \hat{\mathcal{P}}^{DM} = \text{TN} \cup \text{FN} \quad , \tag{2}$$

where TP, FP, TN, and FN represent true positive, false positive, true negative, and false negative point sets, respectively. Using this equation, we can redefine the problem as reducing the number of FP and FN points within the static and dynamic estimates.

Indeed, from the fact that the FN is equal to a subset of whole TP points if we assume a point always belongs to the static or dynamic set, our goal can be understood as detecting FN points from $\hat{\mathcal{P}}^{DM}$ and moving them to $\hat{\mathcal{P}}^{SM}$. Our algorithm performs this process repeatedly and augments the predicted static map so that it converges closer to the true static map. In short, this iterative static map enhancing strategy is the main idea behind our "remove-then-revert (*Removert*)" algorithm. The outline is given in Fig. 2 and the details will be provided in §III-D.

B. Range image-based map comparison

Sharing the analogous philosophy to existing approaches [14, 15], we perform dynamic point discrimination using *visibility* rather using voxel ray tracing. Unlike existing methods, we check the visibility of a map point within a projected range image plane. As in Fig. 3, the matrix operations while varying the range image's resolution are straightforward ways to detect discrepancies and to relieve the map point association ambiguities (see §III-D for details). For the visual domain, a similar work used an image-format for 3D change detection [27]. We also provide a range image-based visibility check method for a LiDAR point cloud, and extend to a multiresolution version for better static map prediction by alleviating LiDAR motion ambiguities.

For dynamic object removal, we project the large size map (i.e., $|\mathcal{P}^{M}| > |\mathcal{P}^{Q}_{k}|$) into a fixed-size range image using the relative transformation between the query's kth frame and the map. During the projection, we use a range image of particular resolution (e.g., a single pixel represents 1° for both horizontal and vertical FOV). We call this FOV-restricted and sampled map point cloud a "visible map point cloud". The visible map point cloud $\mathcal{P}^{M}_{k}$ is defined as its equivalent range image $I^{M}_{k} = (I^{M}_{k,ij}) \in \mathbb{R}^{m \times n}$, such that

$$I^{M}_{k,ij} \quad = \quad \min_{\mathbf{p} \in \mathcal{P}^{M}_{ij}} r(\mathbf{p}) \quad , \tag{3}$$

$$\mathcal{P}^{M}_{k} \quad = \quad \{\mathbf{p}^{M}_{k,ij} | \ \mathbf{p}^{M}_{k,ij} = \underset{\mathbf{p} \in \mathcal{P}^{M}_{ij}}{\operatorname{argmin}} \ r(\mathbf{p})\} \quad , \tag{4}$$

where the number of pixels m and n are determined by the given particular resolution per pixel and horizontal and vertical $[\min, \max]$ FOV ranges. $r(\cdot)$ represents a range of a point $\mathbf{p} \in \mathbb{R}^3$ within the query scan's local coordinate. $I^{M}_{k,ij}$ is a pixel (i, j) value of the range image. $\mathcal{P}^{M}_{ij}$ is a subset of $\mathcal{P}^{M}$ such that points with spherical coordinates (i.e., elevation and azimuth angle) reside in the pixel (i, j).

In the same way, the query scan's range image I^{Q}_{k} and its associated point cloud $\mathcal{P}^{Q}_{k}$ are acquired. Then, the visibility of map points is calculated via their matrix element-wise subtraction as

$$I^{\text{Diff}}_{k} = I^{Q}_{k} - I^{M}_{k} \quad , \tag{5}$$

and we assign a map point $\mathbf{p}^{M}_{k,ij} as \in \mathcal{P}^{M}_{k}$ as dynamic if its corresponding pixel value of I^{Diff}_{k} is larger than a certain threshold τ_D. Finally, the dynamic map points are defined as

$$\mathcal{P}^{DM}_{k} = \{\mathbf{p}^{M}_{k,ij} | \text{ its associated } I^{\text{Diff}}_{k,ij} > \tau_D\} \quad , \tag{6}$$

and the static map points are defined as the complementary set of the dynamic map points as $\mathcal{P}^{SM}_{k} = \mathcal{P}^{M}_{k} - \mathcal{P}^{DM}_{k}$. We

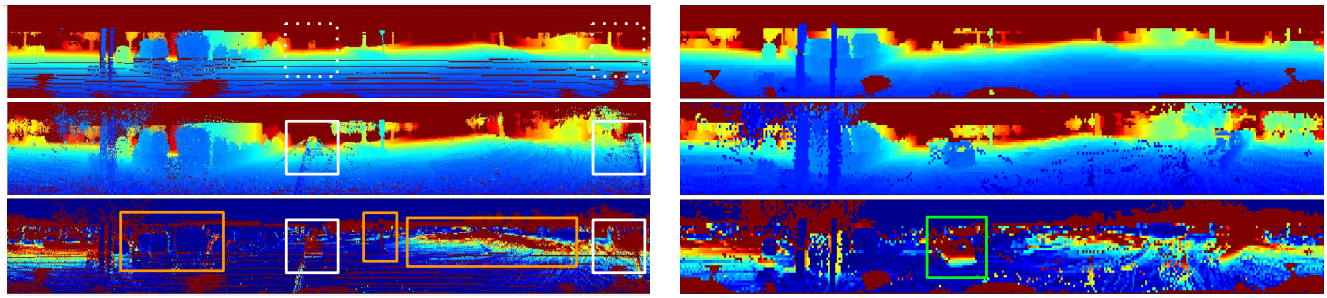

Fig. 3: Left column: high (fine) resolution range image (0.4° for a pixel). Right column: low (coarse) resolution range image (1° for a pixel). From top to bottom, each row represents I_k^Q, I_k^M, and I_k^{Diff}, respectively. The color map shows the distance (range) of a pixel's 3D point with respect to a kth sensor frame; blue is closer and red is further away. Thus, the red pixels in the bottom range images (i.e., I_k^{Diff}) represent a high discrepancy between a query scan and a map. Therefore, the map point at that pixel should be marked as dynamic (see white boxes at the left column). Using the fine resolution range image (left) removes a number of ambiguous points at object boundaries or grounds (orange boxes at left columns); these are actually static. They are first removed and solidly certain static points are retained during the initial iteration. At the coarser step, the aforementioned misclassified dynamic (actually static) information could be recovered. However, points near the ground tend to be incorrectly recovered if we use too coarse a resolution of a range image (see the bottom of the car in the green box in the right bottom range image).

additionally propose to use an adaptive threshold to modulate the sensitivity of our algorithm with respect to the point's distance; the adaptive threshold is defined as a function of the range: $\tau_{D_{\text{ada}}} = r(\mathbf{p}_{ij}^M)\tau_D$.

C. Batch dynamic point removal

Our main problem involves complete removal of dynamic points rather than in-flight processing. In this section, we present a batch version of §III-B to carefully update from multiple scan voting. Doing so allows one to overcome the map point occlusion and FOV limitation in a single scan. The module in this section (§III-C) is called BATCHREMOVAL in our algorithm description of Algorithm. 1.

We assume a set of sequential N scans along robot motions is given in advance. For each scan $\mathcal{P}_k^Q$, where $k = 1, ..., N$, we perform the range image-based dynamic map point detection as described in §III-B and *count* the total number marked as SM or DM for every single point in the map; n_{SM} and n_{DM}. Then, the dynamic map $\mathcal{P}^{DM} \subset \mathcal{P}^M$ is defined using a staticity score $s(\cdot)$, which is a weighted function of n_{SM} and n_{DM}, and the static map is naturally defined as the complement of $\mathcal{P}^{DM}$:

$$\mathcal{P}^{DM} = \{\mathbf{p}^M |\ s(\mathbf{p}^M) < \tau_S\} \quad , \tag{7}$$

$$\mathcal{P}^{SM} = \mathcal{P}^M - \mathcal{P}^{DM} \quad . \tag{8}$$

Here, the staticity score of a single 3D point $s(\cdot) := \alpha_{SM} n_{SM}(\mathbf{p}^M) + \alpha_{DM} n_{DM}(\mathbf{p}^M)$ and α_{SM} and α_{DM} are a positive and a negative weight coefficient, respectively. We note that some uncounted points (i.e., never marked during the batch) are considered static by default.

Furthermore, the comparison and removal should be independent of vehicle motion. For example, a car in front that moves at the same speed may hardly be removed if we sample query scans in the same order as the motion sequence. During the comparison, we conduct the batch while randomly selecting a query scan $\mathcal{P}_k^Q$.

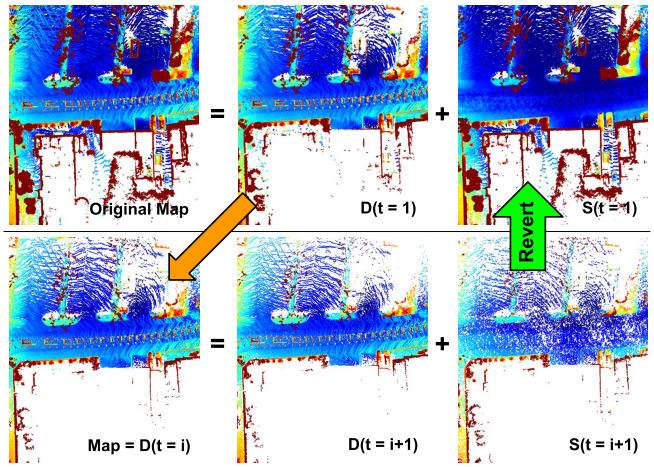

Fig. 4: A visualization of the proposed remove-then-revert algorithm described in Fig. 2 and Algorithm. 1. The colormap shows point's height for visual clarity (red is higher) and we note that all figures showing point cloud maps follow these color codes.

D. Removert: remove, then revert static points

The dynamic map separation in §III-C could be used as it is with a single fixed-size resolution range image. Yet, we found additional refinement is required because the estimated dynamic points may include static points (top row of Fig. 4). As reported in [14], for example, ground points are ambiguous because their range widely varies even when the points' incidence angle differences are small; thus, a static ground point could easily be mismarked as dynamic. To resolve this issue, authors [14, 25] have utilized additional normal, incidence angle of a point, or region growing. Differing from prior works, our method only requires points' 3D locations (x, y, and z). Furthermore, the existing approaches partially solved the issue; if the original robot motion was inaccurate, then the query point and map point could be wrongly associated. The

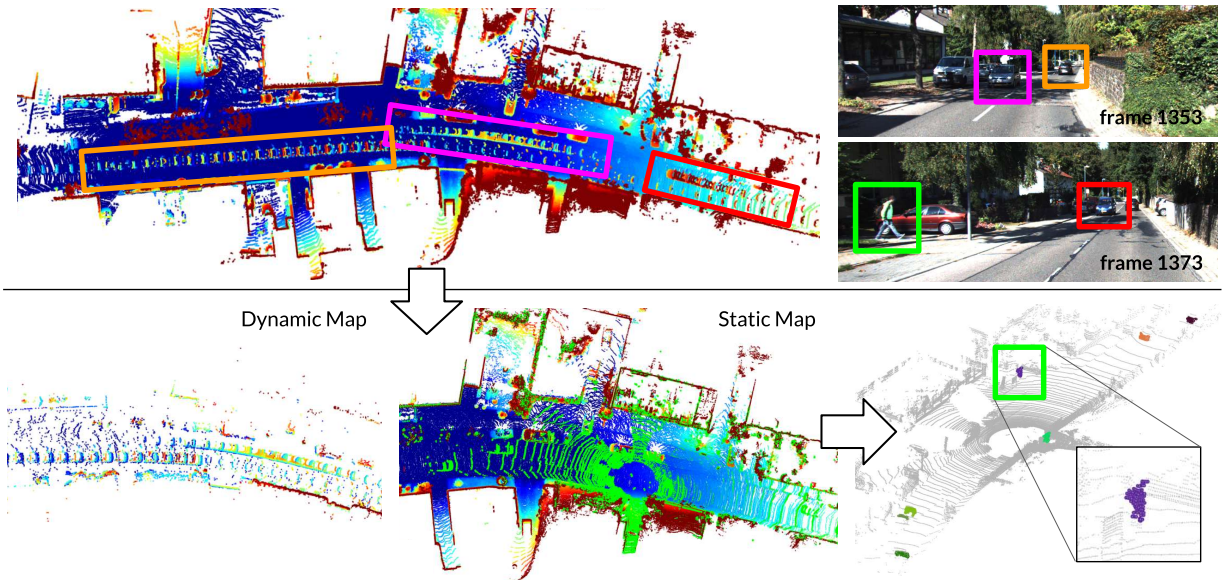

Fig. 5: A visualization of the product (i.e., static/dynamic map segregation) of *Removert* on the KITTI dataset [13] sequence `09` (top view). The top point cloud map is constructed using scans and SE(3) poses estimated using SuMa [8] for every 2 m and has many dynamic points from various types of instances (e.g., cars and a pedestrian). Our proposed method successfully separates dynamic and static maps from the original noisy point cloud. As a possible use-case of our static map, we can automatically self-parse the dynamic objects within a scan (green point cloud) just by checking whether a query point in a scan has a nearest point in a map, without elaborate ground point removal, normal estimation [14], or region growing [25].

same object may be located at the other pixels in the range image, causing false predictions (e.g., removing false static or retaining false dynamic points). We found false estimation frequently occurs with narrow objects (e.g., poles) and at the boundary of an object (e.g., tops of cars and trees; see the orange boxes of the left bottom image in Fig. 3).

Algorithm 1 REMOVERT: Remove-then-revert algorithm.

inputs (from LiDAR odometry or SLAM):
  Query scans and their poses: $\mathcal{S}^Q = \{\mathcal{P}^Q\}$, $\mathcal{T}^Q$
outputs:
  Static/dynamic map points: $\hat{\mathcal{P}}^{SM}$, $\hat{\mathcal{P}}^{DM}$
procedure REMOVERT:
  __ Initialization __
  $\mathcal{P}^M$ = MAKESUBMAP($\mathcal{S}^Q$, $\mathcal{T}^Q$)
  $[\mathcal{P}^{SM}_{t=0}, \mathcal{P}^{DM}_{t=0}]$ = BR($\mathcal{P}^M$, $\mathcal{S}^Q$, $\mathcal{T}^Q$, r_0)
    // BR means BATCHREMOVAL in §III-C
    // r_0 is an initial resolution (the finest)
  $\hat{\mathcal{P}}^{SM} \leftarrow \mathcal{P}^{SM}_{t=0}$
  $\mathcal{P}^M_{t=0} \leftarrow \mathcal{P}^{DM}_{t=0}$
  __ Main __
  `rimgResList` = $[r_1, ...r_{n_r}]$ // finer to coarser
  for $[i,\ r_i]$ in enumerate(`rimgResList`) **do**
    $[\mathcal{P}^{SM}_{t=i}, \mathcal{P}^{DM}_{t=i}]$ = BR($\mathcal{P}^M_{t=i}$, $\mathcal{S}^Q$, $\mathcal{T}^Q$, r_i)
    $\hat{\mathcal{P}}^{SM} \leftarrow \hat{\mathcal{P}}^{SM} + \mathcal{P}^{SM}_{t=i}$
    $\mathcal{P}^M_{t=i+1} \leftarrow \mathcal{P}^{DM}_{t=i}$
  end for
  $\hat{\mathcal{P}}^{DM} = \mathcal{P}^M - \hat{\mathcal{P}}^{SM}$
end procedure

To solve the discrepancy ambiguities in the query to map point association, Palazzolo and Stachniss proposed a window-based approach (i.e., not pixel-to-pixel, but pixel-to-window comparison) [27]. Similar to theirs, but more relaxed, we propose using multiple range images having different resolutions. It has the effect of implicitly associating a query point to a map point with multiple levels of window size without requiring the nearest point be found within a fixed window size.

Fig. 3 shows the different characteristics of two (high and low) resolutions. If we use a coarser pixel resolution, then finding correspondence between a query point and a map point is eased. A point marked as dynamic in the past iteration could be marked as static at the coarser resolution. The finest resolution we propose to use is $r_0 = \frac{\text{vertical FOV}}{\text{\# rays}}$ in Algorithm. 1. We revert its status and merge the point into the static map as in Fig. 4. This reverting algorithm iteratively reduces the number of FN points. During the revert, although the number of FP points slightly increased, we can see the correctly reverted points are dominant in Fig. 9.

IV. EXPERIMENTAL RESULTS

A. Experimental Setups

To evaluate our static map construction performance qualitatively and quantitatively, we used the KITTI odometry dataset [13] and the SemanticKITTI dataset [28]. The SemanticKITTI dataset provides scanwise labeled data and associated LiDAR SLAM-based SE(3) trajectory poses together with synchronized frames with the original KITTI dataset.

Yoon et al.'s dynamic point removal method was tested on their two small simulation datasets [25]. Unlike them,

we selected the KITTI dataset as our evaluation target because the KITTI dataset is widely used as a benchmark. Furthermore, using SemanticKITTI enables us to evaluate the pointwise static and dynamic predictions. They provide not only semantic labels but also movable instances' individual IDs, so we can track which object was moved. If an object moves longer than 0.5 m while a robot is observing a scene, then that object's points are dynamic.

Ground truth static map preparation. Using KITTI scans and SemanticKITTI instance labels, we constructed a moved-objects-excluded map and considered it as a ground truth static map (the left of Fig. 6). For evaluation clarity, we built a certain length of map (e.g., 100 m in Fig. 6) composed of equidistant sampled scans (e.g., 2 m in our experiment) with their poses estimated by SuMa [8]. We note that we did not include SemanticKITTI's unlabeled points, whose label index is zero (e.g., the gray points in Fig. 8), because if we contain them, then some dynamic points also emerge, and SemanticKITTI's map can no longer serve as a ground truth static map.

Our parameters. For fair comparison, we also used LiDAR SLAM-based scan poses predicted by SuMa [8] following the SemanticKITTI. By doing so, we aimed to consider less accurate pose estimation and prove the feasibility of the proposed method without using the original KITTI dataset's ground truth poses.

We define a single iteration as counting static/dynamic marks during a single batch of scans, whose enumeration number is equal to the number of scans used to construct a certain length of submap (e.g., 50 scans for 100 m and 2 m sampling). We calculate the staticity score of a point to conclude whether to remove it. We used the weights $\alpha_{SM} = 0.3$ and $\alpha_{DM} = -0.7$, and thresholds $\tau_D = 0.01$ and $\tau_S = -0.1$ (all experimental sequences used the same values).

We used $0.4°$ (vertically and horizontally equal) as the initial (finest) resolution for a single pixel via r_0; that is, the KITTI dataset's vertical LiDAR FOV (27 °) divided by the number of its rays (64). The resolution grew with $0.1°$ as an iteration evolved.

Evaluation criteria We used both SemanticKITTI's static map and our original map voxel down-sampled with 0.05 m size cell. For the proposed method's estimated static points, we define TP if the estimated static points appear in the SemanticKITTI ground truth map and FP if they do not appear. Co-appearance is considered to have occurred when a nearest point distance is within 0.1 m. In Fig. 8, TP and FP samples are marked as green-blue and red respectively. If a true static point in the ground truth map has no nearest neighbor in the predicted static map, then the point is marked as FN (yellow in Fig. 8).

B. Static map quality

Qualitative analysis. Fig. 6 shows our results and the results from SemanticKITTI. In SemanticKITTI's result, as mentioned already, some static regions are dismissed and marked as unlabeled, which could be a limitation of the supervised human-engaged labeling. We found this dismissed

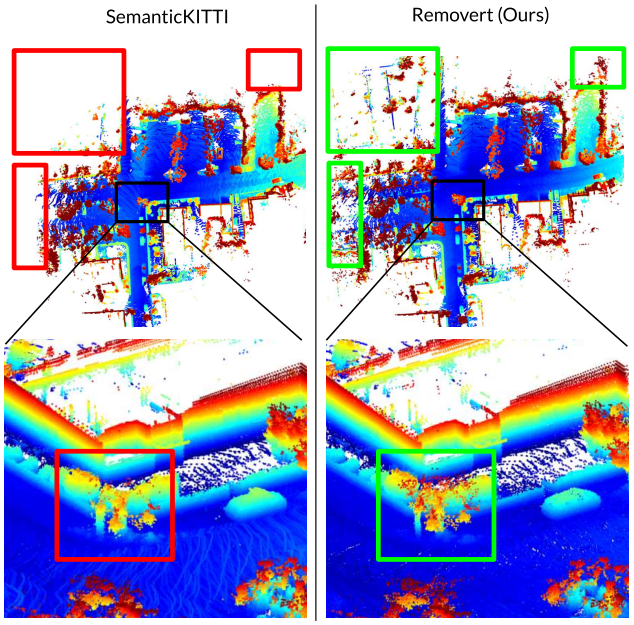

Fig. 6: A qualitative comparison of the SemanticKITTI [28] for the `KITTI 03` sequence (from frame 6 to 199, ranging approximately 100 m). SemanticKITTI has a number of unlabeled points, thus it loses some static points within the dynamic-removed map (top red boxes, top view) and some scattered bush points on a tree (a bottom red box, side view), which are actually static.

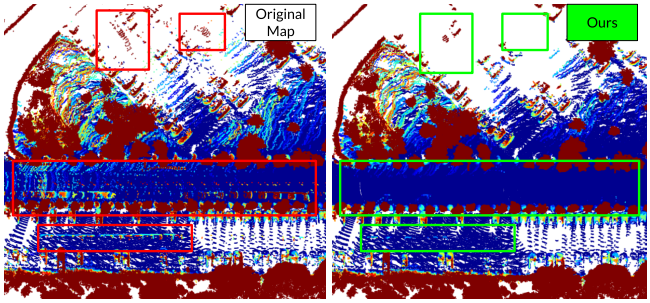

Fig. 7: Results of the `KAIST 02` sequence of the MulRan [29] dataset. The boxes above show the trajectories of moving pedestrians were removed, as well as the cars below.

ambiguous points from objects that have scattered shapes, such as tree leaves (bottom of Fig. 6). Compare to the human-labeled result, our result shows the consistent removing and preserving performance independent of the their sensing range. The more qualitative examples are shown in Fig. 1, Fig. 5, and Fig. 11 for `KITTI 08`, `KITTI 09`, and `KITTI 01`, respectively. See also the attached video `Removert.mp4`. We also validated independence of the LiDAR configuration by evaluating over the MulRan dataset [29] as in Fig. 7. Our algorithm works for various LiDARs having different vertical FOVs (i.e., $45°$ for the MulRan dataset and $27°$ for the KITTI dataset).

Quantitative analysis. Fig. 8 and Fig. 9 show our detailed quantitative results. As seen in Fig. 8, the proposed *Removert* algorithm first radically removes ambiguous points from the original noisy map and iteratively recovers the true positive

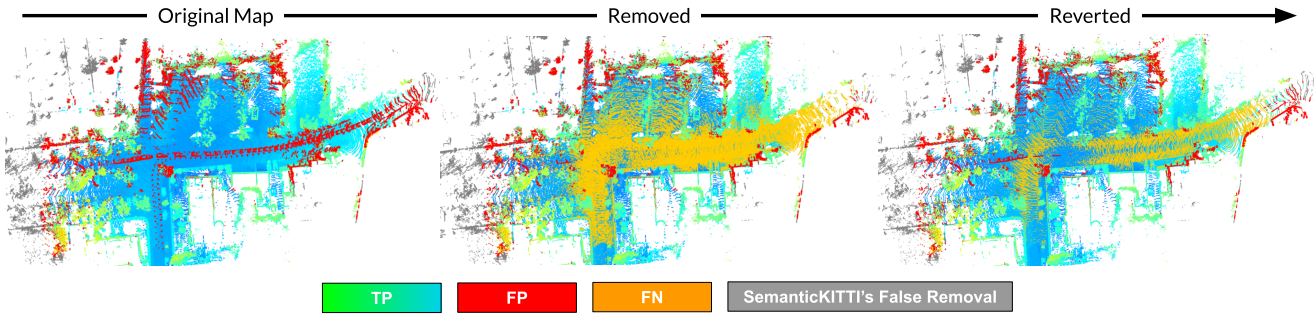

Fig. 8: The visualization of Fig. 6's evolution using the proposed method on KITTI 03. The TP, FP, and FN means correctly estimated static, falsely preserved static (i.e., static that is actually dynamic), and falsely removed static (i.e., static that should not be removed), respectively. Gray points, which exist in our map, seem to be static but do not exist in SemanticKITTI's map; for fair evaluation, the gray points are not involved in the evaluation in Fig. 9.

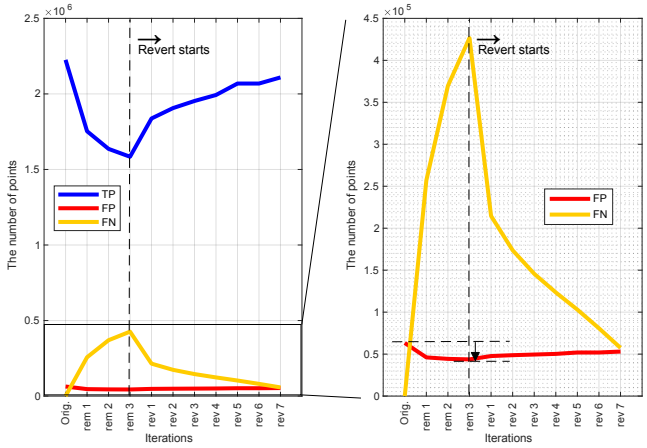

Fig. 9: The quantitative result of Fig. 8 shows the history of the number of TP, FP, and FN points along the iteration evolves. As the reverted iterations with decreasing resolutions of the range image, a large number of FN points disappeared while the number of FP points marginally increased; that is, the recovered static points are dominant.

points while increasing the implicit association window size (i.e., pixel resolution). Fig. 9 presents the historical number of points of the iterations. We removed points in the first three steps from the initial 0.4° resolution, and we reverted points along the following seven iterations. At the right in Fig. 9, we can observe the FN decrease was dominant while the number of wrongly added points was small. However, we argue that some FP points observed in uncertain regions may actually be static, as mentioned at the bottom of Fig. 6, and we can see this phenomenon at the right in Fig. 8. This may slightly disrupt the quantitative result of FP in Fig. 9 and our quantitative result may be better than the report.

C. Labeling pointwise dynamicity

We introduce here a potential application of our method. As mentioned in §I, the performance of dynamic point detection within a scan depends on static map's quality. Through our *Removert* algorithm, we can reliably separate static points from a noisy original map. Fig. 5 shows how we can parse urban dynamic objects using our static map, without an elaborate ground point removal method or shape assumptions. The objects in Fig. 5 are acquired using simple nearest point search in the map and segmenting points based on Euclidean distance; if a cluster has a small number of points (e.g., less than 30), then the object is ignored. More examples from the KITTI dataset [13] are shown in Fig. 10.

D. Limitations

A single nearest point is selected because the map's range image is constructed using visibility. This could be a limitation because we may not be able to catch dynamic points beyond the static points in some environments, such as the concrete median barrier in Fig. 11 (KITTI 01, around frame 256). However, additional sensor measurements from the opposite lanes would easily overcome the limitation.

V. Conclusion

In this paper, we proposed a novel method to recover the false predictions in a dynamic map and simultaneously enhance a static map. We also provided various qualitative results in addition to quantitative analysis of many sequences of the KITTI dataset and showed the proposed method is capable of removing dynamic points while successfully reverting and including static points. Our method has potential applications (e.g., automatic urban dynamic object parsing), and the self-labeled data could support the deep learning-based segmentation methods. In future works, we will apply *Removert* to data from multiple sessions [29, 30, 31] and expand our dynamic removal algorithm to long-term change detection and map update frameworks.

References

[1] M. Bosse, R. Zlot, and P. Flick, "Zebedee: Design of a spring-mounted 3-d range sensor with application to mobile mapping," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1104–1119, 2012.

[2] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments," *Intl. J. of Robot. Research*, vol. 38, no. 6, pp. 642–657, 2019.

[3] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain," in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.* IEEE, 2018, pp. 4758–4765.

[4] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3d lidar inertial odometry and mapping," in *Proc. IEEE Intl. Conf. on Robot. and Automat.* IEEE, 2019, pp. 3144–3150.

Fig. 10: Examples of self-parsed urban dynamic objects, which are automatically extracted from various sequences of the KITTI dataset. From left to right, the objects are a car, a bus, a pedestrian, a pedestrian with a dog, a pedestrian pushing a stroller, bikes, and a pair of pedestrians. As mentioned in Fig. 5, our method does not assume any model's shape, size, or density and does not require point cloud processing (e.g., ground removal, normal estimation, or region growing).

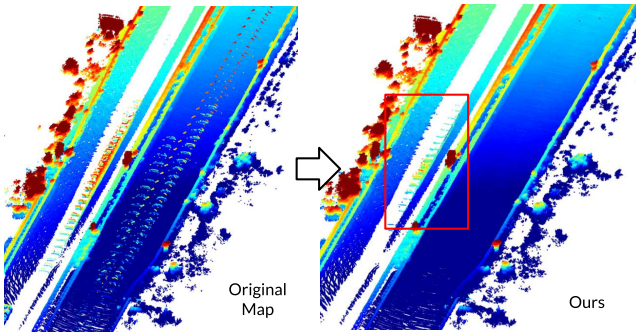

Fig. 11: Our limitations in KITTI 01 around frame 256. The points in the red box were not removed due to the existence of the concrete median barrier.

[5] G. Kim and A. Kim, "Scan Context: Egocentric Spatial Descriptor for Place Recognition within 3D Point Cloud Map," in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, Madrid, Oct. 2018.

[6] R. Dubé, A. Cramariuc, D. Dugas, H. Sommer, M. Dymczyk, J. Nieto, R. Siegwart, and C. Cadena, "SegMap: Segment-based mapping and localization using data-driven descriptors," *Intl. J. of Robot. Research*, 2019.

[7] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *Proc. IEEE Intl. Conf. on Robot. and Automat.* IEEE, 2016, pp. 1271–1278.

[8] J. Behley and C. Stachniss, "Efficient surfel-based slam using 3d laser range data in urban environments." in *Proc. Robot.: Science & Sys. Conf.*, 2018.

[9] J. J. Youngji Kim and A. Kim, "Stereo camera localization in 3D LiDAR maps," in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, Madrid, Oct. 2018, pp. 1–9.

[10] G. Kim, B. Park, and A. Kim, " 1-Day Learning, 1-Year Localization: Long-term LiDAR Localization using Scan Context Image ," *IEEE Robot. and Automat. Lett.*, vol. 4, no. 2, pp. 1948–1955, 2019.

[11] C. Wei, R. Wang, T. Wu, T. Chen, Y. Fang, and H. Fu, "Plane-based scan registration with moving vehicles exclusion," *Robot. and Autonomous Sys.*, vol. 83, pp. 261–274, 2016.

[12] W. Maddern, G. Pascoe, and P. Newman, "Leveraging Experience for Large-Scale LIDAR Localisation in Changing Cities," in *Proc. IEEE Intl. Conf. on Robot. and Automat.* IEEE, 2015, pp. 1684–1691.

[13] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2012, pp. 3354–3361.

[14] F. Pomerleau, P. Krüsi, F. Colas, P. Furgale, and R. Siegwart, "Long-term 3D map maintenance in dynamic environments," in *Proc. IEEE Intl. Conf. on Robot. and Automat.* IEEE, 2014, pp. 3712–3719.

[15] R. Ambruş, N. Bore, J. Folkesson, and P. Jensfelt, "Meta-rooms: Building and maintaining long term spatial models in a dynamic world," in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.* IEEE, 2014, pp. 1854–1861.

[16] C. Chen and B. Yang, "Dynamic occlusion detection and inpainting of in situ captured terrestrial laser scanning point clouds sequence," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 119, pp. 90–107, 2016.

[17] J. Gehrung, M. Hebel, M. Arens, and U. Stilla, "An approach to extract moving objects from mls data using a volumetric background representation," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, p. 107, 2017.

[18] J. Schauer and A. Nüchter, "The peopleremover—removing dynamic objects from 3-d point cloud data by traversing a voxel occupancy grid," *IEEE Robot. and Automat. Lett.*, vol. 3, no. 3, pp. 1679–1686, 2018.

[19] W. Xiao, B. Vallet, M. Brédif, and N. Paparoditis, "Street environment change detection from mobile laser scanning point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 107, pp. 38–49, 2015.

[20] I. Bogoslavskyi and C. Stachniss, "Fast range image-based segmentation of sparse 3d laser scans for online operation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 163–169.

[21] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and Accurate LiDAR Semantic Segmentation," in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, 2019.

[22] P. Ruchti and W. Burgard, "Mapping with dynamic-object probabilities calculated from single 3d range scans," in *Proc. IEEE Intl. Conf. on Robot. and Automat.* IEEE, 2018, pp. 6331–6336.

[23] L. Sun, Z. Yan, A. Zaganidis, C. Zhao, and T. Duckett, "Recurrent-octomap: Learning state-based map refinement for long-term semantic mapping with 3-d-lidar data," *IEEE Robot. and Automat. Lett.*, vol. 3, no. 4, pp. 3749–3756, 2018.

[24] K. Wong, S. Wang, M. Ren, M. Liang, and R. Urtasun, "Identifying unknown instances for autonomous driving," *Conference on Robot Learning (CoRL)*, 2019.

[25] D. Yoon, T. Tang, and T. Barfoot, "Mapless Online Detection of Dynamic Objects in 3D Lidar," in *2019 16th Conference on Computer and Robot Vision (CRV)*. IEEE, 2019, pp. 113–120.

[26] B. Ravi Kiran, L. Roldao, B. Irastorza, R. Verastegui, S. Suss, S. Yogamani, V. Talpaert, A. Lepoutre, and G. Trehard, "Real-time Dynamic Object Detection for Autonomous Driving using Prior 3D-Maps," in *Proc. European Conf. on Comput. Vision*, 2018.

[27] E. Palazzolo and C. Stachniss, "Fast Image-Based Geometric Change Detection Given a 3D Model," in *Proc. IEEE Intl. Conf. on Robot. and Automat.* IEEE, 2018, pp. 6308–6315.

[28] 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 *Proc. IEEE Intl. Conf. on Comput. Vision*, 2019.

[29] G. Kim, Y. S. Park, Y. Cho, J. Jeong, and A. Kim, "MulRan: Multimodal Range Dataset for Urban Place Recognition," in *Proc. IEEE Intl. Conf. on Robot. and Automat.*, Paris, May 2020, accepted. To appear.

[30] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of Michigan North Campus long-term vision and lidar dataset," *Intl. J. of Robot. Research*, vol. 35, no. 9, pp. 1023–1035, 2015.

[31] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner, "The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset," *Proc. IEEE Intl. Conf. on Robot. and Automat.*, May 2020, accepted. To appear.