Change of Scenery: Unsupervised LiDAR Change Detection for Mobile Robots

Alexander D. Krawciw
alec.krawciw@mail.utoronto.ca

Jordy Sehn
jordy.sehn@mail.utoronto.ca

Timothy D. Barfoot
tim.barfoot@utoronto.ca

Abstract—This paper presents a fully unsupervised deep change detection approach for mobile robots with 3D LiDAR. In unstructured environments, it is infeasible to define a closed set of semantic classes. Instead, semantic segmentation is reformulated as binary change detection. We develop a neural network, RangeNetCD, that uses an existing point-cloud map and a live LiDAR scan to detect scene changes with respect to the map. Using a novel loss function, existing point-cloud semantic segmentation networks can be trained to perform change detection without any labels or assumptions about local semantics. The mean intersection over union (mIoU) score is used for quantitative comparison. RangeNetCD outperforms the baseline by 3.8% to 7.7% depending on the amount of environmental structure. The neural network operates at 67.1 Hz and is integrated into a robot's autonomy stack to allow safe navigation around obstacles that intersect the planned path. In addition, a novel method for the rapid automated acquisition of per-point ground-truth labels is described. Covering changed parts of the scene with retroreflective materials and applying a threshold filter to the intensity channel of the LiDAR allows for quantitative evaluation of the change detector.

Index Terms—unsupervised machine learning, change detection

Fig. 1. The Clearpath Warthog UGV with the Ouster OS-1 LiDAR, driving a previously taught path. A section of the path is now blocked and the change detection algorithm proposed in this paper will be used to allow the robot to safely navigate around the obstruction. The mannequin is wearing the retroreflective suit that is used for dataset generation and evaluation.

I. INTRODUCTION

While significant progress has been made towards robot navigation in unstructured environments, challenges with robust perception and terrain assessment remain [1]. These can be exacerbated by a domain gap where methods designed for one location do not transfer reliably to another [2]. On-road autonomous driving benefits from clear rules and common types of objects that assist with risk assessment and detection [3]. Using extremely large, labelled datasets such as the Waymo Open Dataset [4] or SemanticKITTI [5] have allowed deep 3D networks to progress rapidly in the semantic and instance segmentation of known classes [6], [7].

This paper aims to solve a broader problem. We envision a situation where a robot can safely navigate autonomously after an initial mapping step. The map could be generated through manual exploration or prior experience from other systems. We argue that a map is a stronger prior for the types of features and obstacles that may be encountered than a predefined set of classes. It is desirable to perform per-point segmentation because bounding box regression implicitly imposes class-related sizes. Based on these conditions, we formulate the problem as binary change detection; points in the live scan are labelled as either changed or consistent with respect to the

All authors are with the University of Toronto Robotics Institute.

map. A conceptualization of the desired behaviour is provided in Figure 1. Classical change-detection methods use nearest-neighbour distances [8], normal distances [9], or ray tracing [10] to classify changes. However, they are most effective indoors and are less accurate in unstructured environments [11]. Vegetation is often dynamic on small length scales leading to variable scans with small changes irrelevant to planning. We hypothesize that a deep neural network can filter these planning-irrelevant changes to detect higher-quality clusters of changed points. Rather than tackle the full scope of terrain traversability assessment [12], all changes are treated as a threat and are avoided by the path planner. In the absence of a change, regions where the robot has driven before are considered traversable.

A novel loss formulation is used to provide an unsupervised training signal. This loss leverages inductive biases based on the amount of change, distance of changes from the existing map, and temporal permanence of objects.

These factors can be represented as a standard optimization problem. However, finding the optimal solution is computationally intractable. Our approach uses a neural network to learn a good heuristic for the problem instead. An unsupervised network training process can be used to learn an initial,

general network, which can be fine-tuned to new environments using data collected from the robot itself.

Once the network is trained, it is integrated into an existing autonomy stack to detect obstacles on a Clearpath Warthog unmanned ground vehicle (UGV) [13]. The robot can avoid obstacles reliably in real time.

In summary, we make the following contributions:

- an unsupervised deep LiDAR change-detection method for use on mobile robots,
- a loss formulation to train neural networks in an unsupervised manner,
- a rapid evaluation process for per-point semantic labelling,
- a change-detection neural network running in closed-loop on an unmanned ground vehicle.

The corpus of LiDAR datasets that contain multiple trajectories through the same environment is limited. A custom dataset with a combination of on-road and off-road driving is created and used for evaluation. Figure 2 shows the satellite view of the dataset paths.

II. RELATED WORK

A. Change Detection

Change detection in point clouds has been well studied in both mobile robotics [14] and remote sensing [15]. Classical change detection is primarily based on distances between two scans at different times. Some works [16], [10] use the labels *Changed* and *Unchanged*, which carry the same meaning as *Changed* and *Consistent* in this work. Our notation emphasizes that the classifications are not absolute and that points are typically consistent with mapping evidence, not identical. The simplest method evaluates the distance from each point in the first point cloud to the closest point in the second [8]. Distant points are classified as changed if they exceed a threshold value. Wu [11] proposed a Gaussian roughness model of a local neighbourhood to set a dynamic threshold. Alternatively, Underwood et al. [10] use ray tracing to determine which parts of the scene have changed. Any points along a ray that are closer than a previous scan are considered changed. A common

Fig. 2. The dataset trajectories and sample views. In total, the dataset is 8.5 km long with 1.75 km of unique paths.

challenge with these methods is that occluded regions are implicitly assumed to be free space, leading to false positives. Voelsen et al. [17] combine these approaches and use region growing to create instance clusters. However, these clusters require a pre-defined set of classes for the changes.

More recently, change detection has been approached using deep learning. There is a focus on airborne LiDARs that detect the long-term change of large geographic areas [18], [19]. Datasets such as SHREC 2023 [20] contain matched pairs of large point clouds for change detection but do not contain trajectory information. De Gelis et al. [21] present SiameseKPConv: a supervised semantic segmentation network that detects six classes of changes on 3D point clouds of cities. Twin encoders extract features from the map and live-scan point clouds. Once encoded, the feature-space difference of every live point to its nearest map neighbour is used for classification. A follow-up work uses the same network architecture but with self-supervised training [16]. They propose three losses: a contrastive loss, a deep clustering loss, and a temporal consistency loss. The contrastive loss pushes points classified as changed to have different embeddings than those classified as consistent. By clustering the predictions, some misclassifications are corrected to improve pseudo-labels for the next epoch. Finally, the temporal consistency loss encourages predictions to be consistent over time, imposing the bias that the number of changed points should be small. This work focuses on large-scale point clouds and does not consider the local trajectory of a mobile system.

Deep learning for change detection on mobile robots is beginning to take shape. Zhao et al. [22] perform supervised change detection on an indoor robot. They use four classes of changes: unchanged, dynamic, structural change, and temporary change. They show that a range image can be used to make efficient predictions about changes in the environment. To the best of our knowledge, there are no unsupervised deep LiDAR change detection systems for mobile robots.

B. Map Refinement

The construction of the point-cloud map impacts change-detection algorithms. Ideally, maps contain only static structures. Map refinement is the process of updating maps, removing dynamic objects, and rejecting outliers [23]. Pioneering works, Erasor [24] and PeopleRemover [25], use voxelized pseudo-occupancy-grid maps to estimate whether a region is free at a given time. These methods run in batches, allowing information from future scans to improve the detection at every frame. Points that exist within voxels that are often empty are removed from the map. Additionally, Erasor2 [26] uses instance segmentation to reduce the number of misclassifications at contact regions. Maintenance can be achieved by using multiple passes through the same environment and voting.

An alternative approach to creating static maps detects dynamic objects while the robot is moving and never adds them to the map. Pomerleau et al. [9] use the visibility of the map points and estimate every normal vector to determine whether a point is dynamic or static. Static points are added

Fig. 3. Data flow of the training procedure. Only a single map and live-scan pair are used for inference.

to the SLAM problem and dynamic points are rejected. Yoon [27] applies a combination of free-space checks with region growth to detect class-free dynamic points. Deep learning has also been applied to moving-object segmentation. SLIM [28] is a self-supervised moving-object segmentation network that operates on point clouds directly. RVMOS [29] uses sequences of range images and temporal augmentation to train effective supervised classifiers. Moving-object segmentation alone is insufficient for long-term autonomy because dangerous changes may be static during traversal.

C. Teach and Repeat Framework

The methods presented in this paper apply to any system that localizes against a point-cloud map. For the purposes of evaluation, we integrate the detector into the Visual Teach and Repeat 3 (VT&R3) framework[1]. VT&R was designed for stereo cameras and allows robots to precisely localize and repeat a path taught by a human operator [30]. Wu [11] and Sehn [31] modified the VT&R3 framework to use LiDAR for odometry and localization. During the teaching process, iterative-closest-point (ICP) [32] odometry determines the motion of the robot. These frames are collected into a sequence of topologically connected submaps. A relative transformation between each submap is stored for later localization. The repeat phase involves autonomous navigation to any location within the network of paths. By relaxing the path-following constraint to a corridor [31], the robot is given the flexibility to navigate around obstacles on the path. A local cost map is required by the existing planner within the corridor around the robot. The proposed method detects changes and treats their locations as unsafe to drive.

III. METHODOLOGY

A. Problem Definition

We let the problem of LiDAR change detection be one of binary classification. Given two point clouds in a common reference frame, the map M and current scan S, every point $s_i \in S$ is assigned a binary label (l_i) of *Changed* or *Consistent*. Let the number of points in S be n. *Consistent* is loosely defined as a point that could be in M under a slightly different mapping process.

[1]The framework is available at: github.com/utiasASRL/vtr3

To solve the problem without ground-truth labels, a combinatorial optimization problem is defined over a sequential set of two point-cloud pairs ($\{M_0, S_0\}, \{M_1, S_1\}$). The first live scan is recorded at time t_0. Consider $t_1 = t_0 + \Delta t$ where Δt is a hyperparameter of the system. All four point clouds are transformed into a common frame. The optimization has three terms: a chamfer loss [33] between the map and the *Consistent* live scan, a class-balance loss that penalizes the number of points that are considered to be *Changed*, and a temporal-consistency loss that penalizes labels that change between the sequential scans. The total loss function is defined as

$$\mathcal{L} = \mathcal{L}_{\text{cham}} + \lambda_1 \mathcal{L}_{\text{class}} + \lambda_2 \mathcal{L}_{\text{temporal}}, \tag{1}$$

where λ_1 and λ_2 are hyperparameters to be tuned. In theory, the solution to this problem does not require machine learning. For a fixed set of maps and live-scans, a gradient descent method could be iterated to find a local minimum to the problem for each frame starting from random labels. However, with thousands of points, it is intractable to solve as a classical optimization. To respect the run time requirements of a 10 Hz LiDAR scanner, a neural network is used to learn a fast heuristic solution to the optimization problem. To make the loss functions differentiable, the softmax probability of being classified as *Changed* is used instead of the actual prediction. Next, each of the terms in (1) is explained in detail.

1) Chamfer Loss: The chamfer loss uses the nearest-neighbour Euclidean distance between points in live scan ($s_i \in S$) and the map ($m_j \in M$). It is weighted by the likelihood that the given point belongs to the *Consistent* class ($l_i = 0$):

$$\mathcal{L}_{\text{cham}} = \frac{1}{n} \sum_{i=1}^{n} p(l_i = 0) \min_{m_j \in M} ||m_j - s_i||_2. \tag{2}$$

If the weighting of the chamfer loss is large, all points that do not exactly match the map will be classified as *Changed*. The chamfer loss pushes points to be classified as *Changed*.

2) Class-Balance Loss: The class-balance loss is the summation of the likelihood of a point being classified as *Changed* ($l_i = 1$):

$$\mathcal{L}_{\text{class}} = \frac{1}{n} \sum_{i=1}^{n} p(l_i = 1). \tag{3}$$

This loss pushes all points to be classified as *Consistent*. This loss opposes the chamfer loss.

3) Temporal-Consistency Loss: The temporal-consistency loss uses a bidirectional chamfer loss between points in the two consecutive live scans, S_0 and S_1, that are classified as *Changed*. l_{ki} is the label of point s_i in point cloud S_k:

$$
\mathcal{L}_{\text{temporal}} = \frac{1}{n_0} \sum_{i=1}^{n_0} p(l_{0i} = 1) \min_{s_j \in S_1} ||s_i - s_j||_2
$$
$$
+ \frac{1}{n_1} \sum_{i=1}^{n_1} p(l_{1i} = 1) \min_{s_j \in S_0} ||s_i - s_j||_2. \quad (4)
$$

This loss acts as a form of outlier rejection and encourages predictions to be consistent through time. This loss will push the network to classify all points as *Consistent* because an empty obstacle class has a loss of zero.

B. Deep Network Architecture

The novelty of this paper lies in the training method rather than the network architecture used to process the 3D points. In Figure 3, the neural network could be any 3D architecture that can be modified to accept pairs of maps and live scans. A convolutional neural network using range images as the input was chosen in this paper for its fast inference time [34]. Speed is critical to the autonomous operation of the UGV. Preliminary experiments were also performed on a 3D architecture based on KPConv [35] but the run time was too long.

a) Range Image Generation: The inputs to the network are aligned range images from the map and the live scan. These images are rendered by transforming the live scan and local map 3D point clouds into a common spherical coordinate system centred on the LiDAR. Each range image pixel, (u_i, v_i), contains the value $r_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$, to the closest point in the frustum. Dimensions $H \times W = 64 \times 1024$ are used over a $25° \times 360°$ field of view (fov). The lower limit of the fov below the horizon (fov$_{\text{down}}$) is a sensor property. An indexed mapping between the range image and point cloud is stored to reassign labels in the image to the corresponding 3D points. Given a point $(x_i, y_i, z_i) \in \mathbb{R}^3$ the following image mapping $\mathbb{R}^3 \to \mathbb{R}^2$ is used:

$$
\begin{pmatrix} u_i \\ v_i \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(1 - \frac{\arctan(y_i, x_i)}{\pi})W \\ \left(1 - \frac{\arcsin(z_i, r_i) + \text{fov}_{\text{down}}}{\text{fov}}\right)H \end{pmatrix}. \quad (5)
$$

b) Neural Network Architecture: The range image network (RangeNetCD) is based on RangeNet++ [34] but modified to accept both a map and live scan. The input has a dimension of $H \times W \times 2$ with the live scan and map on each channel. The network encoder consists of four double convolutions followed by maxpooling. The decoder uses bilinear upsampling and skip connections from the encoder layers for per-pixel prediction. We observed that effective classification requires non-square convolution kernels on the first and last layers by experimenting with different shapes. The kernel in the first and last layer has dimension 1×2.

Fig. 4. A LiDAR scan of two pedestrians coloured by intensity. Left: A pedestrian wearing regular clothes. Right: A pedestrian wearing the reflective suit.

IV. EXPERIMENTS

Experimental evaluation occurred in forested and off-road areas around the University of Toronto Institute for Aerospace Studies (Figure 2). A Clearpath Warthog UGV [13] equipped with an Ouster-OS1 LiDAR [36] was used for experiments. LiDAR ICP localization [32] aligned the live scan with the corresponding map prior to performing any neural-network inference. The horizon of the local planner is 10 m. Accordingly, detections are limited to this range around the robot. Empirically, it was observed that the sensor-aligned frame with the origin on the ground below the LiDAR scanner performed the best. This may occur because the nearby planar ground points are projected into the same pixel frustum.

A. Retroreflective Semantic Evaluation

While the method presented in this paper is fully unsupervised, for quantitative evaluation of the results, ground-truth labels are required. For change detection, an offline ray-tracing approach similar to Thomas et al. [37] could classify the points corresponding to changes of interest. However, these ray-tracing approaches are challenged in foliage [11] and are computationally expensive. Instead, we propose using a rapid technique for implicitly labelling the data.

During dataset evaluation, all objects introduced into the scene are coated in retroreflective material. Figure 4 highlights the difference in intensity between a reflective suit, regular clothes, and the background. The intensity channel of LiDAR is unique to each distinct sensor and not standardized between different models. For this reason, we discard the intensity values and render the intensity channel out-of-band. However, the intensity can be used for automatic labelling: a threshold filter reliably classifies the points that belong to reflective objects.

The effectiveness of this approach requires a controlled environment without ambient dynamic actors, so public scenarios are unsuitable. However, for many cases when the rapid labelling of an object is required, this method is highly effective. While we use pedestrians with reflective suits in our

TABLE II
PERFORMANCE ON EASY (GRASS YARD) DATA OF UNSUPERVISED
DETECTOR AND CLASSICAL BASELINE

Method	IoU_{ch}	Corridor IoU_{ch}	mIoU
Nearest Neighbour	0.507	0.618	0.745
RangeNetCD (ours)	**0.651**	**0.820**	**0.822**

TABLE III
PERFORMANCE ON MEDIUM (NORTH FIELD) DATA OF UNSUPERVISED
DETECTOR AND CLASSICAL BASELINE

Method	IoU_{ch}	Corridor IoU_{ch}	mIoU
Nearest Neighbour	0.438	0.586	0.712
RangeNetCD (ours)	**0.577**	**0.609**	**0.782**

TABLE IV
PERFORMANCE ON HARD (FOREST LOOP) DATA OF UNSUPERVISED
DETECTOR AND CLASSICAL BASELINE

Method	IoU_{ch}	Corridor IoU_{ch}	mIoU
Nearest Neighbour	0.290	0.547	0.635
RangeNetCD (ours)	**0.607**	**0.656**	**0.673**

Fig. 5. Changed IoU vs Distance Traveled Fine-tuning for a map-voxel of 0.3 m and live voxel of 0.05 m. Pre-training improves the performance.

dataset, in general, retroreflective cloth or tape can be placed on any object where per-point ground truth is required.

B. Results

The network is trained and evaluated on a collection of four different mapping and repeating sequences. In total, the raw dataset consists of $50,000$ frames and the robot traverses 8.5 km over 1.75 km of unique paths. A LiDAR frame is stored every 30 cm of driving rather than every 0.1 s, which reduces the total frames used to $20,568$. This prevents biased training due to interruptions in the data collection when the robot stops moving. The four sequences are traced out in Figure 2. The longest sequence is the Parking Loop, which is uncontrolled and contains moving vehicles, bicycles, and pedestrians. The other three sequences were recorded with reflective obstacles for evaluation. Table I describes the details of each run of the dataset.

For comparison, a classical nearest-neighbour baseline [8] is used as an alternative to unsupervised learning. Table II, Table III, and Table IV provide the intersection-over-union (IoU) score of the *Changed* class as well as the mean IoU (mIoU). Note that, because the labels are highly imbalanced, metrics evaluated on the *Changed* class are the basis of comparison. We adopt an additional planning-oriented metric to represent the ability of the robot to perform path planning. The IoU is re-evaluated within the known planning corridor around the robot, which is 5 m wide for the Warthog running Teach and Repeat. The results in Tables I-III are evaluated with map voxel size of 0.2 m and live-scan voxel size 0.05 m. The loss function hyperparameters during training were $\lambda_1 = 15$ and $\lambda_2 = 1.0$.

On the Grass Yard (Table II), the corridor IoU_{ch} is 16.9% higher than the IoU_{ch} evaluated on the entire scan for RangeNetCD. This highlights the practical effectiveness of the

system because false positives on static structures outside the corridor of the planner do not affect the trajectory.

We find that pre-training a general network followed by environment-specific fine-tuning is the most effective training approach. To pre-train, the unlabelled and uncontrolled data from the Parking Loop is used. Finetuning occurs on a specific sequence with a lower learning rate in the optimizer. In Figure 5, pre-training improves the initial performance to 68%. While the general network is capable in all of the environments tested, fine-tuning with more specific examples boosts detection accuracy further. An avenue of future work is to fine-tune live on the robot once it has been deployed in operation. The detection quality is suitable for autonomous navigation.

C. Ablation Studies

a) Scan and Map Density: The OS-1 LiDAR used for these experiments produces 1.3 million points per second [36]. Using every point is computationally impractical for ICP odometry and localization [11]. Additionally, storing every point in the map is prohibitively expensive. For this reason, the map is voxel downsampled. The voxel sizes of the map and live scan impact the accuracy of change detection. Table V shows the corridor IoU of the *Changed* class for different

TABLE V
CORRIDOR IoU_{CH} FOR DIFFERENT COMBINATIONS OF THE MAP AND LIVE
SCAN VOXEL DOWNSAMPLING ON THE GRASS YARD

		Map Voxel Size		
		0.1 m	0.2 m	0.3 m
Live Voxel Size	0.05 m	0.760	**0.820**	0.811
	0.15 m	0.091	0.111	0.110
	0.3 m	0.050	0.071	0.056

TABLE VI
ABLATION STUDY ON LOSS FUNCTIONS EVALUATED ON THE GRASS
YARD WITH MAP VOXEL 0.2 M AND LIVE VOXEL 0.05 M

Chamfer	Class	Temporal	IoU_{ch}
✓	✗	✗	0.02
✗	✓	✗	0.0
✓	✓	✗	0.388
✓	✓	✓	**0.651**

Fig. 6. The input range images and change detection of a cyclist passing by the robot. The two range images are used by the network to determine changes. Part of the live scan is blocked by the robot's structure, this area is purple. In the result, green regions are classified as changed and grey regions are consistent.

combinations of scan and map densities. Increasing the live-scan density improves performance but changing the map density is less important. Most obstacles in the dataset are smaller than 1 m in their longest dimension so voxel downsampling removes defining points. In contrast, much of the map contains larger objects that can be downsampled without affecting performance. The range-image network performs much better than the baseline as the map density decreases. This is advantageous because smaller point clouds use less storage and are processed faster. For reference, storing the raw scan requires about 50 Gb/km and downsampling to 0.3 m requires about 3.75 Gb/km. Intermediate values scale nonlinearly based on the LiDAR scan pattern and density of nearby static features. A map voxel size of 0.3 m and a live-scan voxel size of 0.05 m is used on the Warthog.

b) Loss Functions: Table VI shows the impact of each loss function on the performance for the 0.2 m and 0.05 m map and live-scan voxelizations of the Grass Yard. As expected, if the class loss is removed, all points are classified as *Changed*. This occurs because the trivial optimal solution classifies all points as *Changed*. Conversely, if only the class loss is used, all points are classified as *Consistent*. The correct balance of the chamfer and class losses accounts for the basic capability of the system. Adding the temporal loss improves the performance by penalizing transient points, reducing the number of false-positive detections.

D. Qualitative Evaluation

Several interesting phenomena are observed in the output of the network. First, this change-detection approach is capable of generalizing to new types of objects. In Figure 6, a cyclist is accurately detected biking past the robot. No cyclists were included in the training data. While RangeNetCD is designed to allow for the detection of any changes in the scene, it is possible to overfit the network to shapes seen during training. For example, when the network was trained on data that changed only by introducing pedestrians, it failed to detect new vehicles at test time. This was solved by using more, diverse training data. With extensive datasets, it is interesting

to consider if these objects could be differentiated in the feature space. We observe that if training sequences contain changes in every frame, the network forces detections in every frame. The most false positives appear when the correct solution is no change at all.

Change-detection performance is limited by the map quality. The map is assumed to capture only the permanent environment but errors appear when it does not. Using the uncontrolled parking lot data, a map was generated containing cars that had moved before later runs were recorded. This led to two types of misdetections. Points on static objects that were previously occluded are false positives. For example, a fire hydrant behind a parked car was later detected as a change even though it was always present. Points in the same location as an object that moved are false negatives. For example, a pedestrian walking through an empty parking space where a car was located during mapping is not detected even though it is a change. These issues rarely impact the planner, because the taught path will not exist inside obstacles in the map. Significant issues with the mapping process can destabilize network training because patterns are less consistent. These errors were eliminated by generating the map when the parking lots were empty. Applying a map-cleaning approach offline would be beneficial as well.

E. Closed-Loop Performance

The range-image network was tested on a laptop NVIDIA RTX A4500 GPU located on the robot. Inference takes 14.9 ± 2.3 ms. Most of this time is spent transferring the tensors between CPU and GPU memory. More optimal pipelines may exist that reduce the amount of transfer. RangeNetCD was exported to TorchScript and executed in C++ as part of the existing LiDAR Teach and Repeat [11] pipeline.

Points detected as *Changed* are passed to a cost map inflation module that projects them into 2D and accounts for the robot's radius. The planner maintains a queue of cost maps to overcome the LiDAR's close-range blind spot. These cost maps allow the robot to avoid local obstacles. The supplementary video [2] contains sequences of the robot navigating around a series of introduced obstacles. In Figure 7,

[2]https://youtu.be/prqVQJHXYWE

Fig. 7. Left: an image view of an obstacle and the path the robot drives to avoid it. Right: The point cloud view of the same scene.

a mannequin was placed on the path as a change. The robot successfully detects and avoids it and then continues to follow the originally planned path. As a side effect of maintaining a queue of obstacles, dynamic changes cause smearing in the cost map leading to suboptimal routes. As a future extension to this work, a self-supervised prediction layer could be used to extrapolate the motion of changes over a short time horizon.

V. Conclusion and Future Work

In this paper, we present a novel method for the unsupervised training of a deep network that performs change detection on a pair of point clouds. We show that by exploiting inductive biases related to the amount of change, distance of changes from the map, and temporal consistency of physical scenes, it is possible to train a network. We demonstrate the applicability of the approach on a range-image convolutional neural network that is trained on data collected on an unmanned ground vehicle. Once trained, the network runs quickly and is added to the robot to improve its ability to navigate for long periods. Future work using a pre-trained contrastive encoder for feature extraction will add semantic differences to the training loss.

Acknowledgment

References

[1] L. Wijayathunga, A. Rassau, and D. Chai, "Challenges and Solutions for Autonomous Ground Robot Scene Understanding and Navigation in Unstructured Outdoor Environments: A Review," *Applied Sciences*, vol. 13, p. 9877, Jan. 2023. Number: 17 Publisher: Multidisciplinary Digital Publishing Institute.

[2] W. M. Kouw and M. Loog, "A Review of Domain Adaptation without Target Labels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 766–785, Mar. 2021. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[3] Z. Ma, Y. Yang, G. Wang, X. Xu, H. T. Shen, and M. Zhang, "Rethinking Open-World Object Detection in Autonomous Driving Scenarios," in *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, (New York, NY, USA), pp. 1279–1288, Association for Computing Machinery, Oct. 2022.

[4] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[5] 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. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.

[6] X. Yan, J. Gao, C. Zheng, C. Zheng, R. Zhang, S. Cui, and Z. Li, "2dpass: 2d priors assisted semantic segmentation on lidar point clouds," in *European Conference on Computer Vision*, pp. 677–695, Springer, 2022.

[7] Z. Zhou, Y. Zhang, and H. Foroosh, "Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[8] D. Girardeau-Montaut, M. Roux, R. Marc, and G. Thibault, "Change detection on points cloud data acquired with a ground laser scanner," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, no. 3, p. W19, 2005.

[9] F. Pomerleau, P. Krüsi, F. Colas, P. Furgale, and R. Siegwart, "Long-term 3d map maintenance in dynamic environments," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3712–3719, 2014.

[10] J. P. Underwood, D. Gillsjö, T. Bailey, and V. Vlaskine, "Explicit 3D change detection using ray-tracing in spherical coordinates," in *2013 IEEE International Conference on Robotics and Automation*, pp. 4735–4741, May 2013. ISSN: 1050-4729.

[11] Y. Wu, "VT&R3: Generalizing the teach and repeat navigation framework," Sept. 2022. MASc Thesis.

[12] P. Papadakis, "Terrain traversability analysis methods for unmanned ground vehicles: A survey," *Engineering Applications of Artificial Intelligence*, vol. 26, pp. 1373–1385, Apr. 2013.

[13] "Clearpath Robotics Warthog UGV." Available Online https://clearpathrobotics.com/warthog-unmanned-ground-vehicle-robot/, 2020.

[14] L. Wellhausen, R. Dubé, A. Gawel, R. Siegwart, and C. Cadena, "Reliable real-time change detection and mapping for 3d lidars," in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, pp. 81–87, 2017.

[15] W. Xiao, H. Cao, M. Tang, Z. Zhang, and N. Chen, "3D urban object change detection from aerial and terrestrial point clouds: A review," *International Journal of Applied Earth Observation and Geoinformation*, vol. 118, p. 103258, Apr. 2023.

[16] D. G. Iris, S. Saha, M. Shahzad, T. Corpetti, S. Lefèvre, and X. Zhu, "Deep unsupervised learning for 3d als point clouds change detection," *ISPRS Open Journal of Photogrammetry and Remote Sensing*, vol. 9, p. 100044, 08 2023.

[17] M. Voelsen, J. Schachtschneider, and C. Brenner, "Classification and Change Detection in Mobile Mapping LiDAR Point Clouds," *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 89, pp. 195–207, June 2021.

[18] U. Okyay, J. Telling, C. L. Glennie, and W. E. Dietrich, "Airborne lidar change detection: An overview of earth sciences applications," *Earth-Science Reviews*, vol. 198, p. 102929, 2019.

[19] U. Stilla and Y. Xu, "Change detection of urban objects using 3d point clouds: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 197, pp. 228–255, 2023.

[20] Y. Gao, H. Yuan, T. Ku, R. C. Veltkamp, G. Zamanakos, L. Tsochatzidis, A. Amanatiadis, I. Pratikakis, A. Panou, I. Romanelis, V. Fotis, G. Arvanitis, and K. Moustakas, "SHREC 2023: Point cloud change detection for city scenes," *Computers & Graphics*, vol. 115, pp. 35–42, 2023.

[21] I. de Gélis, S. Lefèvre, and T. Corpetti, "Siamese KPConv: 3D multiple change detection from raw point clouds using deep learning," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 197, pp. 274–291, Mar. 2023.

[22] H. Zhao, M. Tomko, and K. Khoshelham, "Interior structural change detection using a 3D model and LiDAR segmentation," *Journal of Building Engineering*, vol. 72, p. 106628, Aug. 2023.

[23] M. Yguel, O. Aycard, and C. Laugier, "Update Policy of Dense Maps: Efficient Algorithms and Sparse Representation," *Field and Service Robotics: Results of the 6th International Conference (STAR: Springer Tracts in Advanced Robotics Series Volume 42)*, vol. 42, pp. 23–33, 2008.

[24] H. Lim, S. Hwang, and H. Myung, "Erasor: Egocentric ratio of pseudo occupancy-based dynamic object removal for static 3d point cloud map building," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2272–2279, 2021.

[25] J. Schauer and A. Nüchter, "The Peopleremover—Removing Dynamic Objects From 3-D Point Cloud Data by Traversing a Voxel Occupancy Grid," *IEEE Robotics and Automation Letters*, vol. 3, pp. 1679–1686, July 2018. Conference Name: IEEE Robotics and Automation Letters.

[26] H. Lim, L. Nunes, B. Mersch, X. Chen, J. Behley, and H. Myung, "ERASOR2: Instance-Aware Robust 3D Mapping of the Static World in Dynamic Scenes," in *Robotics: Science and Systems*, (Daegu, Republic of Korea.), July 2023.

[27] J. D. Yoon, *Model-free Setting-Independent Detection of Dynamic Objects in 3D Lidar*. MASc, University of Toronto, 2019.

[28] S. Andreas Baur, D. Josef Emmerichs, F. Moosmann, P. Pinggera, B. Ommer, and A. Geiger, "SLIM: Self-Supervised LiDAR Scene Flow and Motion Segmentation," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, (Montreal, QC, Canada), pp. 13106–13116, IEEE, Oct. 2021.

[29] J. Kim, J. Woo, and S. Im, "Rvmos: Range-view moving object segmentation leveraged by semantic and motion features," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8044–8051, 2022.

[30] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *Journal of Field Robotics*, 2010.

[31] J. Sehn, Y. Wu, and T. D. Barfoot, "Along Similar Lines: Local Obstacle Avoidance for Long-term Autonomous Path Following," in *20th Conference on Robots and Vision*, June 2023.

[32] K. Burnett, Y. Wu, D. J. Yoon, A. P. Schoellig, and T. D. Barfoot, "Are We Ready for Radar to Replace Lidar in All-Weather Mapping and Localization?," *IEEE Robotics and Automation Letters*, vol. 7, pp. 10328–10335, Oct. 2022. Conference Name: IEEE Robotics and Automation Letters.

[33] "Pytorch3d Chamfer Loss." Available Online https://pytorch3d.readthedocs.io/en/latest/modules/loss.html.

[34] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and Accurate LiDAR Semantic Segmentation," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[35] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and deformable convolution for point clouds," *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[36] "Ouster OS1 LiDAR." Available Online [https://ouster.com/products/scanning-lidar/os1-sensor/].

[37] H. Thomas, B. Agro, M. Gridseth, J. Zhang, and T. D. Barfoot, "Self-supervised learning of lidar segmentation for autonomous indoor navigation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14047–14053, 2021.