

# LaserSAM: Zero-Shot Change Detection Using Visual Segmentation of Spinning LiDAR

Alexander Krawciw  
Robotics Institute  
University of Toronto  
Toronto, Canada  
alec.krawciw@mail.utoronto.ca

Sven Lilge  
Robotics Institute  
University of Toronto  
Toronto, Canada  
sven.lilge@utoronto.ca

Timothy D. Barfoot  
Robotics Institute  
University of Toronto  
Toronto, Canada  
tim.barfoot@utoronto.ca

**Abstract**—This paper presents an approach for applying camera perception techniques to spinning LiDAR data. To improve the robustness of long-term change detection from a 3D LiDAR, range and intensity information are rendered into virtual perspectives using a pinhole camera model. Hue-saturation-value image encoding is used to colorize the images by range and near-IR intensity. The LiDAR’s active scene illumination makes it invariant to ambient brightness, which enables night-to-day change detection without additional processing. Using the range-colourized, perspective image allows existing foundation models to detect semantic regions. Specifically, the Segment Anything Model detects semantically similar regions in both a previously acquired map and live view from a path-repeating robot. By comparing the masks in both views, changes in the live scan are detected. Results indicate that the Segment Anything Model accurately captures the shape of arbitrary changes introduced into scenes. The proposed method achieves a segmentation intersection over union of 73.3% when evaluated in unstructured environments and 80.4% when evaluated within the planning corridor. Changes can be detected reliably through day-to-night illumination variations. After pixel-level masks are generated, the one-to-one correspondence with 3D points means that the 2D masks can be used directly to recover the 3D location of the changes. The detected 3D changes are avoided in a closed loop by treating them as obstacles in a local motion planner. Experiments on an unmanned ground vehicle demonstrate the performance of the method.

**Keywords**—Change Detection; LiDAR Semantic Segmentation

## I. INTRODUCTION

Robust perception remains a central challenge for mobile robots operating in unstructured environments. Unlike autonomous cars, there are no rules of the road and out-of-distribution classes of obstructions occur with a higher frequency [1]. Occlusion effects caused by vegetation degrade the quality of many detectors trained in structured environments [2]. These factors make it difficult for supervised perception pipelines to identify hazards reliably enough to maintain the aggressive performance of the vehicles. In addition to obstacle detection, off-road vehicles must perform terrain assessment to understand which local paths are feasible to drive [3]. Visual Teach and Repeat (VT&R)

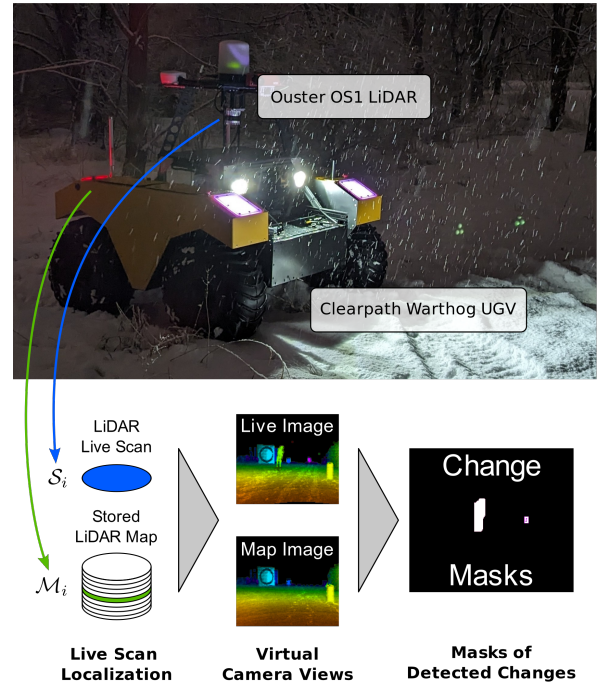


Figure 1. The Clearpath Warthog robot driving at night during snowfall, repeating a path that was previously taught during daytime. This paper proposes LaserSAM to detect and mask environmental changes between teach and repeat paths by creating virtual camera views from LiDAR data and applying deep-learning-based segmentations.

[4] provides a practical alternative to terrain assessment. A human operator manually drives a robot along a network of connected paths. While driving, the robot constructs local submaps, which are used later for localization. After the paths are taught, the robot can autonomously navigate between any two locations in the network. The primary responsibility of terrain assessment is delegated to the human pilot, allowing aggressive maneuvers to be demonstrated to the robot. To allow safe repeats over long periods of time, we propose that it is sufficient to detect changes in the scene that intersect the planned local path of the vehicle [5], [6]. During the teach pass, the operator defines a safe corridor

[7] around the path. When repeating, the robot navigates around new obstacles within the corridor’s boundary. By focusing on detecting changes instead of specific types of hazards, any type of obstacle can be detected and avoided without predefining specific classes. For this reason, a robust change-detection module is desired to improve the system’s capabilities.

Existing work on change detection has focused on images from monocular cameras [8] and 3D LiDAR point clouds [9]. Ding et al. [10] use the FastSAM encoder as part of a visual change-detection approach for satellite images. Recently, change detection of point clouds for remote sensing [11] and mobile robotics [12] using deep learning has emerged as an area of study. Conventional cameras have higher resolution than most spinning LiDAR sensors allowing them to detect shapes and textures more accurately. However, the lack of 3D information within the image makes it difficult to accurately transfer 2D segmentations into 3D for path planning [2]. Stereo vision systems can provide 3D information but must solve an additional data association problem. While progress has been made to allow for direct comparison of images through lighting and seasonal changes [13], illumination variation complicates change detection for cameras.

Spinning 360° LiDAR scanners have emerged as a popular, complementary sensor to cameras [14]. LiDAR provides accurate 3D position and intensity measurements for each point. Most automotive LiDAR units operate at a wavelength of 1550 nm because the atmosphere absorbs most of the sun’s energy at that wavelength [15]. Ouster LiDARs operate at 840 nm allowing them to capture ambient sunlight [15]. The combination of ambient and projected infrared light leads to more natural illumination effects that blur the line between passive camera and active LiDAR.

This work blends the strengths of both modalities by rendering perspective camera images from LiDAR scans. This approach allows computer vision algorithms to be applied to multi-view change detection. Specifically, the Segment Anything Model (SAM) [16] is used to detect semantic regions in the rendered images. Its zero-shot generalization capabilities allow for the segmentation of any object, including unseen ones in new domains. Multi-modal algorithms rely on extrinsic calibration between LiDAR and cameras to define the depth of some pixels in a camera image. In this approach, every pixel in the generated image corresponds exactly to a 3D point. 3D points from the map and the live scan are rendered into a common virtual camera frame for analysis. Finally, the active illumination and signal processing of the 840 nm IR means that change detection works across illumination conditions with no additional processing required. Figure 1 demonstrates the types of conditions that challenge camera-based detection but do not impact the proposed LiDAR pipeline.

In summary, we propose the following contribution: a

change detection pipeline that combines the sensor benefits of LiDAR with a pre-trained foundation model for image segmentation.

## II. RELATED WORK

### A. Change Detection

Detecting changes in images, point clouds, or other rich sensors is often a key task in scene understanding [8]. One of the central challenges is to capture the domain-specific semantics accurately. Differentiating inconsequential changes such as illumination and sensor orientation allows for meaningful downstream processing. Ultimately, these definitions must be determined at the problem level but the binary definitions of *changed* and *unchanged* are common in the literature [8]. Most methods operate at the point/pixel level. Aggregation into objects may not be necessary or occur as a downstream processing task.

1) *3D LiDAR Change Detection*: Change detection in point clouds is most commonly applied in remote sensing [9] to detect large-scale changes such as new buildings [11]. Classical methods are usually based on the geometry of the scene. The simplest point-cloud difference evaluates the distance of every point to its nearest neighbour [17] and thresholds distant points as *changed*. These thresholds can be place-dependent [5] to adapt to the local geometry. Alternative approaches use ray-tracing [18] or normal distances [6] to classify points based on fixed or variable thresholds. Deep learning is successful in the change detection of 3D point clouds as well. de Gelis et al. use parallel encoders to train both a supervised [11] and unsupervised [19] neural network that classifies points as added, removed, or the same. These works operate on point cloud maps that are constructed from many different sensor viewpoints, such as the SHREC 2023 dataset [20]. This makes them less applicable to mobile robots, which are heavily affected by occlusions.

2) *2D Camera Change Detection*: Camera change detection can be classified in many ways, but the most relevant detail to this work is whether or not the method assumes a quasi-static background. Quasi-static methods tend to construct a representation of the scene background that is static [21] or adaptive [8]. These methods perform well in video surveillance. Viewpoint variations from mobile robots make them less applicable to the data analyzed here. Fewer methods consider the temporal information from video data and attempt to track changes from a moving camera [22]. This is difficult in the monocular case because the 3D position of the objects is unknown and warping an image into a new frame requires assumptions about the scene or camera motion.

### B. Zero-Shot Semantic Segmentation

In 2023, Meta Research released the Segment Anything Model (SAM) [16], which aims to perform semantic segmentation on any scene. The model is trained on over

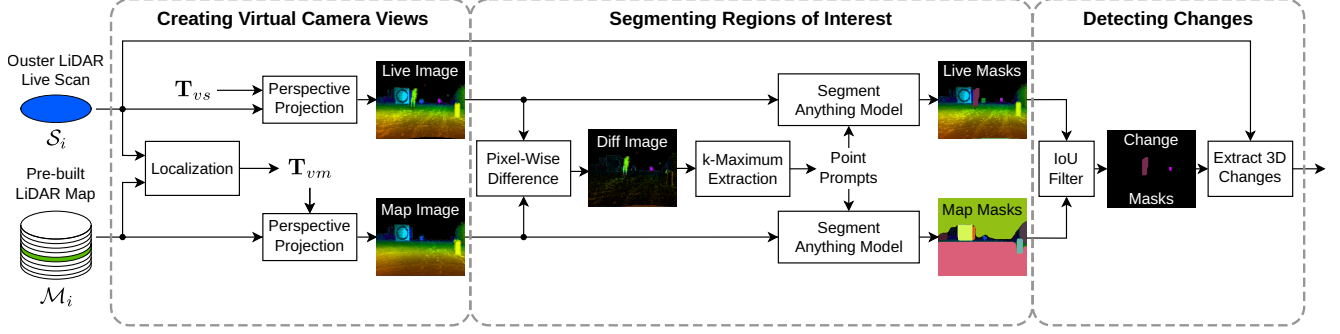


Figure 2. Data processing pipeline of LaserSAM. The pipeline runs for each new frame obtained from the Ouster LiDAR.

eleven million images with more than one billion masks. This vast dataset has enabled the capability to segment regions in a class-agnostic manner. In contrast to common datasets for autonomous driving such as CityScapes [23] or SemanticKITTI [24], this allows the model to generate masks for images that were captured in new environments with different cameras. Additionally, the model decoder accepts geometric prompts such as points of interest or bounding boxes that refine the masks that are reported. While this performance is impressive, its run time is slow for robotics applications.

### C. LiDAR as a Camera

The most common LiDAR sensors used in autonomous vehicle development are 360° units from Ouster (Velodyne) [25], and Hesai [26]. These sensors are characterized by higher resolution along the horizontal spinning axis than in the vertical field of view (FOV). The current state-of-the-art sensors have 128 vertical beams and operate at 10 to 20 Hz [25]. However, low-frame-rate LiDAR sensors have existed with much higher resolutions for more than two decades. McManus et al. [27] showed that lighting-invariant visual odometry could be performed using SURF features on intensity images taken from an Autonosys 2D scanning LiDAR. This sensor produces images with a 30°V × 90°H FOV, with resolution 480 × 360 pixels at 2 Hz. The frame rate is much slower, but the vertical resolution is still significantly higher (12 pixels per degree) than the OS-1 LiDAR (2.85 pixels per degree) used in this paper. This approach was extended to account for the motion distortion caused by the moving sensor [28] and used to control an offroad vehicle in closed loop [29], [30].

## III. METHOD

This paper ties together past progress in change detection and LiDAR-based image processing with state-of-the-art foundation models. By leveraging the strengths of traditional approaches, and the high-quality segmentation generated by SAM [16], our system can detect previously unseen obstacles, improving the autonomy of a mobile robot.

The proposed change-detection pipeline has three primary stages: creating virtual camera views, segmenting regions of interest, and detecting changes. These stages, as well as the internal steps, are illustrated in Figure 2. The problem inputs are two point clouds: the local submap used in localization  $\mathcal{M}_i$  and the live LiDAR scan  $\mathcal{S}_i$ .

### A. Creating Virtual Camera Views

The core idea of this pipeline is to project the two aligned point clouds into images, the format expected by SAM. The natural data format of a spinning LiDAR is equirectangular where each pixel corresponds to a constant angular offset. The natural data format of cameras is the perspective projection. Sample images created using each projection are shown in Figure 3. A larger blindspot around the base of the robot is created in the equirectangular view than in the perspective view when deviating from the path (the map view). In the following, we will use the perspective projection because the image segmentation neural networks were trained on data in this form. We define a virtual camera pose in the vehicle’s frame,  $\mathbf{T}_{cv} \in SE(3)$ , which is used to render the images. The desired virtual camera pose is arbitrary, but aligning its position with the origin of the LiDAR sensor minimizes the amount of interpolation required in the final image. The first step is to align the point clouds in the camera frame. When a new live scan is received, an existing ICP-based localization module [31] is used to extract the relative pose of the vehicle in the map ( $\mathbf{T}_{vm} \in SE(3)$ ). The extrinsic transformation between the LiDAR sensor and the vehicle,  $\mathbf{T}_{vs} \in SE(3)$ , is known from calibration.

The monocular pinhole camera model [32] is used to render the virtual camera image in the camera frame. By convention, the  $z$ -axis extends forward from the camera. The pixel positions are

$$\begin{bmatrix} u_j \\ v_j \end{bmatrix} = \mathbf{g}(\mathbf{p}_j) = \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \end{bmatrix} \frac{1}{z} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (1)$$

The image is the union of the intensity values for each pixel in the map point cloud after it is transformed into the

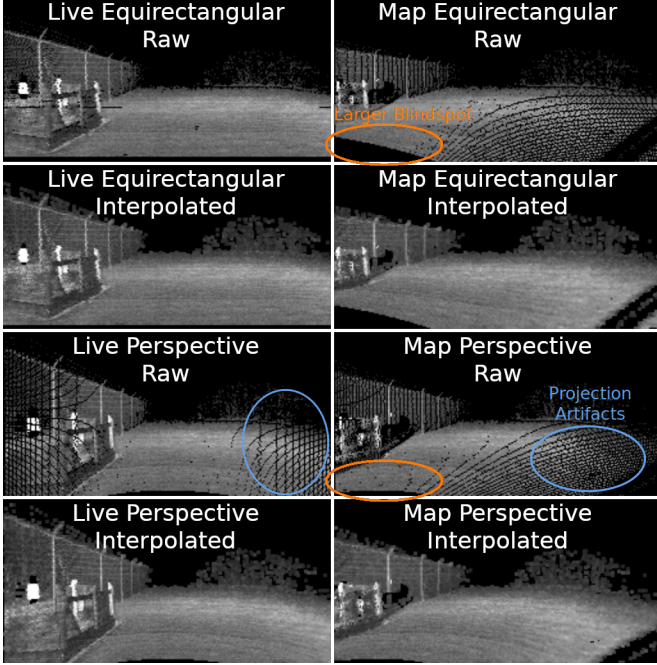


Figure 3. The left column contains equirectangular and perspective images aligned with the LiDAR. The right column shows the two projections with a two-meter lateral offset. The perspective view has a smaller blind spot around the base of the robot.

camera's frame:

$$I_{\text{map}} = \left\{ \begin{bmatrix} u_j \\ v_j \end{bmatrix} = \mathbf{g}(\mathbf{T}_{cv} \mathbf{T}_{vm_i} \mathbf{p}_{m_i}^j) \mid \mathbf{p}_{m_i}^j \in \mathcal{M}_i \right\}. \quad (2)$$

Similarly, the live scan is

$$I_{\text{live}} = \left\{ \begin{bmatrix} u_j \\ v_j \end{bmatrix} = \mathbf{g}(\mathbf{T}_{cv} \mathbf{T}_{vs} \mathbf{p}_{s_i}^j) \mid \mathbf{p}_{s_i}^j \in \mathcal{S}_i \right\}. \quad (3)$$

The image width ( $W$ ) and height ( $H$ ) are set as  $256 \times 128$  pixels based on the resolution of the OS-1 sensor. Horizontal and vertical fields of view are set to be  $\text{fov}_V \times \text{fov}_H = 90^\circ \times 45^\circ$ . An ideal, centred camera is used, leading to

$$c_u = W/2, \quad (4a)$$

$$c_v = H/2. \quad (4b)$$

The focal length is defined based on the desired image size and field of view as

$$f_u = \frac{W}{2 \tan(\text{fov}_H/2)}, \quad (5a)$$

$$f_v = \frac{H}{2 \tan(\text{fov}_V/2)}. \quad (5b)$$

Points beyond the field of view are ignored, and if multiple points are captured by the same pixel, the closest point to the camera is retained.

When a perspective projection is used, gaps in the image are created due to the distortion. These gaps are highlighted in the third row of Figure 3. Interpolating these regions

is critical for the segmentation to work properly. A  $3 \times 3$  kernel is used to interpolate the missing pixels, but only non-zero pixels are considered. The distortion is exacerbated by any offset of the virtual camera's origin from the LiDAR. Conceptually, the virtual camera could be colocated with the LiDAR's position in either the teach or repeat. However, we choose to locate it in the repeat frame because the map point clouds are an accumulation of points from sequential scans. The hue-saturation-value colour model is used to colour the images by range. The  $z$  coordinate in the camera frame is used for hue, mapping over a maximum range of 30 m. The saturation is the constant 255 for all pixels and the value is the LiDAR intensity of the pixel. The first column of Figure 4 shows two sample images rendered from the robot in a common frame.

### B. Segmenting Regions of Interest

After rendering both images in a common frame, segmentation occurs. The pre-trained model `sam_vit_b` [16] is used without fine-tuning for all experiments. SAM requires a geometric prompt on the image to define the anchor point(s) of the mask. To remain as general as possible, we use a single point-prompt for each semantic mask. Prompt selection is a critical aspect of the method. Each prompt adds run time to the model so we are constrained to select a few prompts, rather than a uniform grid that covers the whole image. To select prompts, two methods were compared. The first uses the norm of the pixel difference between the live image and the map image. The top- $k$  local maxima are selected as prompts. Additionally, a minimum distance constraint is imposed to ensure greater image coverage. The second method uses the  $k$ -largest centroids of connected components of the 3D nearest neighbours changes projected into the camera view.

### C. Detecting Changes

After independently segmenting the image pairs, the detected masks are compared. To decide if two masks capture the same object, the intersection-over-union (IoU) score is calculated between them. The bitwise AND ( $\wedge$ ) and OR ( $\vee$ ) operations are used to calculate the IoU of two binary masks ( $b1, b2$ ),

$$\text{IoU} = \frac{\sum_i \sum_j b1[i][j] \wedge b2[i][j]}{\sum_i \sum_j b1[i][j] \vee b2[i][j]}. \quad (6)$$

An IoU score of one indicates a perfect match and zero means no overlap between the individual masks. Masks with a maximum IoU of less than 0.5 are considered to be *changed* between the live scan and the map. A secondary 3D check is used to ensure that the point clouds intersect in 3D as well as in their projection. The remaining masks are considered *unchanged*. All points classified as *changed* in the live scan are then back-projected into 3D. The 3D bounding box, centroid, or raw points can be estimated and

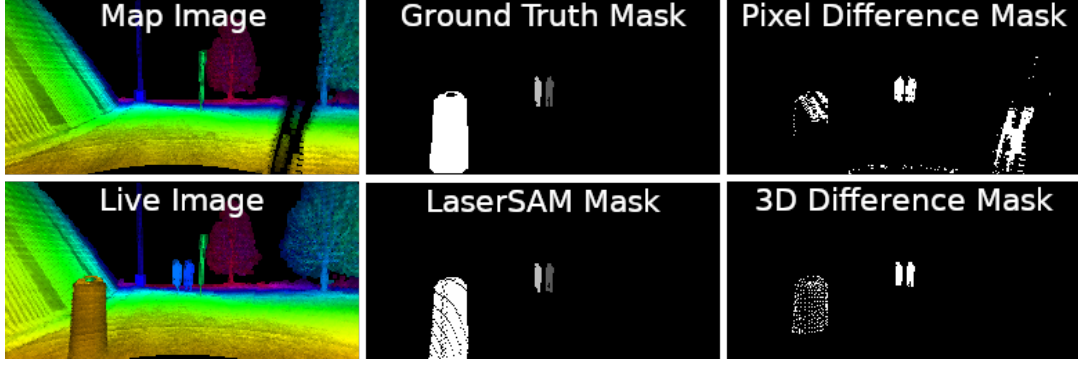


Figure 4. Change-detection masks for a sample frame. The left column shows the input images. There are three changed objects, two pedestrians and a static cone, marked in the ground truth. The segmentation results for each algorithm are shown in their respective panel.

provided to a local planner. A corridor-constrained sampling-based planner [7] inflates the *changed* points to account for the robot’s footprint and avoids them.

#### IV. EXPERIMENTS

Experiments were performed at the University of Toronto’s Institute for Aerospace Studies using a Clearpath Warthog UGV [33] equipped with an Ouster OS-1 128 beam LiDAR [25]. Data were recorded during night and day as well as during snowstorm conditions. Static obstacles placed on and around the original path and pedestrians walking near the robot were introduced as changes in the dataset. A mixture of natural and fabricated items was used to minimize the impact of the chosen materials’ reflectivity on the experiments. Additionally, duplicate items were placed near the path as part of the static scene during mapping. This demonstrates that the algorithm is performing change detection, not obstacle classification. Two different sequences were recorded with changes. The first sequence is 262 m long on relatively flat terrain. The second is 230 m through a wooded area. Changes to be detected include cones, mannequins, pedestrians and road signs.

The dataset has 12,012 live scans to use for testing. One hundred frames were randomly selected for manual annotation to provide ground truth masks. The ground-truth masks were instance segmented to allow for per-object metrics to be evaluated in addition to per-point ones. As discussed in the related work, the precise semantic meaning of a change is problem-dependent. In this dataset, only new

or moved obstacles that would impact the robot’s ability to drive were marked as changed. Small scene changes such as footprints in the snow, or tracks from previous drives were marked as *unchanged*.

##### A. Results

The pipeline was evaluated on the annotated subset of the frames. Two baselines were selected for comparison: a camera-style pixel difference method and a 3D-geometric detector with a Gaussian roughness model [6]. The pixel-wise IoU between the predicted and ground-truth segmentation masks is used for comparison. To quantify the effectiveness of each method for robot navigation, the detection performance is restricted to be within the allowable planning corridor. This planning-oriented metric is useful because errors in change detection that will not block the robot are not as critical to safe operation. The achieved results are summarized in Table I. All computational timing was performed on a laptop NVIDIA RTX A4500 GPU and an Intel i7-12800H CPU. LaserSAM was compared to both baselines, using the five most likely regions of change from each baseline as the input prompts. When prompted using pixel differences, LaserSAM significantly reduces the number of false positives, as demonstrated by the increase in precision. When prompted using 3D distances, there are fewer false positives but the mask-refining capabilities more accurately capture the whole object. This improvement is clear from the recall increase from 59.7% to 84.5% after corridor filtering.

Table I  
POINT-WISE COMPARISON OF LASERSAM TO CHANGE DETECTION BASELINE ALGORITHMS.

Method	Full Field of View			Corridor Filtered			Run Time (ms)
	IoU	Precision	Recall	IoU	Precision	Recall	
Pixel Difference Baseline	13.7%	15.3%	55.9%	21.5%	28.0 %	48.0%	5.7 ± 1.2
LaserSAM with Pixel Difference Prompts	28.2%	39.7%	49.4 %	47.1%	68.0%	60.5%	281.3 ± 27.9
3D Difference Baseline	56.5%	<b>95.5%</b>	58.0%	58.6%	<b>97.0%</b>	59.7%	80.6 ± 13.2
LaserSAM with 3D Difference Prompts	<b>73.3%</b>	86.5%	<b>82.8%</b>	<b>80.4%</b>	94.4%	<b>84.5%</b>	356.2 ± 30.9

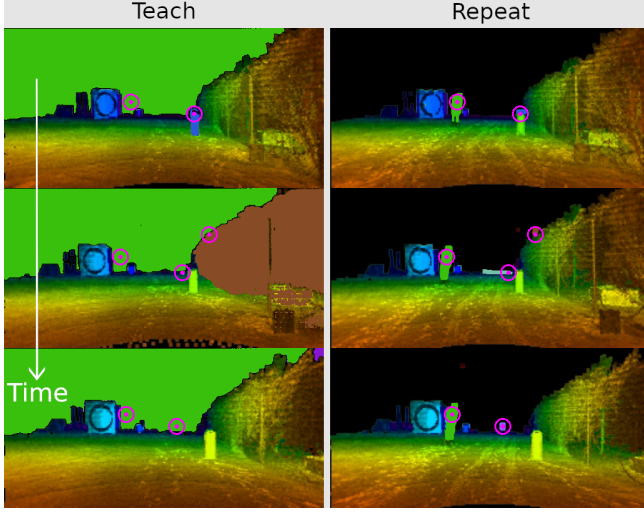


Figure 5. A temporal sequence of semantic regions generated by SAM [16]. The shared prompt point is highlighted in the pink bulls-eye. The mask colours match between the teach and repeat.

Figure 4 shows the segmentation result for a sample frame with two moving pedestrians and a static cone. LaserSAM generates the most accurate segmentation mask, provided that it has been prompted effectively. The gaps in the traffic cone correspond to projection artifacts. While interpolation is used to generate the mask, only those pixels that correspond to a 3D measurement are retained as changes. A secondary benefit of LaserSAM is the instance groupings of points. In Figure 4, each instance is coloured according to the ground truth instance that it matches most closely. The baseline methods provide only binary labels for each point in the scan.

Figure 5 shows an example sequence of masks generated by SAM. When changes are detected, the map image often has a large mask corresponding to the ground or sky. The masks from SAM tend to be consistent over time which is advantageous for robot motion planning. Prompts often exist near the boundaries, which is undesirable. Once the two masks are generated, the IoU threshold filter suppresses false positives.

### B. Closed-Loop Experiments on Warthog UGV

LaserSAM was packaged into C++ and integrated into the Visual Teach and Repeat 3 framework<sup>1</sup>. The run time of LaserSAM with 3D prompting is  $356 \pm 30.9$  ms, which is slower than real-time for the 10 Hz OS-1 LiDAR. A performance compromise is achieved by running lidar odometry at 10 Hz but only running change detection as frequently as possible. The 2 - 3 Hz update rate for the local costmap is fast enough to avoid static and slow-moving obstacles. Much of the computation occurs on the GPU allowing odometry on the CPU to proceed in parallel. The LiDAR

<sup>1</sup>[github.com/utiasASRL/vtr3](https://github.com/utiasASRL/vtr3)

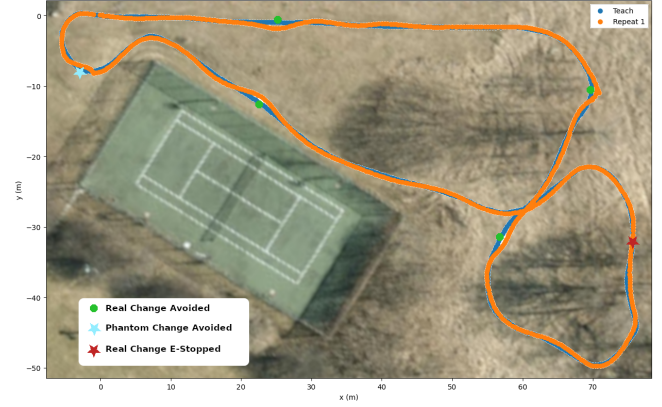


Figure 6. The path of the robot in orange as it detects and avoids changes in real time using LaserSAM. The blue line represents the obstacle-free teach path.

sensor has a near-range blindspot so the planner maintains previously detected changes in a queue, which adds temporal permanence that is not provided by LaserSAM.

In the sample trajectory shown in Figure 6, the robot's obstacle-free path is drawn in blue and is mostly covered by the orange repeat. The robot veers around changes that are detected along the path. Four changes are avoided correctly, and the robot returns to the obstacle-free path afterwards. One small cone was not detected and an emergency stop from the operator was required before continuing. Finally, in one location there was a false positive detection that caused an unnecessary evasive maneuver.

In another experiment, the path was repeated six hours after the mapping was performed after night had fallen. LaserSAM's performance is unaffected by the ambient illumination of the scene. Figure 7 shows the two virtual images in the afternoon and night. The colouring of the images is qualitatively indistinguishable. This stands in contrast to the conventional camera to which the cone is not visible.

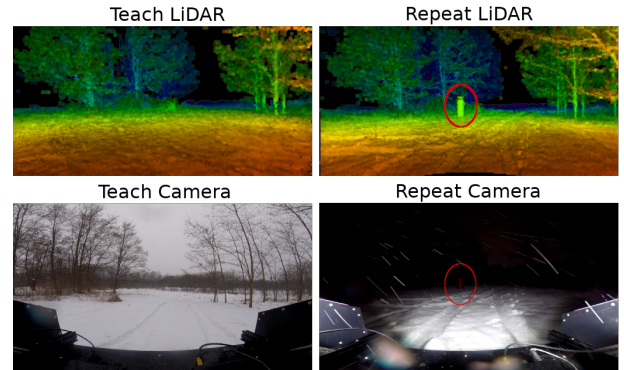


Figure 7. The rendered LiDAR images are not affected by ambient lighting conditions. The teach was performed during sunlight and the repeat was performed in the dark, and during heavier snow.

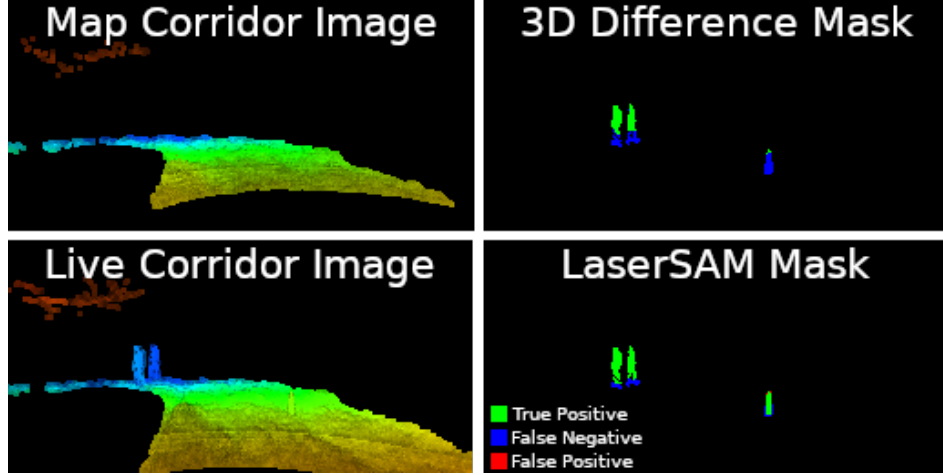


Figure 8. The change detection quality in the planned corridor between the 3D Difference Baseline and LaserSAM. LaserSAM uses semantic information to capture points closer to the object boundary with the map. This leads to more green pixels on all three objects near the ground.

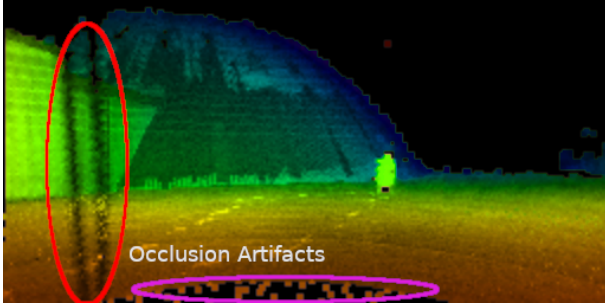


Figure 9. Range-coloured image highlighting the effect of the occlusions and submaps. In the red ellipse, a shadow in the scan is visible from the support strut of the robot that blocked the LiDAR in the teach. In the magenta ellipse, points are visible in the local blindspot of the sensor because the submap contains a union of points from multiple scans.

An attempt to use SAM for change detection on the RGB camera would fail under these conditions.

The two primary benefits of using segmentation over the baselines are a better definition of changes near the map and the suppression of noisy detections caused by mapping artifacts. Figure 8 shows a pedestrian that is partially segmented based on 3D distances, but using the SAM mask instead, the detection reaches the ground interface. Changes smaller than the 3D baseline’s distance threshold can be detected more accurately with LaserSAM because the intensity data defines their boundaries better than distances alone.

### C. Limitations

Similar to works that process point clouds directly, occluded regions remain a challenge. Figure 9 is rendered from a different pose than the LiDAR meaning that occluded parts of the scene are visible from the new view. This causes a shadow effect from the structure of the robot’s sensor supports. The further from the LiDAR’s origin that the virtual camera is rendered, the more gaps there are

in the visual frame. In closed-loop testing, this creates an undesirable feedback loop because when the robot deviates from the path to avoid obstacles detection accuracy can decrease. The submap has fewer occlusion-based gaps because it is constructed from sequential frames. Points within the magenta region of Figure 9 were appended to the local submap during the teaching process, filling in some of the holes caused by displacing the virtual camera. If storage was unlimited, retaining every point from nearby teach frames would allow for denser maps and minimize occlusion effects. In teach and repeat, the robot’s lateral path displacements are typically less than two meters, which minimizes these effects in practice.

### D. Future Work

To improve temporal consistency and add tracking, the centroid of previous obstacles will be projected into the camera at the next timestep and used as a prompt. Changes will likely remain visible in the scene, so prompting on their estimated location should improve temporal consistency. Inconsistent masks could be filtered out as spurious. Lastly, this work only considers positive changes: items added in the live scan. It should be possible to detect items that existed in the teach pass on the path that were removed. In a manual mapping process, it is unlikely that a robot will be driven over items that can disappear so this is left for future investigation.

## V. CONCLUSION

This paper demonstrates that developments in the intensity sensitivity of LiDAR scanners enable algorithms that leverage the benefits of multi-modal detection but use only a single LiDAR sensor. While the resolution of commodity LiDAR-rendered images lags behind conventional cameras, it has reached sufficient thresholds to apply vision algorithms

to field robots. The zero-shot generalization capabilities of the Segment Anything Model allow for visual change detection to be applied to simulated camera images created from LiDAR data. On the test set, using LaserSAM with 3D-based prompting achieves an IoU of 73.3%. When considering only obstacles within the allowable planning corridor performance is higher with an IoU of 80.4%. With LaserSAM running on a Clearpath Warthog UGV, static changes can be detected and avoided reliably. The operational frequency of 2.8 Hz is fast enough for driving up to 1 m/s. By suppressing spurious false-positive detections the local planner can find better routes. LaserSAM naturally works through illumination changes from night to day and day to night. Continued efficiencies in prompt generation and optimization for inference should help decrease the run time and allow LaserSAM to detect dynamic changes.

## REFERENCES

- [1] 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 Conf. on Multimedia*, New York, NY, USA, Oct. 2022, pp. 1279–1288.
- [2] R. Qin, J. Tian, and P. Reinartz, "3d change detection – approaches and applications," *ISPRS J. of Photogrammetry and Remote Sensing*, vol. 122, pp. 41–56, 2016.
- [3] P. Papadakis, "Terrain traversability analysis methods for unmanned ground vehicles: A survey," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 4, pp. 1373–1385, 2013.
- [4] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *J. of Field Robotics*, 2010.
- [5] L.-P. Berczi and T. D. Barfoot, "It's like déjà vu all over again: Learning place-dependent terrain assessment for visual teach and repeat," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 3973–3980.
- [6] Y. Wu, "VT&R3: Generalizing the teach and repeat navigation framework," Sep. 2022, MASC Thesis.
- [7] J. Sehn, J. Collier, and T. D. Barfoot, "Off the beaten track: Laterally weighted motion planning for local obstacle avoidance," *arXiv preprint arXiv:2309.09334*, 2023.
- [8] C. Stauffer and W. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747–757, 2000.
- [9] 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.
- [10] L. Ding, K. Zhu, D. Peng, H. Tang, K. Yang, and B. Lorenzo, "Adapting segment anything model for change detection in hr remote sensing images," *arXiv preprint arXiv:2309.01429*, 2023.
- [11] I. de Gélis, S. Lefèvre, and T. Corpetti, "Siamese KPConv: 3D multiple change detection from raw point clouds using deep learning," *ISPRS J. of Photogrammetry and Remote Sensing*, vol. 197, pp. 274–291, Mar. 2023.
- [12] A. Krawciw, J. Sehn, and T. D. Barfoot, "Change of scenery: Unsupervised lidar change detection for mobile robots," *arXiv preprint arXiv:2309.10924*, 2024.
- [13] Y. Chen, B. Xu, F. Dömbgen, and T. D. B. Barfoot, "What to learn: Features, image transformations, or both?" in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023.
- [14] 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, no. 17, p. 9877, 2023, number: 17 MDPI.
- [15] Ouster. How multi-beam flash lidar works. [Online]. Available: <https://ouster.com/insights/blog/how-multi-beam-flash-lidar-works>
- [16] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.
- [17] 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.
- [18] J. P. Underwood, D. Gillsjö, T. Bailey, and V. Vlaskine, "Explicit 3D change detection using ray-tracing in spherical coordinates," in *2013 IEEE International Conf. on Robotics and Automation*, May 2013, pp. 4735–4741, iSSN: 1050-4729.
- [19] I. de Gélis, 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.
- [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] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "Subsense: A universal change detection method with local adaptive sensitivity," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 359–373, 2015.
- [22] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," in *Proceedings of the Seventh IEEE International Conf. on Computer Vision*, vol. 1, 1999, pp. 255–261 vol.1.
- [23] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [24] 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.
- [25] "Ouster OS1 LiDAR," Available Online [<https://ouster.com/products/scanning-lidar/os1-sensor/>].
- [26] "Hesai LiDAR," Available Online [<https://www.hesaitech.com/>].
- [27] C. McManus, P. Furgale, and T. D. Barfoot, "Towards lighting-invariant visual navigation: An appearance-based approach using scanning laser-range finders," *Robotics and Autonomous Systems*, vol. 61, no. 8, pp. 836–852, 2013.

- [28] S. Anderson and T. D. Barfoot, "Ransac for motion-distorted 3d visual sensors," in *2013 IEEE/RSJ International Conf. on Intelligent Robots and Systems*. IEEE, 2013, pp. 2093–2099.
- [29] C. McManus, P. Furgale, B. Stenning, and T. D. Barfoot, "Lighting-invariant visual teach and repeat using appearance-based lidar," *J. of Field Robotics*, vol. 30, no. 2, pp. 254–287, 2013.
- [30] T. D. Barfoot, C. McManus, S. R. Anderson, H. Dong, E. Beerepoot, C. H. Tong, P. T. Furgale, J. D. Gammell, and J. Enright, "Into darkness: Visual navigation based on a lidar-intensity-image pipeline," in *International Symposium of Robotics Research*, 2013.
- [31] 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, no. 4, pp. 10 328–10 335, Oct. 2022.
- [32] T. D. Barfoot, *State Estimation for Robotics: Second Edition*, 2nd ed. Cambridge University Press, 2024.
- [33] "Clearpath Robotics Warthog UGV," 2020. [Online]. Available: <https://clearpathrobotics.com/warthog-unmanned-ground-vehicle-robot/>