PlaneSDF-Based Change Detection for Long-Term Dense Mapping

Jiahui Fu, Chengyuan Lin, Yuichi Taguchi, Andrea Cohen, Yifu Zhang, Stephen Mylabathula, and John J. Leonard

***Abstract*—The ability to process environment maps across multiple sessions is critical for robots operating over extended periods of time. Specifically, it is desirable for autonomous agents to detect changes amongst maps of different sessions so as to gain a conflict-free understanding of the current environment. In this letter, we look into the problem of change detection based on a novel map representation, dubbed Plane Signed Distance Fields (PlaneSDF), where dense maps are represented as a collection of planes and their associated geometric components in SDF volumes. Given point clouds of the source and target scenes, we propose a three-step PlaneSDF-based change detection approach: (1) PlaneSDF volumes are instantiated within each scene and registered across scenes using plane poses; 2D height maps and object maps are extracted per volume via height projection and connected component analysis. (2) Height maps are compared and intersected with the object map to produce a 2D change location mask for changed object candidates in the source scene. (3) 3D geometric validation is performed using SDF-derived features per object candidate for change mask refinement. We evaluate our approach on both synthetic and real-world datasets and demonstrate its effectiveness via the task of changed object detection.**

***Index Terms*—Mapping, SLAM, range sensing.**

I. Introduction

THE ability to perform robust long-term operations is critical in many robotics and AR/VR applications, such as household cleaning and AR/VR environment scanning. Through multiple traverses of the same place, agents accumulate a more holistic understanding of their working environments. However, in the long-term setting, the working environment is prone to changes over time, e.g., the removal of a coffee mug. Conflicts may then arise when agents try to synthesize scans from different sessions. Therefore, agents are expected to first capture these changes and then obtain the up-to-date 3D reconstruction of the scene after all change conflicts have been resolved.

An intuitive way to conduct change detection is through scene differencing between the two reconstructions of interest. Previous works on change detection leverage scene representations such as point clouds [1]–[4] or Signed Distance Fields (SDF) [5]–[7] and perform point- or voxel-wise comparison [1]–[3], [8] *globally* between the two scenes. To ensure that comparison is carried out at corresponding locations of the two observations, these methods demand consistent and precisely aligned reconstructions, which are hence susceptible to sensor noises and localization errors.

We observe that most scene changes occur at the object level, and that man-made environments can often be modeled as a set of planes with objects attached to them, as opposed to a cluster of unordered points or voxels with no geometric structure. Therefore, we choose to represent the whole scene as a set of planes, each having an associated SDF volume that describes the geometric details of the objects attached to it, which we term as the PlaneSDF representation. Similiar to the idea of dividing the whole environment into submaps, e.g., based on time intervals [7] or objects [6], [9], agents could maintain multiple PlaneSDF volumes of scalable sizes in lieu of a single chunk of global SDF while saving update and memory reload time by updating volumes only in the current viewing frustum. Furthermore, this representation is also more robust to localization drift as local regional correction can be performed patch by patch each time two planes from different traverses are registered via plane pose.

Taking advantage of the PlaneSDF representation, in this paper, we propose a change detection algorithm given a source and a target scene that decomposes the original global comparison in a local plane-wise fashion. Treating each plane as a separator, the local change detection is performed plane-wise as well as at the object level. The global localization drift issue between two scenes is alleviated during plane-pose registration. Through the projection of SDF voxel height values onto the plane, the obtained height map and its value connectivity offers a solid indication about the potential object candidates along with their projected 2D contours, making it possible to conduct 3D geometric validation only on SDF voxels belonging to the potentially changed objects. Our main contributions are as follows:

1) PlaneSDF is proposed as a novel representation for indoor scene reconstruction.
2) A change detection algorithm, consisting of 2D height map comparison and 3D geometric validation, is developed leveraging the PlaneSDF data structure.

Manuscript received 24 February 2022; accepted 26 June 2022. Date of publication 18 July 2022; date of current version 26 July 2022. This letter was recommended for publication by Associate Editor N. Atanasov and Editor J. Civera upon evaluation of the reviewers' comments. The work was done when Jiahui Fu was an intern at Meta. *(Corresponding author: Jiahui Fu.)*

Jiahui Fu and John J. Leonard are with the MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139 USA (e-mail: jiahuifu@mit.edu; jleonard@mit.edu).

Chengyuan Lin, Yuichi Taguchi, Andrea Cohen, Yifu Zhang, and Stephen Mylabathula are with Meta, Menlo Park, CA 94025 USA (e-mail: chengyuanlin@fb.com; yuichi.taguchi@gmail.com; andy.cohen@fb.com; yifuzhang@fb.com; mylabathula@fb.com).

Supplementary video: https://ieeexplore.ieee.org.

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2022.3191794, provided by the authors.

Digital Object Identifier 10.1109/LRA.2022.3191794

3) The effectiveness of the proposed algorithm is demonstrated on both synthetic and real-world datasets of indoor scenes.

II. Related Work

Change detection, as widely discussed in research concerning long-term robotic operations, can be roughly divided into two categories: geometric and probabilistic approaches.

A. Geometric Approaches

Geometric approaches are usually based on comparing geometric features extracted from various environment representations. Walcott-Bryant *et al.* [11] developed Dynamic Pose Graph SLAM, where change detection is performed on the 2D occupancy grid to edit and update the pose graph. Classical 2D feature descriptors, e.g. SURF, ORB, and BRISK [12], [13], were extracted from the grey scale input images and the visual database, respectively. Next, the Euclidean distance between the two features is computed to determine if changes have taken place. There are also many works in the literature which use 3D representations. Finman *et al.* [1] performed scene differencing on depth data among multiple maps and learned segmentation models with surface normals and color edges to discover new objects in the scene. Ambrus *et al.* [2] computed a meta-room reference map of the environment from the collected point cloud, and employed spatial clustering based on global descriptors to discover new objects in the scene. Fehr *et al.* [8] adapted volumetric differencing onto a multi-layer SDF grid and showed its effectiveness in object discovery and class recognition. Kunze *et al.* [14] built and updated a hierarchical map of the environment by comparing object positions between observations and corresponding map contents. Schmid *et al.* [9] proposed a panoptic map representation using multiple Truncated Signed Distance Fields for each panoptic entity to detect long-term object-level scene changes on-the-fly. Langer *et al.* [10] combined semantic as well as supporting plane information, and conducted local verification (LV) to discover objects newly introduced into the scene. The proposed method outperforms several global point- and voxel-based approaches and is selected as the baseline here for comparison.

B. Probability-Based Approaches

Previous works in this category tend to develop statistical models to describe sensor measurement or environment dynamics. Krajnik *et al.* [15] modeled the environment's spatio-temporal dynamics by its frequency spectrum, while [3], [16] exploited probabilistic measurement models to indicate how likely it is for each surface element in the scene to have moved between two scenes. Bore *et al.* [17] proposed a model for object movement describing both local moves and long-distance global motion. Katsura *et al.* [18] converted point clouds and measured data into ND (Normal Distribution) voxels using the Normal Distribution Transform (NDT) and compared voxel-wise distribution similarity.

There are also learning-based change detection approaches [19], [20] that learn geometric features through neural networks trained on pre-registered images or SDF pairs. Considering the potential challenges of training data availability and generalization to unseen changes, this paper focuses only on non-learning based methods.

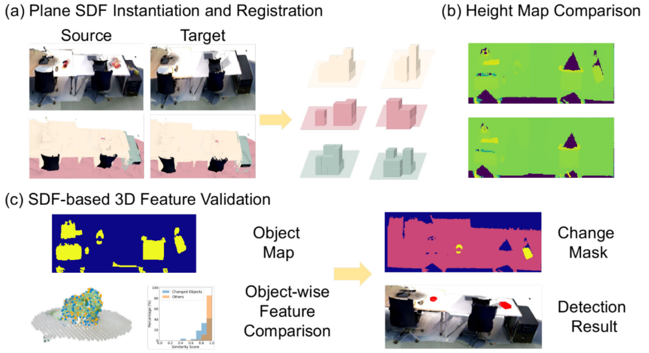

(a) Plane SDF Instantiation and Registration
Source
Target
(b) Height Map Comparison
(c) SDF-based 3D Feature Validation
Object Map
Change Mask
Object-wise Feature Comparison
Detection Result

Fig. 1. System Overview. Input: point clouds of the source and target scene. Output: voxels of objects detected as changes between the two scenes. (a) For the two input point clouds, PlaneSDF volumes are fused and registered using poses of major planes (e.g., desk, cabinet, and the floor, as indicated in different colors). A 2D height map and an associated object map are obtained for each plane through projection and connected component analysis. (b) Height values for corresponding planes are compared, which yields a preliminary 2D change mask for the source plane w.r.t. the target plane. (c) The intersection of the current change mask and the source object map is found to determine changed object candidates. Each of these objects has its SDF-based features extracted and compared against the corresponding one in the target for change mask refinement.

Despite all the results reported, the global point- or voxel-wise geometric comparisons are susceptible to sensor noises and localization errors and the results of probabilistic approaches may not be readily applicable to scene mapping tasks. Hence, in this work, we consider 2D as well as 3D information on the voxel and object level with the proposed PlaneSDF structure, and achieves robust change detection on both synthetic and real-world datasets.

III. Method Overview

Our method (see Fig. 1) leverages the plane-to-object supporting structure through the PlaneSDF representation, thereby enabling us to first perform local pairwise plane pose alignment against global reconstruction errors. We then obtain change detection results via efficient and effective local scene comparison on 2D height map and 3D object surface geometry informed by the SDF volume.

A. PlaneSDF Instantiation

We first generate the PlaneSDF representation for each scene, i.e., representing the input 3D point cloud for the scene as a set of planes and their associated SDF volumes.

For plane detection, when given sequential point cloud streams, we extract planes from each frame with RANSAC and merge them when a new frame arrives, as how SLAM systems commonly proceed when using planes as pose estimation constraints [21]–[23]. When a point cloud for the complete scene is available, we run a spatial clustering algorithm [24] to detect a set of planes out of the cloud.

For each plane detected, we fuse an SDF volume using all the points within a predefined distance to the plane, in the hope that the obtained SDF will record the free space and object geometry solely from objects directly supported by the plane, e.g., the

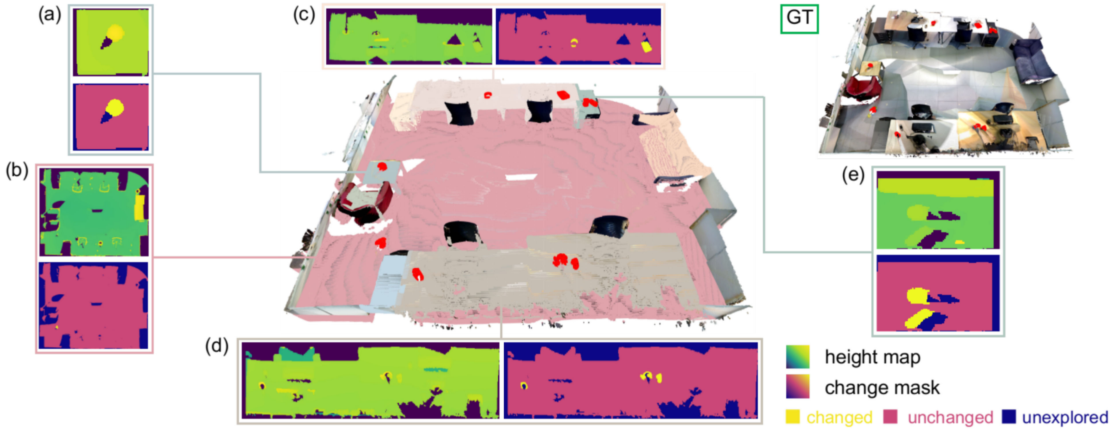


Fig. 2. Change detection results for a complete indoor scene from the object change detection dataset [10]. The whole scene is spatially subdivided into multiple PlaneSDF instances (marked by distinct colors). Note that there could be some overlap among certain SDF volumes (e.g., the seating area of the sofa in the upper right of the scene is also fused into the floor volume). For each plane of interest, i.e., planes with objects newly introduced onto them, the associated height map and the final change mask are shown. The detected object changes are colored in red while the ground truth (GT) changes are rendered in the upper right corner of the figure.

drawings hanging on the vertical wall or the soda can placed on the table. Note that when two detected planes are less than the defined fusing distance away from each other or there are bigger objects supported by multiple planes, a point could be fused into multiple PlaneSDF instances, e.g., the color overlap of the sofa and the floor instances in Fig. 2). We also limit our detection of planes to only horizontal and vertical ones, as they constitute most of the "plane-supporting-objects" cases we encounter in daily lives.

Furthermore, the local 2D height grid map evaluated w.r.t. the plane is computed, where each grid stores the maximum voxel-to-plane distance in the height direction at the current plane location. The height map is non-zero for plane locations occupied by objects, zero for flat unoccupied locations, and -1 for unobserved regions. Building on top of this, as non-zero regions are disconnected from each other by the plane zero-level set, we could easily obtain an "object (or object cluster for multiple small objects close to each other) map" [Fig. 1(d)] preserving relatively accurate object contours through connected component labeling on the height map.

Given two PlaneSDF volumes, a source and a target, instantiated from the two scenes respectively, we define the 2D change mask of the pair as a ternary mask of the same size as the *source* height map, indicating all changed plane locations in the *source* w.r.t. the *target* (Fig. 2).

B. PlaneSDF Registration

Before scene differencing is conducted, PlaneSDF volumes of the two scenes are first registered so that the comparison is guaranteed to be carried out on two observations of the same plane. With the assumption that input point clouds from different sessions share the same world coordinate frame, registration of PlaneSDF volumes is accomplished through plane poses to alleviate the effect of localization drift among reconstructions of the same plane. For each pair of PlaneSDFs, we determine if they belong to the same plane according to the orientation cosine similarity and offset difference of the two plane poses:

$$\begin{aligned} \mathbf{n}^{\mathbf{T}}\mathbf{n}' &\geq \delta_{\mathbf{n}} \\ ||d - d'|| &\leq \delta_d, \end{aligned} \tag{1}$$

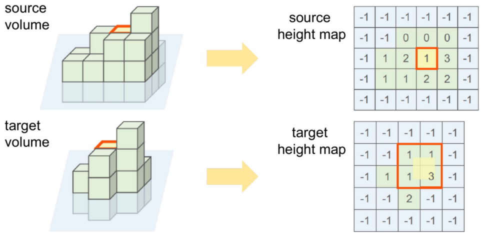
where $(\mathbf{n}, d)$ and $(\mathbf{n}', d')$ are the plane surface normals and offsets from origin of the source and target PlaneSDF volumes, respectively. $\delta_{\mathbf{n}}$ and δ_d are the minimum cosine similarity and maximum offset distance for two planes to be regarded as the same plane. In this way, via associating plane detections of similar orientations and offsets in the pair of reconstructions, small localization drift of the same plane can be mitigated by applying the relative transform between plane poses, from which we are then ready for change detection on each registered PlaneSDF pair.

C. Height Map Comparison

As floating objects are rare in daily scenes, height value discrepancy at the same plane location in different observations can offer informative speculation about the changes on this plane, e.g., when objects are newly removed or added, drastic changes between zero and non-zero height values will occur. In this spirit, we project each location, (x, y), of the source height map H onto the target height map H' using the relative plane pose. If the height value variation is above a threshold δ_h, we mark this plane location as *changed* (see Fig. 3). Oftentimes, the projected location, $(x,' y')$, will not not land exactly onto a grid center in the target map, so comparisons are drawn between the source height and those of the four nearest neighbors of $(x,' y')$:

$$\sum_{i=0,1;j=0,1} \mathbb{1}(|H'(\lfloor x' \rfloor + i, \lfloor y' \rfloor + j) - H(x, y)| \leq \delta_h) = \begin{cases} 0, & \text{changed} \\ \geq 1, & \text{unchanged}. \end{cases} \tag{2}$$

In most cases, as a consequence of measurement noises, the change mask obtained after direct comparison is usually corrupted by small false positive clusters scattered around the map.



-1	-1	-1	-1	-1	-1
-1	-1	0	0	0	-1
-1	1	2	1	3	-1
-1	1	1	2	2	-1
-1	-1	-1	-1	-1	-1

-1	-1	-1	-1	-1
-1	-1	1	1	-1
-1	1	1	3	-1
-1	-1	2	-1	-1
-1	-1	-1	-1	-1

Fig. 3. Height map comparison. For the registered source and target PlaneSDF pairs, each grid in the source height map is projected onto the target height map, with its height value compared against those of its closest 2×2 neighborhood. If all four neighbors have a height difference above a threshold, this grid (plane location) is preliminarily marked as changed.

Therefore, a round of connected component filtering followed by dilation is applied to remove the noise.

D. 3D Voxel Validation

Comparing height values for changes works well when (1) objects are removed or added, inducing significant variation in height values, or (2) camera trajectories have a high observation overlap of the unchanged objects between two runs. However, height implications can fail easily when old objects are replaced with new ones in the same place, or different parts of the same unchanged object are observed due to disparate viewing angles.

Therefore, 3D validation on the SDF of potential changed source plane locations is introduced with the goal of correcting false positives indicated by the change mask. For the overlapping space of two observations, if the same object persists, then the local surface geometry and free space description should be similar, or the target SDF will otherwise be remarkably different from that of the source.

Here, for the sake of selecting key voxels and obtaining corresponding descriptive geometry characterization around the selected locations, the curvature-derived description of the SDF is adopted for its capability to characterize the geometry of both object surfaces and the unoccupied space in between. In addition to indicating the planarity, convexity, or concavity of the object surface, the trend of SDF variation amid object surfaces can reflect inter-surface spatial relations, e.g., the sudden drop of an increasing SDF value along a ray direction can imply the switch of the nearest reference surface for SDF value calculation as the ray marches through surfaces. In contrast, the raw SDF value description and its gradient-derived counterpart are less suitable for the unified goal of key voxel selection and local geometry description. The former, due to the unavailability of ground truth surfaces during point fusion, is prone to slight inconsistency when constructed from different camera trajectories, while the latter returns an indistinguishable magnitude of one by construction in most places.

Additionally, to make the comparison more robust to measurement noises and reconstruction errors, the SDF voxels of interest are extracted and compared in the minimal unit of an object (cluster). This is achieved by selecting voxel blobs in each source PlaneSDF as those whose 2D projected clusters from the change mask have high overlap with the connected clusters in the object map, i.e., the intersection of the change mask and the object map. Through per-blob 3D geometry validation, the final change mask not only preserves a more detailed object contour in cases of adding/removing an object to/from a free space, but also self-corrects false per-voxel height variation induced by sensor noises in a clean way.

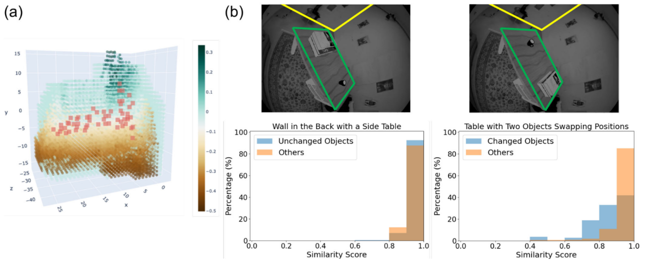


Fig. 4. Key voxel distribution and corresponding similarity score distribution of planes with and without changes. (a) Key voxel (red square dots) distribution within a voxel blob (round dots with colors indicating the SDF value). (b) Key voxels within the same PlaneSDF volume are classified as either "part of an object" or "others" as everything left in the background. Left (PlaneSDF of the yellow plane): Both the side table (object) and the wall (others) are unchanged, hence both similarity scores bias towards higher-valued bins. Right (PlaneSDF of the green plane): The book stack and the coffee mug swap their positions on the table. Their shape distinction leads to scattered distribution of voxel similarity scores at the same 3D position, while the "other" unchanged voxels around the tabletop plane still share high similarity.

1) Key Voxel Selection: Key voxels are selected per object blob so as to offer a more compact and robust characterization of the overall blob shape. Inspired by [25], voxels around regions of high curvature are selected as key voxels, implying neighborhoods of significant shape variations (see Fig. 4(a). We adopt the measure of local extrema of the determinant of Hessian (DoH), $det(Hess(\boldsymbol{v}))$, and calculate the Hessian matrix within a complete $3 \times 3 \times 3$ neighborhood $\mathcal{N}$:

$$Hess(\boldsymbol{v}) = \begin{bmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{yx} & s_{yy} & s_{yz} \\ s_{zx} & s_{zy} & s_{zz} \end{bmatrix}$$

$$s_{ij} = (\mathbf{G}_j * \mathbf{G}_i)(\Phi(\boldsymbol{v})) \quad i, j = x, y, z, \tag{3}$$

where each element s_{ij} in the Hessian matrix of $\boldsymbol{v}$ is obtained via convolution of $\Phi(\boldsymbol{v})$, the $3 \times 3 \times 3$ SDF neighborhood at $\boldsymbol{v}$, with the 3D Sobel filter $\mathbf{G}$ in turn in the i and j direction.

2) Per-Voxel Shape Description: For each key voxel $\boldsymbol{v}_0$ in the object blob $\mathcal{O}$, the three eigenpairs of the Hessian matrix, $\boldsymbol{p}_i = (\lambda_i, \boldsymbol{e}_i), i = x, y, z$, are computed and represent the three principal curvatures (λ_is) and their directions ($\boldsymbol{e}_i$s) at $\boldsymbol{v}$, respectively. This operation is then repeated for each voxel in $\mathcal{N}$ and its corresponding neighborhood $\mathcal{N}'$ in the target map (determined by its projected location $\boldsymbol{v}'$ in the target map). The three eigenvalues are normalized for numerical stability and each principal direction vector $\boldsymbol{e}_i$ is converted into spherical coordinate (θ_i, ϕ_i).

We then construct eigenpair histograms, $\mathcal{H}$ and $\mathcal{H}'$, for the corresponding neighborhood $\mathcal{N}$ and $\mathcal{N}'$. For neighborhood $\mathcal{N}$, we compute three sub-histograms, h_is, for all the eigenpairs $\boldsymbol{p}_{j_i}$

in the i direction, where $i = x, y, z$:

$$\begin{aligned} \boldsymbol{p}_{j_i} &= [\theta_i, \phi_i, \lambda_i], \boldsymbol{p}_j \in \mathcal{N} \\ &\Rightarrow h_i \in \mathbb{R}^{N_\theta \times N_\phi \times N_\lambda} \\ \theta_{h_i} &= [0, 180^\circ], \phi_{h_i} = [-90^\circ, 90^\circ] \\ \lambda_{h_i} &= [\min_{j\in\mathcal{N}}(\lambda_{j_i}), \max_{j\in\mathcal{N}}(\lambda_{j_i})], \end{aligned} \tag{4}$$

where N_θ, N_ϕ, and N_λ are the number of bins, and θ_{h_i}, ϕ_{h_i}, and λ_{h_i} are the bin threshold in the θ, ϕ, and λ directions for h_i, respectively. With each h_i of dimension $N_\theta \times N_\phi \times N_\lambda$, we then concatenate the three to form the final histogram, $\mathcal{H} = [h_1||h_2||h_3]$, describing the local shape distribution around this key voxel in the source. The corresponding $\mathcal{H}'$ for the target neighborhood is computed in the same fashion, while sharing all the histogram thresholds with those of $\mathcal{H}$.

To further enhance its ability of characterizing local shapes, we append the final histogram with a weighted signed distance value s of the neighborhood. The weights are assigned with a Gaussian filter centered at $\boldsymbol{v_0}$ with deviation of $\sigma = 2$, and the weighted SDF s is computed as follows:

$$\begin{aligned} w_i &= \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(\boldsymbol{v}_i - \boldsymbol{v}_0)^2}{2\sigma^2}}, \boldsymbol{v}_i \in \mathcal{N}(\boldsymbol{v}_0) \\ s &= \frac{\sum w_i \Phi(\boldsymbol{v}_i)}{\sum w_i}. \end{aligned} \tag{5}$$

Thus the ultimate feature vector for the key voxel in the source is $f(\boldsymbol{v}_0) = [\mathcal{H}, s]$, which is of dimension $3 \times N_\theta \times N_\phi \times N_\lambda + 1$. We define a similarity score, $sim \in (0, 1)$, at this key voxel between the two features, f and f', of the source and target map, respectively, as:

$$sim(f, f') = 1/(1 + \alpha \|f - f'\|_2), \tag{6}$$

where $f'(\boldsymbol{v}_0) = [\mathcal{H},' s']$ and α is a coefficient for adjusting the contribution of the Euclidean distance between f and f', $\|f - f'\|_2$, to the similarity score.

3) Per-Object Shape Comparison: The distribution of the similarity scores for all key voxels in the current object blob then makes it possible to determine if the space is occupied by the same object across two sessions. We argue that for an *unchanged* space occupied with the same object blob, the similarity scores, as an indication of the local shape, should be concentrating around higher values, whereas for a space with objects later removed, added, or replaced by another object, they should either be low (removed or added) or distributed more evenly around a wider range of bins (replaced) [see Fig. 4(b)]. Therefore, we construct the similarity score histogram for the object blob and compute the histogram mean to determine if the object has changed:

$$\begin{aligned} H_{avg} &= \sum m_i n_i / N \\ isChanged(\mathcal{O}) &= \mathbb{1}(H_{avg} < \delta_{blob}), \end{aligned} \tag{7}$$

where m_i and n_i are the midpoint value and frequency of each bin i, and N is the total number of key voxels in this object blob. The object is then validated as changed if H_{avg} is below a similarity threshold, δ_{blob}, or false positives from 2D comparison can be corrected based on the relatively high H_{avg} value.

Following the plane locations marked as changed in the change mask, all the corresponding voxels along the height direction are extracted, which are the changed part of the source scene w.r.t. the target.

IV. Experiments and Results

In this section, we evaluate our approach on both synthetic and real-world indoor datasets, and demonstrate its strength via tasks revolving around object-level change detection.

A. Datasets

1) Synthetic Tabletop Dataset: For evaluations under controlled environments, we generated synthetic indoor sequences with known object models on a tabletop. We first scanned a static, furnished room with a Lidar scanner to obtain a ground-truth 3D point cloud of the room. A few synthetic daily objects, e.g. mug and book stack, are then arbitrarily placed on a synthetic table in the scene, which are added, removed, or moved across multiple sequences, thus creating the desired changes to be detected. The scenes are rendered by simulating cameras on the Oculus Quest 2 headset moving in a preset trajectory around the table, from which per-frame 3D point cloud observations were generated and used as the input to our algorithm.

2) Object Change Detection Dataset: The object change detection dataset [10] is recorded with an Asus Xtion PRO Live RGB-D camera mounted on an HSR robot, consisting of multiple complete or partial point clouds of five scenes: big room, small room, kitchen, office, and living room. Each scene consists of a reference reconstruction and 5 to 6 other reconstructions obtained using Voxblox [26], accompanied by various levels of permanent structure misalignment and noisy boundaries due to localization and reconstruction errors. Ground truth annotation of 3 to 18 newly introduced YCB [27] objects to the scene is provided.

B. Evaluation Metrics

We adopt the commonly used precision and recall rates as the metrics for change detection evaluation.

For the object change detection dataset, following the measures in [10], we compute precision, recall rate, and F1 score at the *point* level, based on the ground truth changed point annotation and our detection results. Precision is computed as the proportion of total number of detected points that correspond to the ground truth, and recall rate is defined as the proportion of ground truth points that are incorporated in the detection points. The F1 score provides the harmonic mean of the two metrics. Two other metrics, the number of missing objects (changed objects with no points detected as changed) and wrongly detected clusters (clusters generated by the method that do not overlap with any changed objects) are also reported so as to better manifest the approach's performance on the object/cluster level.

C. Implementation Details

We follow the procedures described in III-A for generating PlaneSDF instances, with the RANSAC-based approach for data streams of the synthetic tabletop dataset and the clustering-based approach for scene point clouds of the object change detection dataset. We set the fusing threshold to include points within 0.3 m from the plane, hoping to cover most of the easy-to-move daily objects supported by a plane. The SDF voxel grid resolution is set as 7 mm so as to best preserve the scene geometry, especially for smaller objects.

For PlaneSDF registration, the minimum cosine similarity and maximum offset distance are set as $\delta_{\mathbf{n}} = 0.95$ and $\delta_d = 0.2$ m.

For change detection, the height map difference threshold is set to be $\delta_h = 0.02$ m so as to not miss smaller objects. To construct the 3D feature histogram for each area of interest, the number of bins along each dimension is set to be $N_\phi = 5, N_\theta = 5, N_\lambda = 6$. The α and the threshold δ_{blob} are set as $(\alpha, \delta_{blob}) = (2, 0.9)$ for the synthetic dataset, and further moderately tuned for the object change detection dataset to accommodate certain dataset-defined cases where some slightly moved planes are not marked as changed.

D. Results on the Synthetic Tabletop Dataset

The tabletop dataset captures a relatively complete surrounding view of the various objects on a tabletop, which provides a simple yet effective scene for initial evaluation of the proposed algorithm. The experiments are run on 20 arbitrarily selected source-target sequence pairs, with objects on the tabletop ranging from coffee mug (5- cm in height), toy car (10 cm in height), to 3-layer book stack (30+ cm in height), etc. The output is the 2D change mask of the same size as the height map of the source PlaneSDF volume, indicating all the changed locations on the source plane w.r.t. the target. To prove the robustness of our algorithm, we also run all the experiments in a bi-directional fashion, i.e., detecting changes source-to-target as well as target-to-source. With relatively complete observation of all the tabletop objects, for the 20 pairs we have tested, the algorithm is able to achieve 100% recall and 80% precision rate for detecting changed objects without 3D geometric validation. The precision rate further rises to 100% after incorporating 3D validation, where false positive height differences are corrected by verifying the shape similarity in the SDF field [as for the case of the book stack shown in Fig. 5(b)].

Fig. 5 shows examples of the evolution of change masks out of each stage in the proposed method for three common object changing scenarios: (a) Two objects swap places. (b) One object changes and one remains. (c) Objects are added/removed to/from a free space. We can see that the masks out of height map comparison (3^{rd} column) still contains noisy false positive (FP) clusters, as a consequence of reconstruction errors. The smaller FP clusters are then partially removed by connected-component filtering and dilation, as shown in the 4^{th} column, but bigger FP patches still persist, such as the book stack on the left side of the tabletop in scenario (b). The 3D validation here then plays a significant role in comparing the 3D geometric similarity of all the possible patches and effectively reverting the FP book stack back to unchanged (5^{th} column in (b)). The results also demonstrate bi-directional robustness as the change masks are of similar pattern within each source-target pair.

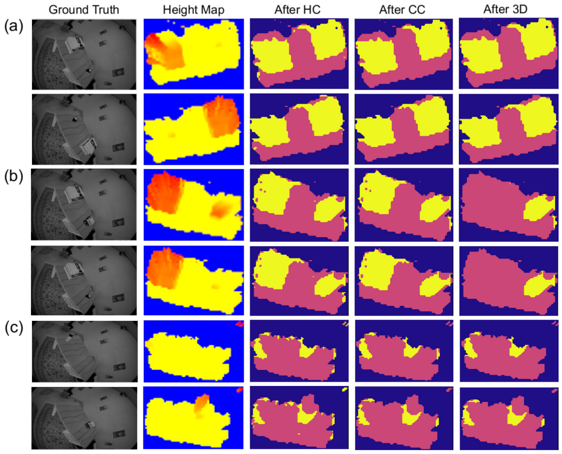


Fig. 5. Sample change detection results on the synthetic tabletop dataset. Each mask showcases the change detection result of treating the sequence in the same row as the source. Here we include snapshots of the actual scene in the first column, the associated height map in the second column, and the evolution of the change mask out of each stage of our approach in the last three columns: (1) height map comparison (HC) (2) connected component filtering and dilation (CC) (3) 3D geometric validation (3D).

E. Results on the Object Change Detection Dataset

In addition to the synthetic tabletop dataset, we further evaluate our algorithm on the more challenging real-world object change detection dataset, which offers scene settings with object changes of more diverse sizes and layouts.

Quantitatively, Table I compares the results of our approach in terms of the five metrics against those of the volumetric/point-based approaches Octomap [28] and Meta-room [2], and the best results of the approach proposed by [10]. The results are computed by projecting the ground truth point clouds into SDF voxels and determining the change state of each point according to that of its corresponding voxel indicated by the 2D change mask from our approach. Note that following dataset definition, we manually exclude all detected changed points resulting from moved furniture and decoration from evaluation.

Moreover, to demonstrate the effectiveness of our blob-level curvature-based SDF description for robust change detection, we provide another baseline (*FPFH* in Table I) with a point-wise variant of the proposed method by replacing the 3D voxel validation step III-D with the point-based FPFH [29] feature matching using the Open3D [30] implementation. As our selected key voxels are *not* located on object surfaces, where off-the-shelf point feature extractors cannot be directly applied, FPFH features are extracted for every point in the original point cloud that contributes to the fusion of the SDF. A point is marked as changed if its source FPFH feature cannot be matched in its target neighborhood.

Here, Fig. 6 illustrates the key voxel distribution and the false positive points detected by FPFH matching for two unchanged sub-scenes: a single green object and two bottle standing closely against a wall. In (b), near-surface key voxels (within 1.5 SDF voxel size to an object point, shown in blue) are distributed around the object surface, giving good characterization of the object geometry, while key voxels farther away from the

TABLE I

RESULT COMPARISON OF THE PROPOSED APPROACH WITH THREE BASELINES PROVIDED BY THE OBJECT CHANGE DETECTION DATASET. BEST VALUES ARE MARKED IN BOLD. (PR = PRECISION, RE = RECALL, F1 = F1 SCORE, M = MISSED OBJECTS, W = WRONGLY DETECTED CLUSTERS)

	Small Room					Big Room				
	Pr	Re	F1	M	W	Pr	Re	F1	M	W
Octomap [28]	0.11±0.05	0.61±0.18	0.19±0.08	15	176	0.07±0.04	0.42±0.15	0.12±0.07	42	434
Meta-room [2]	0.04±0.03	0.44±0.08	0.07±0.04	24	276	0.24±0.30	0.55±0.05	0.25±0.27	31	464
Best of [10]	**0.55±0.36**	0.66±0.17	0.57±0.22	**6**	28	0.78±0.13	0.78±0.04	0.69±0.10	**2**	50
FPFH	0.13±0.14	0.12±0.05	0.11±0.08	32	-	0.13±0.12	0.39±0.12	0.18±0.14	19	-
Ours	0.50±0.24	**0.83±0.14**	**0.59±0.21**	10	**18**	**0.78±0.03**	**0.85±0.15**	**0.81±0.09**	8	**15**
	Living Room (partial)					Office (partial)				
	Pr	Re	F1	M	W	Pr	Re	F1	M	W
Octomap [28]	0.11±0.08	0.50±0.08	0.17±0.10	19	74	0.18±0.07	0.77±0.13	0.28±0.10	8	73
Meta-room [2]	0.13±0.18	0.42±0.10	0.14±0.14	15	122	0.17±0.25	0.39±0.20	0.17±0.18	12	146
Best of [10]	**0.83±0.29**	0.69±0.11	0.72±0.17	**4**	**13**	0.49±0.27	0.83±0.06	0.54±0.20	**0**	16
FPFH	0.11±0.11	0.31±0.14	0.15±0.14	12	-	0.21±0.13	0.50±0.19	0.27±0.13	5	-
Ours	0.80±0.05	**0.87±0.10**	**0.83±0.05**	**4**	**13**	**0.72±0.10**	**0.94±0.08**	**0.79±0.06**	**0**	**4**
	Kitchen (partial)					Average				
	Pr	Re	F1	M	W	Pr	Re	F1	M	W
Octomap [28]	0.43±0.08	0.41±0.08	0.41±0.07	9	40	0.18±0.14	0.54±0.18	0.23±0.13	18.6	159.4
Meta-room [2]	0.56±0.17	0.35±0.12	0.44±0.14	9	70	0.23±0.26	0.43±0.13	0.21±0.20	18.2	215.4
Best of [10]	0.62±0.21	**0.92±0.07**	0.55±0.11	**0**	55	0.64±0.27	0.74±0.14	0.61±0.16	**2.8**	34.2
FPFH	0.57±0.16	0.62±0.11	0.59±0.14	4	-	0.22±0.21	0.38±0.21	0.26±0.21	14.6	-
Ours	**0.77±0.015**	0.85±0.05	**0.81±0.03**	2	**3**	**0.72±0.16**	**0.86±0.12**	**0.76±0.13**	4.8	**10.6**

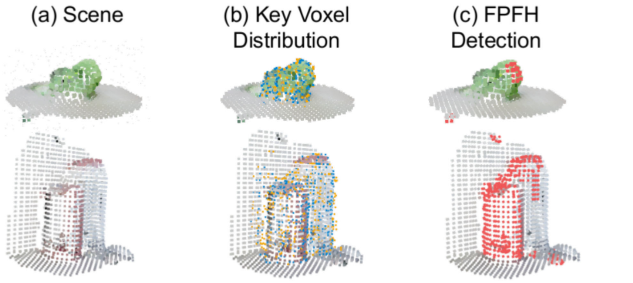


Fig. 6. Illustration of key voxel distribution and detected false positive points from the per point FPFH feature matching baseline for two unchanged subscenes. (a) Scene rendering. Above: an isolated green object in the center of a tabletop in the "small room" scene. Below: two bottles standing together against a wall in the "kitchen" scene. (b) The scene point clouds are rendered in bigger colored squares, and the key voxels are in smaller squares with blue ones as those near object surfaces and orange ones farther away in the unoccupied space amid object surfaces. (c) Falsely detected changed points from the FPFH feature matching baseline are rendered in red.

surface are more frequently witnessed in spaces amid surfaces, e.g., the area around the top of the shorter bottle and the left gap between the bottles and the wall, acting to unravel the spatial relations of these adjacent surfaces. The effectiveness of considering both object surfaces and inter-surface regions is then demonstrated by (c). While our method correctly recognizes the two scenes as unchanged, FPFH shows a small ratio of false positive points for the less noisy, single-object scenario but induces considerable amounts of false positives for the two-bottle case given a partial and warped reconstruction of the shorter bottle and the wall.

From Table I, we see that our approach achieves the highest values in terms of the five aforementioned metrics in most scenes. The point-wise FPFH matching baseline, while not eligible for wrongly detected clusters measurements as no cluster-level operations are involved, results in worse performance in the rest of the four metrics. This can be ascribed to its sensitivity to reconstruction noises, e.g., residual points or warpings that are prevalent around boundaries.

In comparison to the baseline approaches, our better performance could be attributed to the more distinct object contours and more robust neighborhood geometry verification enabled by the PlaneSDF representation. First, finding intersections between the preliminary change mask and the object map ensures that most of voxels extracted for 3D validation belong to *part of* an object and *all* voxels of the potentially changed objects are selected for 3D validation, hence unaffected by the common artifacts, e.g., noisy and incomplete object boundaries, in 3D clustering and segmentation in [10]. Second, local geometry verification, as opposed to point-wise nearest neighbor searching, offers additional robustness for detecting smaller objects and rejecting false positives, especially in the face of undesired point cloud residuals, such as when reconstruction quality is poor and objects are close to fixed structures such as walls.

Qualitatively, Figs. 2 and 7 display examples of qualitative change detection results of each of the five scenes. From Fig. 7, we can see that the proposed algorithm is able to extract point clouds belonging to most of the newly introduced objects, with some points missing from the planar parts that are attached to the plane, such as the bottom of the skillet in the kitchen scene (the last row of Fig. 7).

While the proposed algorithm has been shown to be effective in object change detection both quantitatively and qualitatively, we point out the failure case as when the height discrepancy between the object and the plane is ambiguous. Two typical examples within the dataset are: (1) The new object is partially occluded by a fixed structure in the height direction, e.g., the baseball placed under the table is missing from detection as its height is not correctly reflected in the height map. (2) The object is close to some noisy plane boundaries such as those caused by non-rigid deformation, e.g., missing object detection on the sofa

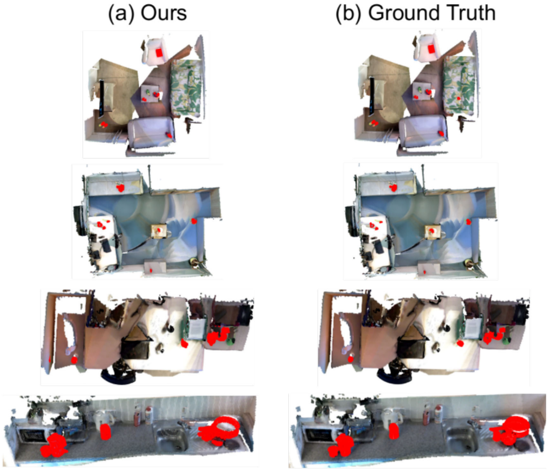


Fig. 7. Qualitative examples of the change detection results (red) for the four scenes in the object change detection dataset, from top to bottom: living room (partial), small room, office (partial), kitchen (partial). (a) Detected objects from our algorithm. (b) Ground truth.

(first row in Fig. 7) and our lower precision scores on the "small room" and "living room" scenes with new objects on the sofa.

V. Conclusion

In this paper, we have presented a new approach for change detection based on the newly proposed PlaneSDF representation. By making the most of the plane-supporting-object structure, our approach decomposes the common noise-sensitive global scene differencing scheme in a local plane-wise and object-wise manner, demonstrating enhanced robustness to measurement noises and reconstruction errors on both synthetic and real-world datasets.

References

[1] R. Finman, T. Whelan, M. Kaess, and J. J. Leonard, "Toward lifelong object segmentation from change detection in dense RGB-D maps," in *Proc. IEEE Eur. Conf. Mobile Robots*, 2013, pp. 178–185.

[2] 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 Int. Conf. Intell. Robots Syst.*, 2014, pp. 1854–1861.

[3] E. Herbst, P. Henry, and D. Fox, "Toward online 3-D object segmentation and mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 3193–3200.

[4] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1352–1359.

[5] R. A. Newcombe *et al.*, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, 2011, pp. 127–136.

[6] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level SLAM," in *Proc. IEEE Int. Conf. 3D Vis.*, 2018, pp. 32–41.

[7] V. Reijgwart, A. Millane, H. Oleynikova, R. Siegwart, C. Cadena, and J. Nieto, "Voxgraph: Globally consistent, volumetric mapping using signed distance function submaps," *IEEE Robot. Automat. Lett.*, vol. 5, no. 1, pp. 227–234, Jan. 2020.

[8] M. Fehr *et al.*, "TSDF-based change detection for consistent long-term dense reconstruction and dynamic object discovery," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 5237–5244.

[9] L. Schmid *et al.*, "Panoptic multi-TSDFS: A flexible representation for online multi-resolution volumetric mapping and long-term dynamic scene consistency," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 8018–8024, doi: 10.1109/ICRA46639.2022.9811877.

[10] E. Langer, T. Patten, and M. Vincze, "Robust and efficient object change detection by combining global semantic information and local geometric verification," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 8453–8460.

[11] A. Walcott-Bryant, M. Kaess, H. Johannsson, and J. J. Leonard, "Dynamic pose graph SLAM: Long-term mapping in low dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 1871–1878.

[12] E. Derner, C. Gomez, A. C. Hernandez, R. Barber, and R. Babusška, "Towards life-long autonomy of mobile robots through feature-based change detection," in *Proc. IEEE Eur. Conf. Mobile Robots*, 2019, pp. 1–6.

[13] E. Derner, C. Gomez, A. C. Hernandez, R. Barber, and R. Babusška, "Change detection using weighted features for image-based localization," *Robot. Auton. Syst.*, vol. 135, 2021, Art. no. 103676.

[14] L. Kunze *et al.*, "SOMA: A framework for understanding change in everyday environments using semantic object maps," in *Proc. AAAI Conf. Artif. Intell.*, 2018.

[15] T. Krajnik, J. P. Fentanes, G. Cielniak, C. Dondrup, and T. Duckett, "Spectral analysis for long-term robotic mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 3706–3711.

[16] L. Luft, A. Schaefer, T. Schubert, and W. Burgard, "Detecting changes in the environment based on full posterior distributions over real-valued grid maps," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 1299–1305, Apr. 2018.

[17] N. Bore, J. Ekekrantz, P. Jensfelt, and J. Folkesson, "Detection and tracking of general movable objects in large three-dimensional maps," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 231–247, Feb. 2019.

[18] U. Katsura, K. Matsumoto, A. Kawamura, T. Ishigami, T. Okada, and R. Kurazume, "Spatial change detection using voxel classification by normal distributions transform," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 2953–2959.

[19] P. F. Alcantarilla, S. Stent, G. Ros, R. Arroyo, and R. Gherardi, "Street-view change detection with deconvolutional networks," *Auton. Robots*, vol. 42, no. 7, pp. 1301–1322, 2018.

[20] J. Wald, A. Avetisyan, N. Navab, F. Tombari, and M. Nießner, "RIO: 3D object instance Re-localization in changing indoor environments," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7658–7667.

[21] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for hand-held 3D sensors," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 5182–5189.

[22] L. Ma, C. Kerl, J. Stückler, and D. Cremers, "CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 1285–1291.

[23] M. Hsiao, E. Westman, G. Zhang, and M. Kaess, "Keyframe-based dense planar SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 5110–5117.

[24] J. Straub, T. Campbell, J. P. How, and J. W. Fisher, "Small-variance nonparametric clustering on the hypersphere," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 334–342.

[25] A. J. Millane *et al.*, "Freetures: Localization in signed distance function maps," *IEEE Robot. Automat. Lett.*, to be published, doi: 10.1109/LRA.2021.3052388.

[26] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3D euclidean signed distance fields for on-board MAV planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 1366–1373.

[27] B. Calli *et al.*, "Yale-CMU-berkeley dataset for robotic manipulation research," *Int. J. Robot. Res.*, vol. 36, no. 3, pp. 261–268, 2017.

[28] E. Langer, B. Ridder, M. Cashmore, D. Magazzeni, M. Zillich, and M. Vincze, "On-the-fly detection of novel objects in indoor environments," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2017, pp. 900–907.

[29] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2009, pp. 3212–3217.

[30] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," 2018, *arXiv:1801.09847*.