3D Change Detection Using 2D Segmentation Masks

Aikaterini Adam[1] and Torsten Sattler[2]

[1] Archimedes Unit, Athena RC, Athens, Greece
[2] Czech Institute of Robotics and Cybernetics, Prague, Czech Republic

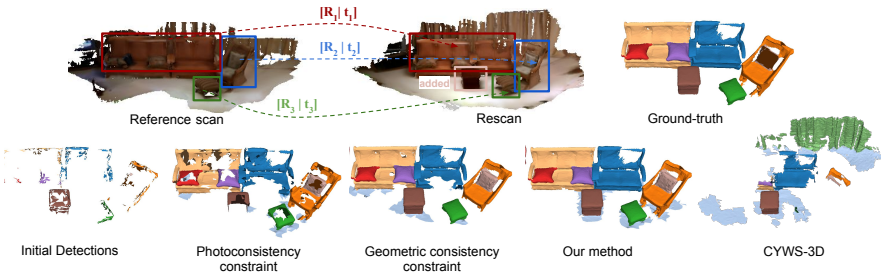

Fig. 1: 3D change detection. Given two scans of the same scene, captured at different times and thus including changes over time, we aim to segment the changed 3D regions. We use a novel segmentation consistency constraint to segment out the changed parts of the scene. Our approach outperforms prior work based on photometric [43] and geometric constraints [1], and CYWS-3D [39]. The colors are obtained from the instance segmentation of the scene provided by the dataset and are solely used for a clearer visualization.

Abstract. With 3D scanning technologies becoming more readily available, we are starting to have more and more scans of indoor scenes captured over time. Comparing the contents of these scans through 3D change detection can unlock interesting information about the changing elements (objects) of the scene. State-of-the-art 3D change detection algorithms typically obtain initial change detections in the 2D image space via render-and-compare techniques. These initial detections are typically incomplete and are thus propagated to the 3D space via a graph optimization. To this end, photometric or geometric constraints are used to define the optimization problem. In this paper, we propose a simple alternative based on the use 2D segmentation masks to guide the 3D segmentation process. We show that this approach, which requires minimal adaptation of existing methods and works with generic, pre-trained 2D masks, significantly improves change detection performance. (Submitted to ECCV 2026).

Keywords: 3D Change Detection · Dynamic Scenes · 3D scene understanding · Vision and Learning

1 Introduction

With the advancements in affordable depth-sensing devices operating in everyday life (*e.g.* Microsoft Kinect [12] and Apple's iPhone), we continuously observe the proliferation of applications based on 3D scene maps. Examples include indoor agents and AR/VR applications. Typically, these applications repeatedly build new models of the scenes that they operate in over time and need to be able to deal with changes in their environment. Revisiting a particular environment enables us to use domain knowledge and check for scene consistency. Change detection through scene consistency is crucial for many applications, *e.g.*, cleaning and surveillance robots, or warehouse logistics. In this paper, we consider the task of change detection between 3D scans: Given two scans of the same scene with occurred changes in between, we are interested in detecting and segmenting all 3D scene parts that have changed.

3D change detection remains an understudied topic, likely due to the required data starting to become more readily available only now. Only recently have end-to-end approaches [39] emerged, operating on pairs of images in the wild, and yielding 3D change detections when backprojected into the 3D space. On the other hand, the main solutions operating directly in the 3D space deploy a simple scene comparison on registered RGB or depth images [32, 33, 43] or directly on the point clouds [2, 3]. However, these methods fall short and might not segment the whole changed region when the observed depth and color do not change drastically (*cf.* Fig. 1-initial detections). To address that shortcoming, subsequent methods [1, 43] enforce a post-processing optimization step in the 3D space, to refine this initial, yet incomplete set of detections acquired through scene comparison or render-and-compare. More specifically, Taneja *et al.* [43] propagate the change to the scene parts sharing the same texture, *i.e.*, by enforcing a photoconsistency constraint. Given that photoconsistency constraints are inadequate when the changed scene parts do not share the same texture, Adam *et al.* [1] enforce a geometric consistency constraint by propagating change within scene parts that undergo the same motion between scans. However, these constraints remain sensitive to computing the transformations induced by the moving objects (*i.e.*, associating the two scans).

To overcome these shortcomings, we propose a new constraint for optimizing incomplete detections based on 2D segmentation masks. Using the same change detection framework as [1, 43], our method works with two 3D scans captured at different points in time and aims to segment any changed (added, moved, or removed) 3D scene parts between the two 3D scans. An initial set of changed parts is detected via depth-based render-and-compare [1, 43]. These initial detections are later optimized by graph cut-based optimization to segment out the whole changed part. The main technical difference between our approach and [1, 43] lies in the way the pairwise term of the optimization problem is formulated. Whereas [1, 43] rely on geometric and photometric consistency, respectively, we rely on the concept of segmentation consistency: Given segmentation masks extracted from the images used to construct the 3D scans, parts of the scene should be jointly labeled as changing/non-changing if they project into the same

segment in one of the images. This simple and intuitive formulation directly links the 2D segmentation task with the 3D segmentation task that is part of the change detection process. Our approach is enabled by recent advances in 2D foundation models [6] that provide us with precise and highly reliable 2D segmentation masks (*i.e.* 2D segments) [20], generalizing well to previously unseen scenes. These 2D segmentation masks typically segment (or oversegment) generic parts of the scene without being limited to a pre-defined list of classes. Compared to the geometric consistency constraint from [1], our approach does not require correspondences between scans (which in itself is a challenging task), thus simplifying the change detection pipeline.

Contributions. Our main contribution is a novel constraint for the graph cut-based optimization stage of current 3D change detection pipelines [1, 43]. Our constraint, based on 2D segmentation consistency, is easy to implement and can readily replace the constraints used in existing change detection pipelines. Integrating our constraint into a state-of-the-art 3D change detection framework [1] significantly boosts performance, demonstrating its effectiveness. Our approach works comparably well using either images or just depth maps, thus removing the need to store images together with the 3D scans for change detection, *i.e.* it can also be applied to colorless 3D representations such as LIDAR point clouds. Overall, our approach sets the new state-of-the-art on the challenging 3RScan dataset [46]. We will release our code.

2 Related Work

3D change detection. Discovering changes between scans captured at different times remains an understudied topic. Among the early works, the methods of [32, 33, 43–45] target change detection by detecting depth and color inconsistencies in paired images, associated with a 3D model. Directly working in the 3D space, Ambrus *et al.* [2, 3] propose a change detection system through 3D scene comparison. However, [2, 3] demand full overlap between the different scans. Contrary to that, our method handles partial observations when revisiting the scene. Most importantly, all the methods relying only on scene comparison in the 3D space fail to extract the whole 3D changed region when the depth does not change drastically. In an attempt to refine the initial results acquired through scene comparison, Taneja *et al.* [43] further impose a photoconsistency constraint to propagate the change to partially missed areas sharing the same texture. However, the method falls short when the changed scene parts do not have a homogeneous texture. In the same vein, Adam *et al.* [1] propose change propagation to all the scene parts moving coherently. Similar to [1, 43], our work addresses 3D change detection by enforcing constraints. Our novel constraint propagates the change to all the parts sharing the same 2D segmentation masks. Our method is one of the few methods [1, 43] overcoming simple scene comparison towards effective 3D change detection and operating on two 3D scans acquired at different times. While the difference between our method and existing approaches [1, 43] might look small, this simple change to segmentation-based

constraints significantly improves performance. *I.e.*, we show that high-quality generic segmentation masks can improve upon existing constraints.

Pair-wise and multiview change detection. So far, common change detection setups use a pair of RGB images captured before and after an event. However, contrary to our work, they do not incorporate any reasoning in 3D [4,18,25,38]. Most importantly, in most cases, the problem is simplified by using fixed cameras [17,42]. With the onset of deep learning, traditional convolutional neural network architectures have been successfully adopted for 2D change detection as in [29,40]. Later, transformer architectures [8,48] and the integration 2D foundation models [27,30] also showed promising results. Closer to our work, addressing change detection of 3D scenes, but only considering change detection on a pair-view image basis, recent methods address more challenging and realistic scenarios. Such setups adhere to large viewpoint shifts, severe occlusions, and disocclusions. More specifically, CYWS-3D [39] introduces a "register-and-difference" method, leveraging frozen embeddings from a pre-trained backbone and feature differences towards scene change detection between two "in the wild" views. Even though originally proposed for scene change detection between images, we modified the method accordingly to propagate the 2D changes in the 3D space, and evaluate on the task of 3D change detection. In 4, we show that the proposed approach significantly outperforms this modified baseline of [39]. In the same vein, OmniPoseAD [7] and SplatPose [21] deploy recent 3D representations to associate 2D images and account for anomaly detection in pairwise views. The former by deploying Nerfs [31] and the latter Gaussian Spatting [19]. Finally, [11] is the first work to extend change localization across multiple views. More specifically, the authors deploy 3D Gaussian splats as an intermediate 3D representation to localize scene changes between two sets of images acquired before and after change.

Long-term instance relocalization and modeling. Related to the 3D change detection problem is the problem of instance relocalization [13, 46, 49], *i.e.* the problem of localizing the same 3D object in multiple scans captured over time. By definition, this requires prior knowledge about the objects in a scene. Correspondences are then used to estimate the 6DoF relative pose of the same object instance between two different scans [46, 49]. Zhu *et al.* [49] also reconstruct these dynamic objects, on top of instance relocalization. In a related task, [13] leverages a semantic instance segmentation of the scene to build a temporal model, which captures the placements of objects at different points in time. More recently, BYE [16] performs instance association in dynamically changing scenes in a class-agnostic manner. Whereas these works focus on relating the movement of object instances over time, we are considering the task of detecting all types of changes, without any prior knowledge about the types of objects in the scene.

Handling dynamic scenes. Rather than considering discrete points in time, another line of work considers the problem of handling scenes with dynamic objects, *e.g.*, as part of SLAM systems [9, 15, 26, 36, 37]. In [26, 37], a common

assumption is that the movement of these objects is directly observed. Thus, the task of handling moving objects is solved via tracking. More similar to our approach, [9, 15, 36] use SLAM approaches to associate new observations and subsequently, use simple scene comparisons to account for changes. Most similar to our work, Finman *et al*. [10] deploy a simple 3D scene comparison to identify changes directly between two 3D maps. In contrast to our work, they do not apply any post-processing optimization steps.

3 3D Change Detection

Our method aims at 3D change detection between scans of the same scene recorded at different times. Essentially, we assume that we are given a set of camera poses (and potentially the corresponding images) together with the 3D model. Our approach, illustrated in Fig. 2, follows the same outline as prior work [1, 43]. We first compute a set of initial change detections via depth-based render-and-compare. Intuitively, regions with a large change in depth likely correspond to regions undergoing changes. These detections serve as "seeds" for change propagation. More precisely, we solve a graph cut problem to identify regions that have changed between both scans. The initial detections are thereby used to define the unary costs of the nodes in the graph. The pairwise costs of the optimization problem are defined by constraints. After the optimization, a connected component analysis is applied to form the final changed 3D scene parts.

We directly build upon the open source code of [1]. The main difference to [1, 43] is the definition of the constraint imposed by the pairwise costs. Observing that modern approaches such as Segment Anything [20] produce highly accurate segmentation masks that are rather consistent between views, we propose to use a segmentation-based constraint rather than the photometric consistency [43] and geometric consistency [1] constraints. Intuitively, if the set of 3D points belonging to a 2D mask contains points initially marked as changed, then it is likely that all these points belong to a change. We thus encourage points belonging to the same segment in one of the images to obtain the same changed/not-changed label through a graph optimization [22], which propagates labels from the initial change detections to all points belonging to the same mask. The used 2D segmentation masks, and thus our method, do not require any prior knowledge about the scene or the objects in it. In the following, we describe the pipeline.

3.1 Obtaining Initial Change Detections

Assuming that the two scans (the reference scan S and the rescan R) are aligned[3], we can use the same set of poses to render the depth maps from both scans.

[3] This assumption is in line with the literature [1,13,46,49], and we use the registrations provided by the dataset used for evaluation.

Since the rendered maps are in the same coordinate frame, the image pairs can be subtracted directly. The subtracted pairs are thresholded with the help of [5]. Backprojecting the thresholded regions into the 3D space leads to an initial set of detections, *i.e.*, regions showing a large depth difference that indicate potential changes. See [1] for details on this stage.

Supervoxels. As [1], our method operates on supervoxels [34]. Supervoxels group 3D points together and offer a coarse over-segmentation of the scene while preserving object boundaries. Please see [1] for details.

3.2 2D Segmentation Masks

The previous stage provides a set of initial detections belonging to potentially changing parts of the scene. As shown in [1,43], not all the parts of the scene that underwent a change will be detected in this stage. *E.g.*, slightly moving a couch such that the new and old positions overlap can lead to regions without changing depths. Hence, only a part of the couch (where the depth changes drastically) is initially labeled as changed. The graph cut optimization, described in Sec. 3.3, thus aims to propagate the initial change detections to all changed scene parts. To this end, we employ 2D segmentation masks, propagated to the 3D space, to indicate which regions should receive the same label.

For each image belonging to a camera pose used to calculate initial change detections, we compute Segment Anything (SAM) [20] segmentation masks. We then associate 3D primitives (supervoxels) with each other depending on whether they belong to the same segment in one of the images. We thus implicitly create a label-consistent 3D segmentation via a graph optimization. To this end, we check whether two supervoxels project into the same mask in at least one of the images and use these pairwise preferences to produce a segmentation in the 3D space. More specifically, we project supervoxels into the images to identify which of them share the same SAM mask, *i.e.*, which of them are likely to belong together. This imposes our segmentation mask constraint that encourages that supervoxels sharing a SAM mask (even if only in one image) should be labeled consistently. The graph optimization will attempt to propagate the change from the initially changing supervoxels to all the supervoxels sharing the same segmentation mask.

To compute the masks, we apply SAM [20] out-of-the-box without any fine-tuning/training. The automatic mask generator is applied to both color and depth images. We deploy SAM on the depth images, achieving good results (*cf.* Fig. 6), even though it was never explicitly trained on depth maps. Indeed, as individual scene parts often are defined by a relatively coherent depth compared to other parts, we find the twist of applying SAM to depth images powerful enough to fulfill our requirements of discovering consistent 2D regions (described by the inferred segmentation masks). Through this modification, our approach can also be applied to colorless 3D representations, *e.g.*, collected through LIDAR.

Given that the images are posed, along with the depth image, we can associate pixels with the corresponding 3D points through backprojection. Thus, we can assign a SAM mask to each 3D point p of the scene. Consequently, we can

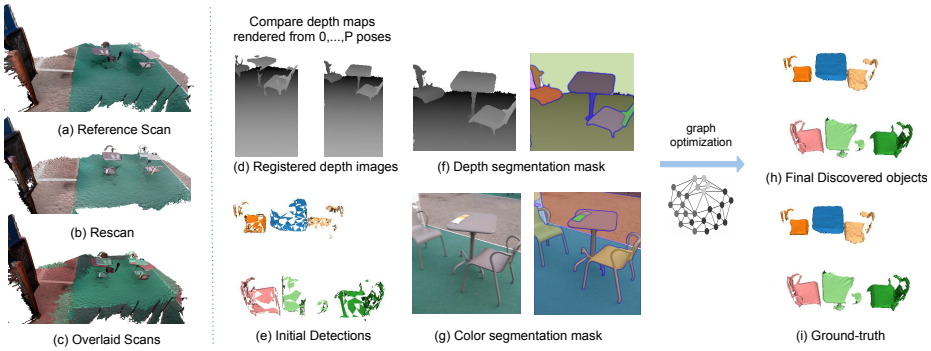

Fig. 2: Overview of our method. Given two scans, (a) a reference scan, and (b) a rescan and associate camera poses, we acquire an initial but incomplete set of change detections via render-and-compare (d), (e). The initial detections are then refined through our novel segmentation mask constraint, which propagates the change to the whole region sharing the same 2D mask (f),(g). The final results in (h) prove the effectiveness of the segmentation mask constraint when compared against the ground-truth (i).

also assign a SAM mask to each supervoxel V, as the most frequent mask of the points $p \in V$. Since we have calculated a SAM mask for each supervoxel, we can now associate these primitives (supervoxels), sharing the same SAM mask in at least one image. When considering depth and RGB separately, two supervoxels belong together as long as they project to the same mask in one or more images/depth maps. When combined (depth + RGB), two supervoxels belong together as long as they project into the same mask in either a depth map or an RGB image.

3.3 Optimization of the Initial Changes

The previous two stages compute the initial change detections and gather information about which parts of the scene are likely to belong together (*i.e.*, share the same segmentation mask). In the final stage, we combine this information to segment out 3D changed regions. To this end, as in [1], we deploy a graph optimization on a supervoxel representation of the 3D scene. The graph optimization distills the information from the SAM segmentation masks into the 3D space. It thus encourages that the change is propagated from the seeds to all the scene parts sharing the same SAM masks and, consequently, to the whole changed region.

Keeping the same configuration as in [1], we apply the optimization on an undirected graph $\mathcal{G} = (V, E, w)$. Each node $v_i \in V$ corresponds to a supervoxel, and the edges $e_i \in E$ with weight w_i connect adjacent supervoxels. The goal of the optimization is to calculate a labeling P^*, assigning a changing or non-changing label to each supervoxel. The labeling P^* should remain as close as possible to the initial solution P (in terms of the supervoxels initially labeled as

changing) while respecting that supervoxels with the same segmentation mask should have the same label.

The graph optimization is solved via the method of "Generalized Minimal Partition" [22], which can be perceived as a continuous-space version of the Potts energy model:

$$P^* \in \arg\min_{Q \in \Omega} \{\Phi(P, Q) + \lambda \Psi(Q)\} \ . \tag{1}$$

In Eq. (1), the first (unary) term Φ stands for the fidelity term, and the second (pairwise) term $\Psi(Q)$ for the penalty function. Ω stands for the search space of the final solution.

The initial labeling P, sourced from Sec. 3.1, is calculated as in [1]. As described above, the difference between our approach and [1] lies in the pairwise term, which encourages that the labeling is consistent with the segmentation masks. λ is a regularization parameter between the two terms. The penalty (pairwise) term can be decomposed as:

$$\Psi(Q) = \sum_{(i,j) \in E} w_{i,j} \, \phi(v_i - v_j) \ , \tag{2}$$

where $w_{i,j}$ stands for the weight across the edge $e_{i,j}$. The term $\phi(v_i - v_j)$ is a piece-wise constant function, based on the Potts model:

$$\phi(v_i - v_j) = \begin{cases} 1 & \text{if } v_i, v_j \text{ same segment. mask} \\ 0 & \text{otherwise} \end{cases} \ . \tag{3}$$

Eq. (3) assigns a high cost to the edges connecting supervoxels v_i, v_j sharing the same segmentation masks. Hence, parts of the scene sharing the same segmentation mask are penalized if they do not get the same label.

Eq. (1) is non-convex due to the Potts penalty term. Thus, its global minimum cannot be realistically solved for large point clouds. We thus use the ℓ_0-cut pursuit algorithm introduced in [22] that quickly approximates a solution with a few graph-cut iterations.

4 Experimental Evaluation

Datasets. We evaluate the effectiveness of our proposed method on the 3RScan dataset [46]. For a thorough discussion of datasets, please refer to the supplementary material (SM). The SM highlights the lack of proper datasets for our task, which leads to most of the similar works being evaluated on a **single, individual dataset** [1, 10, 13, 23, 24, 47] rather than standardized dataset(s).

The 3RScan dataset [46] was originally designed to benchmark instance relocalization. Previous work [1] used the dataset for 3D change detection by exploiting the changes naturally recorded in the data. The dataset consists of several 3D scans of indoor scenes collected at various times and where the scenes change

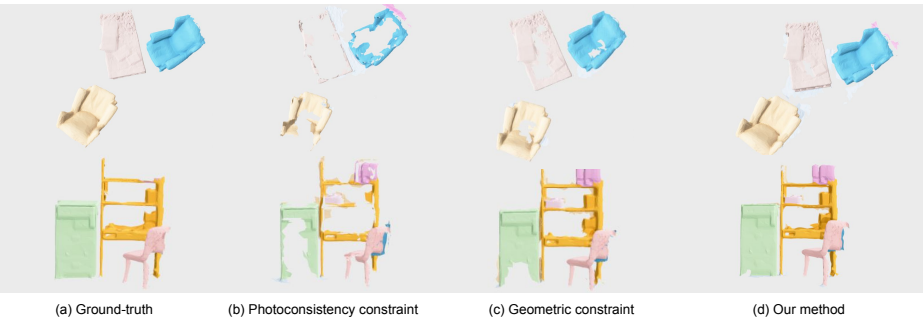

Fig. 3: Comparative results between the different change propagation constraints. (a) Ground-truth, (b) Photoconsistency constraint [43], (c) Geometric constraint [1], and (d) our novel segmentation mask constraint. The changed scene parts segmented from the segmentation mask constraint are much more complete. The colors are obtained from the instance segmentation of the scene provided by the dataset and are solely used for a clearer visualization.

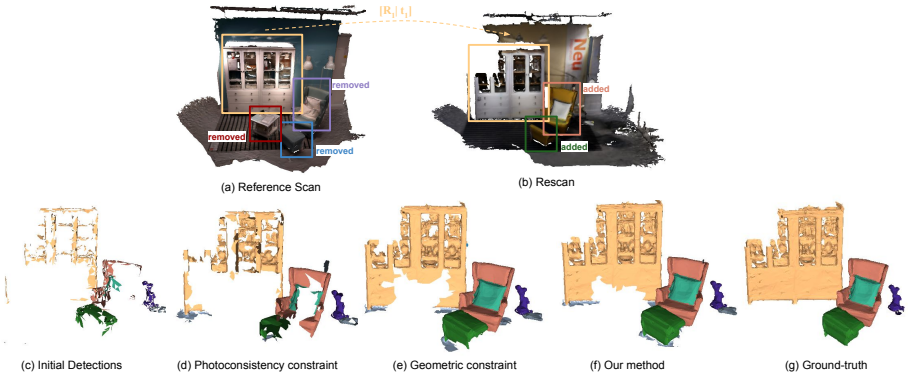

Fig. 4: Qualitative comparison of different change propagation constraints. (a) Reference Scan, (b) Rescan, (c) Initial Detections [33], (d) Photoconsistency constraint [43], (e) Geometric constraint [1], (f) our segmentation mask constraint, and (g) Ground-truth. The objects discovered from the segmentation mask constraint are much more complete. The colors are obtained from the instance segmentation of the scene provided by the dataset and are solely used for a clearer visualization.

between recordings. Each scene contains a reference scan and multiple partial rescans, recorded after changes. For a fair comparison, we use the same splits defined in [1], and evaluate our method on the validation set. This split includes 47 different scenes (and thus reference scans), and 110 rescans, covering a wide range of mostly household scenes.

Metrics. We evaluate our method using recall@kIoU to capture the success of the 3D change detection. The recall@kIoU $= \frac{1}{N}\sum_{i=0}^{N} u_i$, where $u_i \in \{0, 1\}$ is set

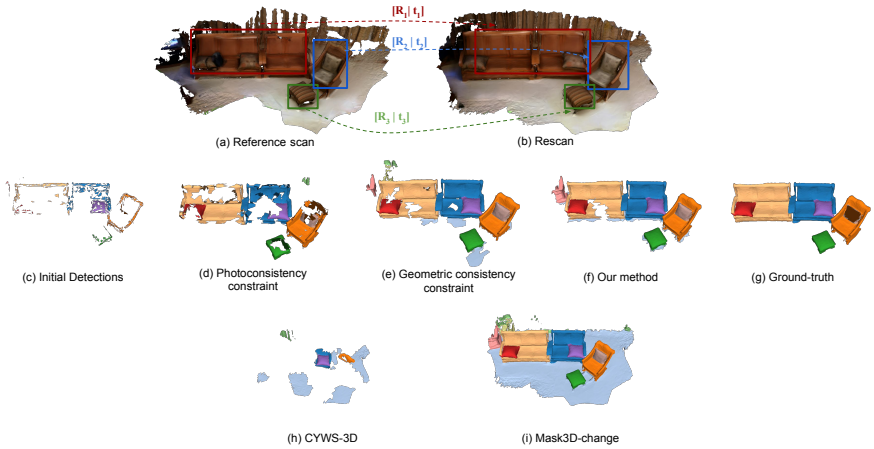

Fig. 5: Comparative results between the different change detection methods. (a) Reference Scan, (b) Rescan, (c) Initial Detections [33], (d) Photoconsistency constraint [43], (e) Geometric constraint [1], (f) our novel segmentation mask constraint, (g) Ground-truth, (h) CYWS-3D, and (i) Mask3D-change. The colors are obtained from the instance segmentation of the scene provided by the dataset and are solely used for a clearer visualization.

Fig. 6: The 2D segmentation masks [20] when applied on color and depth images. Individual scene parts tend to have consistent depth versus the background, thus leading to larger 2D segments on the depth images.

to 1 if the IoU score for the $i-th$ changed box is greater than k, otherwise 0. Thus, the metric returns the percentage of correctly retrieved changed parts at kIoU, out of all the changed scene parts N. Concerning the thresholds k, we use the IoU@0.25 and IoU@0.50, as largely used in [35]. For completeness, we also tabulate results for the IoU@0.20, as in [1].

Variants of our method. Our paper introduces a novel segmentation mask constraint, propagating change from the seeds to all the regions sharing the same SAM mask, towards effective 3D change detection. We present results when the

Table 1: Recall@kIoU for our segmented changed 3D scene parts vs baselines on the 3RScan validation set. Baselines: render-and-compare only [33], graph-cut based [1,43], 2D reasoning [39], pre-optimization Mask3D-change and Mask R-CNN change [1].

Method	Recall@20	Recall@25	Recall@50
Palazzolo *et al.* [33]	54.23	40.00	31.10
Photoconsist. [43]	48.10	33.07	28.78
Geom. constraint [1]	68.40	61.83	42.54
CYWS-3D [39]	37.64	33.00	13.59
Mask3D-change	29.30	27.26	24.10
Mask R-CNN [1]	52.96	44.96	35.20
Ours [Geom. + SAM depth]	71.34	64.80	**44.24**
Ours [Geom. + SAM color]	**74.52**	**65.70**	43.30
Ours [Geom. + SAM depth + SAM color]	**74.52**	**65.70**	43.30
Ours [SAM depth]	71.34	64.80	**44.24**
Ours [SAM color]	**74.52**	**65.70**	43.30
Ours [SAM depth + SAM Color]	**74.52**	**65.70**	43.30

segmentation masks are inferred on depth images (SAM depth) and on color images (SAM color), and when this information is combined (SAM depth + SAM color). We also examine how adding the geometric constraint [1] on top of the segmentation mask constraint affects the performance (Geometry + SAM depth, Geometry + SAM color, Geometry + SAM depth + SAM color). Finally, we present the results of our method, with a different algorithm introducing off-the-shelf masks, *i.e.*, Mask R-CNN [14]. We apply the 2D Mask R-CNN object detector, trained on the COCO dataset [28], on the color images of each rescan without any extra fine-tuning. We thus replace our segmentation mask constraint from SAM with the segmentation masks of Mask R-CNN, and utilize the masks in the same way: a Mask R-CNN label S_i is assigned to each supervoxel v_i, as the most common label of points that fall within the supervoxel.

Baselines. We want to measure the impact of our newly introduced segmentation mask constraint versus the constraints imposed by geometric transformations [1] and photoconsistency [43]. In the baseline comparisons, we also include [33], which is equivalent to our initial detection step. We also compare against CYWS-3D [39]. CYWS-3D represents a baseline where we use a powerful model to detect changes in 2D and propagate them into 3D, without any optimization in the 3D space. The method operates between two images of the same 3D scene. To use this baseline in our context, *i.e.*, for detecting changes between two 3D scans, we have rendered 20 images for each pair of reference scan and rescan. We use the same pose to render the image from both scans, and we ensure that the field of view covers a large area of the room (*i.e.*, contrary to the raw images provided by the dataset that focus on details rather than providing a

larger view of the scene), since in our experience, [39] performs better if it sees a larger part of the scene. The final changes are inferred from the trained model on synthetic data, using the rendered RGB and depth images, as well as their poses. Since CYWS-3D [39] operates in 2D, to acquire the final 3D changes, we back-project the results in 3D using the depth images. We have also created another baseline, denoted as Mask3D-change, that leverages state-of-the-art algorithms for semantic instance segmentation [41]. Mask3D-change represents a baseline that computes 3D segmentations offline in a pre-processing step, rather than online, as our method. To this end, the semantic instances are computed in the 3D space, and change is propagated to the whole instances that include initial seeds of change (*i.e.*, changing regions acquired through render-and-compare). 3D semantic instances are calculated using the pre-trained model from Mask3D [41] based on 200 ScanNet classes. We do not compare against [2,3], as these methods demand a full observation overlap (between the scans) and thus cannot handle the partial observations between scans, that occur in the 3RScan dataset. Most importantly, these methods rely purely on scene comparison, without any optimization, which is equivalent to our method's initial step.

Ablation Study. From Tab. 1, it is clear that adding the geometric constraint on top of our introduced segmentation mask constraint does not help our method perform better. The performance of the Geometry+SAM depth is capped by the performance of SAM depth. Similarly, the performance of the Geometry+SAM color is capped by the performance of SAM color. Intuitively, this means that the geometric constraint propagates the change to regions already discovered by the segmentation mask constraint and, thus, does not provide any additional information to add any uplift in the metrics. Also, this expresses that the segmentation mask constraint achieves better results with a simpler method than the geometric constraint, since our method does not require 3D matching and transformation estimation.

Moreover, when compared with another off-the-shelf method for mask segmentation (Mask R-CNN masks), it is clear that our method leveraging SAM [20] performs significantly better (*cf*. Tab. 1). Further, we point out that leveraging information from the segmentation masks both on color and depth images does not improve the performance; the metrics seem capped by the information from the color masks, which are more detailed (*cf*. Fig. 6). As such, these sources of information (segmentation masks in depth and color) do not seem to be complementary. Fig. 6 illustrates the inferred segmentation masks for the same scene part in color and depth format. While statistically the color images lead to more segments, our method seems robust towards the "over-segmentation" with both methods (SAM color and SAM depth) performing almost the same. Indeed, the only requirement for treating "over-segmentation" is to identify some initial seeds of change in each of the over-segmented masks comprising the whole changed region. This is typically the case since the initial detections mostly delineate the boundaries of the changed scene parts. We see the main importance of the result that SAM depth performs comparably well to SAM color in a potential simplification of the pipeline: All calculations can be done on raw geometry, without the

need for textures or color information. One potential application scenario would be where change detection is performed by a cloud service, and users want to protect their privacy by only sharing the geometry.

Baseline comparisons. Tab. 1 compares our method against the different baselines. From Tab. 1, it is clear that our method outperforms the most competitive baseline, Geometric constraint [1], in all metrics. When leveraging the constraints from the 2D segmentation masks on color images (SAM color), our method performs +6.12% better compared to the SoTA for the recall@0.20, +3.87% for the recall@0.25, and +0.76% better in terms of the recall@0.50. Similarly, when SAM is deployed on depth images (SAM depth), the recall@0.20 improves by +2.94%, the recall@0.25 by +2.97%, and the recall@0.50 by +1.7%. CYWS-3D [39] presents a significant performance drop compared to the methods operating directly in the 3D. This is mostly attributed to missing small movements of large furniture and not having the ability to handle free space, as shown in the SM. Finally, the results of the proposed baseline Mask3D-change showcase the superiority of gradually propagating the change from the initial seeds of change to neighboring regions sharing the same segmentation mask via graph optimization. Indeed, having a 3D instance segmentation at our disposal and directly propagating the change to the whole segment can result in large instances being falsely retrieved as changing. *E.g.*, the whole floor, can be selected as changing due to a small portion of erroneous initial detections belonging to the floor segment. These erroneous detections acquired through render-and-compare are attributed to small misalignments (*cf*. Fig. 5 and the SM). Qualitative results for the most competitive baselines are shown in Figs. 3 and 4 and for all the baselines in Fig. 5. As can be seen, the segmentation mask constraint leads to more complete detections of the changed 3D regions, showcasing the significantly improved performance of our method.

Table 2: Recall@0.20 for segmenting novel scene parts between two 3D scans of the same scene. The three baselines refer to Langer *et al*. [24], Adam *et al*. [1] and our method.

Method	Recall@20
Langer *et al*. [14]	73.85
Adam *et al*. [1]	75.08
Ours [SAM color + SAM depth]	76.11

Novel Scene Parts. Detecting novel (added) scene parts between two different captures of the same scene remains an open problem, of broad interest to many robotic applications. Given that existing methods, such as Langer *et al*. [24], focus on detecting only added objects, we decide to create a relevant subtask to check the performance of our method. Tab. 2 tabulates the results for the

recall@0.20 (*i.e.*, the threshold for the IoU is 20%) for the three baselines, *i.e.*, the method of [24], the most competitive method [1], modified to focus only on novel scene parts, and our proposed method modified to segment out only added parts of the scene. It is important to note here that [24] was initially designed to segment added scene parts on the floor representing small objects. However, we modified the code accordingly to segment out added scene parts irrespective of their size and position to ensure a fair comparison. After close inspection of Tab. 2, it is clear that our method outperforms the most competitive baseline [1] in the sub-task of segmenting out added scene parts between two 3D scans. Indeed, [1] calculates the rigid transformations between moved object instances in the reference scan and the rescan in order to propagate the change to the whole changed area. However, when a novel instance is added or removed, this scenario cannot be applied, and thus, change detection solely relies on simple scene comparison via render-and-compare. On the contrary, our method can optimize initial changes even in added or removed objects, since it simply relies on identifying the 2D segments of these scene parts.

Limitations. Common failures of our method and [1,43] include wrong detections attributed to misalignments. Errors in the scan's registration can lead to significant depth differences in render-and-compare and thus an erroneous set of initial changes. Our newly introduced segmentation mask constraint can also be limited by benchmarks sourced from hand-held devices. In these cases, there might be a deficiency of poses covering scene parts lying next to the floor, which are often occluded (*e.g.* the legs of a table). Limited coverage of such parts leads to fewer 2D masks/segments and less successful change propagation. This limitation can be tackled by generating a variety of poses, ensuring that the whole scene is covered, and rendering new images. For more information, please refer to the SM.

5 Conclusion

We present an adaptation of existing 3D change detection approaches, introducing a novel segmentation mask constraint. The novel constraint leverages off-the-shelf, pre-trained 2D segmentation masks to propagate the change from an initial, incomplete set of detections acquired by render-and-compare to the whole changed region. Despite its simplicity, the constraint is much more effective than existing methods, achieving state-of-the-art performance on the 3RScan dataset, as showcased by our experiments.

References

1. Adam, A., Sattler, T., Karantzalos, K., Pajdla, T.: Objects can move: 3d change detection by geometric transformation consistency. In: Eur. Conf. Comput. Vis. pp. 108–124. Springer (2022) 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 13, 14
2. Ambruş, R., Bore, N., Folkesson, J., Jensfelt, P.: Meta-rooms: Building and maintaining long term spatial models in a dynamic world. In: 2014 IEEE/RSJ international conference on intelligent robots and systems. pp. 1854–1861. IEEE (2014) 2, 3, 12
3. Ambrus, R., Folkesson, J., Jensfelt, P.: Unsupervised object segmentation through change detection in a long term autonomy scenario. In: 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids). pp. 1181–1187. IEEE (2016) 2, 3, 12
4. Bandara, W.G.C., Patel, V.M.: A transformer-based siamese network for change detection. In: IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium. pp. 207–210. IEEE (2022) 4
5. Barron, J.T.: A generalization of otsu's method and minimum error thresholding. In: Eur. Conf. Comput. Vis. pp. 455–470. Springer (2020) 6
6. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021) 3
7. Cutler, K.J., Stringer, C., Lo, T.W., Rappez, L., Stroustrup, N., Brook Peterson, S., Wiggins, P.A., Mougous, J.D.: Omnipose: a high-precision morphology-independent solution for bacterial cell segmentation. Nature methods **19**(11), 1438–1448 (2022) 4
8. Fang, S., Li, K., Li, Z.: Changer: Feature interaction is what you need for change detection. IEEE Transactions on Geoscience and Remote Sensing **61**, 1–11 (2023) 4
9. Fehr, M., Furrer, F., Dryanovski, I., Sturm, J., Gilitschenski, I., Siegwart, R., Cadena, C.: Tsdf-based change detection for consistent long-term dense reconstruction and dynamic object discovery. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 5237–5244. IEEE (2017) 4, 5
10. Finman, R., Whelan, T., Kaess, M., Leonard, J.J.: Toward lifelong object segmentation from change detection in dense rgb-d maps. In: 2013 European Conference on Mobile Robots. pp. 178–185. IEEE (2013) 5, 8
11. Galappaththige, C.J., Lai, J., Windrim, L., Dansereau, D., Sunderhauf, N., Miller, D.: Multi-view pose-agnostic change localization with zero labels. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 11600–11610 (2025) 4
12. Glocker, B., Izadi, S., Shotton, J., Criminisi, A.: Real-time rgb-d camera relocalization. In: 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). pp. 173–179. IEEE (2013) 2
13. Halber, M., Shi, Y., Xu, K., Funkhouser, T.: Rescan: Inductive instance segmentation for indoor rgbd scans. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 2541–2550 (2019) 4, 5, 8
14. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Int. Conf. Comput. Vis. pp. 2961–2969 (2017) 11, 13
15. Herbst, E., Henry, P., Fox, D.: Toward online 3-d object segmentation and mapping. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). pp. 3193–3200. IEEE (2014) 4, 5

16. Huang, C., Burgard, W.: Bye: Build your encoder with one sequence of exploration data for long-term dynamic scene understanding and navigation. Conference on Robot Learning (2024) 4
17. Jhamtani, H., Berg-Kirkpatrick, T.: Learning to describe differences between pairs of similar images. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 4024–4034 (2018) 4
18. JST, C.: Change detection from a street image pair using cnn features and superpixel segmentation. In: Proc. Brit. Mach. Vis. Conf. pp. 61–1 (2015) 4
19. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., et al.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023) 4
20. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Int. Conf. Comput. Vis. pp. 4015–4026 (2023) 3, 5, 6, 10, 12
21. Kruse, M., Rudolph, M., Woiwode, D., Rosenhahn, B.: Splatpose & detect: Pose-agnostic 3d anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3950–3960 (2024) 4
22. Landrieu, L., Obozinski, G.: Cut pursuit: Fast algorithms to learn piecewise constant functions on general weighted graphs. SIAM Journal on Imaging Sciences **10**(4), 1724–1766 (2017) 5, 8
23. Langer, E., Patten, T., Vincze, M.: Robust and efficient object change detection by combining global semantic information and local geometric verification. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 8453–8460. IEEE (2020) 8
24. Langer, E., Ridder, B., Cashmore, M., Magazzeni, D., Zillich, M., Vincze, M.: On-the-fly detection of novel objects in indoor environments. In: 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO). pp. 900–907. IEEE (2017) 8, 13, 14
25. Li, J., Xu, X., Gao, L., Wang, Z., Shao, J.: Cognitive visual anomaly detection with constrained latent representations for industrial inspection robot. Applied Soft Computing **95**, 106539 (2020) 4
26. Li, K., Rezatofighi, H., Reid, I.: Moltr: Multiple object localization, tracking and reconstruction from monocular rgb videos. IEEE Robotics and Automation Letters **6**(2), 3341–3348 (2021) 4
27. Lin, C.J., Garg, S., Chin, T.J., Dayoub, F.: Robust scene change detection using visual foundation models and cross-attention mechanisms. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 8337–8343. IEEE (2025) 4
28. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Eur. Conf. Comput. Vis. pp. 740–755. Springer (2014) 11
29. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015) 4
30. Martinson, E., Lauren, P.: Meaningful change detection in indoor environments using clip models and nerf-based image synthesis. In: 2024 21st International Conference on Ubiquitous Robots (UR). pp. 603–610. IEEE (2024) 4
31. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021) 4

32. Palazzolo, E., Stachniss, C.: Change detection in 3d models based on camera images. In: 9th Workshop on Planning, Perception and Navigation for Intelligent Vehicles at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS) (2017) 2, 3
33. Palazzolo, E., Stachniss, C.: Fast image-based geometric change detection given a 3d model. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 6308–6315. IEEE (2018) 2, 3, 9, 10, 11
34. Papon, J., Abramov, A., Schoeler, M., Worgotter, F.: Voxel cloud connectivity segmentation-supervoxels for point clouds. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 2027–2034 (2013) 6
35. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: Int. Conf. Comput. Vis. (2019) 10
36. Rowell, J., Zhang, L., Fallon, M.: Lista: Geometric object-based change detection in cluttered environments. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 3632–3638. IEEE (2024) 4, 5
37. Rünz, M., Agapito, L.: Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 4471–4478. IEEE (2017) 4
38. Sachdeva, R., Zisserman, A.: The change you want to see. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3993–4002 (2023) 4
39. Sachdeva, R., Zisserman, A.: The change you want to see (now in 3d). In: Int. Conf. Comput. Vis. pp. 2060–2069 (2023) 1, 2, 4, 11, 12, 13
40. Sakurada, K., Wang, W., Kawaguchi, N., Nakamura, R.: Dense optical flow based change detection network robust to difference of camera viewpoints. arXiv preprint arXiv:1712.02941 (2017) 4
41. Schult, J., Engelmann, F., Hermans, A., Litany, O., Tang, S., Leibe, B.: Mask3d: Mask transformer for 3d semantic instance segmentation. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 8216–8223. IEEE (2023) 12
42. Sousa, R.B., Sobreira, H.M., Moreira, A.P.: A systematic literature review on long-term localization and mapping for mobile robots. Journal of Field Robotics **40**(5), 1245–1322 (2023) 4
43. Taneja, A., Ballan, L., Pollefeys, M.: Image based detection of geometric changes in urban environments. In: Int. Conf. Comput. Vis. pp. 2336–2343. IEEE (2011) 1, 2, 3, 5, 6, 9, 10, 11, 14
44. Taneja, A., Ballan, L., Pollefeys, M.: City-scale change detection in cadastral 3d models using images. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 113–120 (2013) 3
45. Ulusoy, A.O., Mundy, J.L.: Image-based 4-d reconstruction using 3-d change detection. In: Eur. Conf. Comput. Vis. pp. 31–45. Springer (2014) 3
46. Wald, J., Avetisyan, A., Navab, N., Tombari, F., Nießner, M.: Rio: 3d object instance re-localization in changing indoor environments. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 7658–7667 (2019) 3, 4, 5, 8
47. Wald, J., Sattler, T., Golodetz, S., Cavallari, T., Tombari, F.: Beyond controlled environments: 3d camera re-localization in changing indoor scenes. In: Eur. Conf. Comput. Vis. pp. 467–487. Springer (2020) 8
48. Wang, Z., Zhang, Y., Luo, L., Wang, N.: Transcd: Scene change detection via transformer-based architecture. Optics Express **29**(25), 41409–41427 (2021) 4

49. Zhu, L., Huang, S., Schindler, K., Armeni, I.: Living scenes: Multi-object relocalization and reconstruction in changing 3d environments. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 28014–28024 (2024) 4, 5