Objects Can Move: 3D Change Detection by Geometric Transformation Consistency

Aikaterini Adam[1,2], Torsten Sattler[1], Konstantinos Karantzalos[2], and Tomas Pajdla[1]

[1] Czech Institute of Informatics, Robotics and Cybernetics, CTU in Prague
{Aikaterini.Adam,Torsten.Sattler,pajdla}@cvut.cz
[2] National Technical University of Athens, Greece
karank@scentral.ntua.gr

Abstract. AR/VR applications and robots need to know when the scene has changed. An example is when objects are moved, added, or removed from the scene. We propose a 3D object discovery method that is based only on scene changes. Our method does not need to encode any assumptions about what is an object, but rather discovers objects by exploiting their coherent move. Changes are initially detected as differences in the depth maps and segmented as objects if they undergo rigid motions. A graph cut optimization propagates the changing labels to geometrically consistent regions. Experiments show that our method achieves state-of-the-art performance on the 3RScan dataset against competitive baselines. The source code of our method can be found at https://github.com/katadam/ObjectsCanMove.

Keywords: 3D change detection, object discovery, graph optimization

1 Introduction

The ability to detect and interact with objects is critical to AR/VR applications and for multiple robotics tasks, such as surveillance, robotic manipulation, and maintaining order. All these tasks are operated in the same setting. Thus, the robot, or the AR/VR device stores a reference map and builds a new map upon each revisit. However, in-between the revisits, certain objects may have changed. Checking for scene consistency and detecting changes on an object-level can thus lead to 3D object discovery, without the need of labeled data.

Motivated by the above, we explore an object discovery approach, based on examining scene consistency on an object-level and without using annotated data. We are aiming at discovering entities (objects) that have changed when revisiting a place. We show that it is possible to detect 3D objects purely geometrically, without a predefined notion of objects. The underlying idea is that objects, unlike the static background of a scene, can be moved. This is an intuitive definition of "objectness" that does not need any annonated data.

Segmenting dynamic objects in temporal observations is a long-standing challenge. There are two ways to apply this idea: (1) segment objects from the

background by actively observing their motion, e.g., by reconstructing dynamic objects during SLAM [38], or (2) revisit the same scene after a (longer) period and detect potential objects as changes between two maps [12]. We follow the latter approach, i.e., we model the problem as a change detection task.

Detecting potential scene changes based on direct data analytics is a task attracting much attention since affordable 3D scanning technology [11, 36, 2] makes such data widely available. However, a straightforward approach to detecting changes between two scans based on voxel occupancy or inconsistency maps [25] would often miss changes, e.g., when an object rotates around an axis passing through the object or when it is "slid along itself". An alternative approach employs the comparison of visual features and relies on photoconsistency constraints [32]. Yet, this approach does not perform well in our setting since there can be significant illumination changes between the two maps.

To tackle the aforementioned shortcomings, we introduce a novel change detection framework, depicted in Figure 1, that uses geometric transformation consistency towards object discovery (i.e., change detection on an object-level). 3D objects are thus discovered without the need to encode what an object is. We consider a scenario where we have two 3D maps, i.e., a reference scan (recorded at time t_0) and a rescan (recorded at time t_1), of a scene, as well as the associated camera poses. Initial change detections are computed as differences in the depth

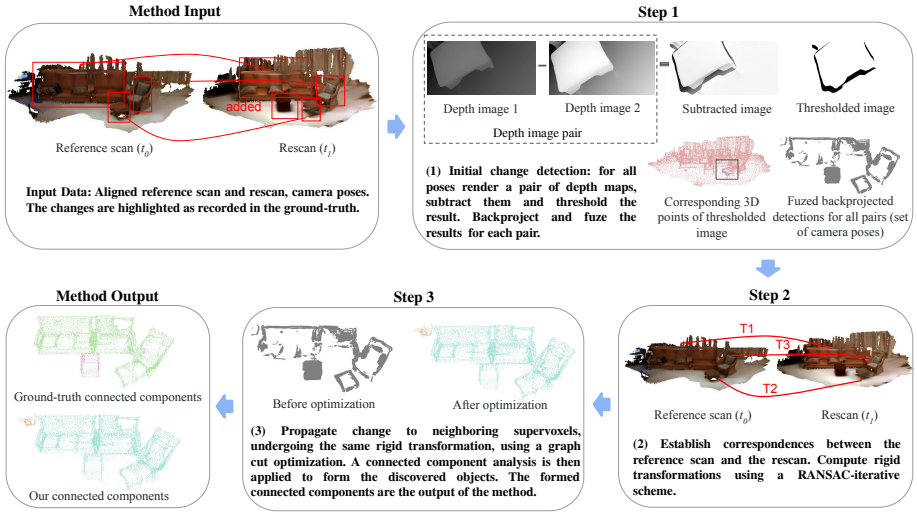

Fig. 1. Workflow of the proposed method: given two scans recording changes and the associated camera poses, we discover all objects that have been added, moved, or removed from the scene. Initial geometric changes are detected as differences in depth maps (Step 1). The dominant transformations are then computed (Step 2). The initial set of detections is incomplete and thus refined, using a graph cut-based optimization on a supervoxel representation, propagating change to all regions undergoing the same transformation (Step 3). Discovered objects are presented as the extracted connected components of the refined detections

maps. As shown in Figures 1 and 2, the initial detected points mainly delineate the boundaries of the moved objects. To recover all parts, we propagate changes from regions where we can detect them to parts where no changes were seen, but which belong to the same object. Our local robust feature matching between parts of the two scans generates motion hypotheses for the scene parts, induced by the moved objects. These motions can measure consistency as scene parts that undergo the same rigid transformation.

Contributions. We introduce a novel 3D change detection framework via geometric transformation consistency. As change detection is performed on an object-level, this novel framework serves as an object discovery method in 3D scenes, without needing any strong priors or definition of what objects are. We showcase that even though we target rigid objects/changes, our method can also handle non-rigid changes, as shown in Figure 4. The proposed method achieves state-of-the-art performance on the 3RScan dataset [36], against competitive baselines.

We evaluate our framework on the 3RScan dataset [36], initially designed for benchmarking object instance relocalization. Our evaluation shows the potential of the dataset to assess 3D change detection. We provide code to generate the ground truth annotations.

2 Related Work

Change Detection. 3D Change detection is directly related to our method since the presented workflow is modeled in this concept. Change detection has been traditionally treated mostly by geometric approaches [33, 32, 24, 25, 39, 35]. Similar to our initial detection step, [32, 33, 24] detect changes based on inconsistency maps from RGB or depth projections. Many change detection algorithms [32, 26] are based on the concept of initial change detection (e.g., though color consistency, comparing depth values, etc.), followed by propagating these detections to identify all regions that have changed. [32, 26] propagate change using spatial and photoconsistency constraints. Our approach follows the same outline, but differs in the key step of change propagation, through a novel geometric twist. Thus, our method is illumination invariant and can be applied to complex, open-set environments under varying illumination conditions.

SLAM Methods for Dynamic Object Segmentation. When addressing dynamic scenes, tracking dynamic objects can be part of SLAM-based techniques. In [1], dynamic parts of the scenes are recovered and a classifier is trained on them to distinguish between static and non-static parts. Semantic SLAM for dynamic environments is presented in [7, 8]. In [29], the authors first segment objects and track them separately. In a similar vein to our research, [10, 14, 23] aim at discovering objects through change observation on an object-level. However, these works build their methods upon a SLAM-based basis. Our method is complementary to SLAM-based techniques since these methods demand the recording of the object's actual movement in front of the camera. On the other

hand, our method needs two 3D models (reference scan and rescan), and the associated camera poses, which are acquired over long time intervals. Thus, objects might have moved, appeared, or disappeared without their movement being explicitly recorded.

3D Object Discovery. Our problem can be conceived as a 3D object discovery technique when declaring as an object everything that can be moved, since movement is an inherent property of objects. Concerning unsupervised object discovery, the authors of [16] focus on identifying parts of the input mesh as candidate objects. They then classify them as an object or clutter. More similarly to our work, [21, 20] extract as objects all the novel additions to the scene. Indeed, by scene comparison, they discover and label as an object anything that has been added between two scans. In contrast, our proposed method does not restrict itself only to added objects, but rather discovers all the objects that have changed (added, moved or removed).

3 Detection via Geometric Consistency

Our method aims at detecting changes on an object-level, thus leading to object discovery, without relying on annotated data. Given two 3D scans, i.e., a reference scan and a rescan, and the associated camera poses, we propose three discrete steps, as illustrated in Figure 1: (1) initial change detection, i.e., compute the locations where a change might have occured, (2) compute dominant transformations, and (3) graph optimization to ensure geometrical transformation consistency. Differences in depth maps provide an initial but incomplete set of detections later refined using a graph cut-based optimization. The central insight is that scene parts that belong to the same object should undergo the same physical transformation, which we model through a novel geometric transformation consistency measure. A connected component analysis is then applied to form the discovered objects.

Initial changes are calculated by depth map comparison. Given a reference scan S of the scene and a rescan R aligned to each other, we render and subtract depth maps. Their subtraction records changing depth values and thus indicates changed regions. However, due to the way the objects move, it is difficult to retrieve the whole object via this single step (as illustrated in Figure 2). To tackle this limitation, we integrate graph optimization [19], performed on supervoxels [27]. Instead of using a simple voxel representation of the scene, we firstly compute supervoxels, i.e., irregular clusters of 3D points sharing common geometrical and color characteristics. Optimizing this representation leads to more accurate results, since supervoxels separate the 3D space into elements, by clustering points with same properties. This is not the case for voxels that are created solely on spatial relations of the 3D points. Moreover, as supervoxels are irregular patches of 3D points, they can preserve objects' boundaries, contrary to the simple voxel representation.

Graph optimization aims to enforce consistency for all the regions undergoing the same rigid transformation. This will help us discover parts of the moving

object that may have been missed during the initial detection step. The change is propagated to all the supervoxels undergoing the same movement. From the above, it is clear that two steps are needed before the optimization: (1) initial change detection, and (2) computation of all the dominant transformations induced by moved objects. Towards the latter goal, learned descriptors [28] are extracted for each point in the scans. We use a pre-trained model, trained on a completely different task (i.e., semantic segmentation). Matches are then computed using nearest neighbor search [15]. The resulting correspondences are used to calculate the 3D transformations.

Scan Alignment. Works and datasets [36, 12] exploring changing indoor scenes demand the two scans to be registered. These datasets provide information for the alignment since registering the scans is outside the scope of their research. Similarly to these works, we use the initial alignment provided by the dataset, which was obtained via manual annotations and is imperfect. In practical applications, the alignment could be provided by re-localization to the previous scan, or by estimating the overall transformation via feature matching [3].

3.1 Initial Change Detection

The first step of our method is identifying changing regions, which we will refine via a graph optimization. Initial change detection is based on depth map comparison. We render depth maps $\mathcal{D}_{S,1,..,N}$, $\mathcal{D}_{R,1,..,N}$ from the reference scan S and the rescan R respectively, for all the viewpoints $i = 1, 2, .., N$, using the $\mathbf{P}_{1,..,N}$ projection matrices. Multiple poses cover the whole 3D scene. We use the same poses to render both the reference scan S and the rescan R, as we assume that both scans are already aligned, even if captured from different viewpoints. We render the depth images rather than simply use raw depth measurements captured by a device to ensure the best possible quality. Moreover, we use depth maps instead of directly working on the mesh, allowing us to handle occlusions and partial observations more naturally than in the 3D space. Working on depth images provides information about free space, which is not directly included in the mesh. Indeed, rendering depth allows us to know if the corresponding 3D region has been scanned or not. If one of the depth images does not contain information, we exclude this region from the initial change detection. This procedure is not straightforward in the mesh, as an intermediate step such as calculating the bounding box of the scan or computing overlapping regions would be needed to ensure that partial observations and free space is taken into consideration. Rendered maps from the reference scan and the rescan are shown in Figure 1. Lighter regions correspond to regions that lie closer to the camera. The paired depth maps are subtracted, and the result is thresholded using [4].

The result is a binary mask, encoding information about changing regions, which are back-projected to the 3D space:

$$[X, Y, Z]^T = \mathbf{R}^T \cdot (\mathbf{K}^{-1} \cdot [x, y, 1]^T \cdot D(x, y) - \mathbf{Tr}) \ , \tag{1}$$

where $[X, Y, Z]$ stands for the world-coordinates of the 3D point, $\mathbf{R}$ for the rotation matrix, $\mathbf{K}$ for the calibration matrix and $\mathbf{Tr}$ for the translation vector, all forming the projection matrix $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{Tr}]$. Vector $[x, y, 1]$ represents the pixel coordinates and $D_{1,..,N}(x, y)$ the depth value stemming from the combination of the depth of the reference scan and the depth of the rescan:

$$D(x,y)_{1,..,N} = \begin{cases} D(x,y)_{S,1,..,N} & \text{if } D(x,y)_{S,1,..,N} < D(x,y)_{R,1,..,N} \\ D(x,y)_{R,1,..,N} & \text{if } D(x,y)_{S,1,..,N} > D(x,y)_{R,1,..,N} \end{cases} . \quad (2)$$

$D(x, y)_{1,..,N}$ represents the depth value at position (x, y) for the combined masks $1, .., N$, $D(x, y)_{S,1,..,N}$ the depth value at position (x, y) for the $1, .., N$ rendered depth maps of the reference scan S and $D(x, y)_{R,1,..,N}$ the depth value of the rendered $1, .., N$ depth maps of the rescan R. This formulation always selects the closest objects to the camera. After experiments, we concluded that in 97% of examined cases, the smaller depth value corresponds to an object. In contrast, the larger depth corresponds to the static background. Figures 1 and 2 depict all the initial points labeled as changing for example scenes. Finally, as the graph optimization is applied to the supervoxel representation, the supervoxels for the scan and the rescan are extracted [27] and the number of changing points belonging to each supervoxel is computed.

3.2 Computing Dominant Transformations

As seen in Figures 1-Before optimization, 2-Initial detected points, our initial detection step may miss changes due to occluded parts of the objects, objects partially captured in one of the scans or due to the way objects have moved. Indeed, when an object is only slightly moved or rotated, there can be regions where the depth values do not change, e.g., when a couch is only slightly shifted. As a result, the initial detections might only cover part of the object. To this end, we use a graph optimization to propagate change detection to the rest of the objects based on consistency under geometric transformations $\mathbf{T}$. We thus compute the different 3D rigid transformations $\mathbf{T}$, induced by the moving objects. Towards that, we match feature descriptors between scans.

Descriptors can be computed using hand-crafted features, such as FPFH [30] and SHOT [34], or learned features. In our case, we use features extracted from the encoder part of a pre-trained deep network. A forward-pass was deployed to densely extract descriptors from the scans, using the weights of pre-trained models on a semantic segmentation task. The models we are using are trained for a completely different task and dataset. Yet, using learned features does not affect our assertion of presenting an unsupervised method.

Correspondences were then computed over the entire scene, using the nearest neighbor search. Visualizing correspondences for the different features showed that the pre-trained Dynamic Graph CNN (DGCNN) [37] had the best preliminary results. To remove outliers from the matches, and given that we want to establish correspondences only between moving objects, we eliminate correspondences lying within a predefined distance of each other in 3D, as these points

are considered part of the static background. All the valid correspondences are then employed to compute the potential transformations using RANSAC [5]. We iteratively apply RANSAC on the remaining set of matches after removing the inliers of the previous estimate and stop once less than three matches remain. Since this method will generate more transformations $\mathbf{T}$ than the real ones due to limitations in establishing correspondences, we will continue selecting the top k transformations $\mathbf{T}$, with the most inliers, to propagate change during graph optimization.

3.3 Supervoxel Graph Optimization

From the initial change detection (Section 3.1), we obtain an initial soft labeling L, based on the fraction of changing 3D points belonging to each supervoxel. Supervoxels with more points labeled as changing, during the initial change detection, are more likely to belong to an object. Thus, the initial labeling L determines the probability ρ of each supervoxel v_i to be labeled as changing $\rho(v_i, l_i = 1)$, or non changing $\rho(v_i, l_i = 0)$ as:

$$\rho(v_i, l_i = 1) = \begin{cases} 0.8 & \text{if changing points} \in v_i \\ 0.5 & \text{if changing points} \notin v_i \end{cases}, \tag{3}$$

$$\rho(v_i, l_i = 0) = \begin{cases} 0.2 & \text{if changing points} \in v_i \\ 0.5 & \text{if changing points} \notin v_i \end{cases}. \tag{4}$$

The weights used in Equations 3 and 4 were chosen based on experiments with different values on a set of scans used for tuning hyperparameters (cf. Section 4). From the above, it is clear that we treat supervoxels with no detected changing points as equally likely of having changed. Indeed, supervoxels without any detected changing points do not necessarily correspond to static scene parts. We thus decide whether a supervoxel belongs to a moving object or not by solving a graph optimization problem [19] that allows us to propagate changes between supervoxels, conditioned on a rigid transformation $\mathbf{T}$.

To deploy the optimization, we create an undirected graph $\mathcal{G} = (V, E)$. Each node $v \in V$ corresponds to a supervoxel in the scene. Two nodes v_i and v_j are connected through an edge $\{v_i, v_j\} \in E$ if the corresponding supervoxels are adjacent to each other. Given a rigid transformation $\mathbf{T}$ (cf. Section 3.2) between the rescan and the reference, our goal is to compute an optimised binary labeling $\mathcal{L}^* = \{l_i^*\}_i$. This labeling indicates for each supervoxel whether it belongs to a changing object consistent with $\mathbf{T}$ (and thus is labeled as $l_i^* = 1$) or not ($l_i^* = 0$). We compute this labeling by solving the graph optimization problem [19]:

$$L^* \in \underset{Q \in \Omega^v}{\arg\min} \{\Phi(L, Q) + \lambda \Psi(Q)\} \ . \tag{5}$$

Φ stands for the fidelity term (here, we use the Kullback-Leibler fidelity function [18]), Ψ for the regularizer, λ for the regularization strength, and Ω for

the search space. The fidelity term $\Phi(L, Q)$ enforces the influence of the initial labeling L, i.e., it decreases when Q lies closer to L. The regularizer $\Psi(Q)$ favors geometrically smooth solutions, i.e., it enforces smoothness to all neighboring supervoxels v_i and v_j, undergoing the same transformation $\mathbf{T}$. $\Psi(Q)$is based on a Potts penalty function:

$$\Psi_{Potts}(Q) = \begin{cases} 1 & \text{if } v_i, v_j \text{ consistent under } \mathbf{T} \\ 0 & \text{otherwise} \end{cases} . \tag{6}$$

It is important to note here that the energy function from Equation 5 is conditioned on each computed transformation $\mathbf{T}$. Thus, we iterate through the top k computed dominant transformations and we solve a series of graph cuts problems. Each iteration segments out the object undergoing the specific transformation $\mathbf{T}$. Objects added or removed, for which a transformation $\mathbf{T}$ is not established, are solely retrieved, based on the unary potentials of the optimization. The results of the iterative procedure, i.e., the set of points labeled as changing, are finally fused. A connected component analysis is finally applied to the fused results, to discover the final 3D objects. Connected component analysis is crucial to form the 3D added or removed objects that are not conditioned on a transformation, but also to overcome the problem of over-segmentation when slightly different transformations are computed for the same object. Optimization results are illustrated in Figures 1, 2 and 3.

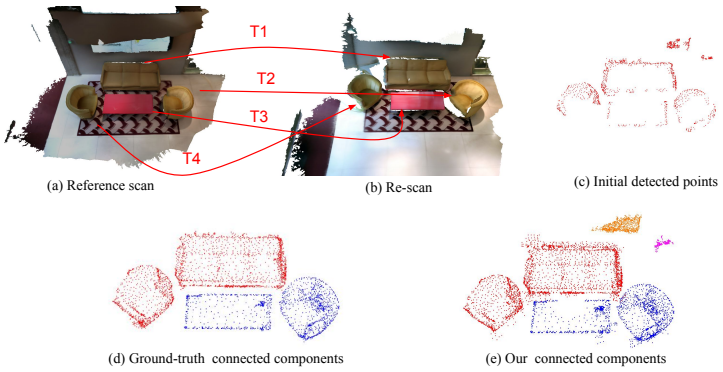

Fig. 2. Our approach: given two scans depicting a scene that has potentially changed, we discover all changes on an object-level. We initially detect potentially changed scene parts by comparing depth maps. We then propagate changes and segment out changed regions based on the principle of geometric transformation consistency. (a) Reference scan (b) Rescan (c) Initially detected areas, with false detections on the wall due to misalignments between the scans. (d) Ground truth connected components and (e) connected components detected by our approach

4 Experimental Evaluation

Datasets. To assess the performance of the proposed approach, we have conducted experiments using the 3RScan dataset [36]. The dataset comprises individual rooms, capturing natural changing indoor environments. It provides, apart from the 3D meshes of the reference scans and the rescans, a series of RGB-D images captured by a Google Tango mobile phone and information concerning objects that have changed between the scenes, along with corresponding transformations. The experiments have been conducted on the validation subset of the dataset comprising 47 different reference scans and 110 rescans. It is important to note that the 3RScan dataset was built initially for object instance relocalization tasks. Therefore, we had to generate the ground truth data for the changing objects based on the dataset's supplementary information. The code is publicly available at https://github.com/katadam/ObjectsCanMove, to enable the usage of this dataset for benchmarking indoor 3D change detection.

To the best of our knowledge, there is no other appropriatbe benchmark to assess 3D indoor change detection and 3D object discovery. Relevant works evaluate their methods on their own datasets, which are either not publicly available [10, 21, 17] or are very small and require manual labeling, as they do not provide appropriate annotations [9, 23, 1]. Please refer to the SM for specific information on discarded datasets.

Hyperparameter Tuning. Ten randomly selected scans from the training split of 3RScan were used for parameter tuning, while the validation split was used for evaluation. The validation split covers many different scene types (i.e., offices, restaurants, living rooms, kitchens, etc.) to assess the generalization performance and robustness to challenging conditions and unseen environments. In our method, the main hyperparameters that need to be tuned are: the RANSAC inlier threshold t for computing transformations $\mathbf{T}$, the number k of transformations $\mathbf{T}$ to compute, and the weights for the graph optimization (as described in Section 3.3). The threshold t can be set intuitively by the desired resolution of the transformations. The number k of transformations should be set to the number of objects that change in a scene. Overestimating k is not an issue as beyond the actual number of objects, RANSAC will be applied to outliers. Alternatively, one could also just stop once only a few matches are left or once the best model found by RANSAC only has a few inliers.

Baseline Methods. To compare the performance of our novel framework against a competitive set of other methods, we have searched for appropriate baselines. However, we had to discard some works treating indoor change detection and unsupervised object discovery, since they are not directly comparable to our method. More particularly, the input to our method are two scans, with changes between them but no recorded actual motion in front of the camera. As such, it is complementary to SLAM methodologies and technologies built upon SLAM systems [10, 14, 1, 9]. Moreover, even though [12, 36] deploy their methods on changing indoor scenes, they focus on instance segmentation and object instance re-localization, respectively. Thus, they cannot be evaluated against our

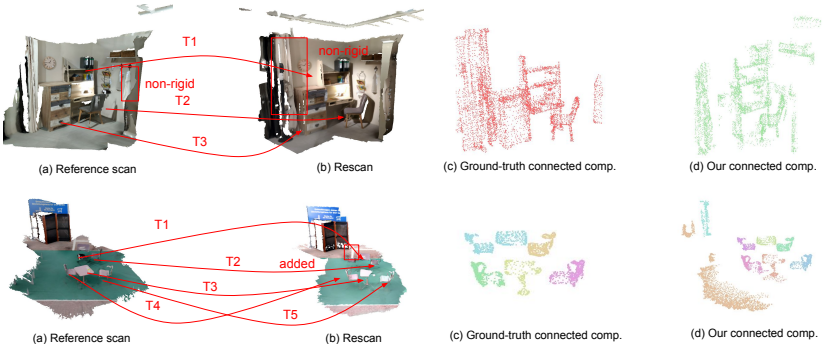

Fig. 3. Qualitative evaluation of the proposed method. Given two scans (a reference scan (a) and a rescan (b)), we perform change detection on an object-level basis, to discover 3D objects. We visualise the final results after applying connected component analysis to the ground truth (c) and to our detected changes (d)

change detection task. Approaches like [20, 23] integrate semantics, contrary to our approach that discovers object-level changes without having a predefined notion of what an object is. The input of [17] is data from range sensors and a highly precise 3D map created by a 3D laser scanner, which is not the case for the 3RScan dataset. [21] aims only at discovering novel objects in the scene, while our approach retrieves all the changed objects. Towards that, we have a created a sub-task of discovering only added objects and compare against [21]. Results are available in the SM. Finally, since we aim at change detection on an object-level towards unsupervised object discovery, we compared against an unsupervised 3D object discovery method [16]. [16] first discovers segments and then classifies them into objects and non-objects. However, the segments obtained via the authors' code after tuning parameter were not meaningful and we were not able to avoid a severe oversegmentation. Thus, we did not include the metrics in our experimental results. For visualizations please refer to the SM.

Our approach is mainly inspired by the change detection approach from [32, 24, 25]. To the best of our knowledge, these are the most closely related baseline and one of our motivations to redefine this problem in a new framework by taking advantage of modern representations (i.e., supervoxels) and more recent graph optimization algorithms [19]. Similar to our work, [25, 24, 32] are also focusing on unsupervised change detection. Taking all the above into consideration, we have decided to create two main baselines inspired by these works.

In [32], change detection is based on inconsistency maps, formed by subtracting pairs of images taken at different points in time. The newly acquired image is warped into the old one, using the known 3D scene geometry and the known poses of both images with respect to the scene. Assuming similar illumination conditions, the two images should be identical if no change in the geometry has happened. In turn, changes in scene geometry will lead to inconsistent projections from one image into the other. Change detection is then optimized via a

graph cut on the voxelized representation. The inconsistency maps are used to calculate the unary term of the graph, while the binary term accounts for color smoothing. Similar to the first step of [32], [25, 24] are discovering changes by formulating inconsistency maps. These works augment the number of inconsistency maps to achieve better results without any further optimization.

Since two 3D models are available in our case, we use the initial change detection step from Section 3.1 to create the inconsistency maps for the two baselines inspired by [32, 24, 25]. We go one step further and resort to depth images instead of RGB images to ensure robustness to illumination conditions. This initial change detection step (i.e., our method before optimization) serves as the 1st baseline, namely **Papazzolo et al.**, as it is equivalent to the work presented in [25, 24]. In these works, estimation of 3D change detection results from back-projecting inconsistencies from multiple 2D maps.

A 2nd baseline (**Taneja et al.**) is formed, following [32], where the initial change detection is optimized ensuring color consistency on a voxelized representation of the scene via a graph cut optimization (solved by max-flow algorithm [6]). The binary term of the graph is computed as described in Equation 7:

$$\psi_{ij}(l_i, l_j) = [l_i \neq l_j] \cdot \gamma / (\sum_{I_t} ||v_t^i - v_t^j||^2 + 1), \tag{7}$$

where $||v_t^i - v_t^j||^2$ accounts for the L2-norm between RGB values of voxels v_t^i and v_t^j and γ is a regularization factor. Comparing against this baseline shows the impact of using geometric consistency for propagating change, which is the main technical contribution of this work.

Ablation Study. Three more baselines are formed in the form of an ablation study, for a better insight into the proposed method. Ablation baselines are reporting intermediate results of our framework. They also calculate the metrics when the method has access to more information, in order to test its robustness with respect to different parameters. Removing the optimization part of our method and relying only on initial change detection is equivalent to [25] and thus reported in Table 1. The first ablation baseline (**ground truth transforms.**) ensures geometric consistency using the ground truth transformations provided by the dataset instead of our computed ones. This gives an upper bound to the performance we can achieve and helps measure the impact of estimated transformation's accuracy on the overall system's efficiency.

As the 2nd ablation baseline (**RANSAC inliers**), closely related to [31], we present the metrics of the non-static points used to form the matches and compute the rigid transformations $\mathbf{T}$. This is equivalent to only the second step of our method (Section 3.2), without the initial change detection and the graph optimization. Ideally, each set of inliers consistent with each RANSAC execution would form the corresponding object moved under this transformation.

Finally, as the 3rd ablation baseline (**Mask-RCNN**), we add a semantic component to the formulated algorithm, as we would like to get an idea of how well our approach performs with respect to a supervised method. Thus, we replace our novel geometric consistency-based term with a term based on

Fig. 4. A non-rigid change (curtain) is not recorded in the ground truth. The curtain is different between the reference (a) and rescan (b), as shown when the two scans are overlaid in (c). The detected changes are shown with red colour in (d), overlaid on the reference scan in blue

Table 1. Mean IoU and mean recall for the proposed method and the published baselines

Method	IoU(%)	Recall(%)
Palazzolo et al.[25] / Ours bf optim.	54.23%	31.48%
Taneja et al. [32]	48.10%	44.50%
Our method	68.40%	76.05%

Table 2. Mean IoU and mean recall for the proposed method and the ablation study's baselines

Method	IoU(%)	Recall(%)
Our method	68.40%	76.05%
Ground truth transforms.	72.40%	93.89%
Mask-RCNN	52.96%	89.22%
RANSAC inliers	10.82%	29.50%

the instance labels of Mask-RCNN [13], propagating the change to all regions sharing the same semantic label. Mask-RCNN is a powerful method for 2D object detection. We deploy the 2D object detector, trained on the COCO dataset [22], on the RGB images of each rescan.

Experimental Results. In addition to the qualitative results presented in Figures 1, 2, 3, we rigorously evaluate our method by using metrics that capture the success of 3D change detection and 3D object discovery. Since we are aiming at object discovery through change detection on an object-level basis, we should first evaluate the efficiency of our change detection results. Thus, we calculate the metric of recall, on a voxel basis. Recall aims to calculate how many of the ground truth changed voxels have been correctly retrieved.

Moving on to 3D object discovery, a 3D connected components analysis is applied to the change detection results. To assess the efficiency of the proposed method as an object discovery pipeline, we deploy the metric of Interestion Over Union (IoU) per discovered object, as it encapsulates both the metrics of precision and recall. To calculate this metric, the connected component analysis is also applied to the ground truth changes. For our scenario, this analysis was performed on a voxel grid of 10 cm, which could sometimes merge objects that

lie together into a single component. This does not affect our metrics since the same connected component analysis is applied both to the ground truth and our solution. However, a smaller step size would lead to a more refined and detailed object discovery. The parameter can be tuned based on the size of the objects we want to discover. We consider an object as successfully discovered when the metric of IoU is more than 20%. The metrics are calculated at a voxel-level since we are interested in measuring how two objects (volumes) intersect.

Tables 1 and 2 show the mean recall over all the scans and the mean IoU of discovered objects. After close examination, it is clear that our method outperforms the most competitive baseline based on [32] by roughly 30% in terms of recall. It also improves the mean IoU by almost 20%. This shows that not only supervoxels constitute a more efficient representation compared to single voxels, when it comes to graph optimization, but also that the novel geometric transformation consistency is much more successful for propagating change, compared to photoconsistency. Moreover, evaluation metrics before and after graph optimization, demonstrate the importance of the optimization, as it improves the mean IoU by 14.17% and the mean recall by 44.57%. As explained above, the method presented in [24, 25] is equivalent to the first step of our method, thus showing improved performance of our presented framework over all published baselines. Integrating a voxel graph cut optimization, propagating change to color-consistent regions [32], leads to better recall rates, but lower IoU, as change is in some cases overpropagated, resulting in low precision, an thus failure of discovering the objects, in terms of IoU.

Concerning the ablation, as denoted by the results of the MASK-RCNN baseline, adding a semantic component is not improving the overall performance. The MASK-RCNN baseline is capable of achieving a mean IoU of 52.96% and a mean recall of 89.22%. This can be attributed to noisy RGB-D images, leading to inaccurate segmentations. Indeed, background patches are falsely detected as foreground objects. Thus, change is propagated into a large percentage of the scene's background, leading to a higher recall rate, compared with a relatively low precision, and thus IoU. The solution using the ground truth transformation (baseline ground truth transforms.) is in close proximity with our method in terms of recall. Even a coarser estimation of the rigid transformation of the scene is capable of achieving close to the best possible results. However, there is still space for improvement, regarding the computation of transformations. The mean IoU of 72.40% in this baseline, is explained due to initial false detections, caused by occlusions and misalignments between scans. The experimental results indicate that the two scans need to be correctly registered to avoid false initial detections. False initial detections are merged with correctly estimated regions, reducing the IoU score. Finally, it is worth mentioning that using only non-static parts discovered by RANSAC iterations leads to results worse than our solution before graph cuts optimization. This explicitly demonstrates that the straightforward approach of feature matching and computing sets of motion-consistent points is insufficient.

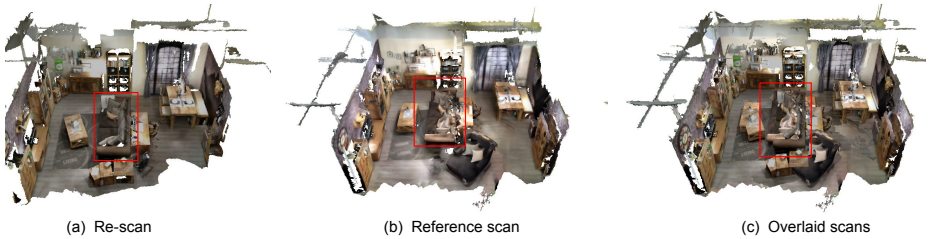

Fig. 5. A moved couch between the reference scan (a) and the rescan (b) that is not part of the ground truth annotations. Overlaid scans in (c)

Finally, 3RScan is a dataset built towards assessing object instance relocalization and not exhaustive change detection. Thus, our method uncovers changes between the scans not recorded in the ground truth. Such cases would affect the evaluation metrics, and we wanted to check their extent. We randomly selected a subset of 10 rescans and visually inspected them. In 60.00% of cases, we discovered an unrecorded change (see, for example, Figure 5). The proposed approach has correctly detected 66,67% of these cases. Moreover, an example of a non-rigid and not recorded change is depicted in Figure 4.

Limitations. By definition of discovering objects via change detection, we will miss objects that do not undergo a substantial enough change. Using a stricter threshold for distinguishing between inliers and outliers in the RANSAC scheme could help recover even small motions. Moreover, depth map subtraction could lead to false initial change detection when the two scans are not entirely aligned. A typical example is illustrated in the second row of Figure 3. Parts of the floor are labeled as changing, forming a 3D object due to the scan's misalignment.

5 Conclusion

The presented method achieves state-of-the-art performance on the object discovery task, via change detection on an object-level basis, for the 3RScan dataset against a competitive set of baselines. The method shows the surprising effectiveness of using scene change for high-recall object discovery and of using motion constraints to achieve precise detections. The very general assumption that objects are connected and move in a coherent way is used to propagate initial detections. Importantly, these geometric cues can be discovered directly from unannotated data, so they do not introduce strong priors or any memorization of what objects are.

Acknowledgment. This research was supported by projects EU RDF IMPACT No. CZ.02.1.01/0.0/0.0/15_003/0000468, EU H2020 ARtwin No. 856994 and the EU Horizon 2020 project RICAIP (grant agreement No 857306).

References

1. 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. Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S.: 3d semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1534–1543 (2016)
3. Bai, X., Luo, Z., Zhou, L., Chen, H., Li, L., Hu, Z., Fu, H., Tai, C.L.: Pointdsc: Robust point cloud registration using deep spatial consistency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15859–15869 (2021)
4. Barron, J.T.: A generalization of otsu's method and minimum error thresholding. In: European Conference on Computer Vision. pp. 455–470. Springer (2020)
5. Bolles, R.C., Fischler, M.A.: A ransac-based approach to model fitting and its application to finding cylinders in range data. In: IJCAI. vol. 1981, pp. 637–643. Citeseer (1981)
6. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE transactions on pattern analysis and machine intelligence **26**(9), 1124–1137 (2004)
7. Brasch, N., Bozic, A., Lallemand, J., Tombari, F.: Semantic monocular slam for highly dynamic environments. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 393–400. IEEE (2018)
8. Cui, L., Ma, C.: Sof-slam: A semantic visual slam for dynamic environments. IEEE Access **7**, 166528–166539 (2019)
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)
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)
11. 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)
12. Halber, M., Shi, Y., Xu, K., Funkhouser, T.: Rescan: Inductive instance segmentation for indoor rgbd scans. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2541–2550 (2019)
13. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
14. Herbst, E., Henry, P., Ren, X., Fox, D.: Toward object discovery and modeling via 3-d scene comparison. In: 2011 IEEE International Conference on Robotics and Automation. pp. 2623–2629. IEEE (2011)
15. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with gpus. arXiv preprint arXiv:1702.08734 (2017)
16. Karpathy, A., Miller, S., Fei-Fei, L.: Object discovery in 3d scenes via shape analysis. In: 2013 IEEE International Conference on Robotics and Automation. pp. 2088–2095. IEEE (2013)

17. Katsura, U., Matsumoto, K., Kawamura, A., Ishigami, T., Okada, T., Kurazume, R.: Spatial change detection using normal distributions transform. ROBOMECH Journal **6**(1), 1–13 (2019)
18. Kullback, S., Leibler, R.A.: On information and sufficiency. The annals of mathematical statistics **22**(1), 79–86 (1951)
19. 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)
20. 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)
21. 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)
22. 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: European conference on computer vision. pp. 740–755. Springer (2014)
23. Mason, J., Marthi, B.: An object-based semantic world model for long-term change detection and semantic querying. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 3851–3858. IEEE (2012)
24. 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)
25. 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)
26. Palma, G., Cignoni, P., Boubekeur, T., Scopigno, R.: Detection of geometric temporal changes in point clouds. In: Computer Graphics Forum. vol. 35, pp. 33–45. Wiley Online Library (2016)
27. Papon, J., Abramov, A., Schoeler, M., Worgotter, F.: Voxel cloud connectivity segmentation-supervoxels for point clouds. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2027–2034 (2013)
28. Phan, A.V., Le Nguyen, M., Nguyen, Y.L.H., Bui, L.T.: Dgcnn: A convolutional neural network over large-scale labeled graphs. Neural Networks **108**, 533–543 (2018)
29. Runz, M., Buffier, M., Agapito, L.: Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In: 2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). pp. 10–20. IEEE (2018)
30. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: 2009 IEEE international conference on robotics and automation. pp. 3212–3217. IEEE (2009)
31. Steinhauser, D., Ruepp, O., Burschka, D.: Motion segmentation and scene classification from 3d lidar data. In: 2008 IEEE Intelligent Vehicles Symposium. pp. 398–403. IEEE (2008)
32. Taneja, A., Ballan, L., Pollefeys, M.: Image based detection of geometric changes in urban environments. In: 2011 International Conference on Computer Vision. pp. 2336–2343. IEEE (2011)

33. Taneja, A., Ballan, L., Pollefeys, M.: City-scale change detection in cadastral 3d models using images. In: Proceedings of the IEEE Conference on computer Vision and Pattern Recognition. pp. 113–120 (2013)
34. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: European conference on computer vision. pp. 356–369. Springer (2010)
35. Ulusoy, A.O., Mundy, J.L.: Image-based 4-d reconstruction using 3-d change detection. In: European Conference on Computer Vision. pp. 31–45. Springer (2014)
36. Wald, J., Avetisyan, A., Navab, N., Tombari, F., Nießner, M.: Rio: 3d object instance re-localization in changing indoor environments. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7658–7667 (2019)
37. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog) **38**(5), 1–12 (2019)
38. Wong, Y.S., Li, C., Niessner, M., Mitra, N.J.: Rigidfusion: Rgb-d scene reconstruction with rigidly-moving objects. Computer Graphics Forum **40**(2) (2021)
39. Xiao, W., Vallet, B., Paparoditis, N.: Change detection in 3d point clouds acquired by a mobile mapping system. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences **1**(2), 331–336 (2013)