DOI: 10.1111/cgf.142651
EUROGRAPHICS 2021 / N. Mitra and I. Viola
(Guest Editors)

Volume 40 (2021), Number 2

RigidFusion: RGB-D Scene Reconstruction with Rigidly-moving Objects

Yu-Shiang Wong[1] Changjian Li[1] Matthias Nießner[2] Niloy J. Mitra[1,3]

[1]University College London [2]Technical University of Munich [3]Adobe Research

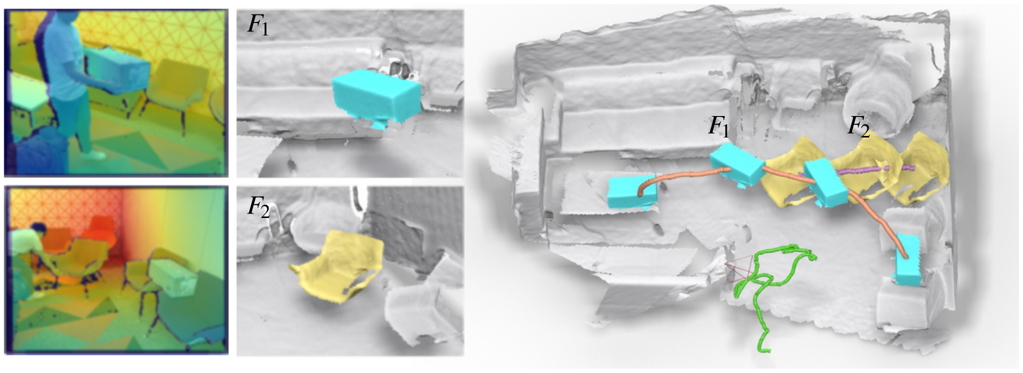

Figure 1: *Given a dynamic scene with rigidly moving objects, RigidFusion performs 4D reconstruction from RGB-D frames (left) and outputs camera motion (shown in green curves), fused object geometries (rendered with light blue and golden yellow), and their respective trajectories (shown in brown/purple curves). Two novel-view reconstructions from two time steps are shown on the middle panel, with frame numbers F_i corresponding to different time steps.*

Abstract

Although surface reconstruction from depth data has made significant advances in the recent years, handling changing environments remains a major challenge. This is unsatisfactory, as humans regularly move objects in their environments. Existing solutions focus on a restricted set of objects (e.g., those detected by semantic classifiers) possibly with template meshes, assume static camera, or mark objects touched by humans as moving. We remove these assumptions by introducing RigidFusion. Our core idea is a novel asynchronous moving-object detection method, combined with a modified volumetric fusion. This is achieved by a model-to-frame TSDF decomposition leveraging free-space carving of tracked depth values of the current frame with respect to the background model during run-time. As output, we produce separate volumetric reconstructions for the background and each moving object in the scene, along with its trajectory over time. Our method does not rely on the object priors (e.g., semantic labels or pre-scanned meshes) and is insensitive to the motion residuals between objects and the camera. In comparison to state-of-the-art methods (e.g., Co-Fusion, MaskFusion), we handle significantly more challenging reconstruction scenarios involving moving camera and improve moving-object detection (26% on the miss-detection ratio), tracking (27% on MOTA), and reconstruction (3% on the reconstruction F1) on the synthetic dataset. Please refer the supplementary and the project website for the video demonstration (`geometry.cs.ucl.ac.uk/projects/2021/rigidfusion`).

CCS Concepts
• ***Computing methodologies*** $\rightarrow$ ***Reconstruction; Tracking;*** *Video segmentation; Image segmentation;*

1. Introduction

Capturing accurate 3D scene geometry from RGB-D input in uncontrolled setups is a long-standing challenge in shape acquisition. Robust solutions now exist for capturing *static* scenes by fusing raw depth scans across multiple frames to recover from incomplete and noisy measurements [CL96; RHL02; NIH*11; CBI13; NZIS13; KPR*15; KDSX15; DNZ*17]. However, there are only limited options for capturing *dynamic* scenes (e.g., requiring background

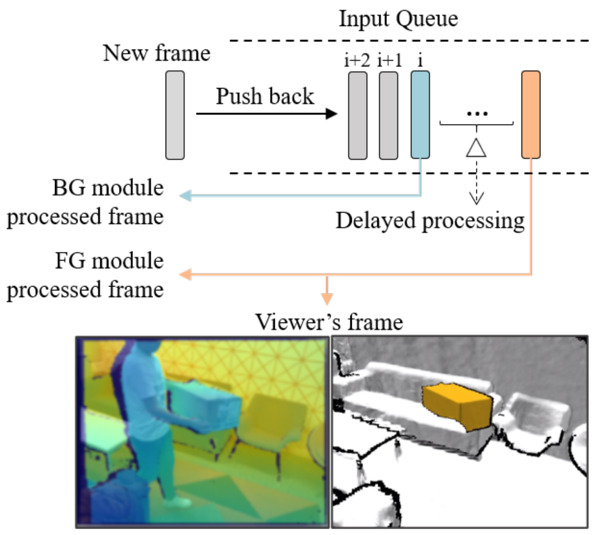

Figure 2: *RigidFusion's reconstruction preview and timeline. Our method runs at 1 fps with a delay of* Δ *frames. The short delay between the background and foreground module allows the system to accumulate more free space information.*

initialization [MS14; RA17], using semantic priors [RBA18; XLT*19a; SS19; HGDG17; XLT*19b; SS19; MCB*18], exploiting scene flow information [BIZ18; GKM*17], or handling deformable objects [NFS15; IZN*16; DKD*16; SBCI17]). This is rather surprising since our surroundings are mostly dynamic as objects are moved around in course of our regular interactions, for instance, a person moves a box, table, or chair. The ability to faithfully record indoor environments with moving objects would open up many possibilities to capture our environments in their natural settings using commodity hardware and, in turn, provide rich data priors in terms of object shape, motion, and interactions.

The primary challenge in reconstructing dynamic scenes is to mask out any human movements, and factorize individual object(s) and camera motion from raw depth scans to produce 4D reconstructions in the form of fused static objects along with their respective motion trajectories in addition to the inferred camera trajectory. This information allows generating a motion-compensated fused output that integrates information from multiple frames, while accounting for individual objects' motions. Note that in absence of prior knowledge about object shape, its deformation model, or motion trajectories, dynamic scene reconstruction amounts to segmenting the scene into (moving) objects and tracking their motion through time. Here, we focus on scenes with rigidly-moving objects, arising due to human interactions. The problem is particularly difficult due to a cyclic dependency: on the one hand, in order to track and group pixels into segments, we have to rely on the coherence of their estimated (rigid) movement; on the other hand, in order to estimate per-pixel trajectories, we need to know segment groupings.

In order to account for the cyclic dependency above, previous work breaks cyclic dependency by discovering foreground objects using motion residuals [RA17] or grouping local motions [GKM*17]. However, the motion cues can be ambiguous, especially when the camera are moving, and the local patches of indoor objects are usually under-constrained [GIRL03]. Our key idea is to discover the foreground object through the accumulated free space information and the background model from multiple frames, which has been shown as an effective signal for outlier removing [PBL*19]. The free-space grid and the background model are used to separate moving elements using a frame-to-model back-projection of the input frames. Instead of performing an expensive global optimization for solving a complete background model, we employ a delayed process at the beginning to regularize the problem, as shown in Figure 2. In addition, we segment out pixels of humans in image space to facilitate real-world scanning scenarios with an operator interacting with moving objects. The prediction of human segmentation may be inaccurate or missing. To purge outlier pixels, we utilize free space information in the foreground model. Aside from this off-the-shelf human segmentation method, our method requires no learning or training data, is agnostic to the underlying volumetric fusion framework, and runs at interactive frame rates after the initialization step.

In addition to real-world scanning cases for qualitative evaluation and comparisons on established datasets (Co-Fusion, MaskFusion), we introduce a new benchmark dataset using synthetic scene models and suitably adapted motion planning library [ŞMK12] to generate physically plausible object motions recorded as RGB-D frames from synthesized camera trajectories. This synthetic data provides ground truth geometry and motion trajectories that we use for quantitative comparison against related methods (Co-Fusion, MaskFusion). The dataset consists of a training set for fine tuning segmentation network for evaluation purpose and a test set as a benchmark dataset. The training set covers $39,068$ scenes and has a total of $151,335$ RGB-D frames and the test set covers 20 scenes and containing 7851 frames. An evaluation protocol is also proposed for quantifying performance of 4D reconstructions. In a series of experiments, we show that RigidFusion outperforms state-of-the-art methods by a significant margin, can handle significantly more challenging real-world cases, and provide reconstruction preview at the run-time.

In summary, our main contributions are:

- a novel joint formulation of segmentation-by-reconstruction and object pose tracking that enables 4D reconstructions through decomposing the underlying TSDF rotations in individually reconstructed fused volumes;
- an end-to-end pipeline that runs at interactive rates combined with a human semantic segmentation network to address a wide range of real-world scanning cases;
- in addition to challenging real-world scenes, we introduce a new dataset for quantitative evaluation (20 scenes) and training semantic priors (39,068 scenes), where our method shows substantial improvements, especially for foreground reconstruction and tracking, over competing methods.

2. Related Works

RGB-D Reconstruction. With the wide availability of consumer-grade RGB-D sensors, such as the Microsoft Kinect, a wide range of dense, RGB-D reconstruction approaches have been introduced. The core idea of these methods is to accumulate a series of RGB-D frames into a shared voxel representation, often a signed distance field [CL96], even at interactive rates [RHL02]. Most recent approaches are highly parallelized and run on the GPU with KinectFusion being one of the most popular methods [NIH*11]. In order to tackle larger environments, sparse scene representations, such as hierarchical data structures [CBI13] or spatial voxel hashes [NZIS13] were introduced that even run on mobile devices [KPR*15; KDSX15]. While these approaches have achieved remarkable success, the main shortcoming is the inability to handle dynamic scenes.

Background Reconstruction in Dynamic Scenes. Generalizing static methods to handle dynamic scenes requires handling of a significant number of background outliers (i.e., moving objects), which must be detected. Therefore, tracking with robust kernel based methods have been proposed [SJP*18; PBL*19] and space information embedded in distance field is used as additional outlier rejection signals [Zha13; PBL*19]. This family of methods, however, does not track and reconstruct foreground objects.

Sparse Motion Estimation and Transformation Clustering. Feature-based tracking has demonstrated impressive results in SLAM applications [MMT15] and provides another avenue to address multiple motions in dynamic scenes, such as clustering sparse point trajectory [HYZ*19] or solving via a graph labeling formulation [JGN18]. Dense motion segmentation problem, however, still remains unsolved. Bertholet *et al.* [BIZ18] proposed an offline method that invokes multiple optimization steps, including sparse optical flow clustering, sparse-to-dense association using Euclidean distance, dense pose estimation, and sparse label assignment using a graph formulation. However, sparse-to-dense association is non-trivial and limited to discriminative texture regions.

Dense Reconstruction in Dynamic Indoor Scenes. Reconstructing dynamic surfaces is also related to our approach. In the context of RGB-D scanning, we have seen several popular reconstruction frameworks running in real-time, including DynamicFusion [NFS15], VolumeDeform [IZN*16], Fusion4D [DKD*16], KillingFusion [SBCI17]. These works focus on the joint reconstruction on tracking of a single, deformable object using either a single or multiple RGB-D stream input with known object segmentation. Our work is complementary to this line, as we aim to reconstruct large scenes in uncontrolled settings with rigidly-moving objects and do not require object segmentation in the input.

Dense Motion Segmentation with Rigid Transformation. Dense reconstruction in the dynamic indoor scenes with rigid motion constraint is still a relatively new research topic. The main challenge is simultaneously handling the uncertainty of detecting unknown moving objects and tracking static background under a moving camera. Ma and Sible [MS14] proposed to detect moving objects in frame-to-frame fashion by examining the background model built in the static initialization stage. Multiple moving objects are extracted by finding the disjoint region on the outlier mask. CoFusion [RA17] follows a similar concept but requires a static period to initialize background models. Then, moving objects are detected using a conditional random field (CRF) with alignment errors, color, depth, and 2D position as features. Note that the above methods [MS14; RA17] relies on using the alignment errors to find moving objects, which implicitly assumes camera motion is static or distinguishable from object motions.

Dense Motion Segmentation with Object Priors. More recently, MaskFusion [RBA18] leverages semantic labels from MaskR-CNN [HGDG17] for detecting objects and identifies moving objects using color, geometry cues, and motion residuals. While such semantic prior is useful, it only provides a rather weak initialization for motion segmentation in complex scenes with rigidly moving objects (i.e., many semantic objects but only a few moving ones) and suffers from low recall. To alleviate the low recall issue, MID-Fusion [XLT*19b] and EMFusion [SS19] accumulate instance prediction and back-project instance priors using volumetric grids. These methods require to know the semantic classes of moving objects and still assume background objects can be filtered out using alignment errors. Other interesting directions include incorporating dense scene flow [MWH*19], or using known object shapes as foreground priors [RPMR13; RPK*17]. In contrast, RigidFusion can handle non-static camera motion, without relying on user input, and is not limited to particular object classes or shapes.

Datasets. In an effort to generate large-scale repositories along with ground truth data in the context of 3D indoor setting, efforts have been devoted to static scenes [ASZS17; DCS*17] or for simulating robotic perception [XRH*18]. Recent frameworks have started investigating synthetic setups with dynamic objects [KMG*17; PRB*18]. However, they either focus on a handful of interactions using static cameras [PRB*18] (e.g., making a cup of coffee), or target on simulating specific object-agent interactions such as Open-and-Pickup [KMG*17]. We introduce a new dataset of synthetic objects rigidly moving along with ground truth trajectories and object reconstructions, and also a set of real world scenes where objects are rigidly moved by human operators.

3. Algorithm

3.1. Overview

Our method takes as input a sequence of RGB-D frames $\{\mathcal{F}_i\}$ captured using a moving camera recording a dynamic scene with rigidly moving object(s). We assume the setup to satisfy two conditions: dense recording, i.e., the input frames come as a continuous RGB-D video; and objects either remaining static or rigidly being moved, one at a time, by a human operator. Note that although we assume a single object to be moving at any time, we do *not* require any object segmentation priors, such as in MaskFusion [RBA18]. As output, we produce a 4D reconstruction in the form of consolidated static background mesh, camera trajectory, and the consolidated foreground object meshes along with the respective object trajectories over time.

In order to break the cyclic dependency between rigid tracking and motion segmentation, RigidFusion proceeds in the following steps: (i) background reconstruction and camera estimation, (ii) asynchronous foreground reconstruction, (iii) optional post-processing, and (iv) mesh extraction. We now describe the individual steps.

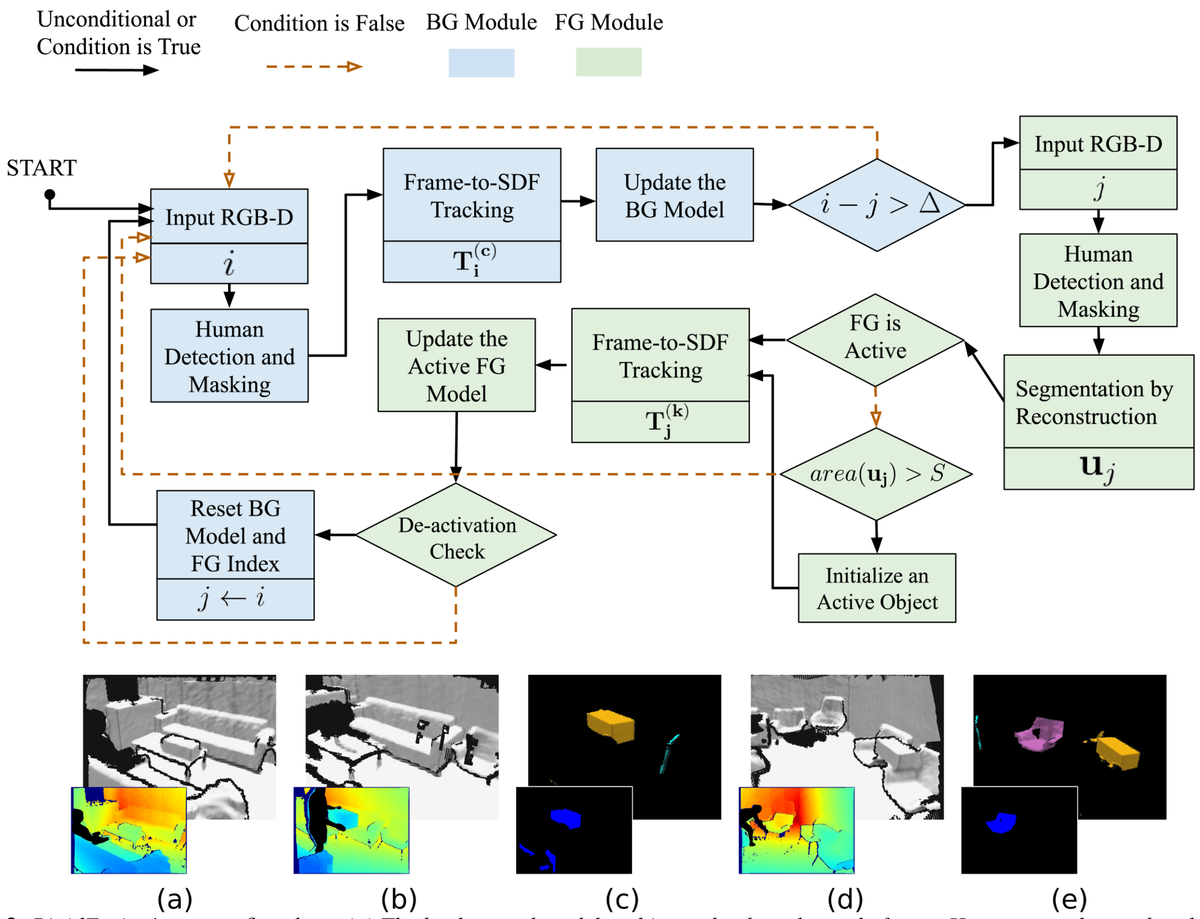

Figure 3: *RigidFusion's system flowchart. (a) The background model and input depth at the early frame. Humans are detected and masked out.(b) The background model and input depth at the time i (equal to $j+\Delta$). (c) The foreground model and the unknown segmentation $\mathbf{u}_j$. (d) The background model is reset (due to the status change of the tracked object trigger the model de-activation). (e) The foreground models and the unknown segmentation at the later frame.*

3.2. Background Reconstruction and Camera Estimation

Human Detection and Masking Since we focus on capturing background and rigidly moving objects, we first perform human detection in each frame $\mathcal{F}_i$. We apply a DensePose [GNK18; WKM*19] detector to acquire a human mask. We reset the depth of the pixels under the detected mask to be *zero* for each frame to serve as the input for subsequent processing. This masking stage not only avoids the existence of non-rigid objects interfering with the camera tracking but also removes the requirement that humans should continuously move, as commonly assumed in [PBL*19; SJP*18].

Free-Space Aware TSDF Fusion To reconstruct a (static) scene, we employ volumetric fusion using sparse voxel hashing on an occupancy grid [CL96; NZIS13]. Specifically, in a volumetric grid, at each voxel we store a signed distance to the closest (recorded) surface as calculated by back-projection from the input depth.

Recall that in the standard implementation, only the voxels near the recorded surface are stored, i.e., a truncated signed distance field (TSDF) is progressively built. Additionally, RigidFusion maintains, using voxel hashing, free space counts to identify free space (outside) voxels, i.e. the voxels that have been frequently observed and have positive distance larger than the truncation margin. In Figure 4, red/blue denote negative/positive SDF values (underlying surface marked in green) that are within the truncation range, while gray cells denote outside free cells. We use this free-space information to prevent foreground signals from polluting the background model.

Camera Tracking Optimization In our tracking formulation, we use 6D vectors $\xi \in \mathfrak{se}_3$, encoded as instantaneous velocity using Lie algebra, to represent a 4×4 rigid transformation T. To transform a spatial point, we convert ξ to a 4×4 transformation matrix

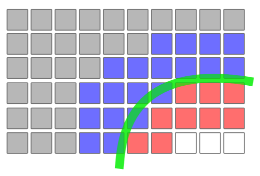

Figure 4: *TSDF illustration. The green curve, gray, blue, and red cells represent the surface, free space, positive and negative truncation regions, respectively.*

using exponential map, denoted by *exp*, as described in [NIH*11; MSL94]. To perform camera tracking, we employ frame-to-SDF registration [BSK*13; PBL*19]. Given the input as a RGB-D frame $\mathcal{F}_i$ at time i, we solve for the best increment transformation ξ to align the input frame to the TSDF model as,

$$\mathbf{T_i^{(c)}} = exp(\xi) \cdot \mathbf{T_{i-1}^{(c)}}, \tag{1}$$

where $\mathbf{T_i^{(c)}}$ represents the estimated camera pose at the frame i. Our tracking objective E includes both geometry and color intensity cost functions and is defined as

$$E(\xi) = E_{SDF}(\xi) + \alpha E_I(\xi), \tag{2}$$

where α is a scalar weight for balancing the two terms. The geometric cost term E_{SDF} minimizes the signed distance value of the transformed input point set P, as any surface voxel has zero singed distance values, and hence a perfectly aligned point set should have a zero residual. The input points are firstly transformed into the world space using the accumulated pose $\mathbf{T_{i-1}^{(c)}}$.

$$E_{SDF}(\xi) = \sum_{p \in P} (\psi_g(\mathbf{T_i^{(c)}} p))^2 = \sum_{p \in P} (\psi_g(exp(\xi) \cdot \mathbf{T_{i-1}^{(c)}} p))^2, \tag{3}$$

where ψ_g represents a signed distance function $\psi_g : R^3 \to R$ that takes a 3D point and returns a truncated signed distance value from the TSDF model. The photometric cost term E_I minimizes the difference of color intensity between the transformed input points $p \in$P to the corresponding voxels in the model, where I represents the input intensity map by converting the input RGB to relative luminance as,

$$E_I(\xi) = \sum_{p \in P} (\psi_I(\mathbf{T_i^{(c)}} p) - I(p))^2. \tag{4}$$

Here, ψ_I represents a color interpolation function $\psi_I : \mathbb{R}^3 \to \mathbb{R}$, which takes a point and returns an interpolated color intensity from the TSDF model. To optimize our objective (2), we apply Gaussian Newton solver while linearizing the objective around the initial ξ.

Background Model Update We take the estimated pose and transform the input frame to integrate the new information. Note that we prevent any surface (depth pixels) being integrated into free voxels. Please refer to [CL96] and our supplementary for details about the integration steps.

3.3. Asynchronous Foreground Reconstruction

We regularize the foreground segmentation problem by assuming only one rigidly moving object at any point. Note that our proposed scheme still captures multiple moving objects over an entire scan session, and perform well when camera motion is large. We tested our method on medium-scale scenes, e.g., 30-100 sqm.

Segmentation by Reconstruction We define the unknown foreground segmentation $\mathbf{u}_j$ at the time j as the depth pixels that unproject to the free space of the background model. We apply connected component filtering to remove small blobs and initialize a new active object if the area of $\mathbf{u}_j$ is larger than a threshold. We empirically set the threshold to 1% of image size. Note that remaining false positives are later removed in our foreground deactivation step (see later) due to the lack of correspondences detected using tracking failures.

Delayed Processing For motion segmentation, a frame-to-frame approach, such as [RA17; SJP*18], cannot guarantee the observed motion signals are sufficient because an object may move slowly. Instead, we delay foreground tracking by a pre-defined window size Δ (60 frames), leading to a delayed reconstruction in the foreground modules at run-time to gather more background information as shown in Figure 2. In other words, when the foreground module is processing the frame at the time i, the background module has processed the frames until the time $i+\Delta$. Hence, the extracted unknown foreground segments $\mathbf{u}_j$ at the time j can directly access the *future* background model during segmentation-by-reconstruction step. This step is similar to doing scene completion during registration [SC18], in the sense that the foreground module can use a completed background model. Note that our asynchronous processing does not add any extra memory overhead and only has a short delay in the beginning.

Foreground Tracking and Reconstruction To estimate the pose $\mathbf{T_j^k}$ of the active object k at the time j, as well as reconstruct its geometry, we apply the same tracking optimization and free-space aware fusion used in background module (in Sec. 3.2). Note that, during foreground tracking, we still perform frame-to-SDF tracking without an instance mask since the non-foreground pixels will receive zero gradients in the foreground TSDF model.

Handling Multiple Objects by Deactivation In RigidFusion, we assume multiple objects do not move simultaneously. We deactivate any active object under any of the following conditions: (i) an active object becomes static over a period of time (i.e., accumulated pose difference is less than 1e-4 over the last $\Delta/2$ frames); (ii) an active object moves out of the camera view; or (iii) an active object cannot be successfully tracked for more than ten frames.

Once a deactivation event is triggered, we temporarily reset the background model (i.e., allocate a new TSDF model) and freeze foreground detection for Δ frames to prevent inactive objects being re-detected in the unknown segment proposal $\mathbf{u}_j$.

3.4. Post-Processing

After processing all input frames, we can perform an optional post-process step (PS). During this step, we re-optimize full 4D reconstruction. First, given the captured foreground models and trajectories, we perform backward tracking for each object from its first detected frame to the first frame. This step can recover some missing frames. Second, we re-build background reconstruction by using a high-resolution grid and re-optimize the camera trajectory $\left\{T_i^{(c)}\right\}$. During this optimization, the already inferred foreground models are used to mask out foreground pixels by depth ray-casting. Thus it improves the camera tracking accuracy.

3.5. Mesh Extraction

To output surface reconstruction, we perform Marching Cube on both foreground and background models, and extract respective 3D meshes along with their corresponding trajectories.

4. Benchmark Dataset

In this section, we introduce our data generation pipeline and benchmark tasks supported by our dataset. We derive our dataset from a synthetic collection of 3D scene layouts and a set of publicly-sourced CAD models. Our goal is to create ground truth data where we virtually move objects around by simulating real-world scanning scenarios with dynamic environments with rigidly-moving objects. To this end, we simulate the motion of moving objects over a series of time steps (frames). We observe the scene from a virtual static/moving camera in a given scenery to render out RGB-D frames, along with ground truth semantic segmentation, and per-object instance labels with associations over frames. Our frame resolutions for RGB-D and labels are 640×480. Along with the respective camera poses and camera intrinsics, this forms the ground truth of our benchmark dataset. In total, our dataset contains 39K scenes, consists of 159K RGB-D frames, poses, and segmentations.

4.1. Data Generation

We developed an automatic data synthesis toolkit to generate motion in indoor scenes. By considering the complexity of the 4D reconstruction problem, our design choices are made to avoid some object tracking challenges, including object re-identification, heavy occlusion, and tracking small objects, to focus on the joint tracking-and-segmentation problem. We selected scenes with sufficient scene complexity with objects that are being moved around, identify a target motion position, and compute a representative motion pattern from source to target motion positions. For each such scene, we select a start camera position and generate a virtual camera trajectory.

In the following, we detail each step:

- **Room selection:** we calculate free space ratio of a room as the amount of empty space left on the floor mask and reject the overly cluttered ones.
- **Object selection:** we (uniform) randomly select objects to be moved using the surface normals to avoid selecting flat objects (e.g., a whiteboard), which does not has good geometry signals for tracking.
- **Goal position sampling:** we sample collision-free positions by moving the selected model in the room.
- **Motion generation:** based on the original object position and the goal position, we generate a motion trajectory while avoiding collisions. We use a tree-based path planner [ŞMK12] to solve for such motions. Note that we restrict movements to 2D rigid rotation and translation on the XZ plane (floor plane) in order to create a more reasonable motion (instead of flying motion).
- **Camera selection:** we sample several camera positions in a room by estimating both 2D and 3D occlusion ratios of the target object using rendered binary images and 3D bounding box of the target. We prefer less occluded views.
- **Generate camera motions:** we generate random translation to move a camera locally and rotate the XZ direction of a camera by using the dynamic object's position and gradually following the moving object.

4.2. Dataset Analysis

By using our toolkit, we generated 39,068 motion sequences spanning 151,335 RGB-D frames and 14 object categories (such as chair, sofa, and bed). The generated sequences are used as a training data for fine-tuning a segmentation network [HGDG17], which is required for instance segmentation based methods [RBA18; XLT*19a; SS19]. Then, for the evaluation purpose (in Section 5), we generated another 20 sequences with longer motion as a test set, which contains 7851 frames in total. The average length of camera movement is 5.1m and object trajectory length is 8.6m. The rooms have an average of 6 models leaving out walls, floors, and ceilings. We select moving objects from the sofa, table, desk categories in order to focus on joint tracking-and-segmentation instead of handling small objects. Two examples are shown in the supplementary.

4.3. Benchmark Tasks

Reconstruction (Precision/Recall) In order to assess surface reconstruction quality on our benchmark, we reconstruct ground truth fusion [CL96] using the ground truth poses, ground truth segmentation, and depth frames. We consider the fusion result to be the ground truth reconstruction mesh M_g. Note that this mesh M_g is different from the actual 3D mesh in the synthetic model since it contains occlusions from the virtual scanning. This usage of M_g aims to evaluate the reconstruction quality instead of the hidden surface prediction. We formulate the evaluation metric between this ground truth mesh M_g and the reconstruction result M_r using a bi-directional *Chamfer* distance (denoted as CD). Specifically, we calculate Precision and Recall based on the squared distance between every vertex in one mesh and the corresponding nearest vertex in the other mesh; i.e., a point is considered to be captured if there is a nearest neighbor within 3cm (Recall); a point is considered to be an outlier if there is no neighbor within 3cm (Precision). Please refer Figure 2 in the supplementary for the visualization.

Tracking (MOTA/MISS/BAD/MOTP) Our dataset provides ground truth trajectories for both the camera and the moving objects. We employ multiple object tracking metrics including multiple object tracking accuracy (MOTA) and precision (MOTP) [BS08]. MOTA shows the percentage of frames are tracked using ground truth mesh centroid c as a landmark (i.e., the frames have position error smaller than 5cm) and penalize drift (BAD), and missing detection (MISS); MOTP estimates the average position error of the tracked frames,

$$MOTP(\{T_i^e\}) := \sqrt{\frac{1}{N}\sum_i^N \|T_i^e c - T_i^g c\|^2}, \tag{5}$$

where T_i^e and T_i^g respectively represent an estimated pose and a ground truth pose of at each frame $\mathcal{F}_i$ and N being the total number of tracked frames. Note that MOTP does not include the errors from the lost tracked frames (BAD frames).

Note that the standard MOTA and MOTP measures are designed for 2D visual tracking and allow trackers to switch targets. In contrast, we enforce one-to-one mapping between the ground truth and the output by calculating MOTA on each trajectory independently and selecting the best pair to evaluate its tracking and reconstruction. Please refer to Section 2 and Figure 3 in the supplementary for corresponding visualization.

5. Results and Discussions

We compare RigidFusion with different state-of-the-art reconstruction methods, including VoxelHashing (VH) [NZIS13], BundleFusion (BF) [DNZ*17], StaticFusion (SF) [SJP*18], ReFusion (RF) [PBL*19], CoFusion (CF) [RA17], MaskFusion (MF) [RBA18], EM-Fusion (EM) [SS19], and MID-Fusion (MID) [XLT*19a] on both synthetic and real-world datasets. We use the above abbreviation of comparison throughout the rest of the paper.

To evaluate with the prior based approach [RBA18], we fine-tuned MaskRCNN on our synthetic dataset (Section 4.2) and pre-computed instance segmentation. Our trained model reaches 70.2 average precision (AP) on instance segmentation task, which shows the model is capable of providing reasonable object priors. For evaluating on real-world data, we used the pre-trained model on CoCo dataset [LMB*14] and manually select the semantic class of moving objects for each testing example.

Computational Time and Implementation Details We measure the computational time of RigidFusion for processing a single frame in each module (Figure 3 and Section 3) as follows - Human detection: 0.38s. BG tracking and model updating: 0.07s. FG tracking and model updating: 0.10s. Segment-by-Reconstruction takes: 0.44s. Due to the asynchronous process (as shown in Figure 2), our system has $(0.38+0.07)\times\Delta$ seconds delay at the beginning (includes human detection and background tracking). Each frame contains an active foreground object takes around one second $(0.38+0.07\times 2+0.44)$ to process. The post-process step in Section 3.4 takes 0.45 second for each frame. We summarize the system parameters in the supplementary. All the experiments are performed on a desktop machine with Intel Core i7-6700K 4.00 GHz CPU, 32GB memory, and GTX 1080Ti GPU.

5.1. Evaluation on Our Synthetic Benchmark

In Table 1 and Figure 5, we conduct a quantitative and qualitative evaluation on our synthetic dataset along the two main axes: reconstruction and tracking, as introduced in Section. 4. The reconstruction F1 score shows the balance between reconstructing surfaces and removing noises, and the MOTA metric shows the accuracy and the completeness of tracking results. The missing detection metric (MISS) is only employed for the foreground because a system must first detect a target then perform tracking.

Background Tacking and Reconstruction In Table 1, we evaluate different methods's background tracking and reconstruction performance. VoxelHashing [NZIS13] and BundleFusion [DNZ*17] are the off-the-shelf methods designed for static scenes, which can be seen as a performance reference. Their reconstruction and tracking performance are low due to the existence of dynamic elements.

ReFusion [PBL*19] is an example of robust background reconstruction methods and reaches a F1 of 0.63 on background reconstruction and a MOTA of 66% on camera tracking. ReFusion averages out foreground voxels in the TSDF model by additionally integrating free-space voxels and remove high residuals pixels using a flood-filling algorithm. However, its performance is still affecting by pixels from the foreground.

CoFusion [RA17] and MaskFusion [RBA18] are state-of-the-art methods based on ElasticFusion [WSG*16]. Interestingly, their MOTA for background tracking is 16-19% lower than ReFusion. We found this is due to two reasons: The false positives of the moving object detection remove many geometry signals for camera tracking. And the weighting mechanism adapted from ElasticFusion, which prefers not using newly observed points.

RigidFusion achieves the best F1 (0.74) and MOTA (68%) on the background tasks without performing the post-processing step (w/o PS, Section 3.4). This is benefited from our aggressive outlier purging strategy (Section 3.2, free-space aware fusion). With the post-processing step, our system's performance can be further enhanced by 0.12 on background reconstruction F1 and 2% on background tracking MOTA.

Foreground Detection, Tracking, and Reconstruction In Table 1, we evaluate RigidFusion's foreground reconstruction, detection, and tracking accuracy against other methods. To reconstruct foreground objects in a RGB-D sequence, a system needs to firstly detect the moving objects. This step is non-trivial because too many false positive detections not only increase the computation costs but also remove the geometry signals for background tracking.

Both CoFusion [RA17] and MaskFusion [RBA18] detect a new foreground object by examing the estimated camera motion residuals. This approach's major disadvantage is being sensitive to the actual camera motion (moving direction and magnitude) and the selection of the detection threshold. This causes a high missing detection ratio (MISS) and makes these two methods require a static or slow camera setup. This observation is also reported in the previous work [SJP*18].

MaskFusion utilizes a very good segmentation prior (73 mAP) and hence achieves a higher recall and F1 on foreground reconstruction. However, the semantic segmentation does not directly solve the motion segmentation. When the motion segmentation is given, such as TUM f3w sequences in Table 4 (major moving objects are humans), MaskFusion has better background tracking than CoFusion and achieves competitive performance as ReFusion [PBL*19]. Inversely, when motion segmentation is non-trivial, i.e. input masks may contain static objects, such as our dataset (Table 1 and Figure 6) and CoFusion dataset (Table 5), MaskFusion does not have performance gains on the tracking.

Notably, RigidFusion's foreground miss detection is significantly lower than the comparisons (15% and 26% lower than CoFusion and MaskFusion). This is benefited by employing the segment-by-reconstruction strategy, which is insensitive to the relative motion between the moving object and the camera. Hence, RigidFusion can detect foreground quickly when both foreground and background have non-static movements. With the post-process step (Section 3.4), the reconstruction F1 and MOTA can be enhanced by 0.05 and 4% respectively. In Figure 5, we shows a qualitative comparison.

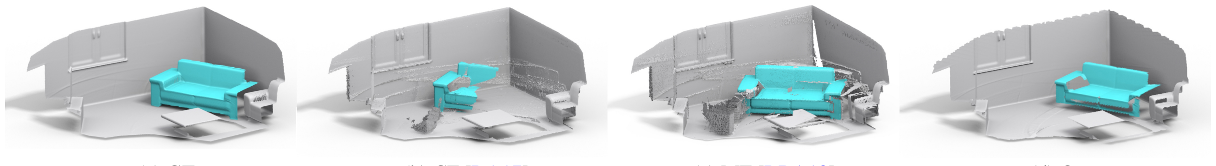

Figure 5: *Result comparison on RigidFusion dataset. Note that the moving object is positioned at the first detected frame. In contrast to ours, both CoFusion and MaskFusion result in ghosting in the background due to delayed moving object detection.*

Table 1: *4D reconstruction quality on our benchmark comparing against state-of-the-art methods. Overall, our method achieves a significantly lower missing detection ratio and better recall/outlier balance than the other approaches. Symbol '-' denotes that the metric is non-applicable. Please refer to supplemental material for the examples of the employed error metrics and 4D reconstruction videos.*

	Object Priors	Background							Foreground							
		Reconstruction				Tracking			Reconstruction				Tracking			
		F1	Recall	Prec.	CD (m)	**MOTA (%)**	BAD (%)	MOTP (m)	**F1**	Recall	Prec.	CD (m)	**MOTA (%)**	MISS (%)	BAD (%)	MOTP (m)
VH	no	0.58	0.66	0.52	0.15	49	51	0.024	-	-	-	-	-	-	-	-
BF	no	0.44	0.61	0.37	0.18	38	60	**0.017**	-	-	-	-	-	-	-	-
RF	no	0.64	0.67	0.62	0.09	66	34	0.023	-	-	-	-	-	-	-	-
CF	no	0.73	0.72	0.74	0.07	50	50	0.030	0.38	0.27	0.70	0.26	32	39	29	0.030
MF	yes	0.67	0.70	0.64	0.08	47	53	0.030	0.53	**0.47**	0.70	0.29	33	50	**17**	0.031
Ours w/o PS	no	0.74	0.69	0.79	0.08	68	32	0.025	0.51	0.41	0.74	0.31	55	24	21	0.026
Ours	no	**0.86**	**0.83**	**0.90**	**0.04**	**70**	**30**	0.025	**0.56**	0.44	**0.88**	**0.13**	**59**	**13**	28	**0.025**

5.2. Evaluation on Real-world Data

Scenes with different levels of camera motion We recorded four dynamic scenes with different settings, including a small desktop scene with a near static camera and medium scale scenes with non-static camera motions, using a Structure Sensor developed by Occipital Inc. mounted to an iPad AIR2. The scene settings are summarized in Table 2. We rank the ambiguity of the camera motion from low to high using the estimated camera trajectories and the annotated foreground masks on the frame that a dynamic object appears or starts to move. In addition, to achieve a fair comparison, we passed to other methods input frames with the human regions masked out.

In Figure 6, we compare RigidFusion against CoFusion [RA17] and MaskFusion [RBA18]. In Table 3, we show the frame index of the foreground being detected by each method and the corresponding delay frame number. The results of Scene 1 show when the camera motion is very small, both CoFusion and MaskFusion can detect foreground objects within a short delay. When a camera has non-static motion, as in Scene 2 to Scene 4, both CoFusion and MaskFusion suffer from long delayed detection and cause severe errors in tracking and reconstruction. This observation coincides with the evaluation results in Table 1, where both CoFusion and MaskFusion suffer from high MISS ratios. RigidFusion produces significantly better reconstruction results in all four examples, output visually completed object trajectories, and maintain low detection latency as shown in 6.

Camera Tracking Accuracy In Table 4, we carry out an evaluation on camera tracking in dynamic scenes using the TUM RGB-D dataset, *freibur3*, and report ATE-RMSE. Note that this dataset contains many far-range pixels. RigidFusion 's camera tracking is comparable to alternate methods, especially in high-dynamic *walking* examples, although not the best.

What if the camera tracking fails? In this case, our method and all comparison [PBL*19; RA17; RBA18; SS19; XLT*19a] will fail to produce a reasonable reconstruction result since camera tracking (or reconstruction) is used for identifying foreground objects.

Does the proposed method mainly benefit from having the foreground movement assumption? While the foreground movement assumption (i.e., objects do not move simultaneously) helps to simplify the instance segmentation step, the evaluation results in Section 5.1 and Section 5.2 show the major limitation of previous work is the moving object detection step in scenes with moving cameras, i.e., when to initialize the first foreground tracking and which pixels to group. The multiple object association-and-tracking problem comes after a system having more than one tracked foreground objects. From Table 3 and Figure 6, we can see a long delayed detection such as CoFusion's result on Scene3 produces a noisy and rather partial reconstruction.

5.3. Evaluation on CoFusion Dataset

In Table 5, we evaluate RigidFusion on the CoFusion's synthetic data [RA17] and report ATE-RMSE. For Airship object in Room4 sequence, both MaskFusion [RBA18] and EMFusion [SS19] fails to associating segments across frames and output a fragmented trajctories. Compared to the other methods, RigidFusion consistently produces better camera tracking and improves foreground tracking in Room4 sequence without using object priors.

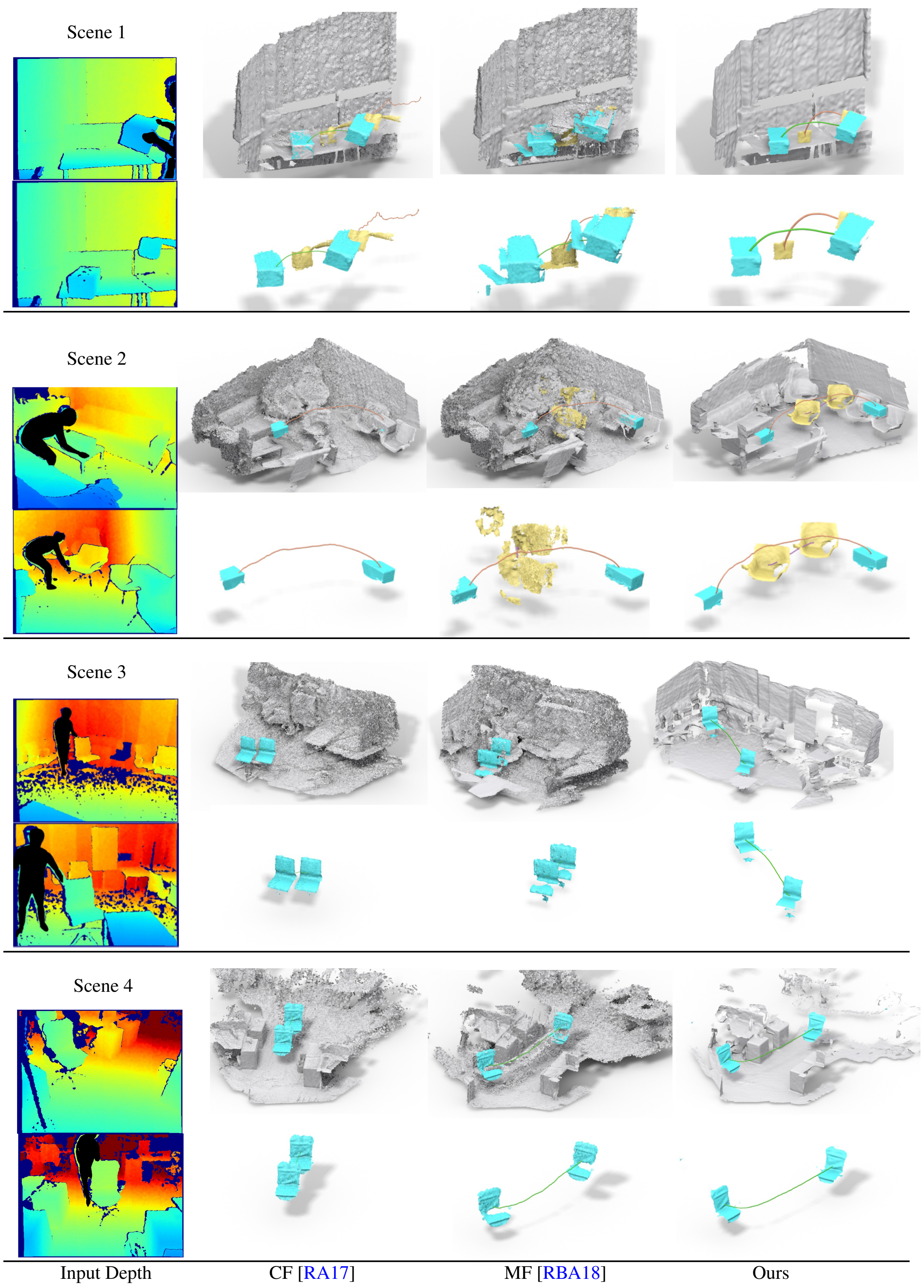

Figure 6: *Qualitative evaluation on real-world data. In each scene, the first row: full 4D reconstructions; the second row: foreground trajectories and reconstruction results. The scene settings are listed in Table 2. The scenes are ordered based on camera motion and the scene size. Our method can work on both low-dynamic and high-dynamic setting and output plausible results.*

Table 2: *The statistics of our real-world examples.*

	Scene1 (two objects)	Scene2 (two objects)	Scene3 (one object)	Scene4 (one object)
Scene size	2.4x0.6 m^2	4.5x3.0 m^2	4.3x2.2 m^2	5.9x4.5 m^2
Camera motion	2.7 cm/s	25.3 cm/s	10.5 cm/s	39.2 cm/s
BG's median motion residuals	4.32E-04	9.65E-05	1.22E-04	4.85E-04
FG's median motion residuals	9.70E-01	1.01E-04	1.43E-04	7.87E-04
Motion ambiguity	Low	Medium	High	High
First dynamic frame / total frame #	11 / 350	134 / 635	26 / 270	1 / 209

Table 3: *The evaluation of moving object detection on our real-world examples. We show the number of delayed detection frames (Delay #) using the output trajectories. Clearly, RigidFusion has the lowest detection latency. Symbol 'n/a' represents the object is not detected and '-' indicates there are no object 2.*

		Scene1 (two objects)	Scene2 (two objects)	Scene3 (one object)	Scene4 (one object)
FG ID	Detection Methods	Delay #	Delay #	Delay #	Delay #
Object1	CF	11	27	145	144
	MF	11	12	120	21
	Ours w/o PS	**3**	10	5	**0**
	Ours	**3**	**0**	**0**	**0**
Object2	CF	3	n/a	-	-
	MF	7	**0**	-	-
	Ours w/o PS	4	6	-	-
	Ours	**3**	**0**	-	-

Tracking methods versus moving object detection methods Table 5 shows that the accuracy of the moving object detection step is more important than the underlay tracking methods. Note that both CoFusion and MaskFusion use surfel tracking [WSG*16], while EMFusion uses volumetric based tracking but employs a similar moving object detection step as MaskFusion using semantic priors. EMFusion utilizes its volumetric model to improve semantic segmentation. Therefore its object tracking is better than MaskFusion. However, EMFusion and MaskFusion still have similar error patterns in camera tracking, and both have higher ATE-RMSE than CoFusion. In contrast, RigidFusion employs segment-by-reconstruction strategy and volumetric based tracking and achieves the best in class camera tracking performance.

What if our foreground movement assumption is not hold? RigidFusion assume foreground objects are not moves simultaneously. If this assumption is not hold, our method will only track the dominated foreground object. One example is the ToyCar3 sequence in Table 5. where the two cars moves simultaneously and Car1 is the dominated foreground object.

6. Conclusion

We investigated the problem of 4D reconstruction of scenes with moving objects as recorded from a moving camera and proposed RigidFusion to simultaneously solve for objects' motion and segmentation. We conducted both qualitative and quantitative evaluations and reported systematic improvement in terms of tracking and reconstruction errors. Our approach, being non-learning based, is not restricted to objects with semantic labels, and hence can be used to collect training data for typical object movements under real-world interactions.

Table 4: *Background tracking on TUM RGB-D dataset. f3s and f3w represent sitting and walking cases in freiburg3 respectively.*

	AT-RMSE (in cm)								
Methods	Human Priors	Object Priors	f3s static	f3s xyz	f3s halfsphere	f3w static	f3w xyz	f3w halfsphere
SF	no	no	1.3	4.0	4.0	**1.4**	12.7	39.1
RF	no	no	**0.9**	4.0	11.0	1.7	9.9	10.4
CF	no	no	1.1	**2.7**	3.7	55.1	69.6	80.3
MF	yes	yes	2.1	3.1	5.2	3.5	10.4	10.6
MID	yes	yes	1.0	6.2	**3.1**	2.3	6.8	**3.8**
EM	yes	yes	**0.9**	3.7	3.2	**1.4**	**6.6**	5.1
Our	yes	no	1.9	5.4	12.9	1.8	9.0	7.6

Table 5: *Comparison of foreground tracking on CoFusion's synthetic dataset. Symbol '†' represents the corresponding moving object is not detected. MaskFusion does not detect Horse due to the delay detection and camera tracking drift, as reported in [SS19]. Our method does not detect Car2 and Horse due to they simultaneously move with other objects, but our approach achieves the best tracking accuracy on Airship, Car and Camera.*

	AT-RMSE (in cm)							
		ToyCar3			Room4			
	Object Priors	Camera	Car1	Car2	Camera	Car	Airship	Horse
CF	no	0.61	7.78	1.44	0.93	0.29	0.96	5.8
MF	yes	20.6	1.53	0.58	1.41	2.66	6.46 (13.6 / 2.3 / 3.5)	†
EM	yes	0.95	**0.77**	**0.18**	1.37	2.1	0.91 (0.6 / 1.4 / 0.8)	3.57
Ours	no	**0.46**	1.9	†	**0.58**	**1.0**	**0.55**	†

Limitations Our approach has two main limitations: (i) *only allows one active object at a time*: we believe this can be addressed by incorporating class-agnostic priors across multiple frames; (ii) *large run time*: as mentioned in Section 5, the major bottlenecks are the human detector [GNK18], which we believe can be improved by adapting a real-time human detector, and the segment-by-reconstruction step, which can be sped up by using parallel processing on GPU as the volumetric fusion step.

Future work We would like to investigate supplementing our active object detection step with motion segmentation priors. Such an approach, however, requires bridging the synthetic real gap in terms of generalization as we do not have supervision datasets with ground truth motion for dynamic scenes, like those we consider in our paper.

Acknowledgement This work was supported by the ERC Starting Grant Scan2CAD (804724), ERC Smart Geometry Grant, and gifts from Adobe.

References

[ASZS17] Armeni, I., Sax, A., Zamir, A. R., and Savarese, S. "Joint 2D-3D-Semantic Data for Indoor Scene Understanding". *ArXiv e-prints* (Feb. 2017). arXiv: 1702.01105 [cs.CV] 3.

[BIZ18] Bertholet, P., Ichim, A.E., and Zwicker, M. "Temporally Consistent Motion Segmentation From RGB-D Video". *Computer Graphics Forum* 37.6 (2018), 118–134. doi: 10.1111/cgf.13316 2, 3.

[BS08] Bernardin, Keni and Stiefelhagen, Rainer. "Evaluating multiple object tracking performance: The CLEAR MOT metrics". *EURASIP Journal on Image and Video Processing* 2008 (Jan. 2008) 6.

[BSK*13] Bylow, Erik, Sturm, Jürgen, Kerl, Christian, et al. "Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions". June 2013. doi: 10.15607/RSS.2013.IX.035 5.

[CBI13] Chen, Jiawen, Bautembach, Dennis, and Izadi, Shahram. "Scalable real-time volumetric surface reconstruction". *ACM Transactions on Graphics (ToG)* 32.4 (2013), 113 1, 3.

[CL96] Curless, Brian and Levoy, Marc. "A volumetric method for building complex models from range images". *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM. 1996, 303–312 1, 3–6.

[DCS*17] Dai, Angela, Chang, Angel X., Savva, Manolis, et al. "ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes". *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*. 2017 3.

[DKD*16] Dou, Mingsong, Khamis, Sameh, Degtyarev, Yury, et al. "Fusion4d: Real-time performance capture of challenging scenes". *ACM Transactions on Graphics (TOG)* 35.4 (2016), 114 2, 3.

[DNZ*17] Dai, Angela, Niessner, Matthias, Zollöfer, Michael, et al. "BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Re-integration". *ACM Transactions on Graphics 2017 (TOG)* (2017) 1, 7.

[GIRL03] Gelfand, Natasha, Ikemoto, Leslie, Rusinkiewicz, Szymon, and Levoy, Marc. "Geometrically Stable Sampling for the ICP Algorithm". Vol. 2003. Nov. 2003, 260–267. isbn: 0-7695-1991-1. doi: 10.1109/IM.2003.1240258 2.

[GKM*17] Golyanik, V., Kim, K., Maier, R., et al. "Multiframe Scene Flow with Piecewise Rigid Motion". *International Conference on 3D Vision (3DV)*. Oct. 2017 2.

[GNK18] Güler, Riza Alp, Neverova, Natalia, and Kokkinos, Iasonas. "Densepose: Dense human pose estimation in the wild". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, 7297–7306 4, 10.

[HGDG17] He, Kaiming, Gkioxari, Georgia, Dollár, Piotr, and Girshick, Ross. "Mask r-cnn". *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE. 2017, 2980–2988 2, 3, 6.

[HYZ*19] Huang, Jiahui, Yang, Shilin, Zhao, Zishuo, et al. "ClusterSLAM: A SLAM Backend for Simultaneous Rigid Body Clustering and Motion Estimation". *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, 5874–5883 3.

[IZN*16] Innmann, Matthias, Zollhöfer, Michael, Niessner, Matthias, et al. "VolumeDeform: Real-time volumetric non-rigid reconstruction". *European Conference on Computer Vision*. Springer. 2016, 362–379 2, 3.

[JGN18] Judd, Kevin, Gammell, Jonathan, and Newman, Paul. "Multimotion Visual Odometry (MVO): Simultaneous Estimation of Camera and Third-Party Motions". Oct. 2018, 3949–3956. doi: 10.1109/IROS.2018.8594213 3.

[KDSX15] Klingensmith, Matthew, Dryanovski, Ivan, Srinivasa, Siddhartha, and Xiao, Jizhong. "Chisel: Real Time Large Scale 3D Reconstruction Onboard a Mobile Device using Spatially Hashed Signed Distance Fields." *Robotics: science and systems*. Vol. 4. 2015 1, 3.

[KMG*17] Kolve, E., Mottaghi, R., Gordon, D., et al. "AI2-THOR: An Interactive 3D Environment for Visual AI". *ArXiv e-prints* (Dec. 2017). arXiv: 1712.05474 [cs.CV] 3.

[KPR*15] Kähler, Olaf, Prisacariu, Victor Adrian, Ren, Carl Yuheng, et al. "Very high frame rate volumetric integration of depth images on mobile devices". *IEEE transactions on visualization and computer graphics* 21.11 (2015), 1241–1250 1, 3.

[LMB*14] Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, et al. "Microsoft COCO: Common Objects in Context". *Computer Vision – ECCV 2014*. Ed. by Fleet, David, Pajdla, Tomas, Schiele, Bernt, and Tuytelaars, Tinne. Cham: Springer International Publishing, 2014, 740–755. isbn: 978-3-319-10602-1 7.

[MCB*18] McCormac, John, Clark, Ronald, Bloesch, Michael, et al. "Fusion++: Volumetric Object-Level SLAM". *2018 International Conference on 3D Vision (3DV)*. IEEE. 2018, 32–41 2.

[MMT15] Mur-Artal, Raul, Montiel, J., and Tardos, Juan. "ORB-SLAM: a versatile and accurate monocular SLAM system". *IEEE Transactions on Robotics* 31 (Oct. 2015), 1147–1163. doi: 10.1109/TRO.2015.2463671 3.

[MS14] Ma, Lu and Sibley, Gabe. "Unsupervised dense object discovery, detection, tracking and reconstruction". *European Conference on Computer Vision*. Springer. 2014, 80–95 2, 3.

[MSL94] Murray, Richard M., Sastry, S. Shankar, and Li, Zexiang. "A Mathematical Introduction to Robotic Manipulation". *CRC Press* (1994) 5.

[MWH*19] Ma, Wei-Chiu, Wang, Shenlong, Hu, Rui, et al. "Deep Rigid Instance Scene Flow". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 3.

[NFS15] Newcombe, Richard A, Fox, Dieter, and Seitz, Steven M. "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, 343–352 2, 3.

[NIH*11] Newcombe, Richard A, Izadi, Shahram, Hilliges, Otmar, et al. "KinectFusion: Real-time dense surface mapping and tracking". *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*. IEEE. 2011, 127–136 1, 3, 5.

[NZIS13] Niessner, M., Zollhöfer, M., Izadi, S., and Stamminger, M. "Real-time 3D Reconstruction at Scale using Voxel Hashing". *ACM Transactions on Graphics (TOG)* (2013) 1, 3, 4, 7.

[PBL*19] Palazzolo, Emanuele, Behley, Jens, Lottes, Philipp, et al. "ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals". *arXiv* (2019). url: https://arxiv.org/abs/1905.02082 2–5, 7, 8.

[PRB*18] Puig, Xavier, Ra, Kevin, Boben, Marko, et al. "VirtualHome: Simulating Household Activities via Programs". *Computer Vision and Pattern Recognition (CVPR)*. 2018 3.

[RA17] Rünz, Martin and Agapito, Lourdes. "Co-Fusion: Real-time Segmentation, Tracking and Fusion of Multiple Objects". *2017 IEEE International Conference on Robotics and Automation (ICRA)*. May 2017, 4471–4478 2, 3, 5, 7–9.

[RBA18] Rünz, Martin, Buffier, Maud, and Agapito, Lourdes. "MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects". *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. Oct. 2018, 10–20 2, 3, 6–9.

[RHL02] Rusinkiewicz, Szymon, Hall-Holt, Olaf, and Levoy, Marc. "Real-time 3D model acquisition". *ACM Transactions on Graphics (TOG)* 21.3 (2002), 438–446 1, 3.

[RPK*17] Ren, C. Y., Prisacariu, V. A., Kähler, O., et al. "Real-Time Tracking of Single and Multiple Objects from Depth-Colour Imagery Using 3D Signed Distance Functions". *Int. J. Comput. Vision* 124.1 (Aug. 2017), 80–95. issn: 0920-5691. doi: 10.1007/s11263-016-0978-2. url: https://doi.org/10.1007/s11263-016-0978-2 3.

[RPMR13] Ren, Carl, Prisacariu, Victor, Murray, David, and Reid, Ian. "STAR3D: Simultaneous tracking and reconstruction of 3D objects using RGB-D data". Dec. 2013, 1561–1568. doi: 10.1109/ICCV.2013.197 3.

[SBCI17] Slavcheva, Miroslava, Baust, Maximilian, Cremers, Daniel, and Ilic, Slobodan. "KillingFusion: Non-Rigid 3D Reconstruction Without Correspondences". *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017 2, 3.

[SC18] Sommer, C. and Cremers, D. "Joint Representation of Primitive and Non-primitive Objects for 3D Vision". *2018 International Conference on 3D Vision (3DV)*. 2018, 160–169 5.

[SJP*18] Scona, Raluca, Jaimez, Mariano, Petillot, Yvan, et al. "StaticFusion: Background Reconstruction for Dense RGB-D SLAM in Dynamic Environments". May 2018, 1–9. doi: 10.1109/ICRA.2018.8460681 3–5, 7.

[ŞMK12] Şucan, Ioan A., Moll, Mark, and Kavraki, Lydia E. "The Open Motion Planning Library". *IEEE Robotics & Automation Magazine* 19.4 (Dec. 2012). http://ompl.kavrakilab.org, 72–82. doi: 10.1109/MRA.2012.2205651 2, 6.

[SS19] Strecke, Michael and Stückler, Jörg. "EM-Fusion: Dynamic Object-Level SLAM With Probabilistic Data Association". *International Conference on Computer Vision*. Oct. 2019 2, 3, 6–8, 10.

[WKM*19] Wu, Yuxin, Kirillov, Alexander, Massa, Francisco, et al. *Detectron2*. https://github.com/facebookresearch/detectron2. 2019 4.

[WSG*16] Whelan, Thomas, Salas-Moreno, Renato F, Glocker, Ben, et al. "ElasticFusion: Real-time dense SLAM and light source estimation". *The International Journal of Robotics Research* 35.14 (2016), 1697–1716 7, 10.

[XLT*19a] Xu, Binbin, Li, Wenbin, Tzoumanikas, Dimos, et al. "MID-Fusion: Octree-based Object-Level Multi-Instance Dynamic SLAM". May 2019, 5231–5237. doi: 10.1109/ICRA.2019.8794371 2, 6–8.

[XLT*19b] Xu, Binbin, Li, Wenbin, Tzoumanikas, Dimos, et al. "MID-Fusion: Octree-based Object-Level Multi-Instance Dynamic SLAM". May 2019, 5231–5237. doi: 10.1109/ICRA.2019.8794371 2, 3.

[XRH*18] Xia, Fei, R. Zamir, Amir, He, Zhi-Yang, et al. "Gibson env: real-world perception for embodied agents". *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE. 2018 3.

[Zha13] Zhaoyang, Lv. "Kinfuseg: a dynamic slam approach based on kinect fusion". *Department of Computing, Imperial College London, London, England, Msc Thesis,* 2013 3.