

RGB-D SLAM in Indoor Planar Environments with Multiple Large Dynamic Objects

Ran Long^{1,*}, Christian Rauch¹, Tianwei Zhang², Vladimir Ivan¹, Tin Lun Lam^{2,4}, Sethu Vijayakumar^{1,3,*}

Abstract—This work presents a novel dense RGB-D SLAM approach for dynamic planar environments that enables simultaneous multi-object tracking, camera localisation and background reconstruction. Previous dynamic SLAM methods either rely on semantic segmentation to directly detect dynamic objects; or assume that dynamic objects occupy a smaller proportion of the camera view than the static background and can, therefore, be removed as outliers. Our approach, however, enables dense SLAM when the camera view is largely occluded by multiple dynamic objects with the aid of camera motion prior. The dynamic planar objects are separated by their different rigid motions and tracked independently. The remaining dynamic non-planar areas are removed as outliers and not mapped into the background. The evaluation demonstrates that our approach outperforms the state-of-the-art methods in terms of localisation, mapping, dynamic segmentation and object tracking. We also demonstrate its robustness to large drift in the camera motion prior.

I. INTRODUCTION

Simultaneous localisation and mapping (SLAM) has a wide application in autonomous robots or virtual reality. In indoor environments, planes are common man-made features. Planar SLAM methods have used the characteristics of planes to reduce long-term drift and improve accuracy of localisation [1], [2]. However, these methods assume that the environment is static – an assumption that is violated when the robot works in conjunction with other humans or robots, or manipulates objects in semi-automated warehouses.

The core problem of enabling SLAM in dynamic environments while differentiating multiple dynamic objects involves several challenges:

- 1) There are usually an unknown number of third-party motions in addition to the camera motion in dynamic environments. The number of motions or dynamic objects are also changing.
- 2) Static background is often assumed to account for the major proportion of the camera view. However, without semantic segmentation, dynamic objects that occupy a large proportion of the camera view can end up being classified as the static background.

¹Ran Long, Christian Rauch, Vladimir Ivan and Sethu Vijayakumar are with the Institute of Perception, Action and Behaviour, School of Informatics, University of Edinburgh, Edinburgh, EH8 9AB, U.K.

²Tianwei Zhang and Tin Lun Lam is with the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS).

³Sethu Vijayakumar is a visiting researcher at the Shenzhen Institute for Artificial Intelligence and Robotics for Society (AIRS).

⁴Tin Lun Lam is with School of Science and Engineering, the Chinese University of Hong Kong, Shenzhen.

*Corresponding authors: Ran Long (Ran.Long@ed.ac.uk), Sethu Vijayakumar (Sethu.Vijayakumar@ed.ac.uk).



Fig. 1: The top row shows the RGB images from the sequence which is collected from a moving RGB-D camera in an indoor dynamic environment. The dynamic component contains multiple dynamic objects, including rigid boxes and a human. We then segment images into a combination of planes and non-planar areas, where the planes are extracted from the depth map and the non-planar areas are represented by a set of super-pixels. We detect static planes and super-pixels to map the static background. The remaining dynamic components are segmented into multiple planar objects with different motions and non-planar outliers.

- 3) The majority of the colour and depth information can be occluded by dynamic objects and the remaining static parts of the visual input may not be enough to support accurate camera ego-motion estimation.

Many dynamic SLAM methods have considered multiple dynamic objects [3], [4], [5], but either rely on semantic segmentation or assume that the static background is the largest rigid body in the camera view. To concurrently solve these problems, we propose a hierarchical representation of images that extracts plane from planar areas and over-segments non-planar areas into super-pixels (Figure 1). We simultaneously merge planes with similar motions to planar rigid bodies, identify dynamic objects and track the camera motion. For this, we assume that planes occupy a major fraction of the environment, including the static background and rigid dynamic objects. In addition, the camera motion can be distinguished from other third-party motions by a tightly-coupled camera motion prior from robot odometry.

In summary, this work contributes:

- 1) a new methodology for online multimotion segmentation based on planes in indoor dynamic environments,
- 2) a novel pipeline that simultaneously tracks multiple planar rigid objects, estimates camera ego-motion and reconstructs the static background,
- 3) a RGB-D SLAM method that is robust to large-occluded camera view caused by multiple large dynamic objects.

II. RELATED WORK

A. Dynamic SLAM

Dynamic SLAM methods, which enable robots to robustly estimate ego-motion in dynamic environments, can be categorised into outlier-based, semantic-based, multimotion and proprioception-aided methods.

Outlier-based methods assume that the static background is the major component in the camera view and dynamic objects can be robustly removed during camera tracking. Joint-VO-SF (JF) [6] over-segments images into clusters and classifies each cluster as either static or dynamic by comparing the depth and intensity residuals of each cluster against the average residual over the whole image. Static-Fusion (SF) [7] introduces a continuous score to represent the probability that a cluster is static. The scores are used to segment the static background for camera localisation and mapping, while the remaining dynamic parts are discarded as outliers. Co-Fusion (CF) [5] further models the outliers as new objects and tracks them independently using frame-to-model alignment.

Semantic-based methods can directly detect dynamic objects from the semantic segmentation. Based on Mask R-CNN [8], which provides pixel-wise object segmentation, MaskFusion [9] can remove dynamic objects before camera tracking and refine object boundaries using geometric edges. The semantic segmentation can also be combined with intensity and geometric information [10]. ClusterVO [3] can track camera ego-motion and multiple rigidly moving clusters simultaneously by integrating semantic bounding boxes and ORB features [11]. DynaSLAM II [12] further integrates the multi-object tracking (MOT) and SLAM into a tightly-coupled formulation to improve its performance on both problems. However, these methods require that the object detector is pre-trained on a dataset that includes these objects or provide the object model in advance.

Multimotion methods explicitly model the dynamic component as third-party rigid motions and segment dynamic objects by their different motions against the camera motion. Multimotion visual odometry (MVO) [4] is an online multimotion segmentation and tracking method based on sparse feature points. It iteratively samples all sparse keypoints using RANSAC to generate motion hypotheses and automatically merge them when the merging can decrease the total energy. In addition to RANSAC, motion hypotheses can be generated from the increased N-points using a grid-based scene flow [13]. Instead of merging motion hypotheses, DymSLAM [14] computes a residual matrix from hypothetical motion models and directly estimates the number of dynamic objects from the residual matrix. The dense segmentation of multiple dynamic objects based on super-pixels [15] is acquired after the number of objects is specified. However, all these methods assume that the largest body of motion in the camera view is the static background and they have not demonstrated their effectiveness when the major proportion of the camera view is occluded by dynamic objects.

Robot proprioception, such as IMU and wheel odometry, can be fused with visual sensors to improve the accuracy and robustness of localisation in dynamic environments [16], [17]. Kim et al. [16] uses the camera motion prior from an IMU to compensate the camera motion and select static keypoints based on motion vectors. RigidFusion (RF) [17] uses both camera and object motion priors to simultaneously reconstruct the static background and one rigid dynamic object where the major part of the camera view is occluded. However, both of these methods are unable to track multiple dynamic objects independently.

B. Planar SLAM

Planar features have been widely used in indoor dynamic SLAM methods. Infinite planes can be used as landmarks in the pose graph SLAM problem [18]. Based on keyframe management, a global dense planar map can be reconstructed using only a single CPU [19]. Planes can also be combined with keypoints and lines [2], [1] for more robust camera tracking. In structured environments, planes have been demonstrated to significantly reduce accumulated drift by minimising the rotational error between each frame and the underlying Manhattan World (MW) frame [1]. All these methods assume static environments, because planes in the indoor environment, like walls, are often static. However, this assumption is violated when planar objects like boxes are transported or manipulated by humans or robots in places like warehouses.

III. METHODOLOGY

A. Overview and Notation

The overview of our pipeline is illustrated in Figure 2. Our method takes RGB-D image pairs from two consecutive frames. An RGB-D pair at the t -th frame includes a colour image and a depth image D_t . An intensity image I_t is computed from the colour image.

For each image pair (I_t, D_t) , we extract planes directly from depth maps [20]. A plane is represented in the Hessian form $\Pi = (\mathbf{n}^T, d)^T$, where $\mathbf{n} = (n_x, n_y, n_z)$ is the normal of the plane and d is the perpendicular distance between the plane and camera origin. For each image frame, we assume there are P planes: $\mathbf{V}^P : \{V_i^P | i \in [1, P]\}$, where V_i^P represents the set of pixels that belong to the i -th plane. After plane extraction, the remaining non-planar areas are over-segmented into S super-pixels: $\mathbf{V}^{nP} : \{V_i^{nP} | i \in [1, S]\}$.

For the i -th plane, we extract a set of keypoints \mathbf{K}_i using ORB features [21]. We then conduct multimotion segmentation on planar areas \mathbf{V}^P and cluster the planes into M planar rigid bodies of different motions. For simplicity, we name the camera motion relative to objects' egocentric frames as *object egocentric motion* [4] and denote them as ${}^{ego}\mathbf{T} = \{\tilde{\xi}_m \in \mathfrak{se}(3) | m \in [1, M]\}$. Since the static background may not be the largest rigid body in the camera view, we use the camera motion prior $\tilde{\xi}_c$ with potential drift to simultaneously classify all planes and super-pixels into either static or dynamic, and estimate the camera ego-motion.

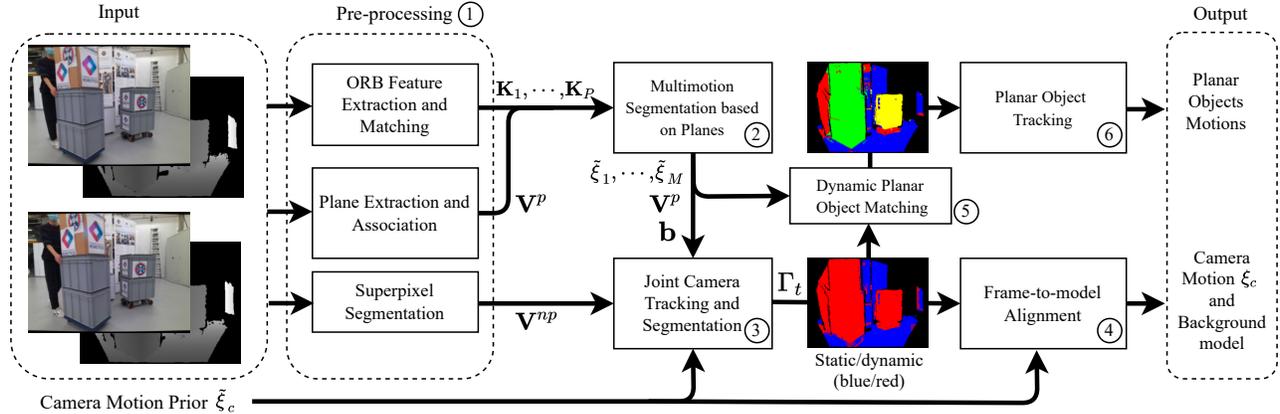


Fig. 2: The pipeline of our proposed method. (1) We first represent the input image in the current frame t as a combination of planes and super-pixels. The ORB features [11] are extracted and matched to the previous frame. (2) Planes with similar rigid motions are clustered into M planar rigid bodies and their corresponding egocentric motions are estimated respectively. However, we are uncertain which planar rigid body belongs to the static background. (3) We, therefore, jointly separate the static background from the planes and super-pixels, and estimate the camera motion via frame-to-frame alignment. (4) The static part is used to reconstruct the background and refine the camera motion. (5,6) Dynamic non-planar super-pixels are removed as outliers, while dynamic planar rigid bodies are matched with planar rigid bodies in the previous frame. The matched planar rigid body is tracked using RANSAC on their ORB features and plane parameters.

We use the score $\gamma_i \in [0, 1]$ to represent the probability that a plane or super-pixel is static. The scores $\boldsymbol{\gamma} : \{\gamma_i | i \in [1, P+S]\}$ are assigned to each plane and super-pixel, where $\{\gamma_i | i \in [1, P]\}$ and $\{\gamma_i | i \in [P+1, P+S]\}$ represent the scores of planes and super-pixels respectively. At time t , the pixel-wise static dynamic segmentation $\Gamma_t \in \mathbb{R}^{w \times h}$ can be estimated from $\boldsymbol{\gamma}$. The static parts of intensity and depth images are used to estimate the camera motion $\xi_c \in \mathfrak{se}(3)$. The dynamic planar rigid bodies are used to track dynamic objects. The non-planar dynamic super-pixels, such as humans, are removed as outliers.

The world-, camera-, and the m -th object-frames are W , C and O^m respectively. The camera motion $T(\xi_c) := \exp(\xi_c)$ is $T_{C_{t-1}C_t} = T_{WC_{t-1}}^{-1} T_{WC_t}$, which transforms homogeneous coordinates of a point in the current camera frame C_t to the previous frame C_{t-1} . The function $\exp(\xi)$ is the matrix exponential map for Lie group $SE(3)$. The m -th object egocentric transformations is the camera motion relative to this object [4]:

$$T(\tilde{\xi}_m) = {}^m T_{C_{t-1}C_t}^{-1} = T_{O_{t-1}^m C_{t-1}}^{-1} T_{O_t^m C_t}. \quad (1)$$

B. Multimotion segmentation based on planes

To extract planes, we transform the depth map to a point cloud and cluster connected groups of points with close normal directions using the method from [20]. To match an extracted plane i in the current frame with one in the previous frame, we first estimate the angle and point-to-plane distance between plane i and all planes in the previous frame. A plane is chosen as a candidate if the angle and distance are below 10 degrees and 0.1 m respectively, which is the same as [2], [1]. Since there can be multiple candidates, rather than choosing the plane with minimal distance [2], [1], we further consider overlap proportion between two planes using the Jaccard index, $J(V_1, V_2) = \frac{|V_1 \cap V_2|}{|V_1 \cup V_2|}$, where $|V_1|$ is the number of pixels in planar segment 1. We choose the candidate plane

that has the maximal Jaccard index as the matched plane for plane i and denote it as plane i' .

To estimate object egocentric motion $T_i = {}^i T_{C_{t-1}C_t}^{-1}$ of plane i , we extract and match ORB keypoints from plane i and i' . The error function is defined as:

$$e_i(T_i) = \sum_{k \in \mathcal{X}_{ii'}} \rho(\|\mathbf{x}_i^k - T_i \mathbf{x}_{i'}^k\|_{\Sigma}) + \lambda_h \|q(\Pi_i) - q(T_i^{-T} \Pi_{i'})\|_2^2, \quad (2)$$

where $\mathcal{X}_{ii'}$ is the set of keypoint matches between planes i and i' . \mathbf{x}_i^k and $\mathbf{x}_{i'}^k$ are homogeneous coordinates $[x, y, z, 1]$ of the two matched keypoints. $\rho(\cdot)$ is the robust Huber error function [21]. λ_h is the parameter to weight the error between the Hessian form of planes. $q(\Pi) = \left[\arctan\left(\frac{n_x}{n_z}\right), \arctan\left(\frac{n_y}{n_z}\right), d \right]$ avoids over-parametrisation of the Hessian form [1].

To cluster planes with similar motions, we introduce a score $b_{ij} \in [0, 1]$ for each pair of neighbouring planes i and j in the current frame. b_{ij} represents the probability that the motion of planes i and j can be modelled by the same rigid transformation. We further introduce a new formulation based on planes to jointly estimate motion of planes and merge planes into rigid bodies:

$$\min_{\text{ego}\mathbf{T}, \mathbf{b}} \sum_{i=1}^P e_i(T_i) + \lambda_1 \sum_{(i,j) \in E_p} b_{ij} f(T_i, T_j) - \lambda_2 \sum_{(i,j) \in E_p} b_{ij}, \quad (3)$$

s.t. $b_{ij} \in [0, 1] \forall i, j$.

$\text{ego}\mathbf{T}$ is the set of egocentric transformations $\{T_1, \dots, T_P\}$ for all planes in the current frame. $\mathbf{b} = \{b_{ij} | (i, j) \in E_p\}$. E_p is the connectivity graph of planes in the current frame and $(i, j) \in E_p$ means that planes i and j are connected in space. The first term $e_i(T_i)$ is introduced in Equation (2). In the second term, we propose $f(T_i, T_j) = [e_i(T_j) + e_j(T_i)] - [e_i(T_i) + e_j(T_j)]$ to quantify the error between two planes with egocentric motion T_i and T_j respectively. The last term penalises the model complexity by maximising the sum of probabilities that neighbouring planes have similar motions.

The novelty of the formulation is that we treat each individual plane as a motion hypothesis and estimate the likelihood \mathbf{b} of any neighbouring two hypotheses having the same motion. This is opposed to MVO [4], which discretely decides whether two motion hypotheses are merged or not.

To minimise Equation (3), ${}^{ego}\mathbf{T}$ and \mathbf{b} are decoupled. We firstly initialise all egocentric motions T_i to identity and all scores b_{ij} to 0. Then, at each iteration, we fix \mathbf{b} and find optimal ${}^{ego}\mathbf{T}$ by optimising each transformation independently. \mathbf{b} is analytically solved subsequently by fixing the optimised transformations. After minimisation, we set a threshold $\hat{b} = 0.9$ and merge planes i and j if $b_{ij} > \hat{b}$. We therefore acquire M planar rigid bodies and use RANSAC to estimate their prior egocentric motions $\{\tilde{\xi}_1, \dots, \tilde{\xi}_M\}$ respectively. However, since the dynamic objects can occupy the major part of the images, we still need to decide which planar rigid body belongs to the static background.

C. Joint camera tracking and background segmentation

We jointly track the camera motion and segment the static background from the combination of all planar and non-planar areas. This is estimated by minimising a combined formulation that consists of three energy terms:

$$\min_{\xi_c, \boldsymbol{\gamma}} R(\xi_c, \boldsymbol{\gamma}) + G(\xi_c, \boldsymbol{\gamma}) + H(\xi_c) \quad \text{s.t. } \gamma_i \in [0, 1] \quad \forall i, \quad (4)$$

where $\boldsymbol{\gamma}$ is the full set of probabilities that each plane or super-pixel is static. $\xi_c \in \mathfrak{se}(3)$ is the camera ego-motion in the world frame. Importantly, planes that belong to the same planar rigid body are assigned with independent scores γ . The first term $R(\xi_c, \boldsymbol{\gamma})$ aligns the static rigid body using weighted intensity and depth residuals. The second term $G(\xi_c, \boldsymbol{\gamma})$ segments dynamic objects by either different motions or high residuals and maintains segmentation smoothness. The last term $H(\xi_c)$ adds a soft constraint on the camera motion.

1) *Residual term:* Following the previous work [7], [17], we consider image pairs (I_{t-1}, D_{t-1}) and (I_t, D_t) from two consecutive frames. For a pixel p with coordinate $\mathbf{x}_t^p \in \mathbb{R}^2$ in the current frame t , the intensity residual $r_I^p(\xi)$ and depth residual $r_D^p(\xi)$ against the previous frame under motion ξ are given by:

$$r_I^p(\xi) = I_{t-1}(\mathcal{W}(\mathbf{x}_t^p, \xi)) - I_t(\mathbf{x}_t^p) \quad (5)$$

$$r_D^p(\xi) = D_{t-1}(\mathcal{W}(\mathbf{x}_t^p, \xi)) - |T(\xi)\pi^{-1}(\mathbf{x}_t^p, D_t(\mathbf{x}_t^p))|_z, \quad (6)$$

where $\pi: \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is the camera projection function and $|\cdot|_z$ returns the z -coordinate of a 3D point. The image warping function \mathcal{W} is:

$$\mathcal{W}(\mathbf{x}_t^p, \xi) = \pi(T(\xi)\pi^{-1}(\mathbf{x}_t^p, D_t(\mathbf{x}_t^p))), \quad (7)$$

which provides the corresponding coordinate \mathbf{x}_{t-1}^p in the previous frame. Similar to SF, the weighted residual term is:

$$R(\xi_c, \boldsymbol{\gamma}) = \sum_{p=1}^N \gamma_{i(p)} [F(\alpha_r w_I^p r_I^p(\xi_c)) + F(w_D^p r_D^p(\xi_c))], \quad (8)$$

where N is the number of pixels with a valid depth value and $i(p) \in [1, P+S]$ indicates the index of the segment that contains the pixel p . α_r is used to weight the intensity residuals. The Cauchy robust penalty:

$$F(r) = \frac{c^2}{2} \log \left(1 + \left(\frac{r^2}{c} \right) \right) \quad (9)$$

is used to control robustness of minimisation and c is the inflection point of $F(r)$. Compared to SF, which assigns scores to each cluster, we represent the image a combination of planes and super-pixels.

2) *Segmentation term:* The objective of $G(\xi_c, \boldsymbol{\gamma})$ is to detect dynamic planar rigid bodies by their motions and dynamic non-planar super-pixels by their high residuals. $G(\xi_c, \boldsymbol{\gamma})$ is computed by the sum of three items:

$$G(\xi_c, \boldsymbol{\gamma}) = \lambda_p G_p(\xi_c, \boldsymbol{\gamma}) + \lambda_{np} G_{np}(\boldsymbol{\gamma}) + \lambda_r G_r(\boldsymbol{\gamma}), \quad (10)$$

where λ_p , λ_{np} and λ_r are parameters to weight different items. The first term $G_p(\xi_c, \boldsymbol{\gamma})$ classifies planes as dynamic when their egocentric motions are different from the camera motion ξ_c :

$$G_p(\xi_c, \boldsymbol{\gamma}) = \sum_{i=1}^P \gamma_i \rho(\|\xi_c - \tilde{\xi}_{m(i)}\|_2^2), \quad (11)$$

where $m(i)$ is the planar rigid body that contains the plane i . $\tilde{\xi}_{m(i)}$ is the egocentric motion prior of the m -th planar rigid body and Huber cost function $\rho(\cdot)$ is used to robustly control the error.

The second term $G_{np}(\boldsymbol{\gamma})$ handles non-planar dynamic areas. We follow StaticFusion and assume they have a significantly higher residual under the camera motion:

$$G_{np}(\boldsymbol{\gamma}) = F(\hat{c}) \sum_{i=P+1}^{P+S} (1 - \gamma_i) N_i, \quad (12)$$

where we only consider super-pixels in non-planar area and N_i is the number of pixels with valid depth in the i -th super-pixel. The threshold \hat{c} is chosen as the average residual over all S super-pixels.

The last term $G_r(\boldsymbol{\gamma})$ maintains the spacial smoothness of segmentation $\boldsymbol{\gamma}$ for both planar and non-planar areas by encouraging neighbour areas to have close scores:

$$G_r(\boldsymbol{\gamma}) = \sum_{(i,j) \in E_p} b_{ij} (\gamma_i - \gamma_j)^2 + \sum_{(i,j) \in E_{np}} (\gamma_i - \gamma_j)^2, \quad (13)$$

where E_p and E_{np} is the connectivity graph for planes and non-planar super-pixels respectively. b_{ij} is directly acquired from the minimisation of Equation (3). This means that rather than directly assigning the same score γ to planes that belong to the same rigid body, we encourage them to have a close score γ .

3) *Motion regularisation term:* We add a soft constraint on the camera motion ξ_c based on the motion prior $\tilde{\xi}_c$:

$$H(\xi_c) = \lambda_c (1 - \alpha_s) \rho(\|\xi_c - \tilde{\xi}_c\|_2^2), \quad (14)$$

where $\alpha_s \in [0, 1]$ is the proportion between the the number of pixels that are associated to the static background over

the total number of pixels with valid depth reading. This means that we rely more on the camera motion prior when the dynamic objects occupy a higher proportion of the image view. The robust Huber cost function $\rho(\cdot)$ is used to handle large potential drifts in the camera motion prior.

The solver of Equation (4) is based on StaticFusion and a similar coarse-to-fine scheme is applied to directly align dense images. Specifically, we create an image pyramid for each incoming RGB-D image and start the optimisation from the coarsest level. The results acquired in the intermediate level is used to initialise the next level, to allow correct convergence. We also decouple the camera motion ξ_c and $\boldsymbol{\gamma}$ for more efficient computation. Concretely, we initialise the camera motion ξ_c as identity and all $\boldsymbol{\gamma}$ to 1. For each iteration, we first fix $\boldsymbol{\gamma}$ and find the optimal ξ_c . The close-form solution for $\boldsymbol{\gamma}$ is then obtained by fixing ξ_c . The solution for the previous iteration is used to initialise the current iteration.

D. Background reconstruction and camera pose refinement

In the current frame t , after the minimisation of Equation (4), we acquire the optimised camera motion $\hat{\xi}_c$ and the static parts of intensity and depth images (I_t^s, D_t^s) . These images are used to reconstruct the static background and refine the camera pose ξ_c using frame-to-model alignment. Concretely, we render an image pair (I_{t-1}^r, D_{t-1}^r) from the current static background model at the previous camera pose. The rendered image pair (I_{t-1}^r, D_{t-1}^r) is directly aligned with (I_t^s, D_t^s) by minimising

$$\min_{\xi_c} R(\xi_c, \boldsymbol{\gamma} = \mathbf{1}) + H(\xi_c). \quad (15)$$

The first term $R(\xi_c, \boldsymbol{\gamma} = \mathbf{1})$ is the same as Equation (8) but the $\boldsymbol{\gamma}$ is fixed to $\mathbf{1}$ because the input should only contain the static background. We append $H(\xi_c)$ in Equation (14) as a soft-constraint for the frame-to-model alignment and α_s is estimated from pixel-wise dynamic segmentation Γ_t . Since we have already solved Equation (4), we directly start from the finest level of the image pyramid and initialise the solver with the camera pose $\hat{\xi}_c$ for the solver of Equation (15). The refined camera pose is used to fuse the static images (I_t^s, D_t^s) with the surfel-based 3D model as described in SF [7].

E. Planar objects tracking

After removing the static planes, for each dynamic planar rigid body m , we match it to the previous dynamic rigid bodies using the plane association and estimate the egocentric motion. If all the currently associated planes are static in the previous frame, we detect the dynamic planar rigid body m as a new object and the initial pose of the object relative to the camera frame is denoted as T_{init} . If the initial time of frame for an object is t_0 , the object pose in the object's initial frame can be acquired by [14]:

$$T_{O_0^m}^{O_t^m} = T_{C_0}^{C_t} T_{C_0}^{O_t^m} T_{C_0}^{-1} T_{init}^{-1}. \quad (16)$$



(a) Mobile manipulator Ada (b) Object 1 (c) Object 2

Fig. 3: (a) An omnidirectional wheeled platform with Vicon markers. (b) The first rigid object is put on a board with wheels and moved by a human. (c) The second rigid object is put on the youBot and is controlled remotely.

IV. EVALUATION

A. Setup

The sequences for evaluation are collected with an Azure Kinect DK RGB-D camera which is mounted on an omnidirectional robot (Figure 3a). The camera produces RGB-D image pairs with a resolution of 1280 x 720 at 30 Hz. The images are down-scaled and cropped to 640 x 480 (VGA) to accelerate the speed of pre-processing (Figure 2), such as super-pixel and plane extraction. In the solver of Equation (4) and (15), we further down-scale images to 320 x 240 (QVGA).

The dynamic objects are created from stacked boxes and are either moved by humans or via a remotely controlled KUKA youBot (Figure 3). The ground truth trajectories of the camera and objects are collected using a Vicon system by attaching Vicon markers on the camera and dynamic objects. The camera motion prior is acquired by adding synthetic drift on camera ground truth trajectories with a magnitude of around 7 cm/s (trans.) and 0.4 rad/s (rot.).

For quantitative evaluation, we estimate the absolute trajectory error (ATE) and the relative pose error (RPE) [22] against the ground truth. The proposed method is compared with Joint-VO-SF (JF, [6]), StaticFusion (SF, [7]), Co-Fusion (CF, [5]) and RigidFusion (RF, [17]). We additionally provide the camera motion prior with drift to CF as the variant CF*. The original RF uses motion priors for both camera and object. Here we only provide RF with the camera motion prior and denote it as RF*, while our method with the camera motion prior is denoted as ours*.

We collect eight sequences with various camera and object movements to simulate different scenarios. For example, in the *sequence 1*, a human moves the taller box to clear way for both the robot and the other object so that the potential collision can be avoided, while in the *sequence 5*, the robot tries to overtake two dynamic objects ahead (Figure 4). All trajectories are designed such that the two dynamic objects and a human can be visible in the image at the same time and frequently occupy the major proportion of the images.

B. Camera Localisation

We estimate the ATE root-mean-square error (RMSE) and RPE RMSE the estimated camera trajectories and

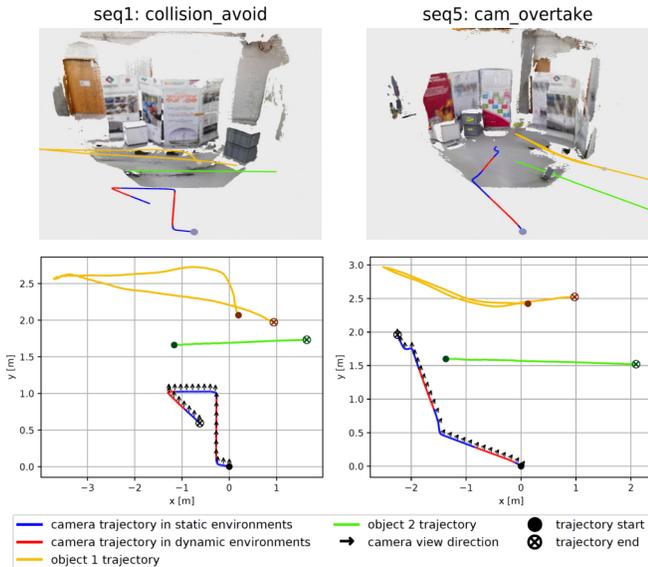


Fig. 4: The ground truth trajectories of camera and dynamic objects in both 2D and 3D perspectives. Trajectories of humans are not plotted. The red segment of camera trajectories represents the part when there are moving objects in the camera view, while the blue segment means the camera moves in static environments. We mark trajectories’ start position with a black solid dot and end position with a circle-cross marker. The black arrows point to the direction of camera view.

ground truth (Table I). The evaluation demonstrates that our method outperforms all other state-of-the-art methods. With the help of the camera motion prior, our method achieves the best performance and corrects the large drift of the camera motion prior. Even without the camera motion prior, we still achieve better performance than the baseline of JF, SF and CF. The error of the estimated trajectories from other state-of-the-art methods mainly comes from short-term tracking failures, while the error of motion priors is mainly caused by long-term drifts. The correct static/dynamic segmentation from our method helps to track camera accurately and reduce drifts from motion priors (Figure 5).

C. Multimotion Segmentation

We visualise the segmentation results of our methods and compare it with SF, RF* and CF (Figure 6). SF is unable to detect all dynamic objects because they as a whole occlude a large proportion of the camera view, while RF* tends to classify parts of the static background as dynamic. Both CF and our method can further distinguish between different dynamic objects. However, the segmentation of CF is not complete and CF tends to have a delay when detecting a new object. we use two different colours (green and purple) to represent that our method treats the taller object as a new one after it passes behind the front object.

D. Background reconstruction

We qualitatively evaluate the reconstruction result in sequence 3 (Figure 7). Since we have no ground truth segmentation, we re-collect a new sequence with the same camera trajectory but no dynamic objects to recover the true

seq	Motion prior	Method						
		JF	SF	CF	CF*	RF*	ours	ours*
1	26.7	30.5	22.9	10.4	10.2	16.5	20.1	4.23
2	49.5	28.2	27.4	26.0	7.30	14.3	6.81	6.32
3	41.7	24.3	74.0	21.6	10.6	4.38	4.01	3.42
4	36.0	28.9	87.2	18.9	20.3	8.39	22.6	8.37
5	16.2	10.3	13.6	4.73	8.35	14.1	25.2	6.74
6	11.7	52.8	23.5	10.1	3.67	7.57	8.37	4.67
7	25.5	34.7	57.6	14.7	8.71	41.3	6.43	7.60
8	28.4	26.5	62.1	69.8	18.9	14.2	8.33	10.3

(a) Trans. ATE RMSE (cm)

seq	Motion prior	Method						
		JF	SF	CF	CF*	RF*	ours	ours*
1	7.64	28.4	17.2	7.58	9.07	9.41	10.9	4.50
2	7.31	26.9	11.2	12.6	5.61	3.99	3.58	3.06
3	7.87	23.5	26.1	6.8	4.22	7.13	3.11	2.78
4	7.38	28.2	64.3	15.9	15.7	6.13	14.2	6.52
5	7.61	31.4	3.31	3.62	6.34	3.90	13.5	4.77
6	7.51	25.4	18.1	7.02	4.67	4.26	4.38	3.18
7	7.52	31.3	62.4	7.54	6.43	25.1	4.73	4.09
8	7.29	24.1	36.4	28.3	11.2	7.54	5.91	4.41

(b) Trans. RPE RMSE (cm/s)

TABLE I: ATE and RPE RMSE for all eight RGB-D sequences. The asterisk (*) symbol represents that the method uses the camera motion prior with drift. Our method achieves the best performance in nearly all sequences.

	seq1		seq4		seq7	
	object1	object2	object1	object2	object1	object2
CF	21.5	10.6	24.2	5.36	33.8	6.57
CF*	20.9	16.3	20.5	6.21	17.1	12.9
ours*	13.1	4.95	4.95	8.84	7.27	3.93

TABLE II: ATE RMSE of the object trajectories estimated from CF baseline, CF* and ours*.

background. As shown in the results, RF* maps the dynamic objects into the static background model. CF has mapped the same static object twice, which is cause by wrong camera pose estimation. Only our proposed method can remove all dynamic objects and correctly reconstruct the background.

E. Planar rigid objects trajectory

For both objects, we compute the ATE RMSE between the estimated and ground-truth trajectories when they are in the camera view (Table II). Since the object can move out of or move into the camera view several times, one object trajectory can be divided into multiple parts. For each object, we, therefore, use the maximal ATE RMSE among the estimated trajectories of different parts for the final result. Our method can provide more accurate and complete object trajectories than CF, but loses track of a dynamic object when the object stops moving or is occluded by other objects (Figure 8).

F. Impact of drift in motion prior

We increase the drift magnitude of the camera motion prior to test our methods’ robustness to different levels of drift. By comparing the RPE RMSE of the camera motion prior and estimated trajectories, we find that our method can outperform Co-Fusion baseline with drift up to 24 cm/s (Figure 9). Even when the motion prior has a drift of nearly 30 cm/s, we can still reduce the drift to around 12 cm/s. Compared to Co-Fusion with camera motion prior, our

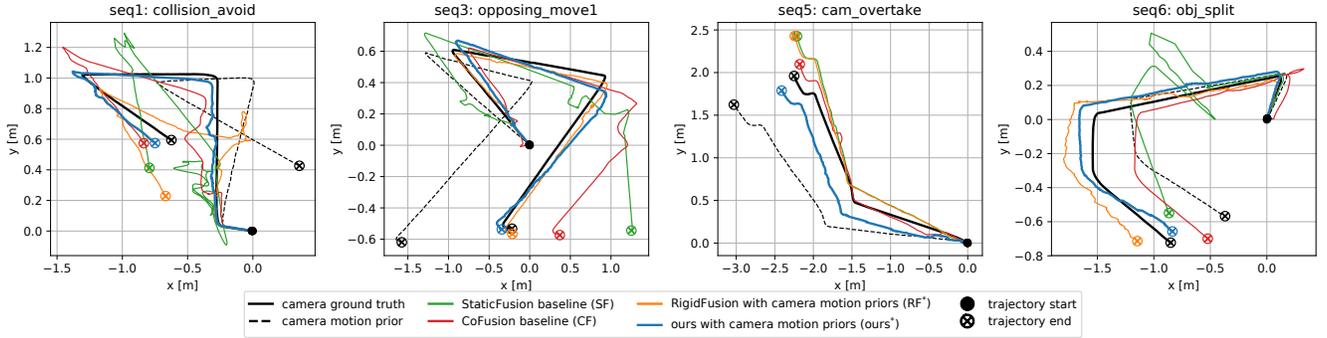


Fig. 5: Visualisation of the estimated camera trajectories, camera motion prior and ground truth. The start position of all trajectories are aligned to the same position and is marked with a black solid dot. Our method (blue) achieves the lowest error compared to the ground truth (black solid) and can correct drift of the camera motion prior (black dashed).

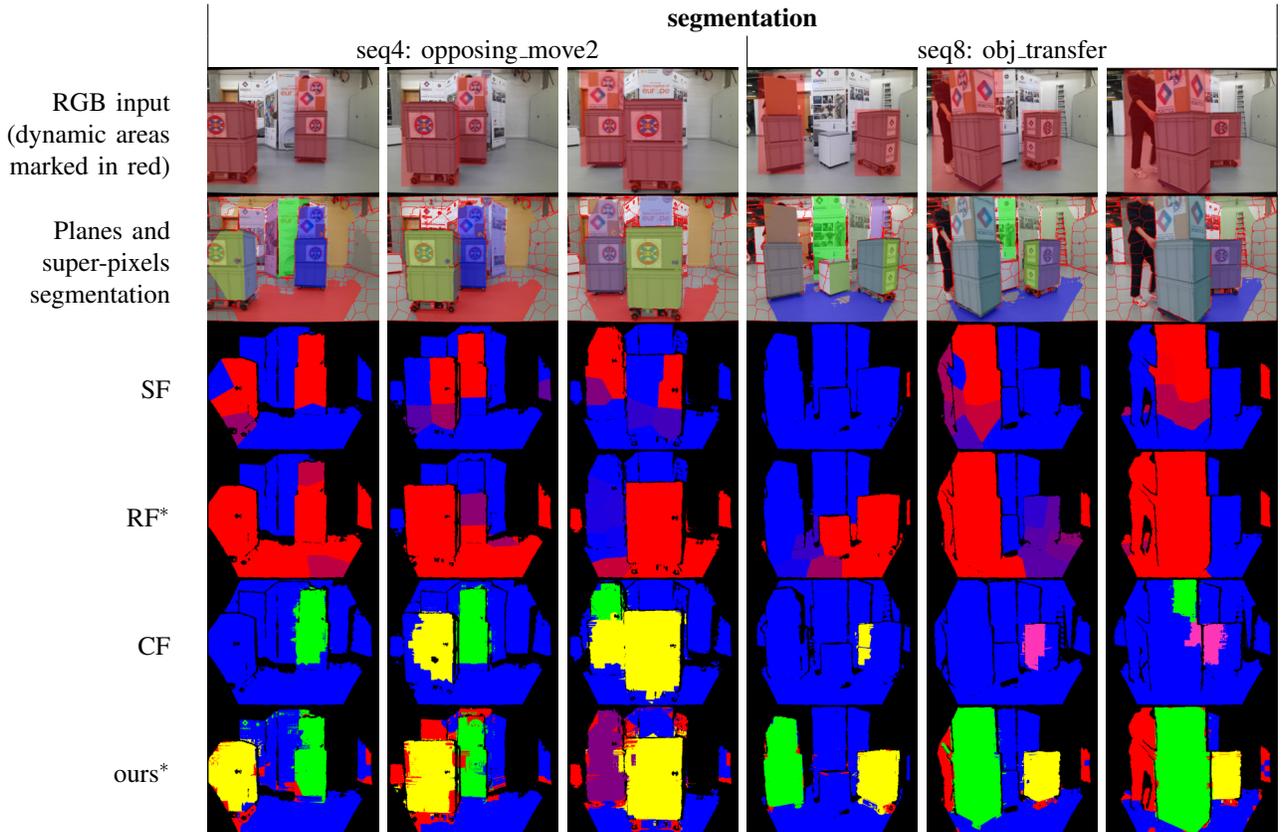


Fig. 6: Segmentation result of the static background and dynamic objects. We visualise the input RGB images and the segmentation of planes and super-pixels in the first two rows. In all four methods, the static part is marked by blue. In SF and RF, we use red to represent dynamic parts. In CF, we use different colours to show different objects. In our method, the non-planar dynamic areas are marked by red, the planar rigid objects are marked by other colours. Results show that only our method can segment multiple dynamic objects correctly and is robust to large occlusion.

method is always better using the motion prior with same magnitude of drift.

V. CONCLUSION

This work presented a dense RGB-D SLAM method that detects and tracks multiple planar rigid objects without relying on semantic segmentation. We also proposed a novel online multimotion segmentation method and a dynamic segmentation pipeline based on a hierarchical representation of planes and super-pixels. The detailed evaluation demonstrates that our method achieves better localisation

and mapping results than state-of-the-art approaches when multiple dynamic objects occupy the major proportion of the camera view.

If one dynamic object is occluded by another, our method fails to track the object but detects the object as new after it reappears in the camera view. Our future work would be re-detecting the dynamic objects based on their models to support long-term object tracking. We also plan to extend our method to dynamic structured environments and improve the accuracy of localisation and mapping.

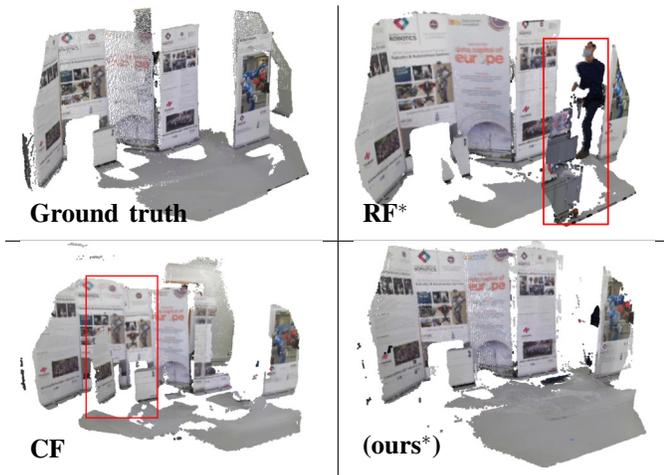


Fig. 7: Reconstruction result of the RGB-D sequence 3. The reconstruction failures are marked with red rectangles. RF has mapped dynamic objects into the background. CF has mapped the same static poster twice, which indicates wrong localisation results.

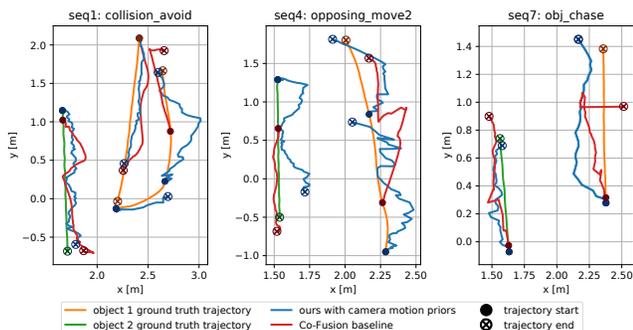


Fig. 8: Comparison between CF baseline (red) and our method with the camera motion prior (blue) in terms of the estimated object trajectories. Our method can detect an object as dynamic immediately when it starts to move and provide a more accurate trajectory than CF.

ACKNOWLEDGMENT

This research is supported by the EPSRC UK Robotics and AI ISCF ORCA Hub (EP/R026173/1), the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101017008, the Alan Turing Institute and the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS).

REFERENCES

- [1] Y. Li, R. Yunus, N. Brasch, N. Navab, and F. Tombari, “RGB-D SLAM with Structural Regularities,” in *IEEE International Conference on Robotics and Automation*, 2021.
- [2] X. Zhang, W. Wang, X. Qi, Z. Liao, and R. Wei, “Point-plane slam using supposed planes for indoor environments,” *Sensors*, 2019.
- [3] J. Huang, S. Yang, T.-J. Mu, and S.-M. Hu, “ClusterVO: Clustering moving instances and estimating visual odometry for self and surroundings,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.
- [4] K. M. Judd, J. D. Gammell, and P. Newman, “Multimotion visual odometry (MVO): Simultaneous estimation of camera and third-party motions,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018.
- [5] M. Rünz and L. Agapito, “Co-Fusion: Real-time segmentation, tracking and fusion of multiple objects,” in *IEEE International Conference on Robotics and Automation*, 2017.

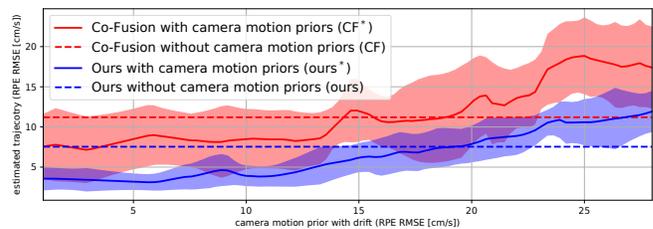


Fig. 9: RPE RMSE of the estimated trajectories against the drift magnitude of wheel odometry. Our method preforms better than CF when using the camera motion prior with the same magnitude of drift and is robust to nearly 24 cm/s odometry drift until it is comparable with CF baseline.

- [6] M. Jaimez, C. Kerl, J. Gonzalez-Jimenez, and D. Cremers, “Fast odometry and scene flow from RGB-D cameras based on geometric clustering,” in *IEEE International Conference on Robotics and Automation*, 2017.
- [7] R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon, and D. Cremers, “StaticFusion: Background reconstruction for dense RGB-D SLAM in dynamic environments,” in *IEEE International Conference on Robotics and Automation*, 2018.
- [8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *Proceedings of the IEEE international conference on computer vision*, 2017.
- [9] M. Rünz, M. Buffier, and L. Agapito, “MaskFusion: Real-time recognition, tracking and reconstruction of multiple moving objects,” in *IEEE International Symposium on Mixed and Augmented Reality*, 2018.
- [10] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, “MID-Fusion: Octree-based object-level multi-instance dynamic SLAM,” in *International Conference on Robotics and Automation*, 2019.
- [11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *IEEE International conference on computer vision*, 2011.
- [12] B. Bescos, C. Campos, J. D. Tardós, and J. Neira, “DynaSLAM II: Tightly-coupled multi-object tracking and SLAM,” *IEEE Robotics and Automation Letters*, 2021.
- [13] S. Lee, C. Y. Son, and H. J. Kim, “Robust real-time RGB-D visual odometry in dynamic environments via rigid motion model,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.
- [14] C. Zhang, B. Luo, Y. Zhang, Q. Zhao, L. Yin, W. Wang, X. Su, Y. Wang, and C. Li, “DymSLAM: 4D Dynamic Scene Reconstruction Based on Geometrical Motion Segmentation,” *IEEE Robotics and Automation Letters*, 2020.
- [15] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” *IEEE transactions on pattern analysis and machine intelligence*, 2012.
- [16] D.-H. Kim, S.-B. Han, and J.-H. Kim, “Visual odometry algorithm using an RGB-D sensor and imu in a highly dynamic environment,” in *Robot Intelligence Technology and Applications 3*. Springer, 2015.
- [17] R. Long, C. Rauch, T. Zhang, V. Ivan, and S. Vijayakumar, “RigidFusion: Robot Localisation and Mapping in Environments With Large Dynamic Rigid Objects,” *IEEE Robotics and Automation Letters*, 2021.
- [18] M. Kaess, “Simultaneous localization and mapping with infinite planes,” in *IEEE International Conference on Robotics and Automation*, 2015.
- [19] M. Hsiao, E. Westman, G. Zhang, and M. Kaess, “Keyframe-based dense planar slam,” in *IEEE International Conference on Robotics and Automation*, 2017.
- [20] C. Feng, Y. Taguchi, and V. R. Kamat, “Fast plane extraction in organized point clouds using agglomerative hierarchical clustering,” in *IEEE International Conference on Robotics and Automation*, 2014.
- [21] R. Mur-Artal and J. D. Tardós, “ORB-SLAM2: An open-source slam system for monocular, stereo, and rgb-d cameras,” *IEEE transactions on robotics*, 2017.
- [22] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of RGB-D SLAM systems,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.