THE UNIVERSITY *of* EDINBURGH

ROBUST SLAM AND MOTION SEGMENTATION UNDER LONG-TERM DYNAMIC LARGE OCCLUSIONS

RAN LONG

Doctor of Philosophy

School of Informatics
University of Edinburgh

2023

Ran Long:
Robust SLAM and Motion Segmentation under Long-term Dynamic Large Occlusions
Doctor of Philosophy, 2023

SUPERVISORS:
Prof. Sethu Vijayakumar, FRSE
Dr. Oisin Mac Aodha

EXAMINERS:
Prof. Andrew Davison
Dr. Hakan Bilen

ABSTRACT

Visual sensors are key to robot perception, which can not only help robot localisation but also enable robots to interact with the environment. However, in new environments, robots can fail to distinguish the static and dynamic components in the visual input. Consequently, robots are unable to track objects or localise themselves. Methods often require precise robot proprioception to compensate for camera movement and separate the static background from the visual input. However, robot proprioception, such as inertial measurement unit (IMU) or wheel odometry, usually faces the problem of drift accumulation. The state-of-the-art methods demonstrate promising performance but either (1) require semantic segmentation, which is inaccessible in unknown environments, or (2) treat dynamic components as outliers – which is unfeasible when dynamic objects occupy a large proportion of the visual input.

This research work systematically unifies camera and multi-object tracking problems in indoor environments by proposing a multi-motion tracking system; and enables robots to differentiate the static and dynamic components in the visual input with the understanding of their own movements and actions. Detailed evaluation of both simulation environments and robotic platforms suggests that the proposed method outperforms the state-of-the-art dynamic SLAM methods when the majority of the camera view is occluded by multiple unmodeled objects over a long period of time.

ACKNOWLEDGEMENTS

Foremost, I want to express my gratitude to Prof. Sethu Vijayakumar, my supervisor, for providing me with assistance and direction throughout my PhD at the Statistical Learning and Motor Control (SLMC) Group. Additionally, I extend my gratitude to my second supervisor, Dr. Oisin Mac Aodha and my annual review examiner Prof. Bob Fisher, for their invaluable input during my PhD and my annual evaluations. Importantly, I would also like to thank my collaborator, Dr. Christian Rauch, who has provided me with significant support, inspired me with creative thought, encouraged me with meticulous effort, influenced me with diligent attitude and led me to novel work.

Next, I would like to convey my sincere appreciation to my colleagues and mentors who have helped me with learning robotics and thinking like a researcher: Vladimir Ivan, Christian Rauch, Henrique Ferrolho, Wolfgang Merkt, Chris Xiaoxuan Lu, Theodoros Stouraitis, Joao Moura, Raluca Scona, Lei Yan and Songyan Xin, Wenqian Du. I have gained a wealth of knowledge from you, and a significant portion of my accomplishments can be attributed to your commitment and support.

I also want to thank the rest of individuals in our group who have assisted me in various ways during my journey: Andreas Christou, Carlo Tiseo, Carlos Mastalli , Christopher Mower, Daniel Gordon, Elle Miller, Jaehyun Shim, Jiayi Wang, Keyhan Babarahmati, Marina Aoyama, Matt Timmons-Brown, Namiko Saito, Ruaridh Mon-Williams, Saeid Samadi, Thomas Corberes, Serena Lambley, Traiko Dinev, Wouter Wolfslag, Yiming Yang.

Besides these individuals, I have been lucky enough to receive assistance from people who are not affiliated with the University of Edinburgh, some of whom I haven't had the opportunity to meet in person. Thank you, Tianwei Zhang and Lin Tin Lam, for our nice discussions related to robotics, SLAM and computer vision.

I would also like to thank my friends: Jiyuan Wang, Nanbo Li, Taichi Hosoi, Wenbin Hu, Xiaofeng Mao, Zhaole Sun, Zhaocheng Liu, Zhengyan Cai, Zhiyuan Zhang, Zonglin Ji, for the fun time we spent together.

Last but not least, my deepest gratitude, from the bottom of my heart, goes to my parents, Ping Long and Zhaohua Fan, my parents-in-law, Bo Zhang and Linying Wang, and my wife Xiaoyu Zhang:

我衷心感谢你们的支持与关心。尽管四年来我总是漂泊在外，但没有你们，这一切都无法完成。

PUBLICATIONS

Parts of the research leading to this thesis have previously appeared in the following peer-reviewed publications. Some passages have been quoted verbatim from the respective sources.

JOURNAL ARTICLES

- Ran Long, Christian Rauch, Tianwei Zhang, Vladimir Ivan and Sethu Vijayakumar, 'RigidFusion: Robot Localisation and Mapping in Environments With Large Dynamic Rigid Objects', in *IEEE Robotics and Automation Letters (**RA-L**)*, vol. 6, no. 2, pp. 3703-3710, April 2021. (Presented at: *IEEE International Conference on Robotics and Automation (**ICRA** 2021)*) (Chapter 4)

  Video: `https://youtu.be/hnyAGv-ZiMM`

- Ran Long, Christian Rauch, Tianwei Zhang, Vladimir Ivan, Tin Lun Lam and Sethu Vijayakumar, 'RGB-D SLAM in Indoor Planar Environments With Multiple Large Dynamic Objects', in *IEEE Robotics and Automation Letters (**RA-L**)*, vol. 7, no. 3, pp. 8209-8216, July 2022. (Presented at: *IEEE/RSJ International. Conference on Intelligent Robots and Systems (**IROS** 2022)*) (Chapter 5)

  Video: `https://youtu.be/iMTop_1glSc`

- Ran Long, Christian Rauch, Tianwei Zhang, Vladimir Ivan, Tin Lun Lam and Sethu Vijayakumar, 'RGB-D-Inertial SLAM in Indoor Dynamic Environments with Long-term Large Occlusion'. arXiv preprint arXiv:2303.13316 (2023). (Chapter 6)

- Christian Rauch, Ran Long, Vladimir Ivan and Sethu Vijayakumar, 'Sparse-Dense Motion Modelling and Tracking for Manipulation Without Prior Object Models', in *IEEE Robotics and Automation Letters (**RA-L**)*, vol. 7, no. 4, pp. 11394-11401, Oct. 2022. (Presented at: *IEEE International Conference on Robotics and Automation (**ICRA** 2023)*)(Chapter 4)

  Video: `https://youtu.be/b8pov4DKLsY`

DECLARATION

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Edinburgh, United Kingdom, 2023

Ran Long
8th August 2023

Dedicated to my wife and my parents,
who have always encouraged me, supported me and guided me.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AR	augmented reality
ATE	absolute trajectory error
BA	bundle adjustment
BEV	bird's eye view
CNN	convolutional neural network
EKF	extended Kalman filter
GRIC	geometrically robust information criterion
IMU	inertial measurement unit
kNN	k-nearest neighbours
LiDAR	light detection and ranging
LO	large occlusion
MAP	maximum a posteriori
MLP	multilayer perception
MW	Manhattan world
NeRF	neural radiance field
PF	PlanarFusion
RANSAC	random sample consensus
RF	RigidFusion
RMSE	root mean squared error
RPE	relative pose error
PTAM	parallel tracking and mapping
SAM	Segment Anything
SLAM	simultaneous localisation and mapping
ToF	time of flight
TSDF	truncated signed distance field

UKF unscented Kalman filter

VIBA visual-inertial bundle adjustment

VINS visual inertial navigation system

VR virtual reality

VO visual odometry

LIST OF SYMBOLS

W	World frame
C	Camera frame
O	Object frame
t	Frame of time
$\mathbf{R}$	3-D rotation in the SO(3) group
$\mathbf{t}$	3-D translation
$\mathbf{T}$	3-D rigid transformation in the SE(3) group
$\mathbf{T}_{AB}$	Transformation from frame A to frame B
$\boldsymbol{\xi}$	Lie algebra of the SE(3) group
$\mathbf{u}$	2-D image coordinate of a pixel
$\mathbf{x}$	3-D position of a point in the world frame
γ	Segmentation score of a cluster
$\boldsymbol{\Gamma}$	Set of per-cluster score over an image
I	Intensity images
D	Depth maps
N	Number of pixels in an image
M	Number of rigid body of motions
K	Number of clusters
$\mathbf{V}$	Connectivity graph of clusters
$r_I^{\mathbf{u}}$	Intensity residual at the pixel $\mathbf{u}$
$r_D^{\mathbf{u}}$	Depth residual at the pixel $\mathbf{u}$
$\mathbf{n}$	Normal direction of planes
d	Perpendicular distance between the plane and camera origin
$\boldsymbol{\Pi}$	Hessian form of plane
$\pi(\cdot)$	Camera projection function
$\lvert\cdot\rvert_z$	z-coordinate of a 3-D point
$num(\cdot)$	Number of elements in a set

$\mathcal{W}(\cdot)$	Image warping function
$F(\cdot)$	Cauchy robust function
$\rho_H(\cdot)$	Robust Huber loss function

1

INTRODUCTION

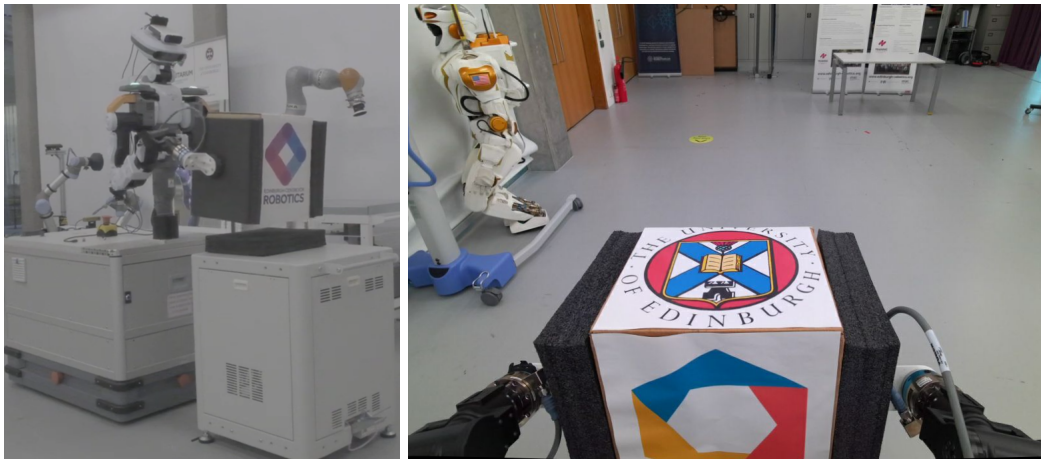
1.1 SCOPE

The use of robots in mobile sensing, mapping and loco-manipulation tasks is becoming ubiquitous across various domains. To advance the automation of complex tasks, it is essential to enhance robots' perception of the surrounding environments and understanding of their ego-motions. Visual SLAM can help robots build a model of previously unseen environments and localise themselves based on visual input only. Recent visual SLAM systems can accurately localise the camera in a large-scale environment [20, 60, 61] and densely reconstruct the environment at a real-time speed [16, 65, 123]. Furthermore, visual sensors can be combined with robot proprioceptive sensors, such as an IMU, to further improve system's robustness to quick camera movement or large-occluded camera views [11, 74, 104]. However, most of these methods are based on a static world assumption and, therefore, struggle to localise the camera accurately in dynamic environments which are closer to real-world application scenarios.

In addition to the academic community, SLAM methods have also been extensively applied to multiple commercial products. For instance, robotic vacuum cleaners can achieve accurate self-localisation on flat surfaces by fusing visual sensors and wheel odometry [82]. VR headsets can utilise depth cameras to reconstruct indoor environments based on the user's head movements. Drones are capable of performing accurate ego-motion estimation even during rapid camera movements. However, SLAM methods implemented on commercial goods are often limited to certain application scenarios. For example, SLAM methods designed for robotic vacuum cleaners are limited to 2D camera motion estimation on planar surfaces. VR headsets have challenges in real-time 3D background reconstruction during long-distance user movements but focus on 3D reconstruction while users are relatively stationary, such as standing or sitting. Moreover, the working environment of drones is usually kept at a distance from other dynamic objects, such as people and vehicles for

safety reasons. Therefore, the majority of the field of view consists of static backgrounds.

One real-world robot application that involves long-distance camera 3D trajectory estimation in highly dynamic environments is the task of mobile manipulation (Figure 1), where a robot needs to simultaneously move the base and manipulate objects which brings dynamic objects into the environment. Moreover, when a dynamic object is large or moves relatively close to the camera itself, it can occupy the major proportion of the visual input. This makes it even more difficult for a robot to localise itself.



(a) (b)

Figure 1: (a) A mobile manipulator manipulates an object in front of the camera, which introduces dynamic objects in the visual input. (b) When an object is large or closely manipulated, it can cause dynamic large occlusion in the camera view.

In order to localise in dynamic environments, dynamic SLAM methods need to detect dynamic objects and mitigate their impact. To achieve it, current state-of-the-art methods either use semantic segmentation to detect pre-defined dynamic objects directly [5, 85, 103, 129]; or assume that moving objects are a smaller part of the overall image and can be removed as outliers compared to the static background [84, 94]. However, if an unmodeled dynamic object is large or moves close to the camera, it can cause dynamic large occlusions that occupy a major proportion of the camera view. Consequently, current state-of-the-art methods can misclassify the unmodeled large dynamic object as static and fail to localise the camera.

Humans, however, can locate themselves in new environments and perceive their ego-motion even when the majority of visual input is occluded by dy-

namic objects. This is because, apart from vision, humans can understand dynamic environments by the perception of their own movements and actions from proprioception. For example, when humans move, they can perceive their body movement via leg odometry. Additionally, when humans push objects, they can approximately estimate the objects' trajectory via hand movement even without vision. This perception of body movement can help humans understand and distinguish static and dynamic objects in the environment.

Similarly, robots can rely on proprioception only, such as robot odometry or IMU, to localise themselves when the visual input is unavailable [7, 12, 46]. However, the wheel or leg odometry of a robot can produce significant drifts on uneven terrain and, therefore, cannot be used for accurate localisation over a long period [126]. To solve this problem, robot proprioceptive sensors are often fused with visual sensors to reduce long-term camera drift and enable accurate localisation in large-scale static environments [74, 126].

This thesis, therefore, specifically targets domains where we need to use relatively inaccurate robot proprioception, such as wheel odometry and IMU to realise robust SLAM and motion segmentation in dynamic environments when the majority of camera view is occluded by multiple unmodeled dynamic objects for a long period.

1.2 PROBLEM FORMULATION

We consider the problem of simultaneous localisation and mapping (SLAM) and motion segmentation in the presence of long-term dynamic large occlusions. Especially, we are interested in using proprioceptive sensors, like inertial measurement unit (IMU) or wheel odometry, to improve the accuracy of robot localisation. However, there are three challenges to completing this task.

1.2.1 *Challenge 1: Large dynamic occlusions*

As mentioned in Section 1.1, dynamic objects can be removed as outliers if they are smaller compared to the static background in the visual input. This is achieved by estimating the rigid transformation of the dominant rigid body and selecting this rigid body as the static background. However, in the scenario of large occlusion, a dynamic object can frequently become the dominant rigid

body of motion in the visual input. Therefore, the largest rigid body in the camera view cannot be pre-determined as either the static background or a moving object.

1.2.2 *Challenge 2: Unmodeled dynamic objects*

In real-world applications, there are usually multiple independently moving dynamic objects in the environment. Even if these moving objects are distinguished from the static environment, a robot needs to further segment and track individual dynamic objects for a more comprehensive understanding of its surrounding environments.

When dynamic objects are unmodeled, robots have no information about the number or appearance of dynamic objects– hence, the classical methods can fail to differentiate one dynamic object from another.

1.2.3 *Challenge 3: Lack of information from the static background*

In the scenario of large occlusion, it is difficult to recover the occluded static background after the removal of dynamic objects. Therefore, the information that belongs to the occluded static areas is lost, including the textures from RGB cameras and depth readings from depth cameras. When dynamic large occlusion lasts for a long period, the loss of information can cause camera drift accumulation and even lead to a tracking failure.

To enable robust SLAM in the scenario of long-term large occlusion, it's important that we resolve these three challenges.

1.3 CONTRIBUTIONS

Next we summarise the key approach and contributions of this thesis towards resolving the above mentioned challenges.

1.3.1 *A novel dense RGB-D SLAM method that is robust to rigid dynamic large occlusions*

Our first contribution is enabling dense SLAM when the major proportion of the camera view is occluded by a rigid dynamic object. Previous state-of-the-art methods are limited to handling pre-defined dynamic objects or assuming that dynamic objects account for a minor proportion of the visual input. Therefore, when unmodeled dynamic objects cause large occlusion, these methods are unable to accurately localise the camera because large dynamic objects can be misclassified as static.

To address this issue, we propose a method that models the entire dynamic component of the visual input with a single rigid transformation. Our approach simultaneously segments, tracks, and reconstructs the static background and a single rigid dynamic object. The novelty of our method is segmenting two rigid bodies with different motions and using motion priors to differentiate the static from the dynamic. Motion priors represent prior rigid transformations of either the camera or dynamic objects. In the mobile manipulation scenario, the camera motion prior can be acquired from wheel odometry when the camera is mounted on the base, while the object motion prior either comes from arm kinematics when the object is manipulated.

Detailed evaluation demonstrates that our method can use motion priors with large drift to segment and track the static and dynamic rigid bodies online. The static rigid body is used to reconstruct the static background, improve the accuracy of camera localisation and furthermore reduce potential drifts from the camera motion prior. We also reconstruct the dynamic object model, which could be used in other robotics applications, like object grasping.

This contribution is presented in:

- Ran Long, Christian Rauch, Tianwei Zhang, Vladimir Ivan and Sethu Vijayakumar, 'RigidFusion: Robot Localisation and Mapping in Environments With Large Dynamic Rigid Objects', in *IEEE Robotics and Automation Letters (**RA-L**)*, vol. 6, no. 2, pp. 3703-3710, April 2021.

- Christian Rauch, Ran Long, Vladimir Ivan and Sethu Vijayakumar, 'Sparse-Dense Motion Modelling and Tracking for Manipulation Without Prior Object Models', in *IEEE Robotics and Automation Letters (**RA-L**)*, vol. 7, no. 4, pp. 11394-11401, Oct. 2022.

1.3.2 *A novel dense RGB-D SLAM method that simultaneously segments and track multiple unmodeled dynamic planar objects*

The major limitation of our first contribution is that it assumes the whole dynamic components can be modelled as a single rigid body. Our second contribution resolves this limitation in planar environments and can simultaneously segment and track multiple rigid planar objects that cause dynamic large occlusions. To improve the robustness of our method, we also detect and remove non-planar dynamic objects as outliers.

To approach this aim, we segment planar areas of an image into planes and the remaining non-planar areas into super-pixels. We then introduce a novel multimotion visual odometry to merge planes with close rigid motions and, therefore, estimate and track multiple planar rigid bodies independently. The non-planar super-pixels are classified into static and dynamic with the camera motion priors. The static part of the visual input is a combination of static super-pixels and the planar rigid object with the closest rigid motion to the camera motion. Similar to the first contribution, we refine camera trajectories and reconstruction the static background with the static parts using frame-to-model alignment.

Compared to other state-of-the-art dynamic SLAM methods, our method is robust to large occlusions caused by multiple unmodeled dynamic objects and provides dense segmentation of these objects separately.

This contribution is presented in:

- Ran Long, Christian Rauch, Tianwei Zhang, Vladimir Ivan, Tin Lun Lam and Sethu Vijayakumar, 'RGB-D SLAM in Indoor Planar Environments With Multiple Large Dynamic Objects', in *IEEE Robotics and Automation Letters (**RA-L**)*, vol. 7, no. 3, pp. 8209-8216, July 2022.

1.3.3 *A novel RGB-D-inertial SLAM method that is robust to Long-term Dynamic Large Occlusions*

In our previous two contributions, we only considered dynamic large occlusion that lasts for a short period. However, an additional problem is caused by long-term dynamic large occlusions when large occlusion persists for the majority of robot operation time. After the removal of dynamic objects that cause large

occlusion, the remaining information from the static background is insufficient to support accurate camera localisation, resulting in camera drift accumulation. In the scenario of short-term large occlusion, robots can either temporarily rely on robot proprioception or relocalise the camera after the visual input from the static background becomes sufficient. This consequently reduces the drift accumulated during dynamic large occlusion. However, when dynamic large occlusion lasts over a long period, both our previous methods are unable to reduce the drift of camera or camera motion prior from wheel odometry.

To address this challenge, our third contribution proposes a novel visual-inertial bundle adjustment method that can effectively segment dynamic objects that cause long-term large occlusions. We also tightly fuse an IMU to the camera to reduce drift and combine sparse and dense information by maintaining a sparse static map while estimating dense segmentation of dynamic objects. Consequently, we can reconstruct the background offline with the static segmentation of images.

Finally, we evaluate our method on a real-world mobile manipulator experiment in a complex dynamic environment and demonstrate its performance in real robot applications.

This contribution is presented in:

- Ran Long, Christian Rauch, Tianwei Zhang, Vladimir Ivan, Tin Lun Lam and Sethu Vijayakumar, 'RGB-D-Inertial SLAM in Indoor Dynamic Environments with Long-term Large Occlusion'. arXiv preprint arXiv:2303.13316 (2023).

1.4 THESIS OUTLINE

It's recommended to read this thesis in sequential chapter order. Each chapter contains a concise introduction and discussion, conveying the primary scientific concepts and contributions of each work. In addition, the chapters are connected through a coherent thread that is summarised in the paragraphs below.

In Chapter 2, we will introduce the widely-used sensors in visual-inertial simultaneous localisation and mapping (SLAM) methods, including monocular, stereo, RGB-D cameras and inertial measurement unit (IMU). We also introduce the mathematical concepts of 3D body motion and the metrics we used to evaluate trajectories.

Chapter 3 provides an overview of related work, including SLAM in static and dynamic environments. For static SLAM methods, we first introduce feature-based visual SLAM and then dense visual SLAM. This is followed by an introduction of planar SLAM methods which extract and use planes in environments. Additionally, we introduce SLAM methods that fuse visual sensors and other proprioceptive sensors, such as IMU and robot odometry. For dynamic SLAM methods,

In Chapter 4, we propose a dense RGB-D SLAM approach to simultaneously segment, track and reconstruct the static background and a single large dynamic rigid object that can occlude major proportions of the camera view.

Chapter 5 extends our previous method to track multiple dynamic objects in planar environments and proposes a dense RGB-D SLAM approach for dynamic planar environments that enables simultaneous multi-object tracking, camera localisation and background reconstruction.

Chapter 6 considers the scenario when the majority of camera view is occluded for most of the time when the camera is in motion. In this scenario, unreliable camera motion priors can lead to wrong object segmentation. Moreover, the colour and depth information from the static parts of images may not be sufficient to support accurate camera localisation. To address these issues, our framework proposes a robust visual-inertial bundle adjustment (VIBA) method that simultaneously tracks the camera, estimates dense

segmentation of dynamic objects based on clusters, and maintains a sparse static map by combining dense and sparse features.

Finally, in Chapter 7, we conclude with multiple directions for improving our approaches and potential future research work.

2

PRELIMINARIES

This chapter introduces the concepts of 3D rigid body motion and gives an overview of the sensors used in visual-inertial simultaneous localisation and mapping (SLAM) methods, including the monocular, stereo, RGB-D cameras and inertial measurement unit (IMU). Additionally, we introduce metrics we used to evaluate estimated camera trajectories. Finally, we discuss factor graphs and explain the relation between this formulation to SLAM.

2.1 3D RIGID BODY MOTION

In this thesis, we use the special euclidean group $\mathbf{T} \in SE(3)$ to represent the 3-D rigid transformation:

$$SE(3) := \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4x4} : \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3 \right\}, \tag{1}$$

where $\mathbf{R}$ denotes the 3-D rotation and $\mathbf{t}$ denotes the 3-D translation. To optimise over 3-D poses, we also use the Lie algebra $\boldsymbol{\xi} \in \mathfrak{se}(3)$, which is the tangent space of SE(3) to represent 3-D poses during optimisation.

The Lie algebra $\boldsymbol{\xi} \in \mathfrak{se}(3)$ is defined as:

$$\left\{ \boldsymbol{\xi} = (\boldsymbol{\rho}, \boldsymbol{\phi})^T \in \mathbb{R}^6 | \boldsymbol{\rho} \in \mathbb{R}^3, \Phi \in \mathbb{R}^3, \boldsymbol{\xi}^\wedge = \begin{bmatrix} \boldsymbol{\phi}^\wedge & \boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4x4} \right\}, \tag{2}$$

where $\boldsymbol{\phi}^\wedge$ generates a skew symmetric matrix from a 3-D vector $\boldsymbol{\phi}$:

$$\boldsymbol{\phi}^\wedge = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix}. \tag{3}$$

Given the Lie algebra $\boldsymbol{\xi}$, we can use the exponential map to calculate the rigid transformation $\mathbf{T} \in SE(3)$ [121]:

$$\mathbf{T} = exp(\boldsymbol{\xi}). \tag{4}$$

Conversely, we can acquire the lie algebra from transformation $\mathbf{T}$ using the logarithmic operation:

$$\boldsymbol{\xi} = log(\mathbf{T}). \tag{5}$$

For more details related to the exponential map, please refer to [121].

2.2 SENSORS FOR VISUAL-INERTIAL SLAM

2.2.1 *Monocular Camera*

Monocular cameras are widely used in SLAM methods and can be modelled by the pinhole camera model. Provided a 3D point $\mathbf{x} = [X, Y, Z]^T \in \mathbb{R}^3$ in the camera frame, we can project the 3D point on the image and acquire the pixel position $\mathbf{u} = [u, v]^T \in \mathbb{R}^2$:

$$\mathbf{u} := \pi(\mathbf{x}) = \begin{bmatrix} f_x \frac{X}{Z} + c_x \\ f_y \frac{Y}{Z} + c_y \end{bmatrix}, \tag{6}$$

where f_x and f_y are the horizontal and vertical focal lengths of the RGB camera respectively. (c_x, c_y) represents the image coordinate of the optical centre. If the depth Z of a pixel is given, we can backproject a pixel to the 3D world:

$$\mathbf{x} = \pi^{-1}(\mathbf{u}, Z) = \begin{bmatrix} \frac{u - c_x}{f_x} \\ \frac{v - c_y}{f_y} \\ 1 \end{bmatrix} Z. \tag{7}$$

However, due to the lack of depth reading, monocular cameras are unable to provide the scale information of the surrounding environments. This can lead to long-term drift of localisation.

2.2.2 *Stereo and RGB-D Camera*

Both stereo and depth cameras can provide depth information about the environment.

A stereo camera is composed of two synchronised RGB cameras with a rigid transformation. Here, we assume that two cameras are well-calibrated and the baseline b which is the distance between the lens of two RGB cameras is estimated beforehand. Given a 3D point $\mathbf{x} = [X, Y, Z]^T$ in the left camera frame, we can use the projection function of stereo cameras π^s to acquire a stereo coordinate $\mathbf{u}^s = [u, v, u^r]^T$ [61]:

$$\mathbf{u}^s := \pi^s(\mathbf{x}) = \begin{bmatrix} f_x \frac{X}{Z} + c_x \\ f_y \frac{Y}{Z} + c_y \\ f_x \frac{X-b}{Z} + c_x \end{bmatrix}, \tag{8}$$

where (u, v) is the image coordinate on the left image and u^r is the horizontal image coordinate of the corresponding pixel on the right image. To estimate depth, a stereo camera first needs to find corresponding feature points between the left and right images, and use triangulation to estimate the depth.

An RGB-D camera combines a monocular camera and a depth sensor. In our first publication (Chapter 4), we collect our own dataset with a structure-light depth camera Asus Xtion Pro Live[1] (Figure 2a). It uses an infrared projector to project a known infrared pattern, which is then perceived and used to estimate depth of each pixel. In all other publications, we use a time of flight (ToF) depth camera Azure Kinect DK[2] (Figure 2b) to collect our own dataset. A ToF camera can emit pulses of infrared light towards the environment in the camera's field of view. The depth is then calculated based on the time it takes for the light pulse to travel from the camera to object and back again.

Compared to stereo cameras, RGB-D cameras can estimate per-pixel depth and have a higher accuracy within the operational range.

1 http://xtionprolive.com/asus-3d-depth-camera/asus-xtion-pro-live
2 https://azure.microsoft.com/en-us/products/kinect-dk

(a) Asus Xtion Pro Live (b) Azure Kinect DK

Figure 2: (a) An Asus Xtion Pro Live RGB-D camera. (b) An Azure Kinect DK RGB-D camera based on ToF.

2.2.3 *Inertial Measurement Unit*

An IMU is an electronic device that is used to measure and report on the acceleration, orientation, and angular velocity of an object in three-dimensional space. An IMU consists of a gyroscope and an accelerometer which provide raw measurements of angular velocity $\hat{\omega}$ and acceleration $\hat{a}$ in the body frame:

$$\hat{\omega}_t = \omega_t + \mathbf{b}_t^{\omega} + \mathbf{n}_{\omega} \tag{9}$$

$$\hat{a}_t = a_t + \mathbf{b}_t^{a} + R_t\mathbf{g} + \mathbf{n}_a, \tag{10}$$

where ω_t and a_t are true values of angular velocity and acceleration at the time t. $\mathbf{b}_t^{\omega}$ and $\mathbf{b}_t^{a}$ are bias terms of gyroscope and acceleration respectively, and they can be modelled by Gaussian random walk, which changes slowly. $R_t\mathbf{g}$ is the direction of gravity in the body frame. $\mathbf{n}_{\omega}$ and $\mathbf{n}_a$ are Gaussian white noise:

$$\mathbf{n}_{\omega} \sim \mathcal{N}(0, \sigma_{\omega}^2), \mathbf{n}_a \sim \mathcal{N}(0, \sigma_a^2). \tag{11}$$

2.3 METRICS FOR TRAJECTORY COMPARISON

The metrics we used are introduced in Sturm et al. [105], which includes relative pose error (RPE) and absolute trajectory error (ATE). Assume that we have two trajectories: one is the ground truth trajectory, which includes a sequence of poses $\tilde{\mathbf{T}}_1, \cdots, \tilde{\mathbf{T}}_N \in SE(3)$, where N is the total number of the poses. The other is the estimated trajectory and includes $\mathbf{T}_1, \cdots, \mathbf{T}_N \in SE(3)$.

The RPE represents the drift of a trajectory over a fixed time interval δt. Here we choose the δt as the time interval between two consecutive frames, then at the i-th point in time, the relative pose error E_i is:

$$(\mathbf{T}_i^{-1}\mathbf{T}_{i+1})^{-1}(\tilde{\mathbf{T}}_i^{-1}\tilde{\mathbf{T}}_{i+1}), \tag{12}$$

and we compute the root mean squared error (RMSE):

$$RMSE(E_{1:N}) = \left(\frac{1}{N-1}\sum_{i=1}^{N-1}||E_i||^2\right)^{1/2}. \tag{13}$$

Compared to the RPE, the ATE is used to evaluate the global performance and consistency of the estimated trajectory. To compute the ATE, first, we need to find the rigid-body transformation $\hat{\mathbf{T}} \in SE(3)$ that aligns the estimated trajectory $\tilde{\mathbf{T}}_{1:N}$ to the frame of the ground truth trajectory $T_{1:N}$. Then the absolute trajectory error at $i-th$ timestamp is computed by: $\mathbf{T}_i^{-1}\hat{\mathbf{T}}\tilde{\mathbf{T}}_i$. Similarly, we compute the RMSE of the absolute translation:

$$RMSE(F_{1:N}) = \left(\frac{1}{N}\sum_{i=1}^{N}||trans(\mathbf{T}_i^{-1}\hat{\mathbf{T}}\tilde{\mathbf{T}}_i)||^2\right)^{1/2}, \tag{14}$$

where $trans(\mathbf{T}_i)$ extracts the translational component of the pose.

2.4 FACTOR GRAPH IN SLAM

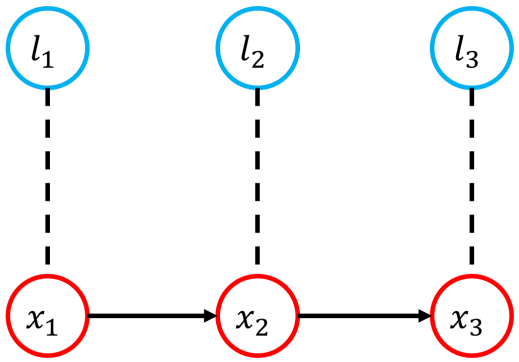


Figure 3: A toy SLAM example with three landmarks (blue) and three camera poses (red). The robot motion is indicated with an arrow and a dashed line represents a measurement.

Factor graphs have been widely used to represent a complex real-world SLAM problem [18, 43, 126]. We show a simple toy SLAM problem where a robot navigate to three poses $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3$ and observes three map points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ in Figure 3. Here, we assume that an absolute measurement $\mathbf{z}_1$ to the initial pose $\mathbf{T}_1$ is given.

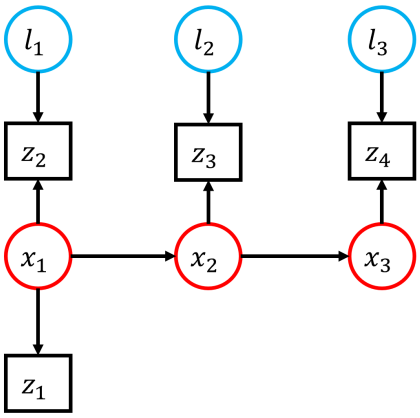


Figure 4: Bayes network for the SLAM problem shown in Figure 3. We illustrate measurement with square boxes.

This SLAM problem can be naturally presented as a Bayes Network (Figure 4). We denote all unknown variables as $\mathbf{X} = \{\mathbf{T}_1, \cdots, \mathbf{T}_3, \mathbf{x}_1, \cdots, \mathbf{x}_3\}$ and all measurements as $\mathbf{Z} = \{\mathbf{z}_1, \cdots, \mathbf{z}_3\}$. In SLAM, we need to find the optimal $\mathbf{X}$ when given a set of measurements $\mathbf{Z}$, which means to find the maximum a posteriori (MAP) estimation $\mathbf{X}^{MAP}$:

$$\mathbf{X}^{MAP} = \underset{\mathbf{X}}{\operatorname{argmax}}\, P(\mathbf{X}|\mathbf{Z}), \tag{15}$$

where $P(\mathbf{X}|\mathbf{Z})$ can be factorised as:

$$\begin{aligned} P(\mathbf{X}|\mathbf{Z}) \propto\ & p(\mathbf{T}_1)p(\mathbf{T}_2)p(\mathbf{T}_3) && (16)\\ & \times p(\mathbf{x}_1)p(\mathbf{x}_2)p(\mathbf{x}_3) && (17)\\ & \times p(\mathbf{z}_1|\mathbf{T}_1) && (18)\\ & \times p(\mathbf{z}_2|\mathbf{T}_1, \mathbf{x}_1)p(\mathbf{z}_3|\mathbf{T}_2, \mathbf{x}_2)p(\mathbf{z}_4|\mathbf{T}_3, \mathbf{x}_3). && (19) \end{aligned}$$

This factorisation can be explicitly expressed as a factor graph (Figure 5). A factor graph $F = (\mathcal{U}, \mathcal{V}, E)$ has two types of vertices: factors $\phi_i \in \mathcal{U}$ and variables $\mathbf{X}_j \in \mathcal{V}$. All edges $e_{ij} \in E$ connect one factor and another variable vertex. Each factor of the factorisation of $P(\mathbf{X}|\mathbf{Z})$ results in a factor vertex ϕ_i

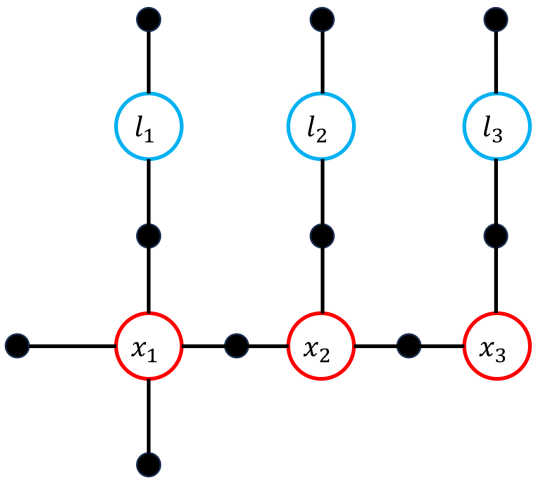


Figure 5: Factor graph of the Bayes network from Figure 4 based on the measurement $\mathbf{Z}$.

and $mathcalN(\phi_i)$ is the set of variables related to ϕ_i. Consequently, every variable vertex $\mathbf{X}_j \in \mathcal{N}(\phi_i)$ is adjacent to phi_i.

We assume that each factor ϕ_i can be modelled by a Gaussian noise with the variance Σ_i:

$$\phi_i \propto exp\{-\frac{1}{2}||h_i(\mathcal{N}(\phi_i)) - \mathbf{z}_{\phi_i}||^2_{\Sigma_i}\}, \tag{20}$$

where $h_i(\cdot)$ denotes the measurement function and $\mathbf{z}_{\phi_i}$ is the measurement of the factor. By taking negative log, Equation (15) can, therefore, be rewritten as:

$$\mathbf{X}^{MAP} = \underset{\mathbf{X}}{\operatorname{argmin}} \sum_i ||h_i(\mathcal{N}(\phi_i)) - \mathbf{z}_{\phi_i}||^2_{\Sigma_i}. \tag{21}$$

Consequently, with the help of factor graphs, we can format a SLAM problem with a non-linear least-square problem [69].

3

LITERATURE REVIEW

3.1 SLAM IN STATIC ENVIRONMENTS

Static simultaneous localisation and mapping (SLAM) methods assume that the whole environment is static. Although this assumption can be violated in real robotics applications, these methods achieve impressive performance to handle agile camera motion and robustness in large-scale environments. We will first introduce the visual SLAM methods based on feature points and traditional dense SLAM methods. We will then introduce planar SLAM methods which use high-level entities, such as lines and planes. Last, we introduce SLAM methods that fuse different sensors.

3.1.1 *Visual SLAM Based on Feature Points*

Initially, probabilistic filters play a significant role in SLAM methods based on fearture points and achieve promising results. Based on a extended Kalman filter (EKF) [41], MonoSLAM [17] proposes an impressive real-time SLAM system with a free-moving monocular camera. Huang et al. [36] proposes an unscented Kalman filter (UKF)-based SLAM system that is robust to large sensor noise and outperforms EKF-based methods in terms of accuracy and consistency. By coupling a particle filter to an UKF, Pupilli et al. [73] introduces a robust camera tracking system that handles unpredictable camera movements, which increases the system's reliability. Nevertheless, the key limitation of these methods is that processing consecutive frames only brings a limited amount of new information and they are unable to mitigate the accumulation of drift.

Compared to filter-based SLAM methods, keyframe-based SLAM methods are able to perform bundle adjustment (BA) optimisation based on keyframes which use computational resources more efficiently. Given multiple images observing a set of 3-D map points, BA can simultaneously optimise the coordinate of map points and estimate relative transformations of different image

frames [116]. Strasdat et al. [101] demonstrated that given the same computational resources, keyframe-based SLAM methods outperform filter-based SLAM methods in terms of accuracy.

Klein et al. [48] proposes a novel keyframe-based SLAM pipeline, parallel tracking and mapping (PTAM), which implements the parallelisation of the tracking and mapping processes. Importantly, it differentiates the concepts of front-end and back-end in visual SLAM: only the front-end is necessary to respond to images at a real-time speed. While the optimisation in the back-end can be processed at a slower rate and updated when new keyframes are inserted. However, PTAM is limited in scenes with a small size and tends to lose camera tracking.

The more recent keyframe-based method ORB-SLAM [60, 61] is one of the most versatile and robust feature-based visual SLAM systems and it outperforms previous SLAM systems in many aspects.

First, it supports various types of cameras, including monocular, stereo and depth cameras, which enables the method to work in both indoor and outdoor environments. In addition, the whole system is based on ORB features [83], including the visual odometry (VO) and loop detection. Compared to SIFT [67] or SURF [4] which require high computational resources, ORB can be estimated at a real-time speed. It also shows better scale and rotational invariance than computationally efficient features like Harris corner detector [31]. Importantly, descriptors provided by ORB features allow robots to relocalise themselves when moving in a large-scale environment. This is achieved by loading an ORB vocabulary file beforehand to detect loop candidates.

Inspired by the pipeline of PTAM, ORB-SLAM is based on three threads: 1) a *tracking* thread which estimates initial relative poses between two consecutive frames at a real-time speed; 2) a *local mapping* thread which conducts local BA based on a novel co-visibility graph; 3) a *loop closing* thread which detects loop candidates and performs full BA to correct loops. Consequently, ORB-SLAM is able to maintain the global consistency of maps and reduce the accumulated drift of cameras.

ORBSLAM-Atlas [19] extends ORB-SLAM [61] with a multi-map system. The advantage of this system is that when the system loses the track of camera, it creates a new map and localise the camera in the new map. Each new map is treated as a submap and will be fused with all previous submaps when loop candidates are detected. In contrast, the previous ORB-SLAM system

can only restart the camera tracking after camera relocalisation. Importantly, by managing a DBoW [26] library of keyframes, ORBSLAM-Atlas is able to seamlessly fuse two sub-maps into a single sub-map when they share a common region.

Despite impressive accuracy and performance in static environments, feature-based static SLAM methods are unable to handle dynamic objects.

3.1.2 *Dense Visual SLAM*

Compared to feature-based SLAM methods, dense visual SLAM is able to densely reconstruct the environment online while localising the camera. Therefore, they have a wider range of applications, such as augmented reality (AR) or virtual reality (VR).

As an early example of dense SLAM methods, DTAM [66] adopts a simple pipeline to track the camera and reconstruct the environments. Given a dense model, it aligns the current RGB image to estimate the camera pose and fuses the image to refine the model once the camera pose is estimated. Based on more recent depth cameras, KinectFusion [65] proposed a real-time dense SLAM approach. Unlike frame-to-frame alignment which is widely used in monocular SLAM, it maintains a model of the whole environment, which can be used to align the current image frame. The disadvantage of KinectFusion is that the computational complexity is proportional to the volume of reconstructed scenes and cannot, therefore, map a large-scale environment. Kintinuous [122], however, extends KinectFusion to an unbounded spatial scale so that the dense SLAM algorithm can be used in large outdoor situations. Both Kintinuous and KinectFusion used a truncated signed distance field (TSDF) as the representation of the map. This data structure can be readily parallel processed by GPU. Nevertheless, it limits the deformation ability of the map.

ElasticFusion [124] uses surface elements (surfel) to represent the map and achieves high accuracy in indoor applications. It produces dense surfel-based maps with global consistency without the help of a pose graph. This is achieved by jointly optimising photometric and depth errors and only using recent frames within a sliding window for camera tracking. The camera poses can also be refined by deforming the dense surfel-based map. Concretely, instead of

applying a rigid transformation to the surfel-based map, ElasticFusion applies a non-rigid deformation which changes the shape of map.

BundleFusion [16] manages to maintain a consistent global map and enables robust camera tracking in large indoor scenarios. To achieve this, it implements a novel pipeline that can hierarchically align global camera pose with a combination of sparse and dense features. In contrast to ElasticFusion, BAD-SLAM [92] proposes a novel direct BA in an online dense RGB-D SLAM system. It projects surfels into keyframes and minimises geometric and photometric errors. During the direct BA, both camera poses and the surfel-based map are refined.

In addition to traditional explicit map representation, like surfels or voxels, neural radiance field (NeRF) [59] becomes a popular implicit dense map representation. iMAP [106] is the first dense RGB-D SLAM system that represents the dense map with one single multilayer perception (MLP) layer. Nice-SLAM [135] introduces a MLP-based hierarchical scene representation that consists of coarse-to-fine geometric models. Compared to explicit scene representation, NeRF-based dense SLAM methods can extrapolate mapping into unobserved areas and inpaint small holes caused by occlusion.

3.1.3 *Visual Planar SLAM*

Apart from sparse feature points, high-level features like lines or planes can also be used in visual SLAM systems. In man-made indoor environments, planes are very common and a large number of man-made objects are comprised of planar surfaces, such as walls or boxes.

Given an RGB-D camera, planes can be directly extracted from the depth map [23, 71]. Point-plane SLAM [110] tracks the camera based on the combination of planes and feature points. It also represents the map with both point and plane landmarks to make the dense map more compact than traditional voxel-based maps. Similarly, the dense map is compressed in dense planar SLAM [88] by representing the non-planar map regions with surfels and the planar regions with planes. Kaess et al. [42] uses infinite planes as landmarks in the pose graph SLAM problem by introducing a novel homogeneous plane parametrisation, which reduces the complexity of optimisation. Ming et al. [35]

reconstructs a global dense planar map online with only a single CPU based on keyframe management.

With a monocular camera, DPPTAM [15] observes that image areas with similar RGB values have low photometric gradients and are mostly planar regions. Therefore, DPPTAM represents high-gradient regions with points while low-gradient regions with planes. This is achieved by segmenting low-gradient regions with super-pixels and actively searching planes based on the normal directions of these super-pixels. TT-SLAM [120] also extracts planes by merging super-pixels extracted monocular images and conducts non-linear optimisation to solve camera poses and plane parameters. In contrast, Pop-up SLAM [130] applies a convolutional neural network (CNN) to directly detect planes from monocular images and refine plane boundaries with line extraction, which enables robust SLAM in texture-less indoor environments.

In addition to plane representations, some planar SLAM methods assume a Manhattan world (MW) in structured environments, where scenes are constructed on a Cartesian grid. Therefore, planes in a MW can be divided into three sets that are perpendicular to each other. Li et al. [52] estimate a MW frame from the extract planes and, therefore, reduce the accumulated drift of the camera. This is achieved by decoupling the translation and rotation estimation of the camera pose. The camera translation is estimated by combining points, lines and planes, while the camera rotation is only estimated by minimising the residual from lines and planes. The MW frame is used to refine the camera rotation to reduce long-term rotational drift.

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, such as boxes, are transported or manipulated by humans or robots.

3.1.4 *SLAM with Sensor Fusion*

Visual SLAM systems can be fused with other proprioceptive sensors, such as inertial measurement unit (IMU), robot odometry or kinematic, to increase the robustness of localisation.

The visual inertial navigation system (VINS) methods fuse visual sensors and an IMU for more accurate localisation. IMUs can be either coupled with feature-

based visual odometry [8, 50, 74] or direct visual odometry [117]. OKVIS uses a sliding window and marginalised out previous useless frames. VINS-Mono [74] combines the measurements from a monocular camera and an IMU in a tightly-coupled manner and proposes an accurate visual-inertial (VI) SLAM system in large-scale environments. DM-VIO [104] delays the marginalisation of previous frames for a certain period to retain the prior information and improves the robustness of the system. ORB-SLAM3 [11] is a real-time tightly-coupled VI SLAM system that maintains multiple maps simultaneously and can reuse all previous information from a co-visible graph when the camera revisits a place. Despite assuming the environment is static, these methods have achieved accurate localisation when the camera view is fully covered for a short period [93]. However, this occlusion is caused by featureless objects, such as a dark tube, instead of large or closely moving dynamic objects with rich features.

Additionally, robot proprioception can also be used for localising robots like quadrupeds or wheeled robots. Fallon et al.[21] introduced a hierarchical process to combine three distinct modalities. These three modalities are vision, IMU and leg odometry respectively. It first coupled leg odometry with IMU, then merged them together with LiDAR. Experiments show that this hierarchical process can reduce the global drift of a humanoid's pelvis states.

Exploiting the high accuracy of robot manipulator kinematics, ARM-SLAM coupled the more accurate kinematic information with the visual input and improved the performance of visual odometry [49]. Similar to the hierarchical structure in Fallon et al. [21], Scona et al. [95] also fused humanoid proprioception into a visual SLAM approach. Additionally, this method weights the proprioception according to the reliance on RGB-D image alignment, and trusts the kinematic and inertial modes more when visual odometry has poor results. Instead of hierarchically fusing different modalities, KO-Fusion [34] directly coupled all three sensors together and summed the errors from kinematic, wheel odometry and dense visual odometry up to one loss function. The dense visual odometry is based on ElasticFusion, and the results proved that KO-Fusion outperformed ElasticFusion in terms of absolute trajectory error (ATE) and produced a better map reconstruction. Similarly, VILENS [125] tightly coupled all three sensors, but used a single factor graph. A sliding window is implemented, and the historic state and unseen landmarks are marginalized

out. Detailed experiments demonstrated that VILENS outperforms both VINS methods and kinematic inertial (KI) systems in all listed datasets.

By merging kinematic, inertial and vision together, the state-of-the-art robot state estimation methods can achieve accurate robot tracking in the real static world. However, proprioceptive state estimation alone cannot self-correct the estimated robot pose and, therefore, the camera drift accumulates. Vision can help reduce the drift of robot proprioception and therefore plays a critical role in those methods.

Nevertheless, in dynamic environments, current static visual SLAM methods are unable to detect dynamic objects. Consequently, dynamic objects can be mapped into the static background reconstruction and cause failures in camera tracking.

3.2 SLAM IN DYNAMIC ENVIRONMENTS

In literature, dynamic SLAM methods can be divided into three categories. The first category of methods can handle unmodeled dynamic objects based on visual sensors only. The second category of methods also relies on visual sensors only, but assumes object classes are pre-defined. Last, visual sensors can be fused with other proprioceptive sensors to help detect dynamic objects.

3.2.1 *Robust Visual SLAM with Motion Segmentation*

Robust visual SLAM systems can detect unmodeled dynamic objects in complex environments by differentiating dynamic objects from the static background based on their different motions. One category of methods not only separates dynamic objects from the static background but also provides separate segmentation of independently-moving objects. The other category of methods detects the whole dynamic areas as a whole.

3.2.1.1 *Multimotion Segmentation Method*

Multimotion segmentation methods that separate the visual input into multiple independently-moving objects have a rich history. Torr et al. [113] extract multiple rigid clusters from two perspectives of a 3D point cloud that consists of multiple objects with different rigid motions. By adopting a 3×3 fundamental

matrix [57] to model the relative transformation of a rigidly-moving 3D point cloud in the camera frame, the method does not require the camera to be calibrated in advance [22, 32]. To achieve it, Torr et al. [113] first find matches between two point clouds taken from two camera positions. It then iteratively applies random sample consensus (RANSAC) [24] to fit a cluster that can be explained by a transformation matrix. After initial cluster generation, it prunes clusters that are repetitive to others or have too few points. It also merges clusters with a close transformation matrix.

Torr et al. [115] further extend the previous motion clustering approach [113] with three image views [115]. Instead of restricting the motion model to a fundamental matrix, Torr et al. [112] propose estimate clusters under the degenerate motion models of a fundamental matrix, such as affine transformations and projective transformations [114]. It also uses geometrically robust information criterion (GRIC) to automatically select the best motion model that fits a segmented cluster. In addition to RANSAC, the normalised cut can also be used to segment an image sequence into partitions that represent different rigidly-moving objects [97]. Schindler et al. [89] introduce Monte-Carlo sampling for motion segmentation from two-view images, which is then extended to handle multiple image views with different camera models [90, 91]. Ozden et al. [70] consider multimotion segmentation and tracking task in practical scenarios when the number of dynamic objects changes. It's able to merge a moving object into the static background when it stops moving and separate one cluster into two when a part of the cluster starts to move differently.

More recently, Sabzevari et al. [86] proposed a theoretical framework based on a multi-RANSAC scheme for simultaneous multibody motion segmentation and reconstruction. Concretely, it repeatedly samples a point subset of k feature points among N feature points according to a certain distribution, where N is the total number of feature points and k is a hyper-parameter. Each point subset is then fitted with a standard SfM model, and the reprojection error of the estimated structure from these k feature points is calculated. The sampling process will repeat until the reprojection error is below a threshold because a low reprojection error suggests that these k feature points have a close movement. Then, this point subset is removed from the N feature points, and the method repeats to sample another point subset among the remaining feature points. The process, which first samples points and then removes

these points, will be repeated until all the feature points have been removed. Consequently, the N feature points are segmented into subsets, and points in each subset agree on a unique motion. These subsets serve as an initialisation of the camera motions and structure of each motion, which are further refined iteratively by factorization.

Based on the previous framework [86], Sabzevari et al. [87] further used ego-motion constraint to help multibody motion segmentation. Specifically, the camera is mounted on a vehicle, and the movement of the vehicle is restricted on a plane. Additionally, they assume that there is an instantaneous center of rotation for the vehicle. The specific movement pattern of vehicles is used to segment the static component apart from all the feature points. Then a similar framework in [86] is applied to the remaining feature points. Co-Fusion [84] proposed a pipeline that can simultaneously track and reconstruct multiple objects. During the segmentation stage, Co-Fusion detects dynamic objects as outliers. However, in contrast to StaticFusion, Co-Fusion models the region of outliers as a new object. The criteria for adding a new model is based on the connectivity of these outliers. Specifically, the outliers are recognised as a new object when the percentage of the connected outliers is larger than 3% of the image. For each input RGB-D image, the tracking phase uses the segmentation from the last image and optimises each object's pose individually. Then the estimated pose was used in the following motion segmentation. Semantic segmentation was also implemented, but only one of the two segmentation methods can be used in the pipeline. After initialising segmentation, the object pose and segmentation are optimised separately.

Judd et al. [39] proposed the first approach that is capable of estimating the full pose of each rigid body in the visual input. Unlike MBSfM which requires all images in advance, Judd et al. [39] applied similar multi-layer RANSAC between two consecutive frames.

Concretely, it first extracts feature points for each frame and finds correspondences between two consecutive frames. Then it randomly chooses three pairs of feature points, calculates the transformations based on the three pairs and counts the number of inliers. This random sample process is repeated a certain number of times, and the transformations with the largest number of inliers are denoted as a trajectory hypothesis. All feature pairs that are inliers of this trajectory hypothesis are removed, and new trajectory hypotheses are repeatedly generated from the remaining feature pairs until all remaining

feature pairs are removed. Additionally, the trajectory hypothesis that has the largest number of inliers is treated as camera ego-motion. This is based on the assumption that the static background is larger than any other rigid body.

Once MVO [39] segments the visual input into multiple rigid bodies with different motions, it chooses the largest rigid body as the static background. Consequently, it is unable to handle dynamic large occlusion.

3.2.1.2 *Dynamic Object Removal Methods*

Sun et al. [108] proposed a dynamic object removal method that serves as a pre-procession for the input of SLAM algorithms. This pre-procession stage can be divided into three steps. Firstly, it takes two consecutive images to estimate camera ego-motion. Specifically, after extracting features from RGB images, a RANSAC algorithm is used to select a homograph transformation that minimizes the reprojection error between the two images. The second step is to project the current image to the previous image frame and then calculate image differences. This image differences then become a motion mask: a higher difference means a higher probability to be dynamic. The third step is to use particle filter and maximum *a posteriori* estimator respectively to refine the results of the first step. Although not stated explicitly, the prerequisite of the first step is that the major part of the image is static. Otherwise, the RANSAC approach would treat the dynamic component as static and therefore lose track of the camera pose.

Kim et al. [45] used a pre-computed transformation to get a difference map between two consecutive images. Then the difference map is used to weight per-pixel photometric error items. This method requires very accurate pre-computed transformations. It assumed that the visual odometry can at least work well in dynamic environments.

Sun et al. [109] extended their previous work [108] to scenarios with multi-cluster dynamics. In addition, instead of directly calculating the difference after reprojection, it used EpicFlow [81] to produce correspondences between two images via edge detection in optical flow. Furthermore, RANSAC was replaced with Least-Median-of-Square which is theoretically more robust to a large percentage of outliers. Nevertheless, this method still assumed that planes in the RGB-D input belong to the static background. This is not true when robots can interact with environments. In addition, once a dynamic object is detected,

it will be accumulated into a set of object models. This means if a dynamic object becomes static, this object would still be treated as dynamic, therefore being removed from the static background.

Li et al. [51] introduced a static pixel/point weighting method to represent the probability of a point is static. This method is different from previous methods that classify image pixels or points as either absolutely static or dynamic. Similar to EpicFlow [81], edges are extracted in the visual input. The key component of this method is static weight estimation. It firstly calculated the reprojection error between the current frame and key-frame, then utilized the Student's t-distribution to model the probabilistic density distribution of the error. Similar to RANSAC, the static weight estimation is based on the assumption that the dynamic component is smaller than the static component.

As an extension to ElasticFusion, StaticFusion [94] applied a similar static point weighting method as [51]. However, rather than estimate the point weight after aligning the consecutive images, StaticFusion jointly optimised the transformation and static/dynamic segmentation in one loss function. Concretely, for each pixel in the previous image, the error or residual is weighted by the static point weight when calculating the reprojection error with respect to the current image. Then the average of the residual is treated as a threshold for the classification of points. By adding this threshold as a penalty term in the loss function, the algorithm encourages the points which have a lower residual error to a higher static probability. To reduce computing complexity, StaticFusion used a k-nearest neighbours (kNN) method as a pre-procession for every image so that the image can be processed at a cluster level. For each image, StaticFusion calculates the similarity between different clusters and uses this similarity in the loss function, forcing similar clusters to have close static weights. In the implementation of solver, StaticFusion decoupled the segmentation and transformation, but they are tightly coupled in the objective.

Compared with StaticFusion, FlowFusion [133] used a novel optical flow residual in addition to intensity and depth residuals. To obtain the optical flow residual, FlowFusion firstly used extracted optical flows from two consecutive images via Pwc-net [107]. The optical flow is defined as the per-pixel movement on the image coordinates. Secondly, a robust visual odometry is used to generate an initial transformation between the two consecutive images. In static environments, the optical flows in images directly come from the camera

motion, while in dynamic environments, the optical flows come from the combination of the camera motion and object motion. Lastly, FlowFusion calculated the difference between the whole optical flows and the optical flows that come from the estimated camera motion. This difference is defined as the optical flow residual and ideally is close to optical flows that come from the movement of dynamic objects.

3.2.2 *Visual SLAM with Pre-defined Dynamic Objects*

In the real world, dynamic objects often have distinct semantic labels, such as bikes, cars or humans. Deep learning methods have achieved impressive improvements in object recognition and segmentation, such as MOTS [118], YOLO [78] and Mask-RCNN [33]. YOLOv4 [9], which is one of the state-of-the-art object detection methods based on bounding boxes, can accurately detect pre-defined objects at a high frequency (nearly 65 FPS). In addition, multi-object dense segmentation methods, such as Mask-RCNN [33], can provide accurate semantic segmentation, therefore supporting robot localisation when dynamic objects are included in the training set [85, 102].

PoseFusion [132] assumed that the moving objects in the environment are highly likely to be humans. Therefore, OpenPose [13] was used in PoseFusion to segment humans apart from the environment. The segmentation was then used as prior information in Min-Cut [30] to remove the humans and recover the static background. Similarly, DS-SLAM [131] used machine learning methods to recognise objects of nearly 20 different classes.

Compared to PoseFusion and DS-SLAM, MaskFusion [85] made a better use of semantic information. Specifically, it managed to use a semantic segmentation method that provides less accurate object boundaries and is less frequent than camera rate to segment the visual input at a real-time speed. To achieve this, MaskFusion applied geometric segmentation to every input image. Concretely, MaskFusion extract boundaries of different object based on the surface normal directions and the depth map discontinuities. When semantic labels are available, the geometric segmentation can be fused with semantic segmentaion to further improve the accuracy of dynamic object segmentation.

After removing dynamic objects by deep learning methods, DynaSLAM [6] further combined multi-view geometry to refine the static/dynamic segment-

ation. Additionally, it introduced a background inpainting method that uses previous images to inpaint the holes in the model, which are cause by the removal of dynamic objects.

Instead of treating all semantic labels equivalently, Cheng et al. [14, 128] assigns weights between 0 to 1 to different semantic labels. For example, humans have a higher possibility to be dynamic than desks, so "human" is assigned a weight of 1 and "desk" is assigned a weight of 0.1. After the semantic segmentation via a deep learning method, they extracted features from the background and the area with a low weight. This is followed by camera pose estimation using feature matching.

Based on Mask R-CNN, EM-Fusion [103] integrates object tracking and SLAM into a single expectation maximisation (EM) framework. ClusterVO [37] can track camera ego-motion and multiple rigidly moving clusters simultaneously by combining semantic bounding boxes and ORB features [83]. DynaSLAM II [5] further integrates the multi-object tracking (MOT) and SLAM into a tightly-coupled formulation to improve its performance on both problems.

However, all these methods require that the object detector is pre-trained on a dataset which includes these pre-defined objects or that the object model is provided in advance.

3.2.3 *Visual SLAM with Unmodeled Dynamic Objects*

To handle unmodeled dynamic objects, dynamic SLAM methods can either remove dynamic objects as outliers or estimate rigid bodies with different motions against the camera motion.

3.2.4 *Dynamic SLAM with Sensor Fusion*

Inspired by VINS algorithms, Kim et al [44] used an IMU to help robot localisation in dynamic environments. It uses estimated movements from an IMU to compensate for the camera movement before featured-based camera tracking. Specifically, after the compensation for camera movement, this method assumed that the movement between feature point pairs is mainly caused by dynamic objects. Therefore, a threshold is chosen to remove feature pairs

that have a large movement. The remaining feature pairs are treated as static and used for camera tracking. The camera motion is estimated after the static background is separated by the pose compensation of an IMU. Therefore, it has a high requirement for the accuracy of the IMU.

Qiu te al. [75] integrates semantic bounding boxes with VI SLAM systems and enables object 3-D motion tracking from 2-D regions of images. Similarly, Dynamic-VINS [54] refines 2D bounding boxes generated from YOLOv3 [79] and removes feature points of dynamic objects with a recourse-limited platform. Ren *et al*. [80] proposes a dense RGB-D-inertial SLAM system that can track and relocalise multiple dynamic objects with the aid of instance segmentation from Mask R-CNN [33].

3.3 SUMMARY

In static environments, visual SLAM methods have achieved promising accuracy even in large-scale complex environments and are robust to agile camera motions. With sensor fusion, static SLAM methods can further reduce long-term camera drift and enable more accurate camera localisation.

However, the static world assumption is often violated in the real world. To reduce the impact of dynamic objects in the environment, current dynamic visual SLAM methods either assume that the static component occupies a major proportion of the observed scene or that the dynamic objects have a semantic meaning. However, in real applications, unmodeled dynamic objects can also occupy a major part of the camera view. In addition, interactions between robots and the environment can introduce dynamics. These dynamic objects cannot, therefore, be handled by current dynamic SLAM methods.

4

DYNAMIC RGB-D SLAM WITH SINGLE LARGE OBJECT RECONSTRUCTION

In this chapter, we focus on enabling robots to localise accurately when the majority of camera view is occluded by a single large dynamic object. We propose a novel RGB-D SLAM approach that can simultaneously segment, track and reconstruct the static background and single large dynamic rigid object that can occlude major portions of the camera view.

Current state-of-the-art approaches [84, 85, 94, 103, 129] treat dynamic parts of a scene as outliers and are thus limited to small amount of changes in the scene, or rely on prior information for all objects in the scene to enable robust camera tracking. Here, we propose to treat all dynamic parts as one rigid body and simultaneously segment and track both static and dynamic components. We, therefore, enable simultaneous localisation and reconstruction of both the static background and rigid dynamic components in environments where dynamic objects cause large occlusion. We evaluate our approach on multiple challenging scenes with large dynamic occlusion. The evaluation demonstrates that our approach achieves better motion segmentation, localisation and mapping without requiring prior knowledge of the dynamic object's shape and appearance.

4.1 INTRODUCTION

Many robotics tasks, such as handling and transporting objects in an unmanned warehouse or collaborative manipulation [100], require a robot to localise against the static environment in which it moves while being robust to distractions from dynamic objects; as well as track the object which they manipulate. While these two problems have been previously addressed separately, only a few strands of work [39, 84] have attempted to solve these two problems together and track the camera and multiple objects at once.

In this work, we argue that localisation against the environment and tracking of objects are fundamentally the same problem, and that solving them

simultaneously reduces ambiguity about the scene and improves localisation in cases of large dynamic occlusions.

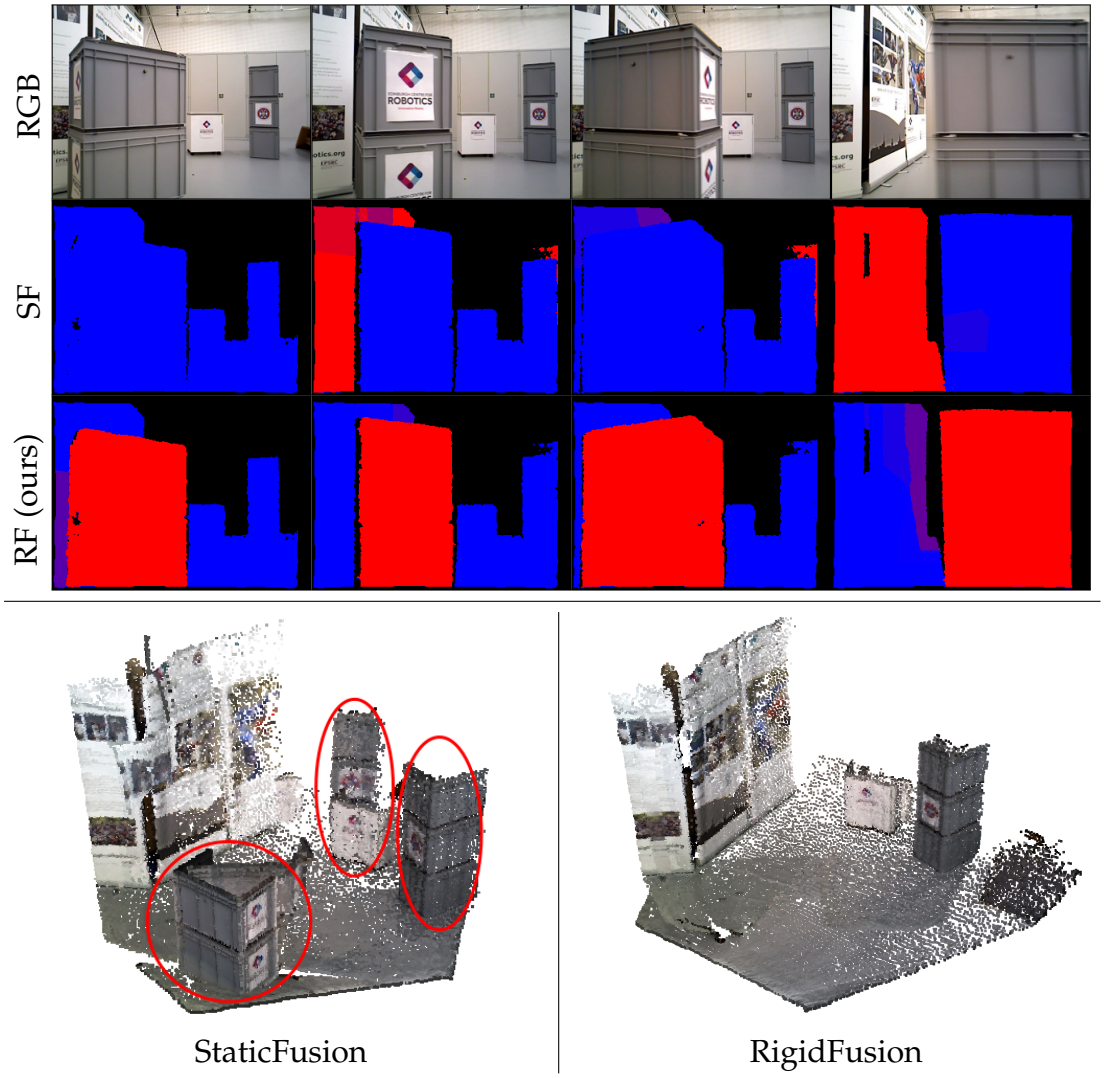


Figure 6: **Top:** Segmentation of a scene with one moving box into static (blue) and dynamic (red) segments. Indirect methods, such as StaticFusion (SF) [94], neglect dynamic parts or incorrectly treat them as static environment while our method, RigidFusion (RF), correctly segments the moving box as dynamic (red). **Bottom:** The reconstruction of the static map in SF contains the dynamic object (red circle) and multiple instances of the same static object (red ellipses), while RF correctly incorporates all static segments.

The core problem – separating the scene into segments of transformations induced by ego-motion and/or moving objects – is challenging due to several factors:

1. **Unknown environments**: Robots may not have prior information about the semantic meaning, 3D model or appearance of the dynamic objects and the background.

2. **Large occlusion**: Dynamic parts of images are often discarded for robust visual odometry; but in many scenarios, they can occlude the majority of a camera view, such as for large moving objects or when manipulated objects are close to the camera. This ambiguity leads to tracking failures where a dynamic object is classified as part of the static environment. This is in contrast to driving/flying, where ego-motion effects dominate.

3. **Mutual static and dynamic transition**: Manipulated objects can transition between static and dynamic with respect to the world at any time during manipulation. Maintaining these state transitions purely with visual odometry can be difficult.

To address all three aspects concurrently, we treat localisation and object tracking as an integrated problem and formalise both as modelling and tracking any rigid movement. Consequently, we achieve improved motion segmentation, localisation and mapping in dynamic environments with large occlusion (Figure 6). For this, we assume that the motions of both static and dynamic components are rigid transformations. These motions can be identified using tightly coupled motion priors from odometry and kinematics.

In summary, the work introduced in this chapter contributes:

1. a new pipeline to simultaneously segment, track and reconstruct the static background and one dynamic rigid body from RGB-D sequences, using motion priors with potential drift,

2. a dense SLAM method that is robust to large occlusions in the visual input (over 65%) without relying on an initialisation of the static and dynamic models,

3. a new RGB-D SLAM dataset[1] with dynamic objects that cause large occlusion in the scenes and ground truth trajectories.

4.2 OVERVIEW

We propose a pipeline that treats the dynamic component as a single rigid body and uses motion priors to segment the static and dynamic components. The

1 http://conferences.inf.ed.ac.uk/rigidfusion/

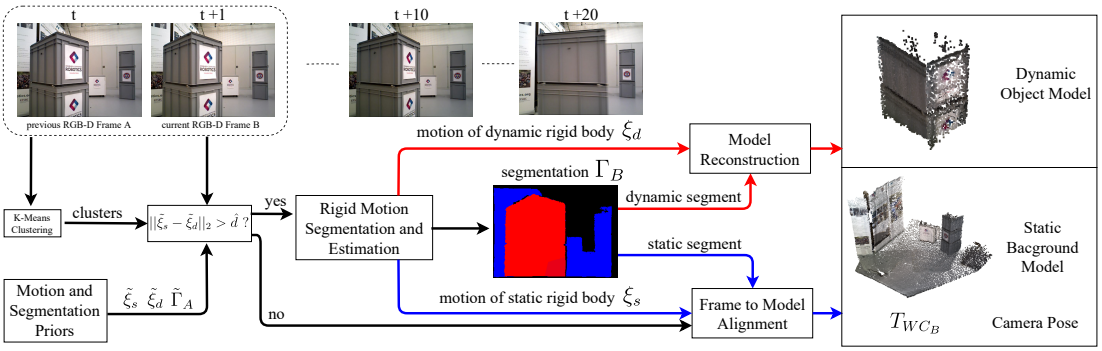


Figure 7: Our method processes two consecutive RGB-D frames (A, B), motion priors ($\tilde{\boldsymbol{\xi}}_s$, $\tilde{\boldsymbol{\xi}}_d$), and the previous cluster-wise segmentation ($\tilde{\boldsymbol{\Gamma}}_A$). We first detect whether the object is dynamic based on motion priors. We then jointly estimate the segmentation $\boldsymbol{\Gamma}_B$ and the rigid body motions $\boldsymbol{\xi}_s$ and $\boldsymbol{\xi}_d$ based on frame-to-frame alignment when the object moves. The segments are used to reconstruct the static environment and the dynamic object, and to localise camera using frame-to-model alignment.

segmentation is used to track the camera, and to reconstruct the background and object models.

The overview of our pipeline is illustrated in Figure 7. Our approach takes two consecutive RGB-D frames, static and dynamic motion priors, $\tilde{\boldsymbol{\xi}}_s, \tilde{\boldsymbol{\xi}}_d \in \mathfrak{se}(3)$, and the previous cluster-wise segmentation.

Similar to [94], each new intensity and depth image pair $(I, D) \in \mathbb{R}^{W \times H}$ is over-segmented into K geometric clusters by applying K-Means clustering [38]. Specifically, we first choose $K = 24$ and uniformly assigns K seeds to the depth map. We then iteratively cluster pixels to different seeds by finding the closest seed to each pixel and optimise the seed position based on the clusters. We hypothesise that each cluster is as rigid as possible, and each rigid body can be approximated by the combination of clusters. We also assign each cluster a score $\gamma_i \in [0, 1]$ which represents the probability that a cluster belongs to the static rigid body: $\gamma_i = 0$ stands for dynamic clusters while $\gamma_i = 1$ means static clusters. $\boldsymbol{\Gamma} = \{\gamma_0, \cdots, \gamma_{K-1}\}$ denotes the segmentation scores of all clusters.

If the difference between two motion priors $||\tilde{\boldsymbol{\xi}}_s - \tilde{\boldsymbol{\xi}}_d||_2$ is less than a threshold $\hat{d}$, we treat all clusters in the image as static and skip motion segmentation. Otherwise, we jointly optimise the cluster-wise segmentation of the current frame and relative motions $\boldsymbol{\xi}_s$ and $\boldsymbol{\xi}_d$ of the static and dynamic rigid bodies (Section 4.3).

Similar to StaticFusion, we compute the masked RGB-D images of both static and dynamic rigid bodies from the segmentation $\boldsymbol{\Gamma}_B$. These masked images

are used to reconstruct models of the background and dynamic object and to refine the estimated camera pose using frame-to-model alignment (Section 4.4).

We denote world-, camera-, and object-frames as F_W, F_C, F_O respectively (Figure 8). Similar to [34], we use $\mathbf{T}_{AB} \in SE(3)$ to transform homogeneous coordinates of a point in coordinate frame F_B to F_A. In image frame A, the camera and object poses are $\mathbf{T}_{WC_A}$ and $\mathbf{T}_{WO_A}$ respectively. Considering two image frames A and B, the relation between $\boldsymbol{\xi}_s$ and camera poses is: $\mathbf{T}(\boldsymbol{\xi}_s) = \mathbf{T}_{WC_A}^{-1}\mathbf{T}_{WC_B} = \mathbf{T}_{C_AC_B}$. The relation between $\boldsymbol{\xi}_d$, camera and object poses is: $\mathbf{T}(\boldsymbol{\xi}_d) = \mathbf{T}_{WC_A}^{-1}\mathbf{T}_{WO_A}\mathbf{T}_{WO_B}^{-1}\mathbf{T}_{WC_B} = \mathbf{T}_{C_AO_A}\mathbf{T}_{C_BO_B}^{-1}$. The motion priors $\tilde{\boldsymbol{\xi}}_s$ and $\tilde{\boldsymbol{\xi}}_d$ can be provided by proprioceptive sensors, such as wheel odometry and arm forward kinematics.

In this paper, the static motion prior $\tilde{\boldsymbol{\xi}}_s$ is computed either from wheel odometry or by adding simulated drift on camera ground truth trajectories. We generate $\tilde{\boldsymbol{\xi}}_d$ by simulating drift on object ground truth trajectories.

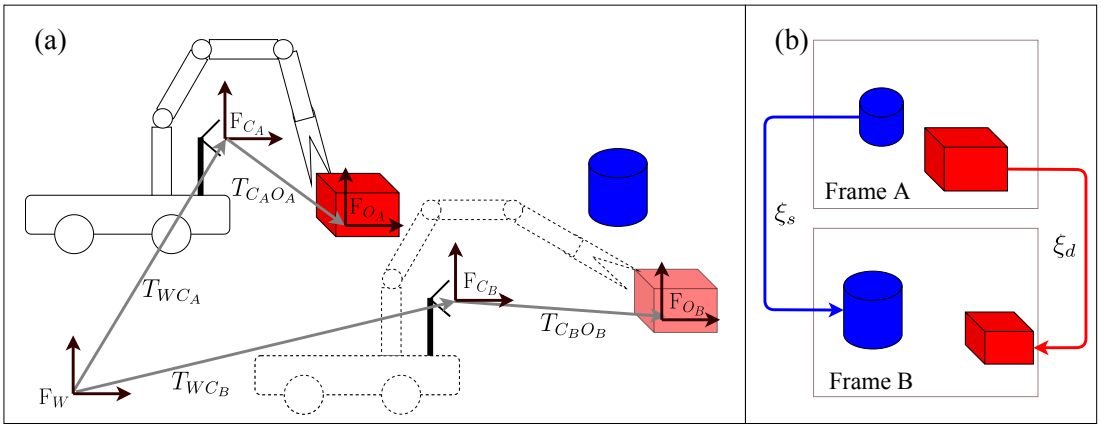


Figure 8: Relation between coordinate frames (F_W, F_C, F_O) and motions ($\boldsymbol{\xi}_s$, $\boldsymbol{\xi}_d$). (a) External camera view. A mobile manipulator simultaneously moves its base and manipulates an object (red box). The camera is fixed on the base. (b) Image view. For the static motion $\boldsymbol{\xi}_s$, we can compute the prior $\tilde{\boldsymbol{\xi}}_s$ from $\mathbf{T}_{WC}$, which can be acquired from wheel odometry. The dynamic motion prior $\tilde{\boldsymbol{\xi}}_s$ can be computed from $\mathbf{T}_{CO}$, which can be acquired from arm kinematics.

At the arrival of each RGB-D pair, we jointly segment and track both static and dynamic rigid bodies by minimising a combined cost that consists of four energy terms:

$$\begin{aligned} &\min_{\boldsymbol{\xi}_s, \boldsymbol{\xi}_d, \boldsymbol{\Gamma}} P_{robust}(\boldsymbol{\xi}_s, \boldsymbol{\Gamma}) + P_{robust}(\boldsymbol{\xi}_d, \mathbf{1} - \boldsymbol{\Gamma}) + R_{seg}(\boldsymbol{\xi}_d, \boldsymbol{\Gamma}) + R_{motion}(\boldsymbol{\xi}_s, \boldsymbol{\xi}_d) \\ &\text{s.t. } \gamma_i \in [0, 1] \quad \forall\, i\,, \end{aligned} \tag{22}$$

where $\boldsymbol{\Gamma}$ represents the scores of all clusters and $\mathbf{1} - \boldsymbol{\Gamma} \coloneqq \{1 - \gamma_0, \cdots, 1 - \gamma_{K-1}\}$. Specifically, the first two robust residual terms align the static and dynamic rigid bodies respectively. The third term $R_{seg}(\boldsymbol{\xi}_d, \boldsymbol{\Gamma})$ adds regularisation on both the spatial and time distribution of scores $\boldsymbol{\Gamma}$ to maintain the smoothness of segmentation. The last term $R_{motion}(\boldsymbol{\xi}_s, \boldsymbol{\xi}_d)$ applies regularisation on motions $\boldsymbol{\xi}_s, \boldsymbol{\xi}_d$ using motion priors $\tilde{\boldsymbol{\xi}}_s$, $\tilde{\boldsymbol{\xi}}_d$.

4.3.1 *Rigid Body Motion Estimation*

Following previous RGB-D SLAM methods [94, 124], in static environments, the relative camera pose between two consecutive image frames $i-1$ and i is estimated by minimising the intensity and depth residuals between the image pairs (I_{i-1}, D_{i-1}) and (I_i, D_i). At a pixel $\mathbf{u}$ in frame i, the intensity residuals $r_I^{\mathbf{u}}$ and depth residuals $r_D^{\mathbf{u}}$ with respect to frame $i-1$ under the camera motion $\mathbf{T}$ are defined as:

$$r_I^{\mathbf{u}}(\mathbf{T}) = I_{i-1}\left(\mathcal{W}(\mathbf{u}, \mathbf{T})\right) - I_i\left(\mathbf{u}\right) \tag{23}$$

$$r_D^{\mathbf{u}}(\mathbf{T}) = D_{i-1}\left(\mathcal{W}(\mathbf{u}, \mathbf{T})\right) - |\mathbf{T}\pi^{-1}(\mathbf{u}, D_i\left(\mathbf{u}\right))|_z\,, \tag{24}$$

where the image warping function W is given by:

$$\mathcal{W}(\mathbf{u}, \mathbf{T}) = \pi\left(\mathbf{T}\pi^{-1}(\mathbf{u}, D_i(\mathbf{u}))\right). \tag{25}$$

where $|\cdot|_z$ indicates the z-coordinate of a 3D point and $D(\mathbf{u})$ is the depth of pixel $\mathbf{u}$. The homogeneous transformation matrix $\mathbf{T} \in SE(3)$ is computed from its Lie algebra $\boldsymbol{\xi} \in \mathfrak{se}(3)$. The projection function $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ projects 3D points onto the image plane using the camera intrinsic matrix.

According to StaticFusion, given the static scores $\mathbf{\Gamma}$ of a rigid body, we can estimate the relative motion of this rigid body by applying the scores to weight residuals. Consequently, only pixels that belong to the rigid body have a high contribution:

$$P_{robust}(\boldsymbol{\xi}, \mathbf{\Gamma}) = \sum_{\mathbf{u} \in U} \gamma(\mathbf{u}) [F(\alpha_I w_I^{\mathbf{u}} r_I^{\mathbf{u}}(exp(\boldsymbol{\xi}))) + F(w_D^{\mathbf{u}} r_D^{\mathbf{u}}(exp(\boldsymbol{\xi})))] \,, \quad (26)$$

where U is the set of image pixels with valid depth reading in one image. The function $exp(\boldsymbol{\xi})$ is the matrix exponential map for Lie group $SE(3)$. $\gamma(\mathbf{u})$ represents the static weight of the cluster that contains the pixel $\mathbf{u}$. α_I is a scale parameter to weight photometric residuals so that they are comparable to depth residuals. The parameters w_I and w_D are computed according to the photometric and depth measurement noise. As in [94], we use the Cauchy robust penalty

$$F(r) = \frac{c^2}{2} log(1 + (\frac{r^2}{c})) \quad (27)$$

to robustly control the minimisation of residuals, where c is the inflection point of $F(r)$.

The novelty of our approach is that in equation 22, we treat the dynamic component as another rigid body with a different motion, where $\mathbf{\Gamma}$ and $\mathbf{1} - \mathbf{\Gamma}$ represents the scores of the static and dynamic rigid body respectively. To simultaneously segment and track the two rigid bodies, we further encourage segmentation smoothness and use tightly coupled motion priors.

4.3.2 *Segmentation Smoothness*

First, to maintain spatial smoothness, we use the regularisation term used in StaticFusion to penalise the score difference between adjacent clusters:

$$R_{spatial}(\mathbf{\Gamma}) = \sum_{(k_1, k_2) \in V} (\gamma_{k_1} - \gamma_{k_2})^2, \quad (28)$$

where V is the connectivity graph for clusters [94] in the i-th keyframe and $(k_1, k_2) \in V_i$ represents the k_1-th and k_2-th clusters of keyframe i are connected in space to each other.

Furthermore, we consider the physical constraint that pixels that belong to the dynamic rigid body at the previous frame are likely to be dynamic at the current frame. Therefore, we use the segmentation result from the previous frame as segmentation prior to encourage segmentation smoothness over time:

$$R_{time}(\boldsymbol{\xi}_d, \boldsymbol{\Gamma}) = \sum_{i=1}^{K} (\gamma_i - \tilde{\gamma}_i(\boldsymbol{\xi}_d))^2 \,, \tag{29}$$

where $\tilde{\gamma}_i(\boldsymbol{\xi}_d)$ denotes the projection of $\tilde{\gamma}_{i-1}$ from the previous frame $i-1$ to the current frame i via $\boldsymbol{\xi}_d$:

$$\tilde{\gamma}_i(\boldsymbol{\xi}_d) = \sum_{\mathbf{u} \in U_k} \frac{\tilde{\gamma}_{i-1}(\mathcal{W}(\mathbf{u}, T(\boldsymbol{\xi}_d)^{-1}))}{num(U_k)} \,. \tag{30}$$

Here, U_k is the set of pixels from the k-th cluster of the current frame i and $num(U_k)$ is the number of pixels in this cluster. $\tilde{\gamma}_{i-1}(\mathbf{u}_{i-1})$ gives the estimated segmentation score of a pixel $\mathbf{u}_{i-1}$ from the previous frame $i-1$. The warping function $\mathcal{W}$ is introduced in equation 25, which uses the estimated motion of dynamic rigid body $\boldsymbol{\xi}_d$ to transform the pixel $\mathbf{u}$ to the previous image frame.

The spatial and time smoothness (equation 28 and 29) are combined and weighted by λ_{seg}:

$$R_{seg}(\boldsymbol{\xi}_d, \boldsymbol{\Gamma}) = \lambda_{seg}(R_{spatial}(\boldsymbol{\Gamma}) + R_{time}(\boldsymbol{\xi}_d, \boldsymbol{\Gamma})) \,. \tag{31}$$

4.3.3 *Tightly Coupled Motion Prior*

Given the motion priors of both static and dynamic rigid bodies $\tilde{\boldsymbol{\xi}}_s$ and $\tilde{\boldsymbol{\xi}}_d$, we add a regularisation term on the motion of each rigid body:

$$R_{motion}(\boldsymbol{\xi}_s, \boldsymbol{\xi}_d) = \lambda_s ||\boldsymbol{\xi}_s - \tilde{\boldsymbol{\xi}}_s||_2^2 + \lambda_d ||\boldsymbol{\xi}_d - \tilde{\boldsymbol{\xi}}_d||_2^2 \,, \tag{32}$$

where parameters λ_s and λ_d weight the regularisation terms. $||\cdot||_2^2$ represents the square of the L_2 norm. Because potential drift and noise in the motion prior could bias the solution, the prior information is ignored if the current estimated state is closer to the prior than a noise-related threshold.

4.3.4 *Solver*

The solver is based on StaticFusion. Since we directly align images in equation 22, the minimisation problem is solved via a coarse-to-fine scheme. We create an image pyramid for each image frame by iteratively down-sampling each image, which reduces the impact of depth noise. The optimisation starts from the coarsest level. The results of intermediate levels are used to initialise the following level.

For each level of the image pyramid, we decouple motions $\boldsymbol{\xi}_s$ and $\boldsymbol{\xi}_d$ from segmentation $\mathbf{\Gamma}$. Specifically, at each iteration, we first fix $\mathbf{\Gamma}$ and optimise $P_{robust}(\boldsymbol{\xi}_s, \mathbf{\Gamma}) + P_{robust}(\boldsymbol{\xi}_d, \mathbf{1} - \mathbf{\Gamma}) + R_{motion}(\boldsymbol{\xi}_s, \boldsymbol{\xi}_d)$ over $\boldsymbol{\xi}_s$ and $\boldsymbol{\xi}_d$. Then $\boldsymbol{\xi}_s$ and $\boldsymbol{\xi}_d$ are fixed, and we optimise $P_{robust}(\boldsymbol{\xi}_s, \mathbf{\Gamma}) + P_{robust}(\boldsymbol{\xi}_d, \mathbf{1} - \mathbf{\Gamma}) + R_{seg}(\boldsymbol{\xi}_d, \mathbf{\Gamma})$ over $\mathbf{\Gamma}$.

4.4 MAPPING AND FRAME-TO-MODEL ALIGNMENT

After the minimisation of equation 22, we use the optimal scores $\mathbf{\Gamma}$ and $\mathbf{1} - \mathbf{\Gamma}$ to compute the weighted images for static and dynamic rigid bodies respectively. The weighted images are fused to the model of rigid bodies, and the estimated motions $\boldsymbol{\xi}_s$ and $\boldsymbol{\xi}_d$ are used to initialise the frame-to-model alignment. We use ElasticFusion without loop closure [124] to build the model and conduct frame-to-model alignment.

4.5 EVALUATION

4.5.1 *Setup*

4.5.1.1 *Synthetic Environment*

In simulated environments, we can control variables, such as object size or speed to accurately analyse the variables that influence the performance. To achieve this, we built a simple synthetic environment in Blender, which contains only two objects from the YCB object dataset [10] and one plane as illustrated in Figure 9.

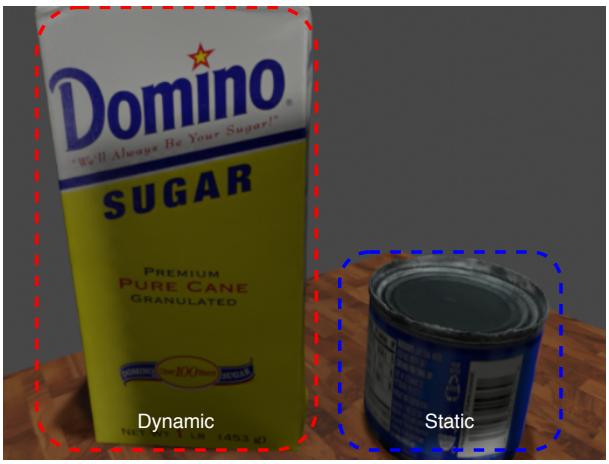


Figure 9: A simple simulated environment used for control variable experiments. Only the object in the blue rectangle can be dynamic and the others are static.

4.5.1.2 *Manipulator with Fixed-base*

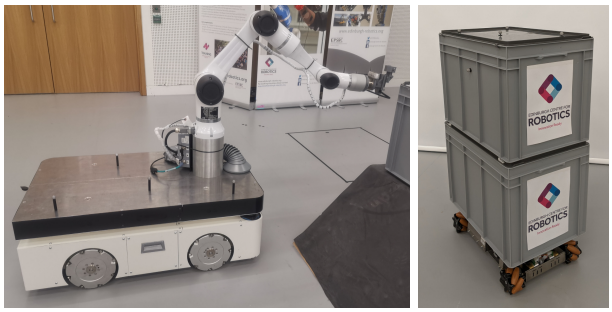
We also collect datasets on a Kawada Nextage robot (Figure 10). This robot is equipped with two 6-DoF arms fitted with custom end-effectors for dual-arm grasping. As for the RGB-D sensor, we use an Azure Kinect DK with a native resolution of 1280×720 after registering the depth to the colour frame. In this dataset, the robot grasps one box each time from the pre-selected objects (Figure 11) and introduces dynamic objects in the camera view.

Figure 10: Experimental setup: A stationary Nextage robot detects moving objects on a conveyor using an RGB-D camera mounted on the head, picks these objects from a conveyor using custom end-effectors and places them on a table.

Figure 11: Objects from left to right: *jaffa*, *oats*.

4.5.1.3 *Wheeled Platform*

Finally, we collect RGB-D sequences with an Asus Xtion PRO Live in plane-parallel movement (2 DoF translation. 1 DoF rotation) showing different characteristic object movements. The camera is either hand-held or mounted on an omnidirectional robot base (Figure 12a). The object is a remote-controlled KUKA youBot with stacked boxes (Figure 12b). The camera and the object are equipped with Vicon markers for ground truth comparisons and to simulate motion prior drift for camera-only sequences.



(a) Mobile manipulator Ada (b) KUKA youBot

Figure 12: Omnidirectional platforms for moving (a) camera and (b) stacked boxes ($0.4 \times 0.6 \times 1\,\text{m}$) with Vicon markers.

4.5.1.4 *Implementation Details*

The motion estimation performance is quantitatively evaluated via the absolute trajectory error (ATE) and the relative pose error (RPE) [105] against the Vicon ground truth for the optical frame. The visualised trajectories are aligned by the initial camera pose.

In the implementation of RF, we set $\lambda_r = 2$, and the thresholds $\hat{d}$ and $\hat{n}$ are both chosen as 0.01. We extend StaticFusion to use motion priors by appending

the regularisation term $\lambda_s||\boldsymbol{\xi}_s - \tilde{\boldsymbol{\xi}}_s||_2^2$ to the loss function. The method that StaticFusion with ground truth camera motion prior is denoted as *SF true*. We control the impact of adding camera motion prior by choosing the same $\hat{n} = 0.01$ for *SF true*. To compare our method with other baseline methods, including JF [38], SF [94] and CF [84], we keep the default parameters of their implementation. This is because they all take real RGB-D images as input and do not require prior information of the environment.

For camera-only sequences, the average simulated drift on camera trajectories is 6 cm/s (trans.) and 0.4 rad/s (rot.), while the average drift on object trajectories is 1.5 cm/s (trans.) and 0.1 rad/s (rot.). The camera and object speed is less than 60 cm/s. In robot experiments, we use wheel odometry as camera motion priors and keep the object motion prior with simulated drift.

4.5.2 *Synthetic Experiments*

We hypothesise that the proposed objective with motion priors improves the estimation for dynamic objects that occupy more than 50% of valid image pixels. To study this effect in a controlled environment, we synthesised a simple scene with an object of varying size moving across the image from left to right. The relation of trajectory error to drift magnitude (Figure 13) supports this hypothesis.

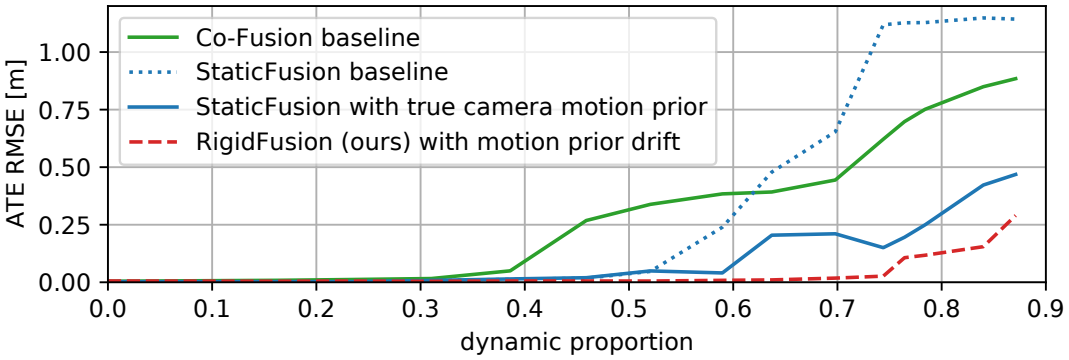


Figure 13: ATE of estimated camera trajectories on a synthetic sequence with different object sizes relative to the amount of valid image pixels. Co-Fusion and StaticFusion break around a dynamic ratio of 0.5 or less. Using the true motion priors in StaticFusion allows larger dynamic objects up to a ratio of 0.6, while our method with drift on the motion priors can track up to a dynamic ratio of 0.75.

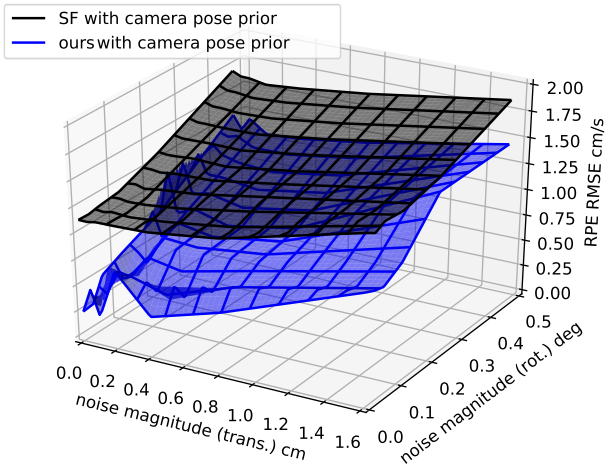


Figure 14: The mean value is illustrated as 2D surfaces over a range of translational and rotational noise magnitudes. Results suggest that when both the proposed methods (blue and green) achieves better performance and is more robust to noise than StaticFusion (black).

We also test the performance of our method when the magnitude of drift in motion priors increases (Figure 14). Results show that the accuracy of estimated trajectories decreases when the motion priors have a higher drift. However, we still outperform the baseline method StaticFusion.

4.5.3 *Camera Experiments*

sequence	**frame motion**		**difficulty**
	camera	textbfobject	
straight	straight	orthogonal crossing	low
orbit	orbit	rotation to camera	medium
overtake	straight	rotation + parallel to camera	medium
sideway	lateral	orthogonal zig-zag crossing	high

Table 1: Camera sequence description.

We collected four sequences involving plane-parallel movement of the camera and the object within the camera frame. These sequences have different characteristics of camera and object motion (Table 1). Figure 15 (top) shows the 2D plane projection of the true trajectories.

RGB-D sequence	Motion prior (drift)	Method				
		JF	SF	SF true	CF	RF (ours)
straight	17.6	48.4	34.8	14.5	**3.84**	7.57
orbit	44.2	52.0	87.7	19.9	14.2	**5.74**
overtake	8.93	59.6	52.6	23.6	23.0	**5.39**
sideway	51.1	55.3	70.1	38.1	48.2	**13.1**

(a) Trans. Absolute Trajectory Error RMSE (cm)

RGB-D sequence	Motion prior (drift)	Method				
		JF	SF	SF true	CF	RF (ours)
straight	6.02	18.5	24.3	12.9	**5.54**	6.05
orbit	6.03	13.4	25.2	5.78	8.47	**5.1**
overtake	6.34	19.1	27.4	11.3	18.9	**4.78**
sideway	6.01	21.7	42.3	9.87	17.0	**8.03**

(b) Trans. Relative Pose Error RMSE (cm/s)

Table 2: ATE and RPE for camera-only sequences. *Motion prior* represents the trajectory computed from prior motion with simulated drift to indicate the performance of simple kinematic odometry. Our method with motion prior drift outperforms the state-of-the-art on difficult sequences, including SF with true motion prior (SF true), while Co-Fusion performs best on the easiest sequence.

Our approach RigidFusion (RF) is compared against Joint-VO-SF (JF, [38]), StaticFusion (SF, [94]), StaticFusion with true motion priors (SF true) and Co-Fusion (CF, [84]). The quantitative evaluation in Table 2 shows that our method outperforms the state-of-the-art on more difficult sequences. Although Co-Fusion achieves best results on the easier *straight* sequence, it tends to over-segment dynamic objects and treats parts of the static background as dynamic. This effect is more dominant in the more difficult sequences, leading to worsen performance of CF.

The visualisation of the estimated trajectories in Figure 15 (bottom) confirms that our method outperforms the state-of-the-art in dynamic scenes. The improved localisation performance stems from a better segmentation of dynamic parts in the image (Figure 16). In our frame-to-frame odometry setting, the improved motion segmentation performance directly affects the estimation performance and additionally leads to better a reconstruction of the static environment.

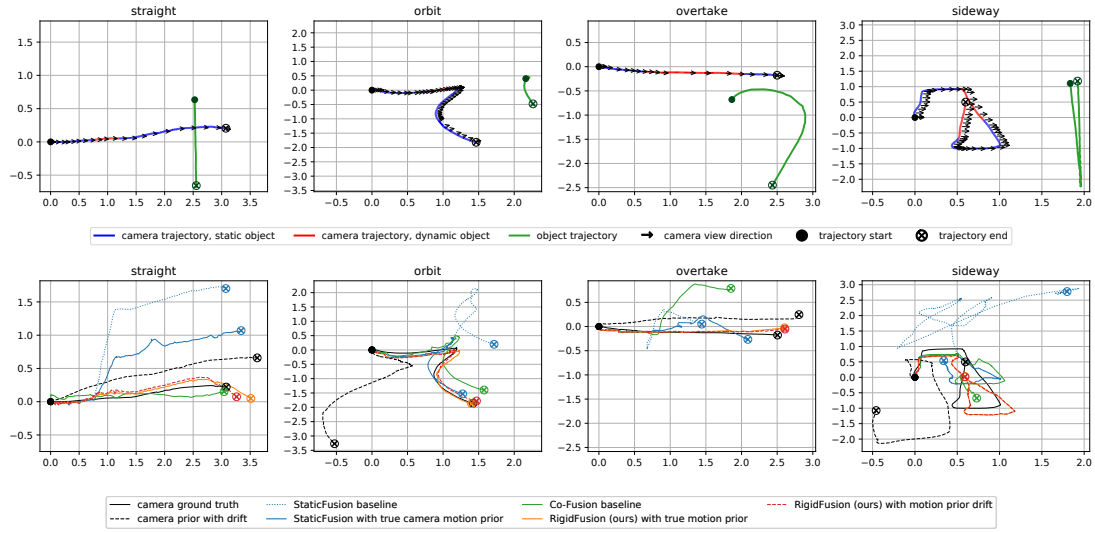


Figure 15: True and estimated trajectories (units in meter). **Top**: Top-down view of true camera and object trajectories in evaluation sequences. The green trajectory represents the true object position in the Vicon reference frame. The red/blue trajectory segments represent the camera trajectory and if the object is static (blue) or dynamic (red) within the image. Black arrows point in the camera view direction. **Bottom**: True and estimated trajectories for our RigidFusion (with and without drift on motion priors), the baselines StaticFusion [94] (with and without true motion priors) and Co-Fusion [84]. Trajectories start at the origin (black solid dot) and end at the circle-cross marker. Our proposed method is closer to the ground truth trajectory even with drift on the motion priors (red, dashed), while StaticFusion fails even with true prior (blue, solid).

4.5.4 *Robot Experiments*

In four additional robotic experiments, we use the camera on the floating base of an omnidirectional robot and replace the simulated drift with wheel odometry. The true trajectories of two of these sequences are shown in Figure 18 (top).

The quantitative results in Table 3 show that using real wheel odometry as motion priors, RF outperforms all other four methods in terms of both ATE and RPE on all four sequences. The estimated trajectories for sequences *sideway1* and *overtake* are shown in Figure 18 (bottom).

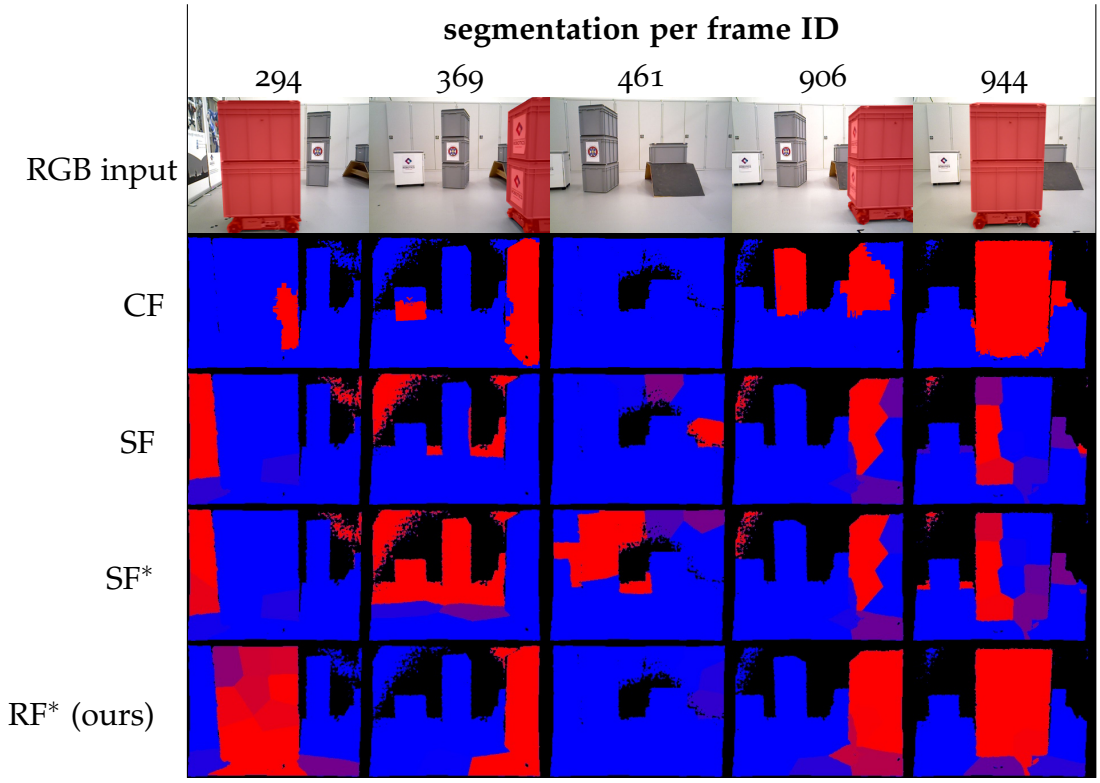


Figure 16: Segmentation and 3D reconstructed background for our proposed algorithm RigidFusion (RF), StaticFusion (SF) [94] (with and without true motion priors) and Co-Fusion (CF) [84] on camera-only sequence *sideway*. Our proposed method is the only one that can consistently segment the large rigid dynamic object (compare first row with highlighted boxes against red dynamic segmentation) and reconstruct the background even the motion priors have a significant drift.

4.5.5 *Object Reconstruction*

We compare the reconstructed dynamic object for CF and RF in Figure 19. Since CF tends to over-segment objects, we only show the first detected model. Results show that RF generates a more complete dynamic model than CF. This suggests that the segmentation estimated by RF is consistent over time and more accurate than CF.

4.5.6 *Object Tracking*

The object tracking is evaluated separately from the camera tracking using two different objects (Figure 11) that are moving on a conveyor belt at 6.8 cm/s. We compare the tracking results of the proposed approach RigidFusion (RF)

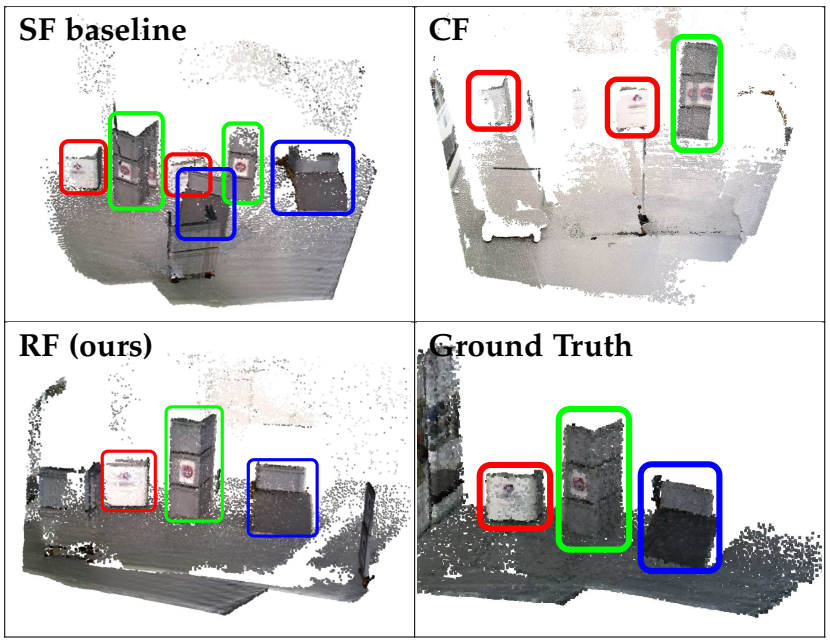


Figure 17: 3D reconstructed background (Figure 12b). The background is similar to Figure 16. Results show that RF generates the most accurate map that the static objects are only mapped once and the dynamic object is detected and thus removed.

RGB-D sequence	Wheel odometry	Method				
		JF	SF	SF true	CF	RF (ours)
sideway1	2.27	37.7	62.8	36.8	32.1	**7.58**
overtake	3.16	23.8	79.1	24.7	16.5	**14.0**
straight	3.64	51.9	86.3	21.9	12.3	**7.98**
sideway2	3.21	53.8	54.2	34.1	32.9	**10.7**

(a) Trans. Absolute Trajectory Error RMSE (cm)

RGB-D sequence	Wheel odometry	Method				
		JF	SF	SF true	CF	RF (ours)
sideway1	0.77	19.4	34.7	15.9	18.6	**3.66**
overtake	0.74	18.7	41.7	10.4	6.78	**2.06**
straight	1.13	39.8	84.2	13.7	10.8	**8.67**
sideway2	1.14	22.2	57.5	18.2	12.9	**6.68**

(b) Trans. Relative Pose Error RMSE (cm/s)

Table 3: ATE and RPE for sequences collected with Ada. The camera motion prior is estimated from the wheel odometry. Our method (RF) outperforms all compared dynamic SLAM methods when using real wheel odometry.

against Co-Fusion (CF) in two configurations. The box-shaped objects can

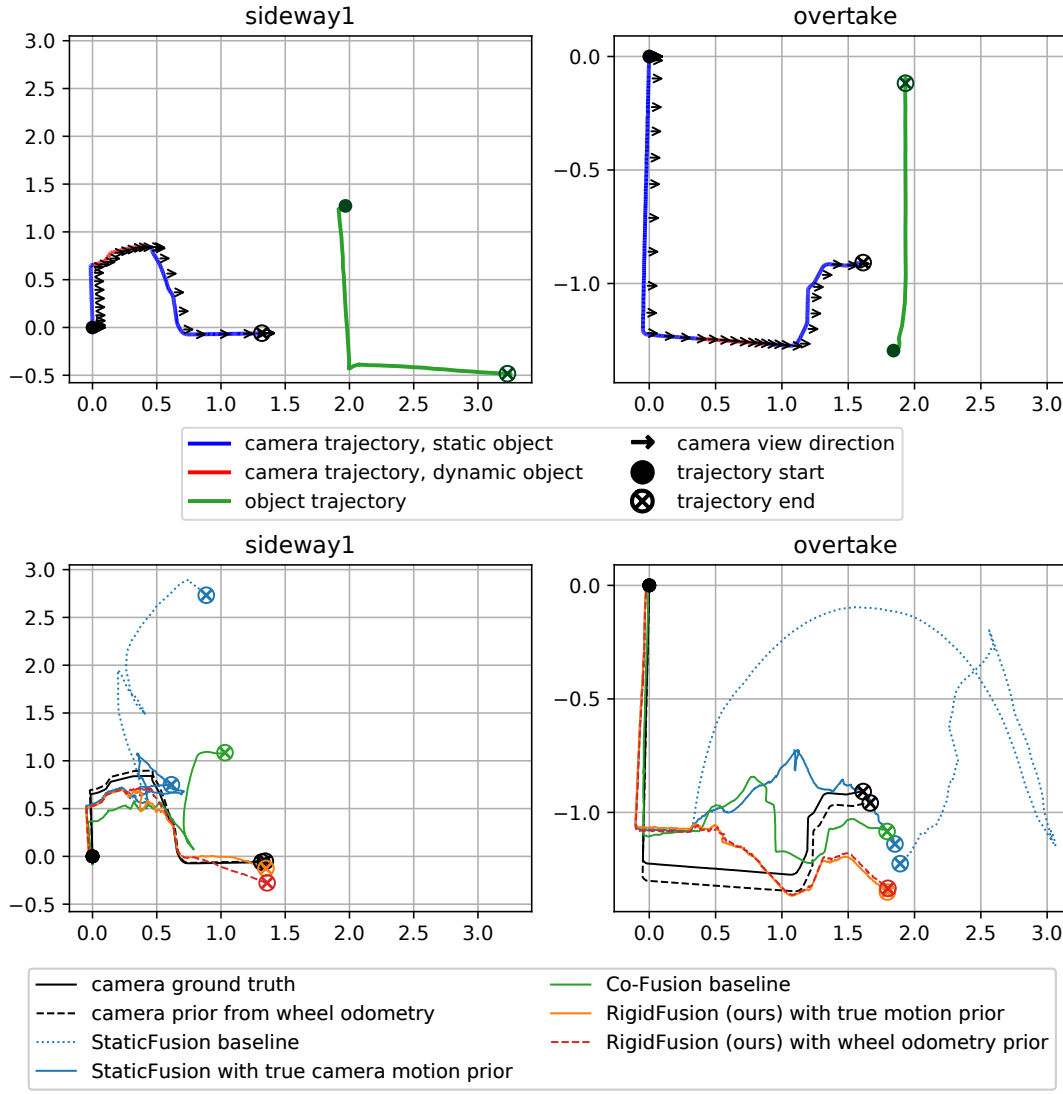


Figure 18: True (top) and estimated (bottom) trajectories (units in meter). Our method (RF) outperforms all state-of-the-art methods. Although CF has a closer end-position in the x-y plane, it has a larger drift in the z position than RF.

either stand *up* with the second largest side orthogonal to the ground, or lay flat *down* with the smallest side orthogonal to the ground.

Results show that our method (RF) is able to detect dynamic objects in both scenarios, while CF fails to detect dynamic objects when they lay flat on the convey belt (Table 4). In addition, our method outperforms CF in terms of the ATE of estimated object trajectories. We further visualise the object trajectories in Figure 20. Although our method has a better performance than CF, the accuracy of object tracking is still not high because our method only relies on frame-to-frame alignment and is unable to reduce drift.

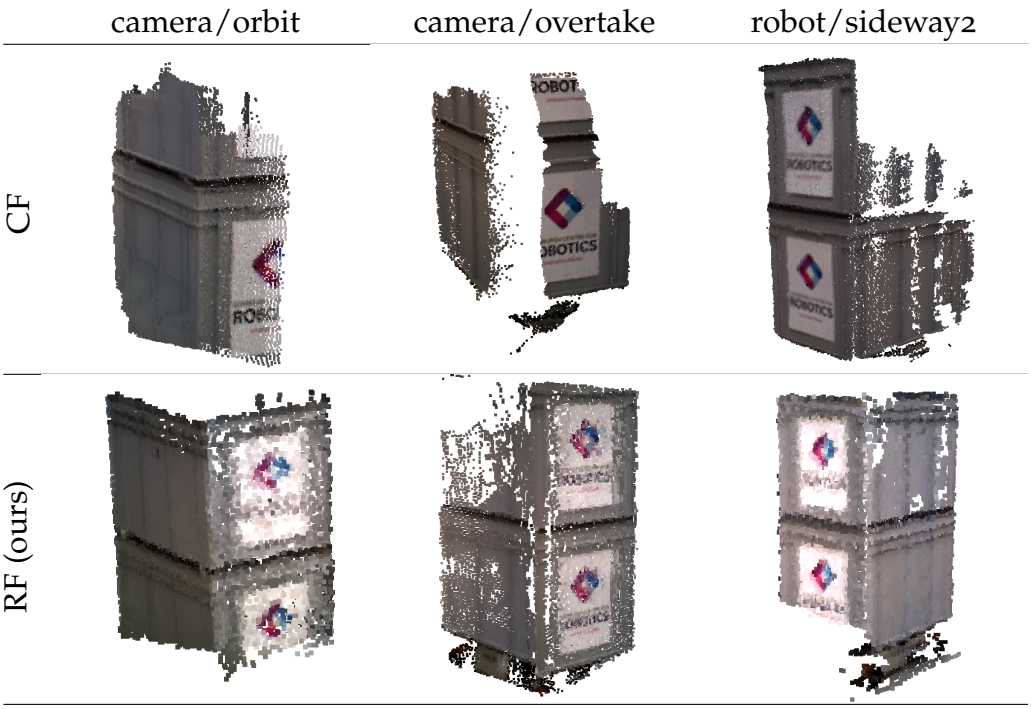


Figure 19: Reconstructed dynamic object. CF can only reconstruct parts of the dynamic object, while RF reconstructs a more complete model with inaccurate wheel odometry.

4.5.7 *Impact of Odometry Drift on Trajectory Estimation*

In robot experiments, we amplify the wheel odometry drift to test RF's robustness against different levels of camera motion prior drift. We also test RF's performance without the object motion prior (fix $\lambda_d = 0$). The relation between the RPE of the estimated trajectories and the drift over all robot sequences is shown in Figure 21. Even without the object motion prior, RF still achieves better performance than CF for up to 17 cm/s drift in terms of average RPE. Using both motion priors, RF performs even better and is more robust to large

seq.	type	CF	RF
jaffa	up	37.50	16.04
	down	—	17.43
oats	up	31.48	14.94
	down	—	17.99

Table 4: Transl. ATE (cm) for tracking the object centre from where they are initially detected up to the grasping position. A dash indicates that no object was detected.

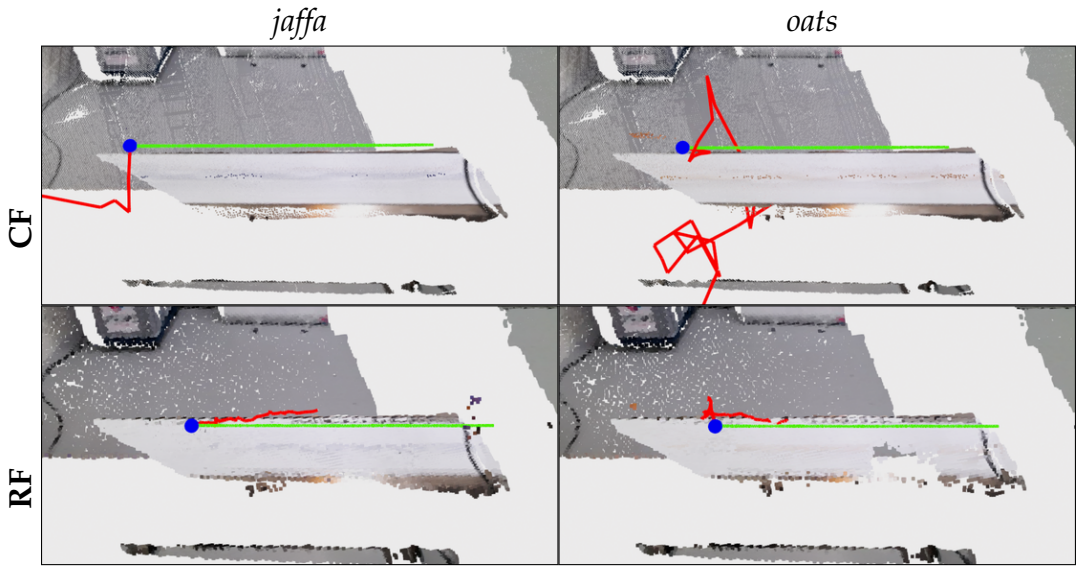


Figure 20: Estimated (red) and true (green) object trajectory on a conveyor belt from the point where an object's segment centre is first detected (blue). RF provides a better object trajectory than CF, but still has a high error against the ground truth.

odometry drift. This demonstrates that our method can handle large odometry drift and the absence of an object motion prior.

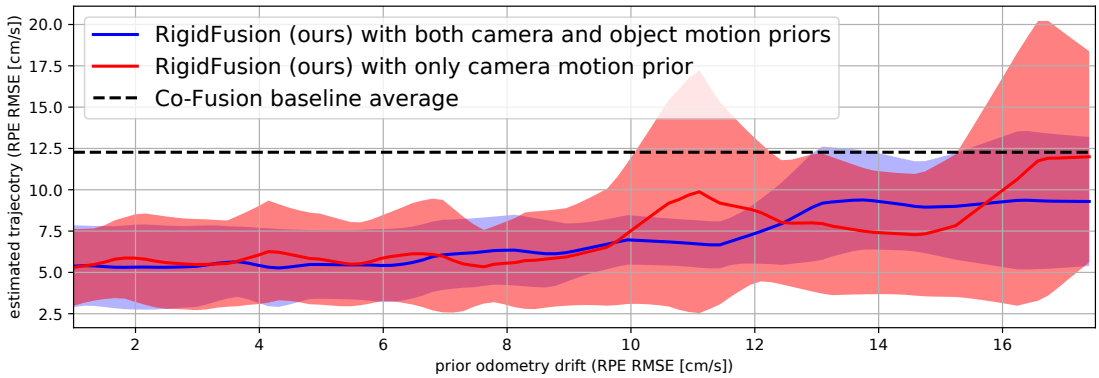


Figure 21: RPE of the estimated trajectories impacted by the drift magnitude of wheel odometry. Our method can handle up to 17 cm/s drift without the object motion prior (solid red) before breaking down to comparable results with CF. Using both motion priors (solid blue), RF has a better performance and stronger robustness.

4.5.8 *Impact of Multiple Dynamic Objects*

RF assumes that the dynamic motion can be explained by a single rigid transformation. To test RF's performance when this assumption is violated, we

conduct qualitative experiments on two OMD sequences [40] where multiple dynamic objects are present (Figure 22).

For sequence *occlusion_2_translational*, which contains one large and one small dynamic object, the motion prior for the larger object is provided. For sequence *swinging_4_translational*, which contains four dynamic objects, the motion prior for the top-left object is provided. Despite this under-representation of the dynamic motion, RF outperforms SF and is able to correctly segment the static environment against all the dynamic objects. However, similar to SF, RF cannot independently track multiple dynamic objects with different motions.

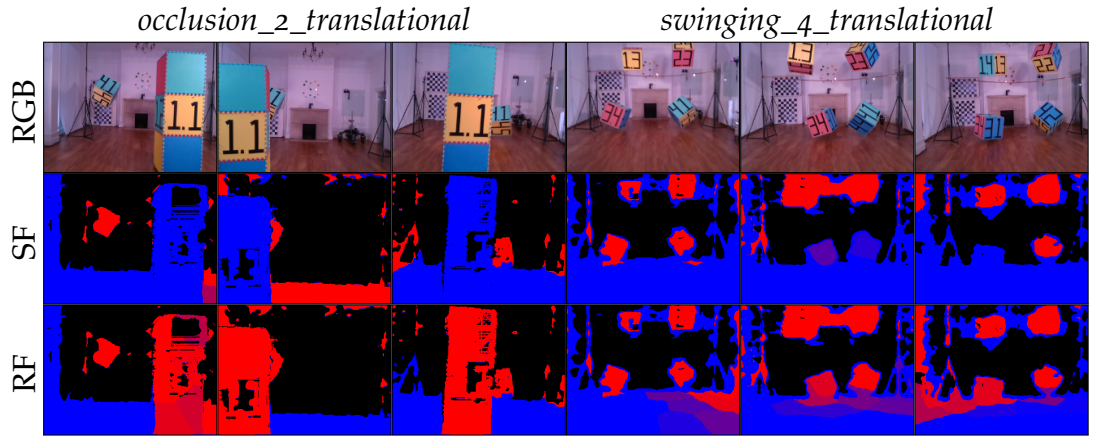


Figure 22: Segmentation results of two OMD [40] sequences with multiple dynamic objects. Although multiple objects can only be represented by a single transformation, RigidFusion (RF) is able to segment the static environment (blue) against multiple dynamic objects (red), while StaticFusion (SF) maps dynamic objects into the static environment.

4.6 CONCLUSION

In this chapter, we have presented an RGB-D simultaneous localisation and mapping (SLAM) approach in environments where dynamic components can occupy the major portions of the visual input. To address this problem, we model the whole dynamic components as a single rigid body, and jointly segment and track the static and dynamic rigid bodies. We also assume that we have prior information about the motion of the camera and dynamic objects.

The detailed evaluation shows that our method RigidFusion outperforms state-of-the-art when a dynamic rigid object occludes more than 65% of the

camera view. We also demonstrate its robustness to odometry drift up to 17 cm/s and the absence of object motion priors.

However, there are still several unresolved issues with our current methods. First, our method is unable to track multiple dynamic objects independently. This restricts the usage scenarios of this method, making it only applicable when there is only one dynamic object in the scene. Moreover, the accuracy of object tracking is low and our method is unable to reduce drift.

Additionally, our method requires both camera and object motion priors to differentiate the static and dynamic environments. This means that the proposed method is unable to automatically detect the number of dynamic objects when object motion priors are unavailable.

Therefore, in the next chapter (Chapter 5), we propose to simultaneously localise the camera, reconstruct the environment and track multiple large dynamic objects independently. We also detect the static background with the camera motion prior only and automatically estimate the number of dynamic objects when the majority of the camera view is occluded.

5

DYNAMIC RGB-D SLAM WITH MULTIPLE LARGE PLANAR OBJECT TRACKING IN INDOOR ENVIRONMENTS

In this chapter, we present 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. With the aid of camera motion prior, our approach enables dense SLAM when the camera view is largely occluded by multiple dynamic objects. 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.

5.1 INTRODUCTION

Simultaneous localisation and mapping (SLAM) is one of the core components in autonomous robots and virtual reality applications. 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 the accuracy of localisation [53, 134]. 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 is 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.

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.

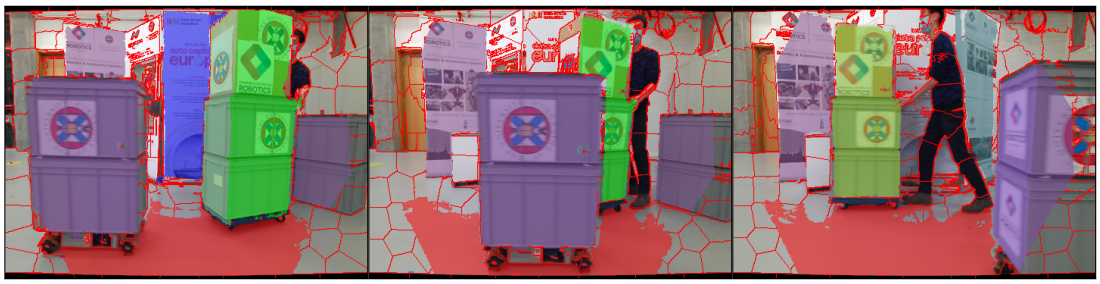


Figure 23: Hierarchical segmentation based on planes and non-planar areas. The planes are extracted from the depth map and the non-planar areas are represented by a set of super-pixels.

Many dynamic SLAM methods have considered multiple dynamic objects [37, 39, 84], 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 planes from planar areas and over-segments non-planar areas into super-pixels (Figure 23). We consequently segment and track multiple dynamic planar rigid objects, and remove dynamic non-planar objects to enable camera localisation and mapping. 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.

5.2 METHODOLOGY

5.2.1 *Overview and notation*

The overview of our pipeline is illustrated in Figure 24. Our method takes RGB-D image pairs from two consecutive frames. At the t-th frame, we have a depth image D_t and an intensity image I_t computed from the colour image.

A plane is represented in the Hessian form $\mathbf{\Pi} = (\mathbf{n}^T, d)^T$, 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 extract planes directly from the depth map using PEAC [23], which can provide the number of planes P, the pixel-wise segmentation of planes $\mathbf{U}^p : \{U_i^p | i \in [1, P]\}$ and their corresponding plane parameters. After plane extraction, the remaining non-planar areas are over-segmented into S super-pixels: $\mathbf{U}^{np} : \{U_i^{np} | i \in [1, S]\}$.

For the i-th plane, we extract a set of keypoints $\mathbf{K}_i$ using ORB features [61]. We then conduct multimotion segmentation on planar areas $\mathbf{U}^p$ 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* [39] and denote them as ${}^{ego}\boldsymbol{\xi} = \{\tilde{\boldsymbol{\xi}}_m \in \mathfrak{se}(3) | m \in [1, M]\}$. Since the static background may not be the largest rigid body in the camera view, we use the camera motion prior $\tilde{\boldsymbol{\xi}}_c$ with potential drift to simultaneously classify all planes and super-pixels into either static or dynamic, and estimate the camera ego-motion.

We use the score $\gamma_i \in [0, 1]$ to represent the probability that a plane or super-pixel is static. The cluster-wise static scores $\mathbf{\Gamma} : \{\gamma_i | i \in [1, P + S]\}$ 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. The static parts of intensity and depth images are used to estimate the camera motion $\boldsymbol{\xi}_c \in \mathfrak{se}(3)$. The dynamic planar rigid bodies are used to track dynamic objects. The non-planar dynamic super-pixels, such as humans, are removed as outliers.

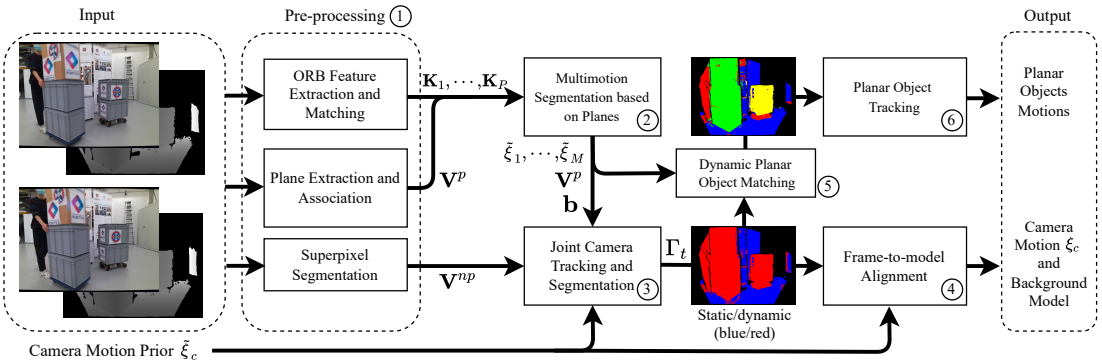


Figure 24: The pipeline of our proposed method. (1) We first represent the input image in the current frame t as a combination of planes and super-pixels. The ORB features [83] are extracted and matched to the previous frame. (2) Planes with similar rigid motions are clustered into M planar rigid bodies 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.

The world-, camera-, and the m-th object-frames are W, C and O^m respectively. The camera motion $\mathbf{T}(\boldsymbol{\xi}_c) := exp(\boldsymbol{\xi}_c)$ is $\mathbf{T}_{C_{t-1}C_t} = \mathbf{T}^{-1}_{WC_{t-1}}\mathbf{T}_{WC_t}$, which transforms homogeneous coordinates of a point in the current camera frame C_t to the previous frame C_{t-1}. The function $exp(\boldsymbol{\xi})$ is the matrix exponential map for Lie group $SE(3)$. The m-th object egocentric transformation is the camera motion relative to this object [39]:

$$\mathbf{T}(\tilde{\boldsymbol{\xi}}_m) = {}^m\mathbf{T}^{-1}_{C_{t-1}C_t} = \mathbf{T}^{-1}_{O^m_{t-1}C_{t-1}}\mathbf{T}_{O^m_t C_t}. \tag{33}$$

5.2.2 *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 [23]. To match 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 [53, 134]. However, rather than choosing the plane with minimal distance [53, 134], we further consider overlap proportion between two planes using the Jaccard index, $J(U_1, U_2) = \frac{|U_1 \cap U_2|}{|U_1 \cup U_2|}$, where $|U_1|$ 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 $\mathbf{T}_i = exp(\boldsymbol{\xi}_i) = {}^{i}\mathbf{T}^{-1}_{C_{t-1}C_t}$ of plane i, we extract and match ORB keypoints from plane i and i'. The error function is defined as:

$$e_i(\boldsymbol{\xi}_i) = \sum_{k \in \chi_{ii'}} \rho_H(||\mathbf{u}_i^k - T\mathbf{u}_{i'}^k||_\Sigma) + \lambda_h ||q(\boldsymbol{\Pi}_i) - q(exp(\boldsymbol{\xi}_i)^{-T}\boldsymbol{\Pi}_{i'})||_2^2, \tag{34}$$

where $\chi_{ii'}$ is the set of keypoint matches between planes i and i'. $\mathbf{u}_i^k$ and $\mathbf{u}_{i'}^k$ are homogeneous coordinates $[x, y, z, 1]$ of the two matched keypoints. $\rho_H(\cdot)$ is the robust Huber error function [61]. λ_h is the parameter to weight the error between the Hessian form of planes. $q(\boldsymbol{\Pi}) = \left[\arctan(\frac{n_x}{n_z}), \arctan(\frac{n_y}{n_z}), d\right]$ avoids over-parametrisation of the Hessian form [53].

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:

$$\begin{aligned} &\min_{{}^{ego}\boldsymbol{\xi}, \mathbf{b}} \sum_{i=1}^{P} e_i(\boldsymbol{\xi}_i) + \lambda_1 \sum_{(i,j) \in V_p} b_{ij} f(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j) - \lambda_2 \sum_{(i,j) \in V_p} b_{ij}, \\ &\text{s.t. } b_{ij} \in [0, 1] \ \forall i, j. \end{aligned} \tag{35}$$

${}^{ego}\boldsymbol{\xi}$ is the set of egocentric transformations $\{\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_P\}$ for all planes in the current frame. $\mathbf{b} = \{b_{ij} | (i, j) \in V_p\}$. V_p is the connectivity graph of planes in the current frame and $(i, j) \in V_p$ means that planes i and j are connected in space. The first term $e_i(\boldsymbol{\xi}_i)$ is introduced in Equation (34). In the second term, we propose

$$f(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j) = \left[e_i(\boldsymbol{\xi}_j) + e_j(\boldsymbol{\xi}_i)\right] - \left[e_i(\boldsymbol{\xi}_i) + e_j(\boldsymbol{\xi}_j)\right] \tag{36}$$

to quantify the error between two planes with egocentric motion $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$ 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 two neighbouring hypotheses having the same motion. This is in contrast to MVO [39], which discretely decides whether two motion hypotheses are merged or not. This means that MVO needs to iterates between label merging, assignment and splitting, while our method unifies these three actions with one set of scores $\mathbf{b}$. Importantly, the score $\mathbf{b}$ is tightly coupled with the motions of planar rigid bodies into a novel loss function (Equation (35)).

To minimise Equation (35), ${}^{ego}\boldsymbol{\xi}$ and $\mathbf{b}$ are decoupled. We first initialise all egocentric motions $\boldsymbol{\xi}_i$ to identity and all scores b_{ij} to 0. Then, at each iteration, we fix $\mathbf{b}$ and find optimal ${}^{ego}\boldsymbol{\xi}$ 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 $\{\tilde{\boldsymbol{\xi}}_1, \cdots, \tilde{\boldsymbol{\xi}}_M\}$ respectively. However, since the dynamic objects can occupy the majority of the images, we still need to decide which planar rigid body belongs to the static background.

5.2.3 *Joint camera tracking and background segmentation*

We jointly track the camera motion and segment the static background based on a hierarchical representation of planes and non-planar super-pixels. This representation is more efficient in planar environments than uniformly sampled clusters used in previous work [56, 94]. In addition, compared to RigidFusion [56], our method only requires the camera motion prior. The dynamic planar objects are detected by their different rigid motions compared to the camera motion while dynamic non-planar areas are removed by their high residuals. To achieve it, we propose to minimise a combined formulation that consists of three energy terms:

$$\min_{\boldsymbol{\xi}_c, \boldsymbol{\Gamma}} \; P_{robust}(\boldsymbol{\xi}_c, \boldsymbol{\Gamma}) + R_{seg}(\boldsymbol{\xi}_c, \boldsymbol{\Gamma}) + R_{motion}(\boldsymbol{\xi}_c) \quad \text{s.t. } \gamma_i \in [0,1] \; \forall i, \tag{37}$$

where $\boldsymbol{\Gamma}$ is the full set of probabilities that each plane or super-pixel is static. $\boldsymbol{\xi}_c \in \mathfrak{se}(3)$ 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 $P_{robust}(\boldsymbol{\xi}_c, \boldsymbol{\Gamma})$ aligns the static rigid body using weighted intensity and depth residuals. The second term $R_{seg}(\boldsymbol{\xi}_c, \boldsymbol{\Gamma})$ segments dynamic objects by either different motions or high residuals and maintains segmentation smoothness. The last regularisation term $R_{motion}(\boldsymbol{\xi}_c)$ adds a soft constraint on the camera motion.

5.2.3.1 *Residual term*

Following the previous work [56, 94], we consider image pairs (I_{t-1}, D_{t-1}) and (I_t, D_t) from two consecutive frames. Similar to Equation (24), for a pixel with coordinate $\mathbf{u} \in \mathbb{R}^2$ in the current frame t, the intensity residual $r_I^{\mathbf{u}}(\mathbf{T})$ and depth residual $r_D^{\mathbf{u}}(\mathbf{T})$ against the previous frame under motion $\mathbf{T}$ are given by:

$$r_I^{\mathbf{u}}(\mathbf{T}) = I_{t-1}\left(\mathcal{W}(\mathbf{u}, \mathbf{T})\right) - I_t\left(\mathbf{u}\right) \tag{38}$$

$$r_D^{\mathbf{u}}(\mathbf{T}) = D_{t-1}\left(\mathcal{W}(\mathbf{u}, \mathbf{T})\right) - |\mathbf{T}\pi^{-1}(\mathbf{u}, D_t\left(\mathbf{u}\right))|_z \,, \tag{39}$$

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

$$\mathcal{W}(\mathbf{u}, \mathbf{T}) = \pi\left(\mathbf{T}\pi^{-1}(\mathbf{u}, D_t(\mathbf{u}))\right). \tag{40}$$

which is the same as Equation (25). Similar to SF, the weighted residual term is:

$$P_{robust}(\boldsymbol{\xi}_c, \boldsymbol{\Gamma}) = \sum_{\mathbf{u} \in U} \gamma(\mathbf{u})[F(\alpha_I w_I^{\mathbf{u}} r_I^{\mathbf{u}}(\boldsymbol{\xi}_c)) + F(w_D^{\mathbf{u}} r_D^{\mathbf{u}}(\boldsymbol{\xi}_c))] \,, \tag{41}$$

which is similar to Equation (26), however, $\gamma(\mathbf{u})$ represents the score of planes or super-pixels that contains the pixel $\mathbf{u}$. α_I is used to weight the intensity residuals. The Cauchy robust penalty:

$$F(r) = \frac{c^2}{2} \log\left(1 + \left(\frac{r^2}{c}\right)\right) \tag{42}$$

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.

5.2.3.2 *Segmentation term*

The objective of $R_{seg}(\boldsymbol{\xi}_c, \boldsymbol{\Gamma})$ adds regularisation on both plane and super-pixel segmentation. Dynamic planar rigid bodies are detected by their motions while dynamic non-planar super-pixels are detected by their high residuals. $R_{seg}(\boldsymbol{\xi}_c, \boldsymbol{\Gamma})$ is computed by the sum of three items:

$$R_{seg}(\boldsymbol{\xi}_c, \boldsymbol{\Gamma}) = \lambda_p R_p(\boldsymbol{\xi}_c, \boldsymbol{\Gamma}) + \lambda_{np} R_{np}(\boldsymbol{\Gamma}) + \lambda_s R_{spatial}(\boldsymbol{\Gamma}), \tag{43}$$

where λ_p, λ_{np} and λ_s are parameters to weight different items. The first term $R_p(\boldsymbol{\xi}_c, \boldsymbol{\Gamma})$ classifies planes as dynamic when their egocentric motions are different from the camera motion $\boldsymbol{\xi}_c$:

$$R_p(\boldsymbol{\xi}_c, \boldsymbol{\Gamma}) = \sum_{i=1}^{P} \gamma_i \rho_H(||\boldsymbol{\xi}_c - \tilde{\boldsymbol{\xi}}_{m(i)}||_2^2), \tag{44}$$

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

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

$$R_{np}(\boldsymbol{\Gamma}) = F(\hat{c}) \sum_{i=P+1}^{P+S} (1 - \gamma_i)|U_i|, \tag{45}$$

where we only consider super-pixels in non-planar area and $|U_i|$ 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 $R_{spatial}(\boldsymbol{\Gamma})$ maintains the spatial smoothness of segmentation $\boldsymbol{\gamma}$ for both planar and non-planar areas by encouraging neighbour areas to have close scores:

$$R_{spatial}(\boldsymbol{\Gamma}) = \sum_{(i,j) \in V_p} b_{ij}(\gamma_i - \gamma_j)^2 + \sum_{(i,j) \in V_{np}} (\gamma_i - \gamma_j)^2, \tag{46}$$

where V_p and V_{np} are the connectivity graphs for planes and non-planar super-pixels respectively. b_{ij} is directly acquired from the minimisation of Equation (35). 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.

5.2.3.3 *Motion regularisation term*

We add a soft constraint on the camera motion $\boldsymbol{\xi}_c$ based on the motion prior $\tilde{\boldsymbol{\xi}}_c$:

$$R_{motion}(\boldsymbol{\xi}_c) = \lambda_c(1-\alpha_s)\rho_H(||\boldsymbol{\xi}_c - \tilde{\boldsymbol{\xi}}_c||_2^2), \tag{47}$$

where $\alpha_s \in [0,1]$ is the proportion between the number of pixels that are associated with 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 $\rho_H(\cdot)$ is used to handle large potential drifts in the camera motion prior.

The solver of Equation (37) 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 are used to initialise the next level, to allow correct convergence. We also decouple the camera motion $\boldsymbol{\xi}_c$ and $\boldsymbol{\gamma}$ for more efficient computation. Concretely, we initialise the camera motion $\boldsymbol{\xi}_c$ as identity and all $\boldsymbol{\gamma}$ to 1. For each iteration, we first fix $\boldsymbol{\gamma}$ and find the optimal $\boldsymbol{\xi}_c$. The closed-form solution for $\boldsymbol{\gamma}$ is then obtained by fixing $\boldsymbol{\xi}_c$. The solution for the previous iteration is used to initialise the current iteration.

5.2.4 *Background reconstruction and camera pose refinement*

In the current frame t, after the minimisation of Equation (37), we acquire the optimised camera motion $\hat{\boldsymbol{\xi}}_c$ 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 $\boldsymbol{\xi}_c$ using frame-to-model alignment. Concretely, we render

an image pair (I^r_{t-1}, D^r_{t-1}) from the current static background model at the previous camera pose. The rendered image pair (I^r_{t-1}, D^r_{t-1}) is directly aligned with (I^s_t, D^s_t) by minimising

$$\min_{\boldsymbol{\xi}_c} \; P_{robust}(\boldsymbol{\xi}_c, \boldsymbol{\Gamma} = \mathbf{1}) + R_{motion}(\boldsymbol{\xi}_c). \tag{48}$$

The first term $P_{robust}(\boldsymbol{\xi}_c, \boldsymbol{\Gamma} = \mathbf{1})$ is the same as Equation (41) but the $\boldsymbol{\Gamma}$ is fixed to $\mathbf{1}$ because the input should only contain the static background. We append $R_{motion}(\boldsymbol{\xi}_c)$ in Equation (47) as a soft-constraint for the frame-to-model alignment and α_s is estimated from pixel-wise dynamic segmentation. Since we have already solved *Equation* (37), we directly start from the finest level of the image pyramid and initialise the solver with the camera pose $\hat{\boldsymbol{\xi}}_c$ for the solver of Equation (48). The refined camera pose is used to fuse the static images (I^s_t, D^s_t) with the surfel-based 3D model as described in SF [94].

5.2.5 *Planar objects tracking*

After removing the static planes, we further track dynamic planar rigid bodies independently. This is different to our previous work RigidFusion [56] which models the whole dynamic component with a single rigid transformation. 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 $\mathbf{T}_{init}$. 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 [119]:

$$\mathbf{T}_{O^m_{t_0} O^m_t} = \mathbf{T}_{C_{t_0} C_t} {}^{m}\mathbf{T}_{C_{t_0} C_t} \mathbf{T}^{-1}_{init}. \tag{49}$$

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

Figure 25: (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.

5.3 EVALUATION

5.3.1 *Setup*

The sequences for evaluation are collected with an Azure Kinect DK RGB-D camera which is mounted on an omnidirectional robot (Figure 25a). 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 24), such as super-pixel and plane extraction. In the solver of Equation (37) and (48), 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 25). 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) [105] against the ground truth. The proposed method is compared with PlanarSLAM (PS) [53], EM-Fusion (EMF) [103], Joint-VO-SF (JF) [38], StaticFusion (SF) [94], Co-Fusion (CF) [84] and RigidFusion

(RF) [56]. We additionally provide the camera motion prior with drift to CF as the variant CF*. Following the previous work (Chapter 4), we use the default parameters of these baseline methods. 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 in different planar environments. For example, in the *seq1*, 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 *seq5*, the robot tries to overtake two dynamic objects ahead (Figure 26). 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 camera view. We also run experiments on sequences *sitting_xyz* and *walking_xyz* from TUM RGB-D dataset [105] which includes a large proportion of non-planar areas and denote them as *seq9* and *seq10* respectively.

5.3.2 *Camera localisation*

We estimate the ATE root-mean-square error (RMSE) and RPE RMSE between the estimated camera trajectories and ground truth (Table 5). In planar dynamic environments (seq. 1-8), the evaluation demonstrates that our method outperforms all other state-of-the-art methods (Figure 27). 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. PS is unable to estimate the correct camera pose because there are dynamic planes in the environment while PS assumes all planes are static. Our method also outperforms EMF because the semantic segmentation method [33] can only detect and segment certain categories of dynamic objects, like humans.

In non-planar dynamic environments (seq. 9-10), EMF outperforms all other methods because the dynamic humans can be directly segmented by Mask R-CNN [33]. However, even without relying on semantic segmentation, our method has close performance compared to StaticFusion. This is because our

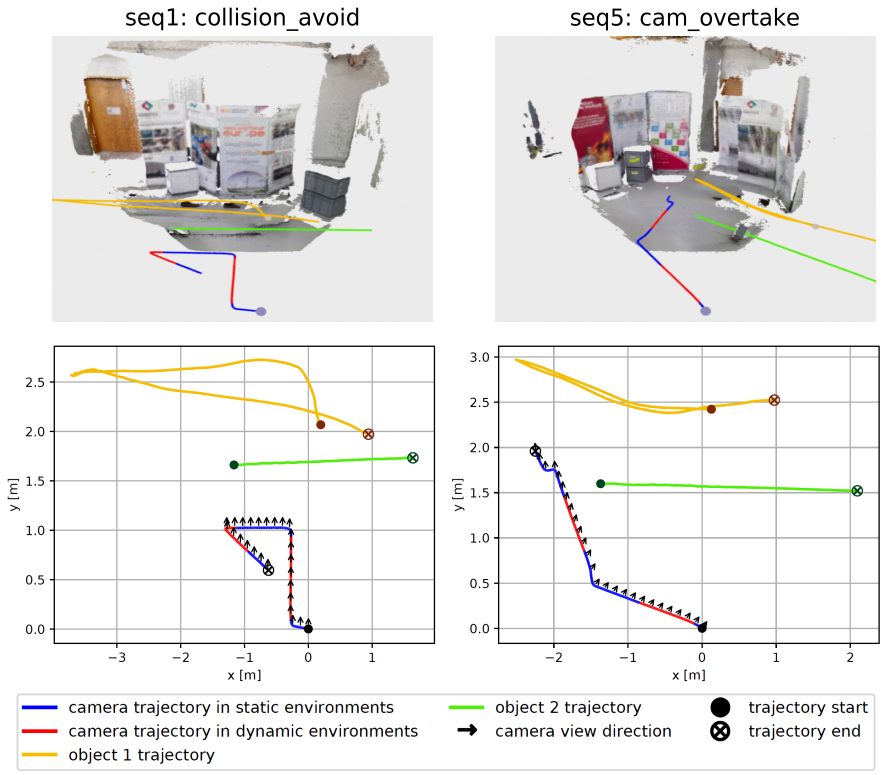


Figure 26: The ground truth trajectories of camera and dynamic objects in both 2D and 3D perspectives. Trajectories of humans are not plotted. The red segment of camera trajectories represents the part when there are moving objects in the camera view, while the blue segment means the camera moves in 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.

method can still detect dynamic super-pixels by their high residuals under the camera motion.

5.3.3 *Multimotion segmentation*

For planar environments, we visualise the segmentation results of our method and compare them with SF, RF* and CF (Figure 29). 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

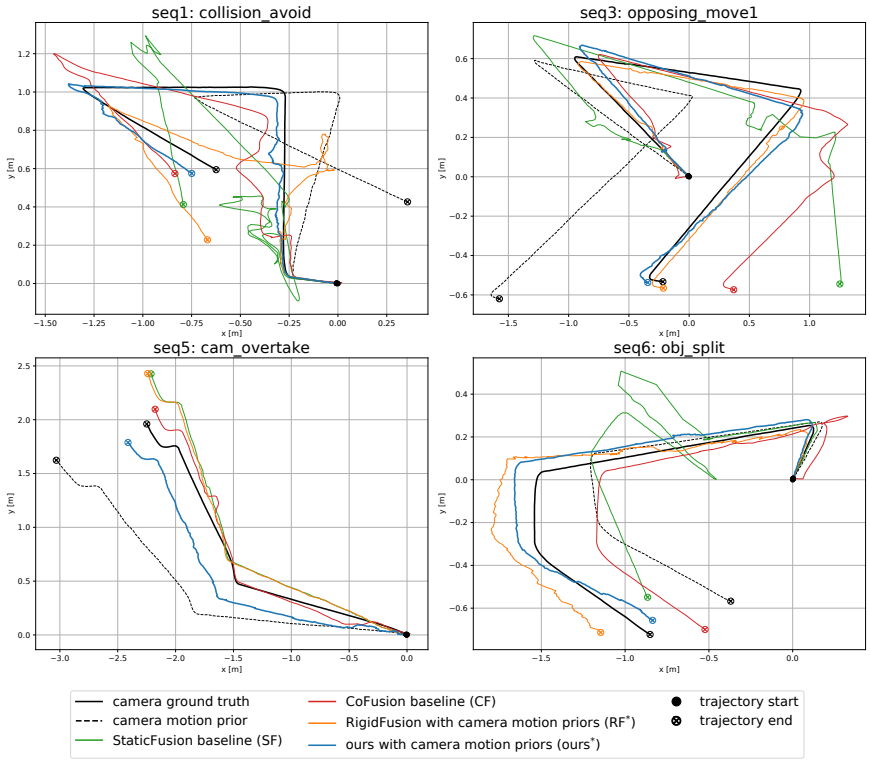


Figure 27: Visualisation of the estimated camera trajectories, camera motion prior and ground truth. The start position of all trajectories is aligned to the same position and is marked with a black solid dot. Our method (blue) achieves the lowest error compared to the ground truth (black solid) and can correct the drift of the camera motion prior (black dashed).

new one after it passes behind the front object. In non-planar environments, our method can still provide correct binary segmentation of the static and dynamic objects (Figure 28). However, we are unable to segment and track different non-planar dynamic objects independently.

5.3.4 *Background reconstruction*

We qualitatively evaluate the reconstruction result of *seq3* (Figure 30). 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 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 caused by wrong camera pose estimation. Only our proposed method can remove all dynamic objects and correctly reconstruct the background.

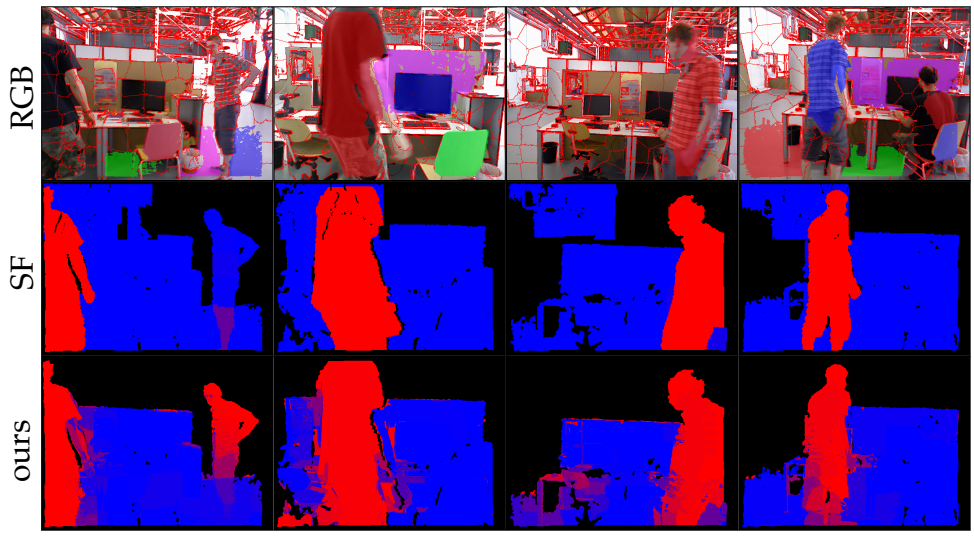


Figure 28: Static/dynamic segmentation results on the *walking_xyz* sequence [105]. The first row shows the RGB images with segmentation of planes and super-pixels. Our method achieves close segmentation performance to SF in non-planar environments.

5.3.5 *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 6). 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 31).

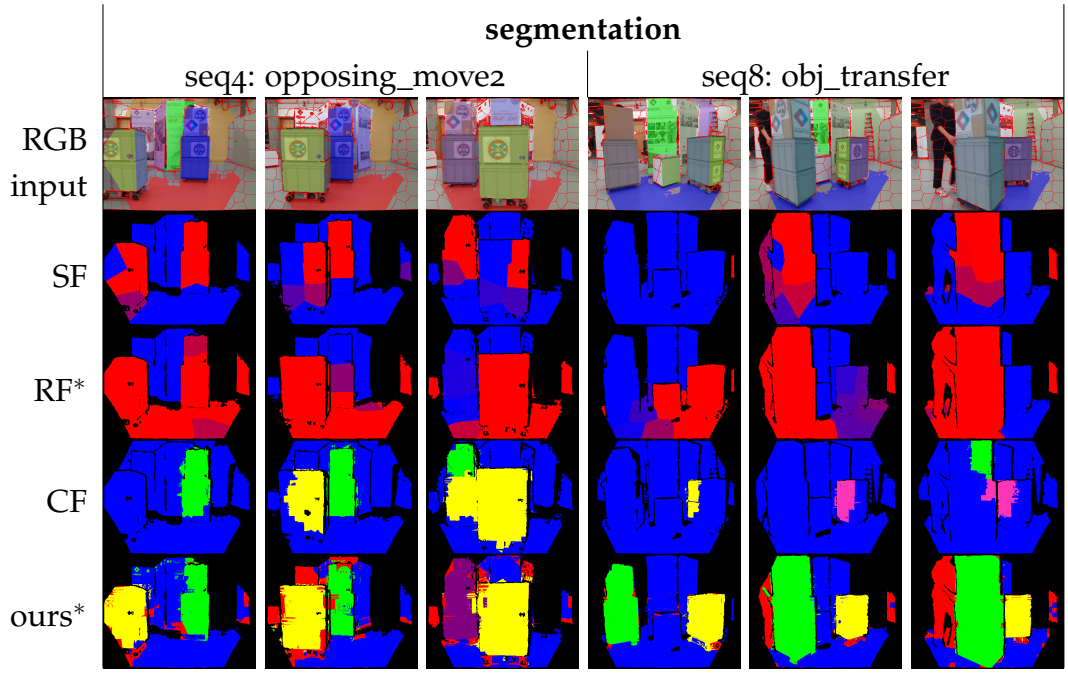


Figure 29: Segmentation result of the static background and dynamic objects. We visualise the input RGB images with the segmentation of planes and super-pixels in the first row. In all four methods, the static part is marked by blue. In SF and RF, we use red to represent dynamic parts. In CF, we use 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.

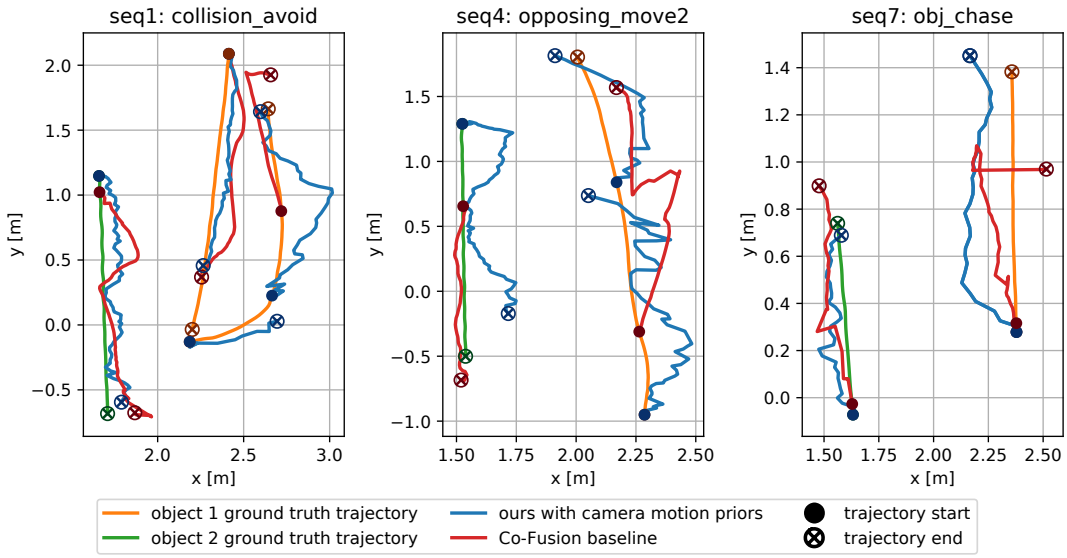


Figure 31: Comparison between CF baseline (red) and our method with the camera motion prior (blue) in terms of the estimated object trajectories. Our method can detect and track an object immediately when it starts to move.

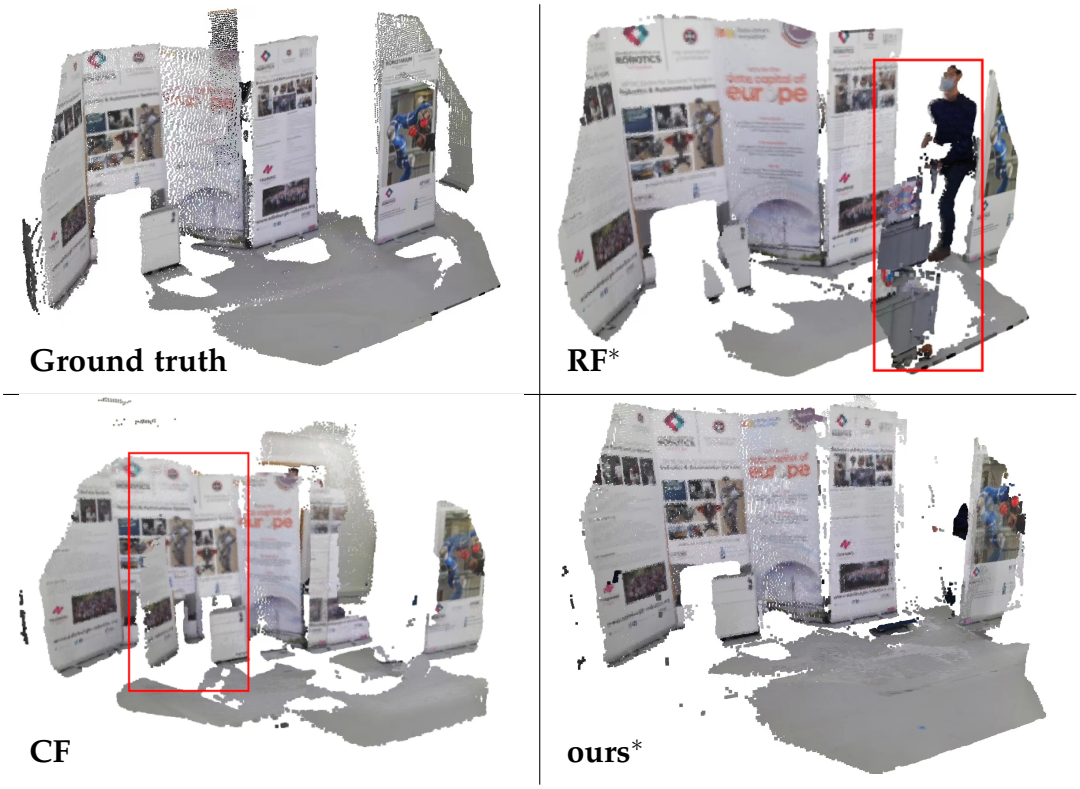


Figure 30: Reconstruction result of the RGB-D sequence 3. The reconstruction failures are marked with red rectangles. RF has mapped dynamic objects into the background. CF has mapped the same static poster twice, which indicates wrong localisation results.

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

5.3.6 *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 32). 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 method is always better using the motion prior with the same magnitude of drift.

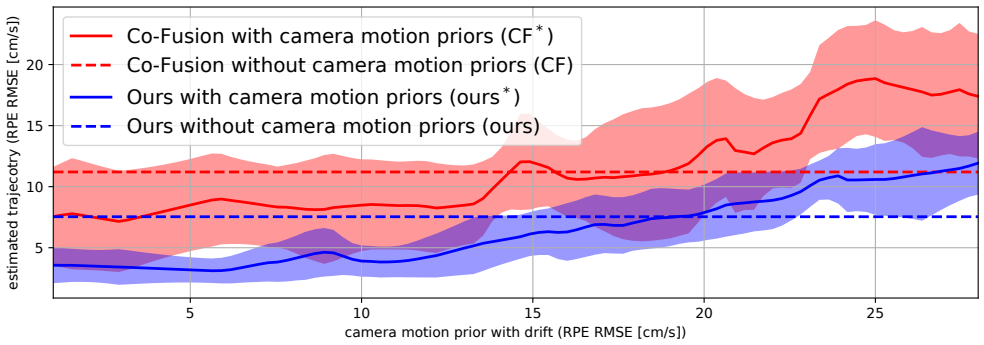


Figure 32: RPE RMSE of the estimated trajectories against the drift magnitude of wheel odometry. Our method performs better than CF when using the camera motion prior with the same magnitude of drift and is robust to nearly 24 cm/s odometry drift until it is comparable with CF baseline.

5.4 CONCLUSION

In this chapter, we present a dense RGB-D SLAM method PlanarFusion (PF) that 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.

Compared to RigidFusion (RF) proposed in Chapter 4, PF is able to track multiple dynamic objects independently and does not rely on object motion priors. PF also achieves higher object tracking accuracy than RF. However, if one dynamic object is occluded by another, PF fails to track the object but detects the object as new after it reappears in the camera view. Moreover, neither RF nor PF demonstrate their performance when the large occlusion lasts for a long period of time. For the solution of re-detecting the dynamic objects based on their models to support long-term object tracking, please refer to [77]. In Chapter 6, we plan to improve our method's robustness to long-term large occlusion and use measurements from an inertial measurement unit (IMU) as camera motion priors.

	MP	SLAM Method								
		PS	EMF	JF	SF	CF	CF*	RF*	ours	ours*
1	26.7	38.5	50.6	30.5	22.9	*10.4*	10.2	16.5	20.1	**4.23**
2	49.5	88.7	63.6	28.2	27.4	26.0	7.30	14.3	*6.81*	**6.32**
3	41.7	53.1	37.0	24.3	74.0	21.6	10.6	4.38	*4.01*	**3.42**
4	36.0	36.8	34.0	28.9	87.2	*18.9*	20.3	8.39	22.6	**8.37**
5	16.2	31.4	14.7	10.3	13.6	***4.73***	8.35	14.1	25.2	6.74
6	11.7	39.6	35.5	52.8	23.5	10.1	**3.67**	7.57	*8.37*	4.67
7	25.5	19.1	25.5	34.7	57.6	14.7	8.71	41.3	*6.43*	**7.60**
8	28.4	46.8	25.6	26.5	62.1	69.8	18.9	14.2	***8.33***	10.3
9	273	**2.15**	3.7†	11.1†	4.0†	2.7†	5.63	9.73	3.81	5.54
10	197	29.8	**6.6**†	87.4†	12.7†	69.6†	48.7	19.5	14.9	11.6

(a) Trans. ATE RMSE (cm)

	MP	SLAM Method								
		PS	EMF	JF	SF	CF	CF*	RF*	ours	ours*
1	7.64	23.8	43.6	28.4	17.2	*7.58*	9.07	9.41	10.9	**4.50**
2	7.31	51.6	22.8	26.9	11.2	12.6	5.61	3.99	*3.58*	**3.06**
3	7.87	25.1	14.8	23.5	26.1	6.8	4.22	7.13	*3.11*	**2.78**
4	7.38	29.9	26.5	28.2	64.3	15.9	15.7	6.13	*14.2*	**6.52**
5	7.61	25.8	6.31	31.4	***3.31***	3.62	6.34	3.90	13.5	4.77
6	7.51	17.1	30.6	25.4	18.1	7.02	4.67	4.26	*4.38*	**3.18**
7	7.52	12.8	15.4	31.3	62.4	7.54	6.43	25.1	*4.73*	**4.09**
8	7.29	20.0	15.6	24.1	36.4	28.3	11.2	7.54	*5.91*	**4.41**
9	7.36	3.12	**2.6**†	5.7†	2.8†	2.7†	3.01	3.48	2.95	2.98
10	7.34	49.0	**6.0**†	27.7†	12.1†	32.9†	41.9	13.4	9.59	8.67

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

Table 5: ATE and RPE RMSE for all ten RGB-D sequences. The asterisk (∗) symbol represents that the method uses the camera motion prior with drift and the dagger (†) symbol means the result is taken from the original paper [103]. Our method achieves the best performance in custom robotic sequences collected from planar environments (seq. 1-8) and estimates correct camera trajectories in TUM RGB-D dataset [105] which contains a large proportion of non-planar areas (seq. 9-10).

6

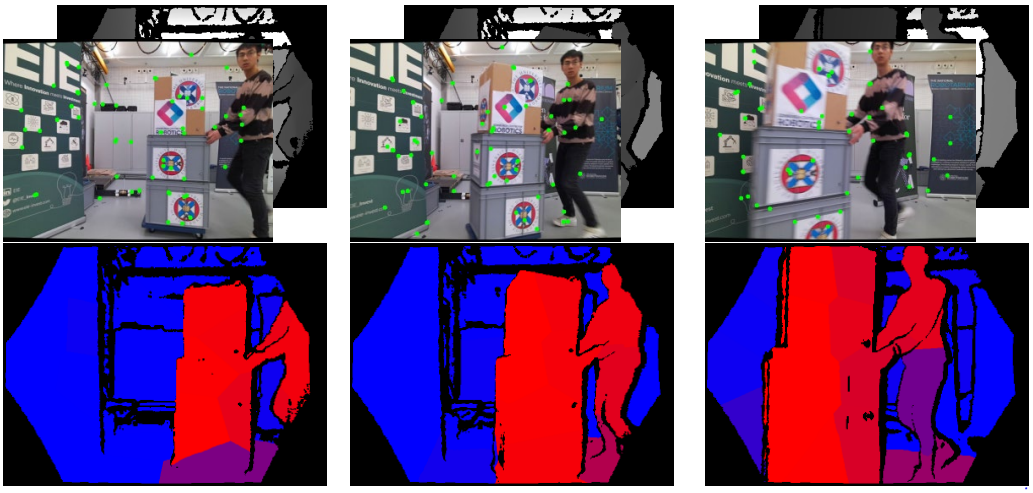
VISUAL-INERTIAL SLAM AND MOTION SEGMENTATION UNDER LONG-TERM LARGE OCCLUSIONS

This work presents a novel RGB-D-inertial dynamic SLAM method that can enable accurate localisation when the majority of the camera view is occluded by multiple dynamic objects over a long period of time. Most dynamic SLAM approaches either remove dynamic objects as outliers when they account for a minor proportion of the visual input, or detect dynamic objects using semantic segmentation before camera tracking. Therefore, dynamic objects that cause large occlusions are difficult to detect without prior information. The remaining visual information from the static background is also not enough to support localisation when large occlusion lasts for a long period. To overcome these problems, our framework presents a robust visual-inertial bundle adjustment that simultaneously tracks camera, estimates cluster-wise dense segmentation of dynamic objects and maintains a static sparse map by combining dense and sparse features. The experiment results demonstrate that our method achieves promising localisation and object segmentation performance compared to other state-of-the-art methods in the scenario of long-term large occlusion.

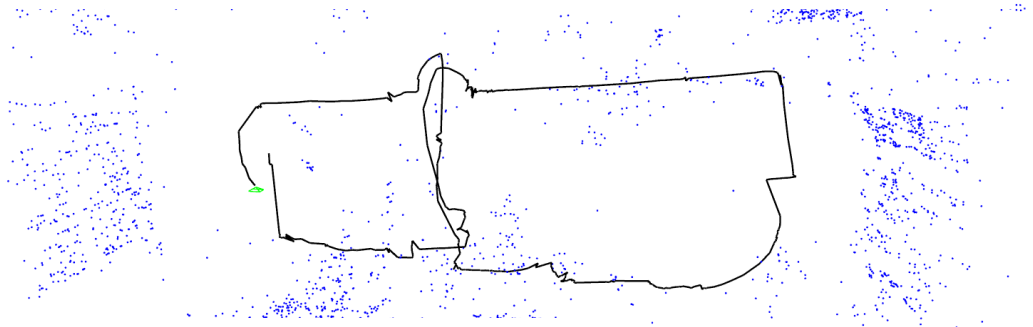
6.1 INTRODUCTION

Simultaneous localisation and mapping (SLAM) is one of the core problems in various robot applications. Despite providing accurate camera motion estimation and mapping in large-scale environments, most existing SLAM systems [61, 123] assume that the environment is static. This assumption can be violated when a robot closely manipulates objects in the scene or collaborates with other humans over a long period. In this scenario, the dynamic objects can cause long-term large occlusion, which means for the majority of time when a robot moves in the environment, the major proportion of the camera view is occluded by multiple dynamic objects.

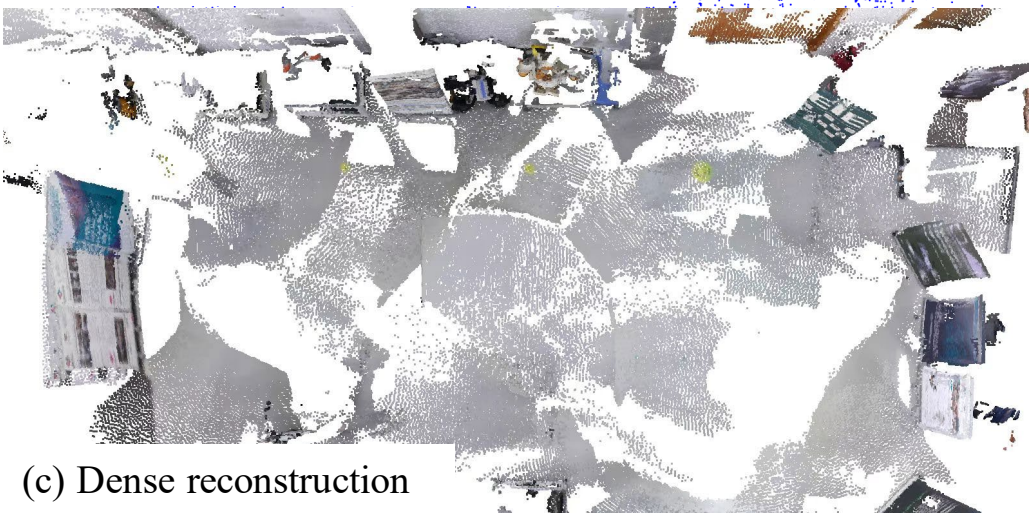
To enable robust SLAM in dynamic environments, many visual SLAM methods [84, 94] detect the areas of dynamic objects by assuming that the



(a) RGB input with large occlusion and dense segmentation



(b) Localisation and sparse mapping



(c) Dense reconstruction

Figure 33: (a) In the scenario of long-term large occlusion, the majority of camera view is occluded for the majority of time frames. Our method can estimate cluster-wise dense segmentation of dynamic objects, and (b) simultaneously localise the camera and create a static sparse map. (c) The dense reconstruction of the static background can be acquired using the estimated camera trajectory and dense object segmentation after the procession of the whole sequence.

static background occupies a major fraction of the camera view. The dynamic objects can, therefore, be removed as outliers during robust camera tracking. On the other hand, when the categories of dynamic objects are predefined, the regions containing these objects can be directly detected using deep learning methods [80].

However, when a priori undefined dynamic objects cause long-term large occlusion, there are still two major challenges. First, robots are unable to differentiate the dynamic objects from the static background because they can neither be detected by semantic segmentation nor be removed as outliers. Moreover, even when the dynamic objects are correctly removed, the remaining colour and depth information from the static background may be inadequate to support accurate localisation or mapping.

Various dynamic SLAM methods [55, 56, 99] have explicitly considered dynamic objects that are dominant in the camera views with the aid of robot proprioception, like wheel odometry or Inertial Measurement Units (IMU). Some static visual-inertial navigation system (VINS) methods [11, 104] have also demonstrated their robustness when the camera view is fully occluded for a short period. However, none of them has shown their performance if the large occlusion lasts for the most of time when the camera is in motion.

This paper is aimed to enable robust dynamic SLAM in the presence of long-term large occlusion. We use motion priors from IMU in a tightly-coupled way to help detect dynamic objects that cause large occlusion by simultaneously estimating camera motion, object segmentation and bias terms of the IMU. However, when the major proportion of camera view is occluded for a long period, the remaining features from the static background are unable to provide an accurate bias estimation of IMU which is important to detect dynamic objects that cause large occlusion. To further improve the method's robustness to long-term large occlusion, we actively remove sparse map points that are generated from the regions of dynamic objects and maintain a sparse model of the static background by integrating both sparse and dense features. The dense reconstruction of the static background is acquired after the processing of the whole sequence (Figure 33).

In summary, this chapter contributes:

1. an RGB-D-inertial SLAM method that is robust to long-term large occlusion caused by multiple undefined dynamic objects.

2. a new bundle adjustment (BA) pipeline that simultaneously provides dense segmentation of dynamic objects, tracks the camera and maps the environments.

3. a novel methodology that combines sparse and dense features for dynamic object detection.

6.2 METHODOLOGY

6.2.1 *Overview and Notation*

The proposed new RGB-D-inertial SLAM pipeline (Figure 34) focuses on accurate localisation and dense segmentation of dynamic objects that cause long-term large occlusion. Our method takes a stream of RGB-D images and readings from a low-cost IMU as input, which means additional variables are needed to be estimated compared to visual SLAM. For a frame i, similar to ORB-SLAM3 [11], we consider the pose of body $\mathbf{T}_{Bi} \in SE(3)$, velocity of body $\mathbf{v}_i$ in the world frame, and the biases of gyroscope $\mathbf{b}_i^g \in \mathbb{R}^3$ and accelerometer $\mathbf{b}_i^a \in \mathbb{R}^3$ respectively. The two biases are modelled by Brownian motions of Gaussian processes. The state vector for frame i is denoted as $\mathbf{S}_i = \{\mathbf{T}_{B_i}, \mathbf{v}_i, \mathbf{b}_i^g, \mathbf{b}_i^a\}$. The camera pose $\mathbf{T}_{Ci}$ can be acquired by $\mathbf{T}_{Ci} = \mathbf{T}_{Bi}\mathbf{T}_{BC}$, where $\mathbf{T}_{BC}$ is calibrated a priori and denotes the rigid transformation between the body (IMU) and camera frame.

For the image frame i, we extract ORB features [83] from the intensity image I_i and the observation is denoted as $\mathbf{u} \in \mathbb{R}^n$, where $n = 2$ for monocular feature points and $n = 3$ for features points with a depth reading. We also over-segment the depth image into K clusters and the region with no valid depth reading is denoted as the $(K+1)$th cluster. For each cluster, we assign a score $\gamma \in [0, 1]$ to represent the probability that the cluster is static. The whole set of scores is denoted as $\mathbf{\Gamma}_i = \{\gamma_{i0}, \cdots, \gamma_{iK-1}, \gamma_{iK}\}$, where $\gamma_{iK} = 1$. To reduce computation complexity, we assume all pixels and feature points in the same cluster have the same score. Concretely, $\gamma(\mathbf{u})$ denotes the score of the pixel $\mathbf{u}$ and is equal to γ_{ik} for any pixel $\mathbf{u}$ that belongs to the k-th cluster at the i-th image frame.

Between two consecutive images $i-1$ and i, we estimate pre-integrated measurements of rotation $\Delta\mathbf{R}_{i-1,i}$, velocity $\Delta\mathbf{v}_{i-1,i}$ and position $\Delta\mathbf{p}_{i-1,i}$ from

their states $\mathbf{S}_{i-1}$, $\mathbf{S}_i$ based on the theory proposed in [25, 58] and denote the inertial residual $\mathbf{r}^I_{i-1,i}$ as $[\Delta\mathbf{R}_{i-1,i}, \Delta\mathbf{v}_{i-1,i}, \Delta\mathbf{p}_{i-1,i}]$.

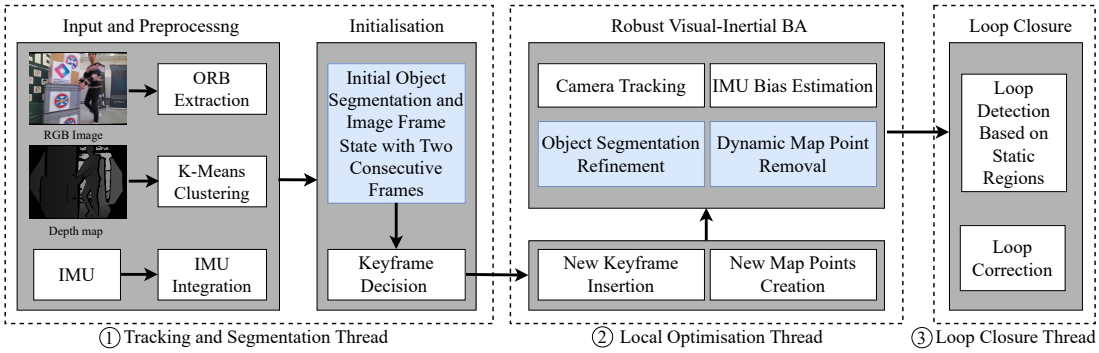


Figure 34: The pipeline of our method is based on ORB-SLAM3 [11] and blue rectangles highlight the functions we implement in addition to ORB-SLAM3. Our pipeline consists of three threads: (1) In the *tracking and segmentation* thread, we extract ORB features [83] from colour images and over-segment the images into clusters by applying K-Means clustering on the depth image. Given IMU bias estimation, we acquire camera motion priors using pre-integrated velocity, rotation and position measurements. We then estimate initial object cluster-wise segmentation and image frame states based on a combination of sparse and dense features (Section 6.2.2). (2) In the *local optimisation* thread, new keyframes are created and sparse map points are generated from the initial static parts of the image. We then conduct robust visual-inertial BA to simultaneously remove dynamic map points, estimate the states of multiple keyframes and refine dense object segmentation (Section 6.2.3). (3) Finally, the static parts of keyframes are used for place recognition and loop closure in the *loop closure* thread (Section 6.2.4).

Our robust BA considers a set of $M+1$ co-visible keyframes within a sliding window and a set of L 3-D map points that are observed by these keyframes [11]. The states of keyframes are denoted as $\hat{\mathbf{S}} = \{\mathbf{S}_0, \cdots, \mathbf{S}_M\}$ and the 3-D positions of map points are denoted as $\hat{\mathbf{X}} = \{\mathbf{x}_0, \cdots, \mathbf{x}_{L-1}\}$. For all keyframes, the set of cluster-wise dense segmentation is denoted as $\hat{\mathbf{\Gamma}} = \{\mathbf{\Gamma}_0, \cdots, \mathbf{\Gamma}_M\}$. In addition, for each map point j, we also assign a score $\beta_j \in [0, 1]$ to represent the probability that the map point is generated from the regions of the static background. The scores of all map points are denoted as $\hat{\mathbf{B}} = \{\beta_0, \cdots, \beta_{L-1}\}$. For simplicity, we denote $\mathcal{X} = \{\hat{\mathbf{S}}, \hat{\mathbf{X}}, \hat{\mathbf{\Gamma}}, \hat{\mathbf{B}}\}$.

6.2.2 *Robust Visual-inertial Bundle Adjustment (BA)*

To handle long-term large occlusion caused by dynamic objects, we simultaneously estimate the segmentation of dense images and sparse map points,

track the camera motions and correct IMU biases. To achieve it, we propose a novel cost function that consists of four terms:

$$\begin{aligned} &\min_{\mathcal{X}} \underbrace{\sum_{i=1}^{M} ||\mathbf{r}^I_{i-1,i}||^2_{\Sigma^{-1}_{i-1,i}}}_{IMU} + P_{robust}(\mathcal{X}) + \underbrace{R_{seg}(\mathcal{X}) + R_{imu}(\hat{\mathbf{S}})}_{Regularision}, \\ &\text{s.t. } \gamma_{ik}, \beta_j \in [0,1] \ \forall i,j,k, \end{aligned} \tag{50}$$

where the first term is the IMU residual term [11] which provides motion priors for any two consecutive keyframes.

The second term $P_{robust}(\mathcal{X})$ is the robust residual error between feature points on keyframes and map points:

$$P_{robust}(\mathcal{X}) = \sum_{j=0}^{L-1} \sum_{(i,k)\in\mathcal{K}^j} \rho(\beta_j, \gamma_{ik}, \mathbf{r}_{ij}) \tag{51}$$

where $\mathcal{K}^j$ includes all clusters that observe the j-th map point and $(i,k) \in \mathcal{K}^j$ represents that the k-th cluster in the i-th keyframe can observe the j-th map point. γ_{ik} is the static weight of this cluster and β_j is the static weight of the j-th map point. $\mathbf{r}_{ij} \in \mathbb{R}^n$ denotes the residual between the j-th map point and its observation $\mathbf{u}_{ij} \in \mathbb{R}^n$ on the i-th keyframe:

$$\mathbf{r}_{ij} = \mathbf{u}_{ij} - \pi(\mathbf{T}_{Ci}\mathbf{x}_j), \tag{52}$$

where $\pi : \mathbb{R}^3 \to \mathbb{R}^n$ is the camera projection function. The robust residual function $\rho(\cdot)$ is inspired by DynaVINS [99]:

$$\rho(\beta, \gamma, \mathbf{r}) = \beta^2\gamma^2\rho_H(||\mathbf{r}||^2) + (1-\beta)^2(1-\gamma)^2\rho_H(\hat{c}), \tag{53}$$

where $\hat{c}$ is heuristically selected as the average of all residuals $||\mathbf{r}_{ij}||^2$ and the robust Huber loss $\rho_H(\cdot)$ is used to reduce the effect of outliers [61]. Compared to DynaVINS, we use the product of static weights β and γ to weight residuals. Therefore, only correspondences between static map points and feature points on static clusters have high static weights in the cost function. We can also acquire the cluster-wise dense segmentation $\hat{\mathbf{\Gamma}}$ for keyframes, while DynaVINS can only provide sparse segmentation on feature points. Additionally, we are able to detect texture-less dynamic clusters that have no extracted feature points.

The third term $R_{seg}(\mathcal{X}) = \sum_{i=1}^{M} \lambda_d R_d(\mathcal{X}, i) + \sum_{j=1}^{L-1} \lambda_s R_s(\mathcal{X}, j)$ adds regularisation on the cluster-wise dense segmentation of images $\hat{\mathbf{\Gamma}}$ and sparse segmentation of map points, where

$$R_d(\mathcal{X}, i) = \sum_{k=0}^{K-1} (\gamma_{ik} - \tilde{\gamma}_{ik})^2 + \sum_{(k_1,k_2)\in V_i} (\gamma_{ik_1} - \gamma_{ik_2})^2, \tag{54}$$

$$R_s(\mathcal{X}, j) = (\beta_j - \tilde{\beta}_j)^2, \tag{55}$$

where $\tilde{\gamma}$ and $\tilde{\beta}$ are the priors of γ and β respectively, λ_d and λ_s are parameters to weight the two terms. V_i is the connectivity graph for clusters [94] in the i-th keyframe and $(k_1, k_2) \in V_i$ represents the k_1-th and k_2-th clusters of keyframe i are connected in space.

Inspired by StaticFusion [94], we assume that the dynamic regions of the depth map have a large depth difference from the static map. However, in contrast to StaticFusion, we estimate the depth difference between sparse map points and dense depth images. The $\tilde{\gamma}_{ik}$ is, therefore, defined as:

$$\tilde{\gamma}_{ik} = 1 - \lambda_p \frac{\sum_{j\in\mathcal{M}^{ik}} |\beta_j (D_i(\pi(\mathbf{T}_{Ci}\mathbf{x}_j)) - |\mathbf{T}_{Ci}\mathbf{x}_j|_z)|}{n(\mathcal{M}^{ik})}, \tag{56}$$

where $\mathcal{M}^{ik}$ is a set of map points that are observed by the kth cluster of the i-th keyframe and $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ is the projection function of a pinhole camera. $D_i(\cdot)$ gives the depth value of a pixel on image i, $|\cdot|_z$ gives the z-coordinate of a 3-D vector and $n(\cdot)$ provides the number of elements in a set. λ_p is a parameter to control the influence of the depth difference to $\tilde{\gamma}$.

Additionally, we assume that a map point belongs to the static background if it is classified as static for most of keyframes that can observe this map point:

$$\tilde{\beta}_j = \phi\left(\lambda_\phi, \frac{\sum_{(i,k)\in\mathcal{K}^j} \gamma_{ik}}{n(\mathcal{K}^j)}\right), \tag{57}$$

where $\mathcal{K}^j$ is defined in Equation (51). $\phi(\lambda_\phi, x)$ denotes $max(0, \frac{x-\lambda_\phi}{1-\lambda_\phi})$, which is inspired by ReLU [2]. $\lambda_\phi \in [0, 1)$ is a parameter and is chosen as 0.5 in implementation, which means a map point j is dynamic if more than 50% of clusters in $\mathcal{K}^j$ classifies the map point j as dynamic.

The last term $R_{imu}(\hat{S})$ adds regularisation on the IMU biases and is defined as:

$$\lambda_{imu} \sum_{i=1}^{M} \left(||\mathbf{b}_i^g - \mathbf{b}_{i-1}^g||^2_{\Sigma_{bg}} + ||\mathbf{b}_i^a - \mathbf{b}_{i-1}^a||^2_{\Sigma_{ba}} \right). \tag{58}$$

Following [25], we assume that the IMU biases are changing slowly over time and can be modelled with a Brownian motion. We, therefore, penalise the difference of IMU biases $\mathbf{b}^a$ and $\mathbf{b}^g$ for any two consecutive keyframes.

The novelty of our robust visual-inertial BA (Equation (50)) is that we actively estimate dense segmentation of input images $\hat{\Gamma}$ and sparse segmentation of map points. This is different to StaticFusion [94] or PlanarFusion [55] which only consider the segmentation of images. We can, therefore, recover a static sparse map and correct IMU biases even if the initial dense segmentation of images are unreliable. The static map and corrected IMU biases can then be used to aid dense segmentation of images in the presence of long-term large occlusion.

To solve the optimisation for a large map, we follow ORB-SLAM3 [11] and consider a sliding window of keyframes with their corresponding map points. We also incorporate observations of these points from covisible keyframes. To optimise Equation (50), we first initialise the state and dense segmentation of the latest keyframe and set $\beta = 1$ for all map points. For each iteration, we fix $\hat{\mathbf{\Gamma}}$ and $\hat{\mathbf{B}}$ while finding the optimal states $\hat{\mathbf{S}}$ for $M+1$ keyframes and and map point position $\hat{\mathbf{X}}$ for L map points. Then, we fix $\hat{\mathbf{S}}$ and $\hat{\mathbf{X}}$, while $\hat{\mathbf{\Gamma}}$ and $\hat{\mathbf{B}}$ are iteratively optimised by fixing one and analytically solving the other. After optimisation, we remove all dynamic map points.

6.2.3 *Initialisation of Segmentation and Image Frame State*

For every two consecutive frames $i-1$ and i, given the state of the previous frame $\mathbf{S}_{i-1}$, we estimate the state of current frame $\mathbf{S}_i$ and a cluster-wise dense segmentation $\mathbf{\Gamma}_i$ of dynamic objects.

Similar to ORB-SLAM [60], we heuristically select a frame as a keyframe to improve the robustness and accuracy of the SLAM method. For example, to avoid the loss of tracking, a new keyframe should have enough overlap area

with the previous keyframe. In addition, to avoid information redundancy, the proportion of the overlap area is lower than a threshold (75%).

If the frame is selected as a keyframe, we use the segmentation and state as the initialisation of the robust BA. The initialisation is a special case of our robust BA such that Equation (50) is a multiple-frame-to-model optimisation while Equation (59) is a single-frame-to-frame optimisation. Therefore, we similarly propose to minimise a cost function with four terms:

$$\begin{aligned} &\min_{\mathbf{S}_i,\mathbf{\Gamma}_i} \underbrace{||\mathbf{r}^I_{i-1,i}||^2_{\Sigma^{-1}_{i-1,i}}}_{IMU} + P^{ini}(\mathbf{S}_i,\mathbf{\Gamma}_i) + \underbrace{R^{ini}_{seg}(\mathbf{\Gamma}_i) + R^{ini}_{imu}(\mathbf{S}_i)}_{Regularisation}, \\ &\text{s.t. } \gamma_{ik} \in [0,1] \ \forall k, \end{aligned} \tag{59}$$

where the first term is the IMU residual term. However, the second term $R_{ini}(\mathbf{S}_i,\mathbf{\Gamma}_i)$ minimises the residuals between the two consecutive dense intensity and depth image pairs (I_{i-1}, D_{i-1}) and (I_i, D_i) so that features from texture-less areas can be considered – a crucial departure from Equation (51). Similar to Equation 26 and 41, $P^{ini}(\mathbf{S}_i,\mathbf{\Gamma}_i)$ is defined as a weighted sum of intensity and depth residuals:

$$\sum_{\mathbf{u}\in U_i} \gamma(\mathbf{u})[F(\alpha_I w^{\mathbf{u}}_I r^{\mathbf{u}}_I(\Delta\mathbf{T}_{Ci})) + F(w^{\mathbf{u}}_D r^{\mathbf{u}}_D(\Delta\mathbf{T}_{Ci}))] \ , \tag{60}$$

where U_i is the set of pixels with a valid depth reading at the current image frame i and $\mathbf{u} \in \mathbb{R}^2$ is the pixel coordinate. $\Delta\mathbf{T}_{Ci} = (\mathbf{T}_{i-1}\mathbf{T}_{BC})^{-1}\mathbf{T}_i\mathbf{T}_{BC}$ denotes the relative camera pose between two consecutive frames. Given a pixel coordinate $\mathbf{u}$, the intensity residual $r^{\mathbf{u}}_I(\mathbf{T})$ and depth residual $r^{\mathbf{u}}_D(\mathbf{T})$ under a transformation $\mathbf{T} \in SE(3)$ can be acquired by:

$$r^{\mathbf{u}}_I(\mathbf{T}) = I_{i-1}\left(\mathcal{W}(\mathbf{u},\mathbf{T})\right) - I_i\left(\mathbf{u}\right) \tag{61}$$

$$r^{\mathbf{u}}_D(\mathbf{T}) = D_{i-1}\left(\mathcal{W}(\mathbf{u},\mathbf{T})\right) - |\mathbf{T}\pi^{-1}(\mathbf{u}, D_i\left(\mathbf{u}\right))|_z \ , \tag{62}$$

where $I_i(\mathbf{u})$ and $D_i(\mathbf{u})$ provide the intensity and depth value of the pixel position $\mathbf{u}$ respectively. $\mathcal{W}$ is the pixel-wise image warping function:

$$\mathcal{W}(\mathbf{u},\mathbf{T}) = \pi\left(\mathbf{T}\pi^{-1}(\mathbf{u}, D_t(\mathbf{u}))\right). \tag{63}$$

In Equation (60), α_I is chosen as 0.15 to re-scale the intensity residual so that it has a similar scale to the depth residual. Parameters w_I and w_D are used to weight the residuals of intensity and depth respectively to penalise measurement noise (σ_I and σ_D), discontinuity, which is similar to SF [94]:

$$w_I = \frac{1}{\lambda_I \sigma_I^2 + |\nabla_{\mathbf{u}} I_i|}, \; w_D = \frac{1}{\lambda_D \sigma_D^2 + |\nabla_{\mathbf{u}} D_i|}. \tag{64}$$

We also use Cauchy robust penalty: $F(r) = \frac{c^2}{2} \log\left(1 + \left(\frac{r}{c}\right)^2\right)$ to improve the robustness of residual minimisation and c is the inflection point of $F(r)$.

The third term of Equation (60) is the same as Equation (54) and $R^{ini}_{seg}(\mathbf{\Gamma}_i) = R_d(\mathcal{X}, i)$. This is because we only consider the dense segmentation $\mathbf{\Gamma}_i$ of the i-th frame and treat all map points as static: $\beta_j = 1, \forall j$. Similarly, the last term $R^{ini}_{imu}(\mathbf{S}_i)$ can be acquired by assigning $M = 1$ in Equation (58).

A coarse-to-fine scheme similar to StaticFusion [94] is applied to align dense intensity or depth images in the solver of Equation (59). Concretely, for each incoming RGB-D image pair, we create image pyramids for both intensity and dense images by iteratively resizing the image to the half-size of the previous level. We start the minimisation from the coarsest level and initialise the next level using the intermediate results from the current level. In addition, the frame state $\mathbf{S}_i$ and dense segmentation $\mathbf{\Gamma}$ are decoupled in the solver. For each iteration, we first fix the dense segmentation $\mathbf{\Gamma}$ and optimise $\mathbf{S}_i$. We then find an analytical solution of $\mathbf{\Gamma}_i$ given the value of $\mathbf{S}_i$.

If the current frame i is selected as a new keyframe, we run our visual-inertial BA (Equation (50)) where the estimated $\mathbf{\Gamma}_i$ and $\mathbf{S}_i$ are used to initialise the segmentation and state respectively.

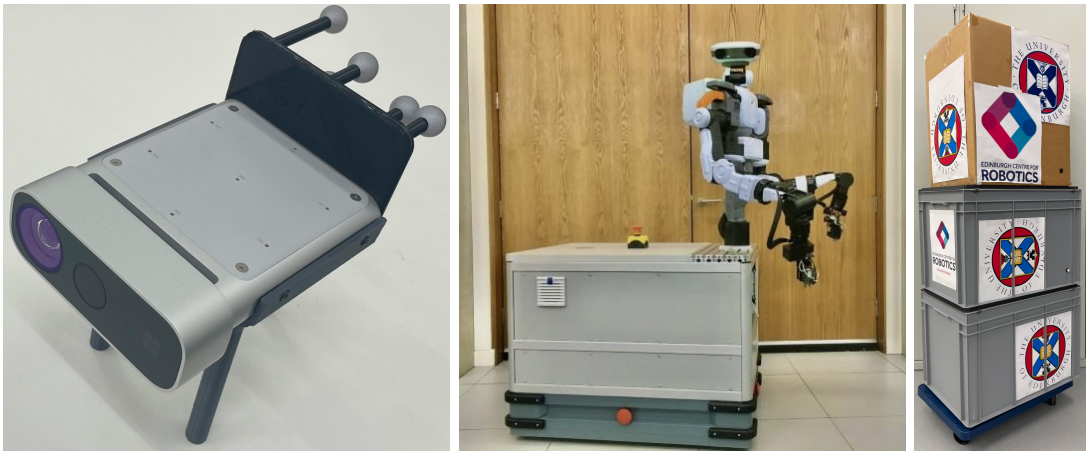
6.2.4 *Place Recognition and Loop Closing*

We adopt a similar place recognition policy to ORB-SLAM3 [11] and use the DBoW2 place recogniser [62] to detect loop candidates based on their appearance. Because of large occlusion, to improve the accuracy of place recognition, we remove the dynamic regions of each keyframe and only consider candidate keyframes when the region of static background is more than 80% of the whole image.

After verification of loop closure matches, we conduct a full vision-only BA based on the static parts of keyframes and all static map points to reduce long-term drift.

6.3 EVALUATION

6.3.1 *Setup*



(a) Azure Kinect DK (b) Wheeled Robot (c) Boxes

Figure 35: (a) An Azure Kinect DK RGB-D camera with attached Vicon markers. (b) The base of an omnidirectional wheeled mobile manipulator on which the camera is mounted. (c) A large rigid object that can be moved by humans to cause large occlusion in the camera view.

We collect evaluation sequences using an Azure Kinect DK RGB-D camera with an embedded low-cost IMU (Figure 35a) . The ground truth trajectories are acquired through Vicon system by attaching Vicon markers to the camera. The Azure Kinect DK can generate registered 1280 $\times$ 720 RGB-D image pairs at the frequency of 30 Hz and IMU readings at around 1700 Hz. To speed up the processing of image frames, we down-scale and crop the RGB-D images to the resolution of 640 x 480 (VGA). In the solver of Equation (59), the images are further down-scaled to 320 x 240 (QVGA) because of dense image alignment.

During data collection, the camera is mounted on an omnidirectional wheeled robot (Figure 35b) and a human moves the stacked boxes (Figure 35c) closely in front of the camera to create large occlusion in the camera view. We collect nine sequences with different camera and object trajectories and they can be categorised into three types (Table 7) based on the proportion of large

occlusion (LO) duration to the whole sequence: short-term (ST), mid-term (MT) and long-term (LT). For each category, we visualise the camera trajectory of one typical collected sequence and highlight the trajectory in red when the camera view is occluded by dynamic objects (Figure 36). For example, in *seq7*, the majority of camera view is occluded for more than 70% of image frames and the camera travels around 20 meters during this period, which increases the difficulty to localise camera. For quantitative evaluation, we follow [105] and calculate the absolute trajectory error (ATE) and the relative pose error (RPE) against the ground truth camera trajectories.

In addition, we also evaluate the localisation performance on a real-world dataset OpenLORIS-Scene [98]. It contains RGB-D-inertial sequences from 5 different scenes, including static (*office*), texture-less (*corridor*) and complex dynamic (*home*, *cafe* and *market*) environments. We remove the sequences *corridor1-3* and *corridor1-4* because of dim lighting.

When compare against other baseline methods [11, 74], we tune their hyper-parameters on static sequences we collected, because the environment in our collected dataset is different from other benchmarks. For example, ORB-SLAM3 [11] and VINS-Mono [74] have achieved high localisation accuracy in outdoor environments, while our dataset was collected in indoor environments.

	Types	Dis. (m)	LO Dis. (m)	Durat. (s)	LO Durat. (s)
1	Static	22.1	0 (0%)	108	0 (0%)
2	ST	15.6	3.09 (19.8%)	58.1	10.2 (17.6%)
3		19.3	3.24 (16.8%)	59.0	9.45 (16.0%)
4	MT	17.2	5.83 (33.9%)	49.9	14.7 (29.5%)
5		17.5	6.01 (34.3%)	62.1	18.6 (30.0%)
6	LT	21.9	14.9 (68.0%)	85.8	48.7 (56.8%)
7		26.5	20.1 (75.9%)	131	97.5 (74.4%)
8		34.1	21.5 (63.1%)	189	101 (53.4%)
9		27.1	21.2 (78.2%)	140	102 (72.9%)

Table 7: Statistics of nine collected sequences. "Static" means there is no dynamic objects in this sequence. Large occlusion (LO) distance or duration represents the distance or duration when the camera view is occluded respectively. Specifically, LTLO means the duration of large occlusion is longer than 50% of the whole sequence duration.

6.3.2 *Camera Localisation*

We first evaluate our method on our collected dataset, and estimate absolute trajectory errors (ATEs) and relative pose errors (RPEs) against ground truth camera trajectories. The results are compared with SF [94], PF [55], VINS-Mono [74], ORB-SLAM3 [11] and Dynamic-VINS [54] in Table 8. In static environments, our method achieves comparable results with other state-of-the-art visual-inertial SLAM methods.

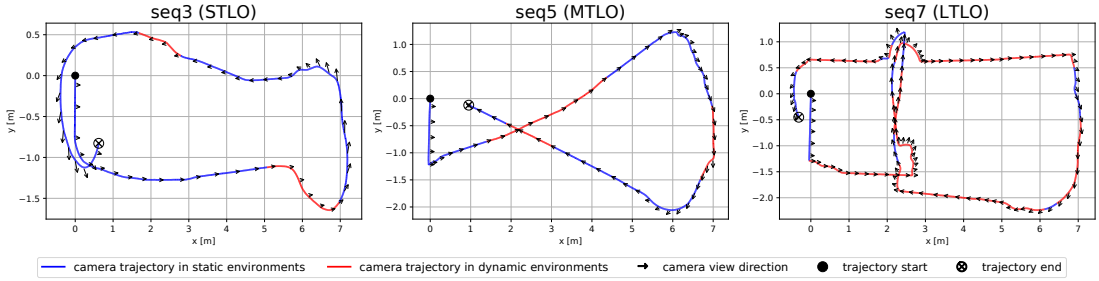


Figure 36: The camera ground truth trajectories from top-down perspective. The blue trajectory segment illustrates the part when there are no moving objects in the camera view. While the red segment represents that dynamic objects can be observed in the camera view. The start position of a trajectory is marked with a black solid dot and the end position is marked with a circle-cross marker. Finally, the black arrows point in the direction of camera view.

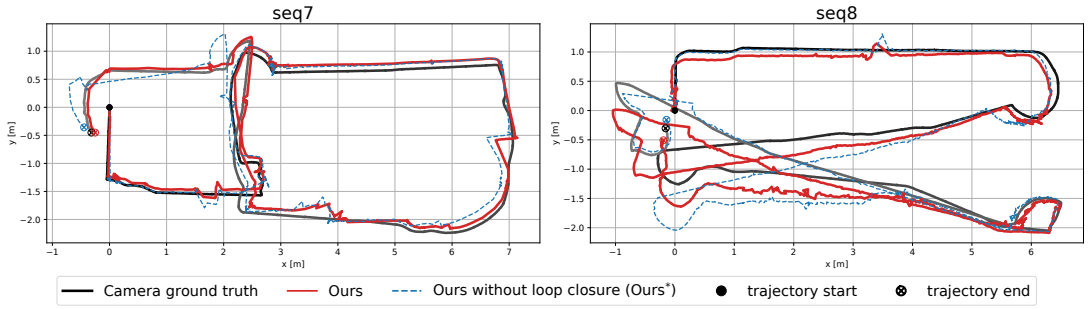


Figure 37: Visualisation of the estimated camera trajectories compared with the ground truth. We align the start position of all trajectory to the same point which is marked with a solid black dot. The colour of the ground truth trajectories gradually changes from black at the start to grey at the end. Results show that our method can robustly handle large occlusion in the camera view and is able to recover correct camera trajectories after drift caused by large occlusion.

Both visual SLAM methods, StaticFusion and PlanarFusion, have a relative high error in static environments, because they are unable to reduce long-term drift with loop closure. In the scenario of LTLO (seq. 6-9), evaluation

		Visual SLAM				Visual-inertial SLAM									
		SF* [94]		PF* [55]		VINS-Mono [74]		ORB-SLAM3 [11]		Dynamic-VINS [54]		Ours*		Ours	
		ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE
1	Static	2.05	0.169	1.97	0.161	0.081	0.025	**0.058**	**0.021**	0.072	0.024	0.156	0.020	0.069	0.022
2	STLO	1.41	0.433	1.18	0.202	>6.0	2.112	0.873	0.321	0.747	0.306	0.170	0.0483	**0.159**	**0.0471**
3		>6.0	3.15	3.40	0.229	5.044	0.473	1.019	0.503	0.949	0.372	0.203	**0.0470**	**0.081**	0.0573
4	MTLO	1.76	0.683	1.78	0.238	>6.0	>6.0	1.934	0.402	1.973	0.375	**0.151**	**0.0319**	0.168	0.0342
5		1.66	0.533	1.54	0.176	1.234	0.345	1.584	0.441	1.181	0.232	**0.178**	**0.0251**	0.182	0.0289
6	LTLO	5.07	1.35	4.02	0.165	>6.0	>6.0	2.381	0.559	2.923	0.527	0.212	0.0771	**0.153**	**0.0557**
7		>6.0	3.46	5.36	0.148	>6.0	0.749	2.049	0.643	1.278	0.608	0.204	0.0844	**0.123**	**0.0727**
8		>6.0	2.07	5.21	0.137	>6.0	1.95	2.661	0.547	2.303	0.518	0.295	**0.0420**	**0.191**	0.0440
9		>6.0	4.49	4.23	0.153	>6.0	3.09	1.995	0.618	2.986	0.628	0.533	**0.0665**	**0.171**	0.0815

Table 8: ATE (m) and RPE RMSE (m/s) for all nine collected sequences. The asterisk (∗) symbol means either the method is unable to close loops or the loop thread is disabled. Our method outperforms all other state-of-the-art methods when the large occlusion lasts for a long period in the camera view. While in static environments, our method has comparable results to other visual-inertial SLAM methods.

demonstrates that our method is able to provide accurate camera trajectories and outperforms all other methods (Figure 37). While neither ORB-SLAM3 nor Dynamic-VINS is able track camera trajectory correctly. This is because ORB-SLAM3 is unable to remove sparse features from dynamic objects and Dynamic-VINS can only remove features from specific categories of dynamic objects like humans. In the STLO and MTLO sequences (seq. 2-5), the state-of-the-art methods have better performance compared to their performance on LTLO sequences, however, our method achieves more accurate results. We further disable the loop closure thread of our method to quantify the accuracy improvement from the dynamic object removal. Results show that the localisation precision decreases as the duration of large occlusion increases.

Additionally, we evaluate our method on OpenLORIS-Scene [98] dataset (Table 9). Results show that our method is able to achieve high localisation accuracy in static environments (*cafe*) and has comparable results with Dynamic-VINS which relies on semantic segmentation in highly dynamic environments (*home*, *cafe* and *market*). However, our method's estimated camera trajectories have a relatively higher ATE in environments with large regions of textureless objects (*corridor*) because we rely on a combination of sparse and dense features.

	office	corridor	home	cafe	market
VINS-Mono [74]	0.193†	3.413†	0.541†	0.486†	1.664†
VINS-RGBD [96]	0.110†	2.412†	2.274†	0.389†	1.817†
ORB-SLAM3 [11]	**0.103**	3.453	0.571	0.359	2.234
Dynamic-VINS [54]	0.110†	**2.367**†	**0.280**†	0.419†	**1.185**†
Ours	0.109	3.279	0.291	**0.349**	1.227

Table 9: ATE RMSE (m) of estimated camera trajectories on OpenLORIS-scene [98] dataset. The dagger (†) symbol represents that the result is taken from the original paper [54].

6.3.3 *Dynamic Object Segmentation*

Figure 38 shows the static/dynamic segmentation results of *seq7* and the majority of camera view is continuously occluded for more than 40 seconds from the time frame 500 to 1746. We compare our segmentation results against StaticFusion (SF) [94], Co-Fusion (CF) [84] and PlanarFusion (PF) [55]. Results show that SF is unable to detect dynamic objects when they occupy the major proportion of the visual input. In addition, the segmentation provided by CF is incomplete and parts of dynamic objects are classified as the static background. Although PF achieves better segmentation results than SF and CF, it is unable to provide accurate results consistently over a long period of time because only two consecutive image frames are used. Our method takes advantage of local optimisation over multiple keyframes and can therefore detect multiple dynamic objects that cause long-term large occlusion.

We further quantitatively evaluate the intersection over union (IoU) of the static background segmentation results. We remove humans using Mask R-CNN [33] and manually remove planes that belong to the dynamic box to obtain static background segmentation ground truth. Results (Figure 39) demonstrate that our method provides more accurate static background segmentation than PlanarFusion [55].

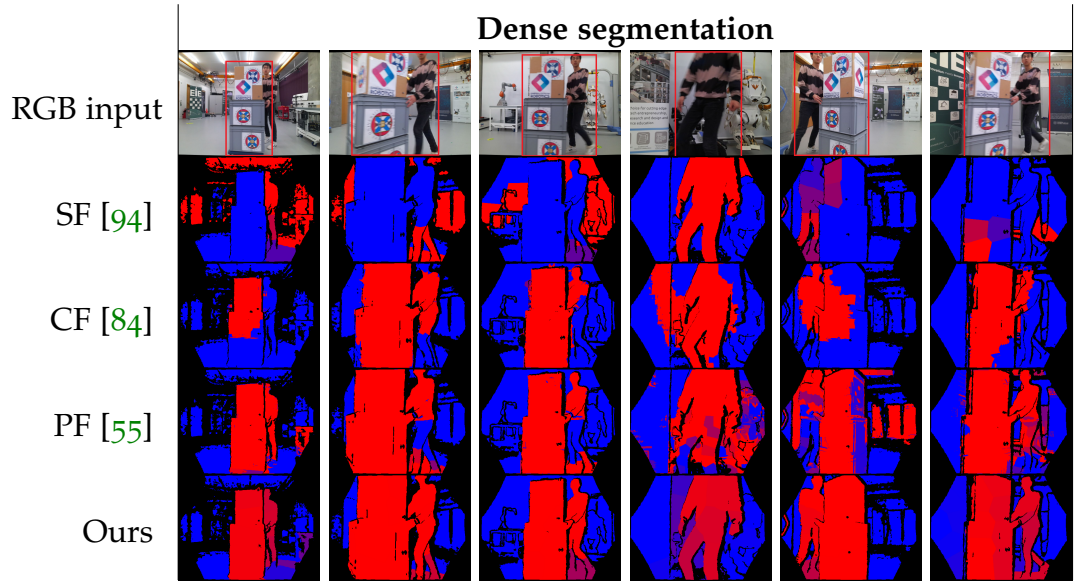


Figure 38: Segmentation result of the static background (blue) and dynamic objects (red) in *seq7*. In the first row, we show the input RGB images and their corresponding time frame ID. The dynamic objects are manually highlighted by red rectangles for better visualisation. Results show that only our method can provide a consistent segmentation of objects that cause large occlusion for a long period of time. In contrast, both SF and CF are unable to segment the dynamic objects correctly, while the segmentation performance of PF is not persistent over time.

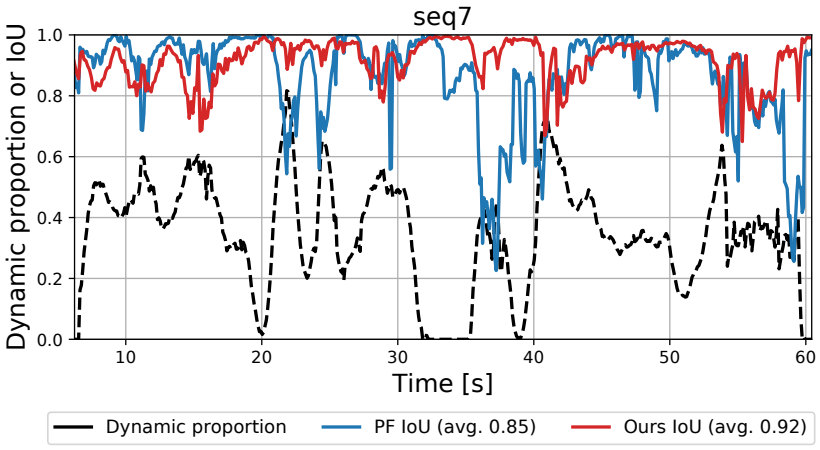


Figure 39: The IoU of the static background segmentation from our method (average 0.92) and PF (average 0.85) for a part of seq7 when dynamic large occlusions last over a long period. The black dashed line illustrates the proportion of dynamic objects to all pixels with a valid depth reading.

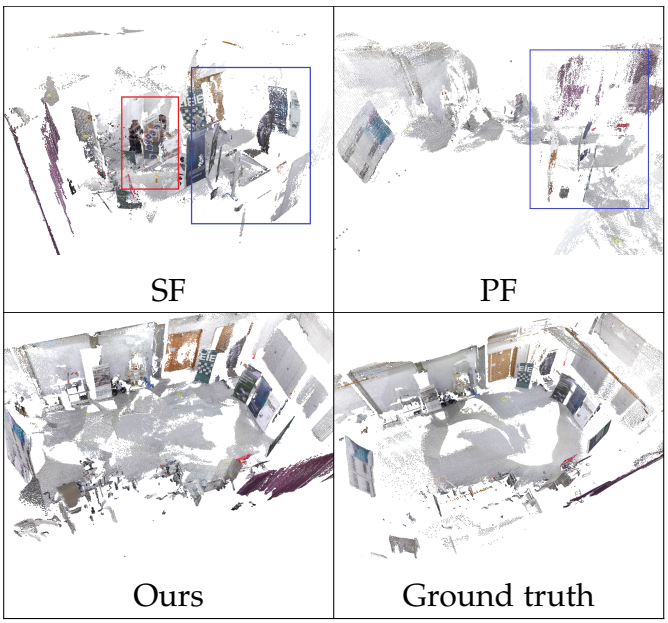


Figure 40: Reconstruction results of the RGB-D sequence 7. We highlight the dynamic objects with red rectangles and the dislocation of static objects with blue rectangles. SF can neither remove dynamic objects nor estimate camera trajectory correctly. In contrast, both PF and our method can detect dynamic objects but PF is unable to accurately localise camera after the removal of dynamic objects.

6.3.4 *Background Reconstruction*

We qualitatively compare our dense background reconstruction results with SF and PF (Figure 40). We reconstruct the background after processing the whole sequence with the estimated dense segmentation and camera trajectory. The ground truth model is acquired by mapping the static environment to the ground truth camera trajectory. Results show that SF maps dynamic objects into the static background model, while PF is able to remove dynamic objects from the model. However, the dislocation of static objects indicates that neither SF nor PF is able to estimate camera ego-motion correctly. Based on consistent dynamic object segmentation and camera localisation, our method outperforms other methods and provides a correct background reconstruction.

6.4 REAL MOBILE MANIPULATION EXPERIMENT

Finally, we evaluate our method on a real-world mobile manipulation experiment in a complex cluttered environment. We also introduce dynamic objects

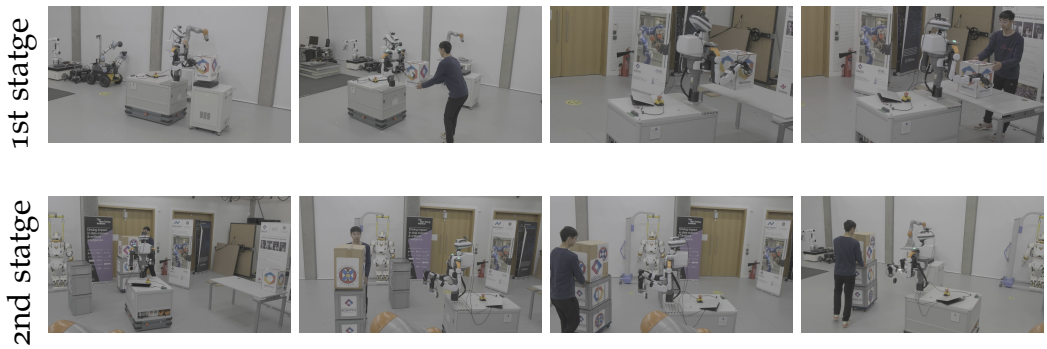


Figure 41: Visualisation of collected sequences from a third-person perspective. During the 1st stage, the mobile manipulator manipulates an object closely while a human works around the robot. In the 2nd stage, the robot returns back to its starting position and a human pushes a large box in front of the camera view, which causes large dynamic occlusions.

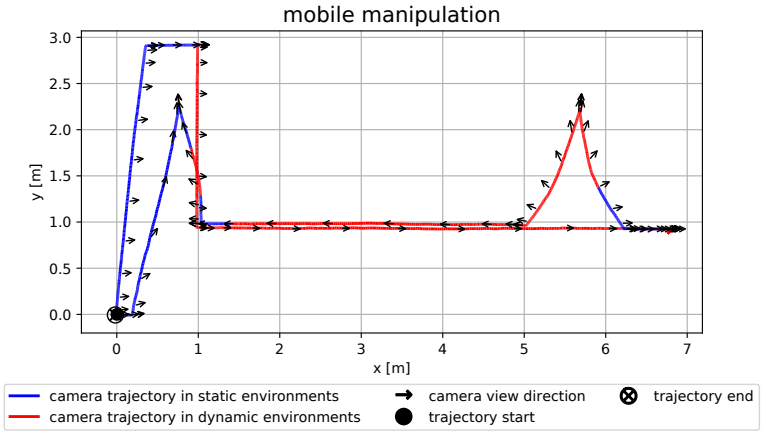


Figure 42: The camera ground truth trajectories from a top-down perspective. 1710 among 3285 (52%) images are occluded by dynamic objects.

in this environment to simulate realistic scenarios. The sequence is collected by the Azure Kinect DK (Figure 35a) mounted on the top of the robot's head and the ground truth camera trajectories are obtained from the Vicon system.

We visualise the sequence from a third-person perspective in Figure 41. The whole sequence can be divided into two stages. In the first stage, the robot picks up a box, transports it to another table and put it down while another human moves in front of the robot. In the second stage, the robot returns back to its previous location while multiple dynamic objects (humans and boxes) cause long-term dynamic large occlusions. We further visualise the camera ground truth trajectory in Figure 42 and around 52 percent of the images are occluded by dynamic objects.

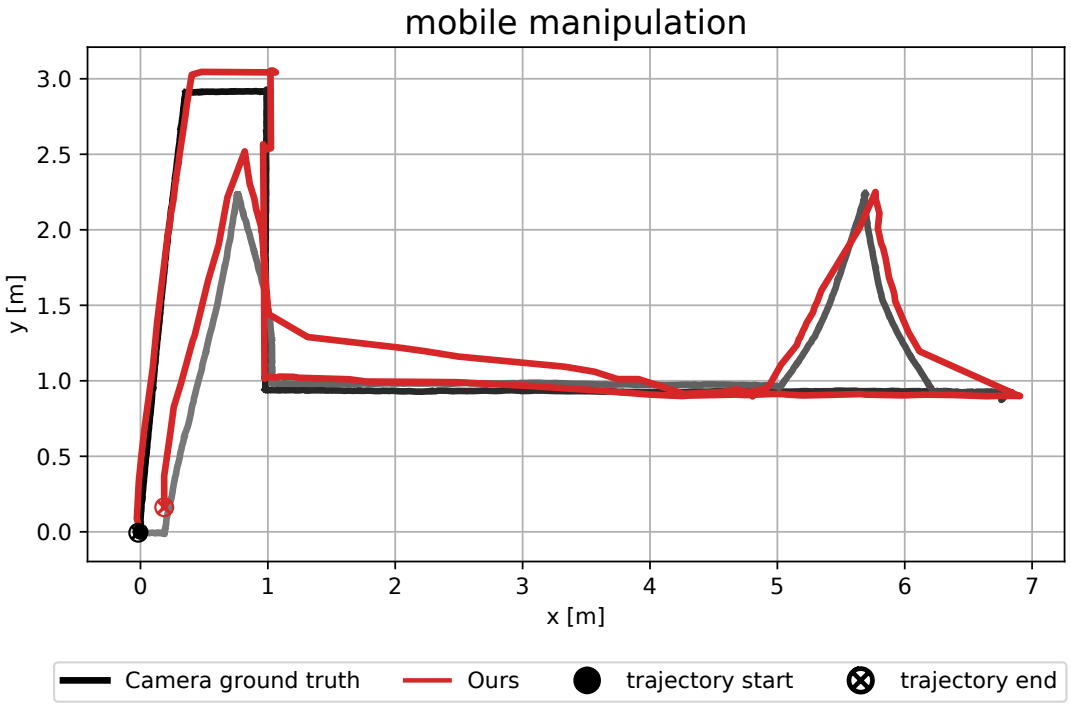


Figure 43: Comparison of our estimated camera trajectory (red) to the ground truth (grey to back). The absolute trajectory error (ATE) root mean squared error (RMSE) of the estimated trajectory is 0.152 m.

We first evaluate the performance of localisation (Figure 43). Results show that our method is able to handle long-term large occlusion in a real-world mobile manipulation scenario and can help robots to accurately localise themselves.

We then evaluate the dense segmentation of our method. In Figure 44, we visualise the cluster-wise dense segmentation of our method in both two stages. In the first stage, when the robot closely manipulates a box, our method is able to detect the manipulated box as dynamic even when it causes large occlusion of camera view. During transportation, the manipulated box is static relative to the camera but we can still detect it due to the use of inertial measurement unit (IMU) measurements. In the second stage, a human moves a large box in front of the camera view and causes large occlusion, our method can also detect dynamic objects correctly.

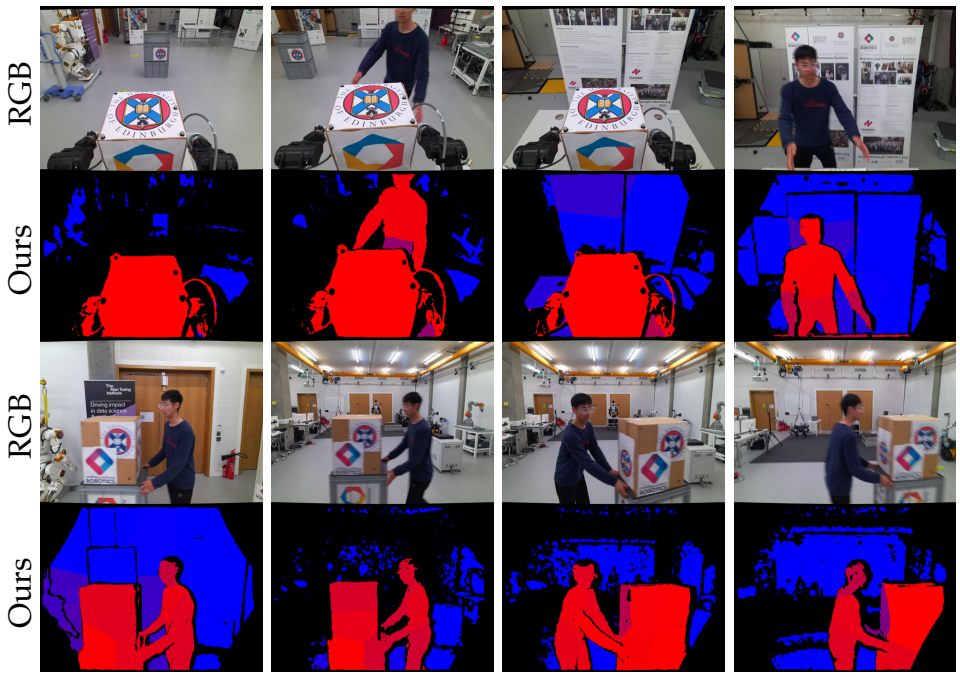


Figure 44: Cluster-wise dense segmentation of our method, where dynamic regions are visualised as ref while static regions are visualised as blue. The first two rows show the RGB images and segmentation results during the 1st stage as described in Figure 41 and the last two rows show the results of the 2nd stage.

6.5 CONCLUSION

In this chapter, we present a novel RGB-D-inertial SLAM method that is robust to multiple undefined dynamic objects that cause long-term large occlusion. Our proposed robust visual-inertial bundle adjustment can simultaneously estimate the dense segmentation of dynamic objects and localise the camera with a combination of dense and sparse features. The dense segmentation can be used to reconstruct the background. The detailed evaluation demonstrates that our proposed approach outperforms other state-of-the-art methods in terms of the dense object segmentation, camera localisation and background reconstruction in the presence of long-term large occlusion.

However, our current method can fail to detect dynamic objects in outdoor or texture-less environments and is unable to track or model dynamic objects. Our future work will aim to address long-term object tracking and modelling, building on the gains realised by the proposed method. We also plan to extend the current method to large-scale outdoor environments.

7

CONCLUSION AND FUTURE WORK

In this thesis, we have presented our latest work in the area of dynamic RGB-D simultaneous localisation and mapping (SLAM), focusing on enabling accurate localisation when the majority of the camera view is occluded by multiple undefined dynamic objects over a long period (Table 10). In our first work, we simplify the scenario and assume there is only one rigid dynamic object in the scene. We then extend our first work to independently track multiple dynamic objects in planar environments. In our third contribution, we consider the scenario when the large occlusion in the camera view lasts over a long period of time and use a low-cost inertial measurement unit (IMU) in a tightly-coupled way to improve localisation robustness and accuracy. Finally, we demonstrate the performance of our method in a real mobile manipulation platform.

	SF [94]	ORB-SLAM3 [11]	RF (ch.4)	PF (ch.5)	LTLO-Fusion (ch.6)
large occlusion (LO)			✓	✓	✓
long-term LO					✓
multiple dynamic objects	✓			✓	✓
large-scale localisation		✓			✓
dense segmentation	✓		✓	✓	✓

Table 10: Comparison between our three contributions and the baseline SLAM methods.

The main outcome of our work in this thesis is an RGB-D SLAM pipeline that is robust to long-term large occlusion with the aid of robot proprioception, such as IMUs or wheel odometry. We have tested our approaches on multiple robot platforms and compared them with other state-of-the-art methods. The detailed evaluation demonstrates that our method is versatile and robust to challenging dynamic environments. We are able to use camera motion priors with a high drift to help separate the static background from dynamic objects and then reduce the drift of camera motion prior with visual odometry. We can also separate multiple undefined planar objects by their different motions. Finally, we show that our method can achieve accurate localisation and mapping in a real-world mobile manipulator experiment.

Additionally, our research work has profound implications for both the industry and academic communities. First, this thesis addresses a challenging real-world SLAM problem in dynamic environments. Our proposed methods enable robots to accurately localise themselves without having prior knowledge of dynamic objects nor assuming that dynamic objects only account for a minor proportion of the camera view. In addition, our method's robustness to long-term large occlusion can extend the application range of robots, particularly when multiple robots work together for a long period. Importantly, our novel formulation for the multimotion segmentation problem can serve as inspiration for other research works.

The limitations of our current approach will be discussed in the following chapter. We will also provide potential ways to resolve these limitations and propose future work.

7.1 LIMITATIONS

There are four main limitations of our current framework.

7.1.1 *Limited to indoor environments*

All three publications are based on RGB-D cameras which have a limited operational range of approximately 0.25 to 5.5 meters[1]. In most scenarios, the dimension of the indoor environments falls within the operational range of depth cameras. However, in outdoor environments [28], the depth can exceed the depth range of a depth camera for the majority of areas in the camera view, resulting in a lack of depth reading.

The depth information is vital to our current framework. For example, our first work RigidFusion (RF) minimises depth residual to estimate camera motions between frames. In addition, our second work extracts planes from depth maps and our last work uses the depth difference between the foreground and background to segment dynamic objects. Therefore, our current framework is unable to achieve desirable performance in outdoor environments.

To resolve this limitation, we can replace the RGB-D cameras with stereo cameras which can estimate the depth of feature points based on triangulation.

1 https://learn.microsoft.com/en-us/azure/kinect-dk/hardware-specification

To increase the density of depth reading, we can use deep-learning methods [29] to inpaint the depth of pixels in depth-missing areas.

7.1.2 *Limited to planar object tracking*

Our second work has extended the first work to track multiple dynamic objects independently. However, it is limited to tracking objects that are comprised of planes. There are many objects commonly found in indoor environments that are not composed of planes, like chairs, sofas or the surface of a cluttered desktop. Importantly, our method is unable to track humans but removes them as outliers.

7.1.3 *Underestimation of the variety of dynamic environments*

In our work, we consider dynamic objects that are constantly moving in front of the camera view. These environments are classified as highly dynamic environments where the movement of dynamic objects is continuous and can be directly observed by the camera.

However, in addition to highly dynamic environments, there are two other categories of dynamic environments. The first is low dynamic environments where dynamic objects move beyond the camera view and remain unobserved during the movement. For example, when a car leaves a parking lot and returns, the other cars in the parking lot may have changed positions during its absence. None of our methods considers low dynamic objects because we detect dynamic objects by their distinct motions compared to static objects.

The other is intermittent dynamic environments where the movement of dynamic objects is discontinuous. Since our method detects a dynamic object by its different motion against the static background, if the status of an object frequently alternates between dynamic and static, our method has difficulty in object segmentation and background reconstruction.

7.1.4 *Requirement of robot proprioception*

Our previous two contributions require either synthetic camera motion priors from ground truth camera trajectories or real camera motion priors from wheel

odometry. Our third contribution only requires camera motion priors from a low-cost IMU. However, an extra proprioceptive sensor is still required in addition to visual sensors. This can limit the scalability of our method to scenarios where only visual sensors are accessible.

7.2 FUTURE WORK

7.2.1 *Extending our method to outdoor environments with multi-modality*

We plan to integrate light detection and ranging (LiDAR) sensors with our visual-inertial system because there are multiple advantages of LiDAR sensors compared to depth cameras. First, a LiDAR sensor has a longer operational range and can even be up to 100 to 200 meters [76], which surpasses the operational range of depth cameras. This makes LiDAR more suitable for outdoor environments. In addition, LiDAR provides a wider field of view than the depth camera. Therefore, LiDAR sensors can be more robust to large occlusions.

7.2.2 *Exploring high-level features*

In our future work, we plan to extend our previous work to independently track multiple non-planar dynamic objects. Instead of segmenting images into super-pixels [1] or clusters [56] and then merging them into different rigid bodies, we propose to take advantage of recent progress in scene understanding [47]. The recent Segment Anything (SAM) system [47] is able to generate a valid dense mask with segmentation prompts. The segmentation prompts can be either semantic labels of a specific object or a single point. Consequently, for unmodeled objects, we can separate them from others with the scene parsing methods. Compared to super-pixels, the segmentation from the scene parsing methods can provide a finer boundary of individual objects. We plan to adopt our multimotion segmentation methods on the finer segmentation and enable multiple unmodeled object tracking.

7.2.3 *4D reconstruction of dynamic environments*

4D reconstruction provides a changing dense model of the environment [63, 127] or a specific object [64, 68, 111], such as a human, over time. In a complex dynamic environment, moving objects can be temporally static while static objects can be moved again. Therefore, a 3D reconstructed model is inadequate to describe this changing environment and we propose to reconstruct a 4D dense model. Wong et al. [127] proposes a 4D reconstruction system for dynamic environments, however, only one rigid dynamic object is allowed to move at one frame of time. Our future work can involve proposing a dynamic SLAM system capable of reconstructing a 4D model of the dynamic environment with multiple unmodeled dynamic objects.

7.2.4 *Representation of dense models*

Compared to traditional explicit dense model representation, the implicit dense representation neural radiance field (NeRF) [59] is able to generate rendered images from a completely new perspective. Importantly, recent developments of NeRF have addressed many limitations of the original NeRF proposed in [59]. FastNeRF [27] is able to render a high-resolution view of objects at around 200 Hz on a Nvidia RTX 3090 GPU, while the original NeRF takes more than 10 seconds to render an image of the same resolution. In addition, D-NeRF [72] proposes a novel neural radiance field for a single dynamic object or simple dynamic scene. Nice-SLAM [135] also uses NeRF to reconstruct the static environment in a SLAM system. However, none of the previous works is capable of representing a complex dynamic scene with NeRF and rendering novel views of images at a real-time speed. Therefore, in our future work, we plan to investigate how to represent the dense model of a large-scale cluttered environment with a NeRF, which can be used to inpaint the significant occluded areas caused by dynamic objects.

7.2.5 *Handling dynamic large occlusions with only visual sensors*

In this thesis, the dynamic large occlusion represents that the major proportion of the camera view is occluded by dynamic objects. However, when this large

occlusion is caused by closely moving objects, these dynamic objects can still account for a smaller proportion in the bird's eye (or orthographic) view [3] of the scene than the static background. Therefore, to detect dynamic objects that cause large occlusions in the camera view based on visual sensors only, we can transform images acquired from cameras into bird's eye view (BEV) images. The dynamic objects are assumed to occupy a minor proportion in the BEV images.

7.2.6 *Handling dynamic large occlusions during multi-robot collaboration*

When multiple robots closely collaborate together, one robot (A) can cause large dynamic occlusion in the camera view of another robot (B). In this scenario, robot B can track robot A, which provides camera motion priors for robot A. Importantly, robot A can be treated as a dynamic object in the camera view of robot B. Therefore, the ego-motion estimation of robot A can be used as an object motion prior. In our first contribution (Chapter 4), we have demonstrated that both camera and object motion priors can help differentiate the dynamic objects from the static background. In future work, we plan to use camera and object motion priors acquired from the multi-robot scenario to help improve the localisation accuracy.

BIBLIOGRAPHY

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua and Sabine Süsstrunk. 'SLIC superpixels compared to state-of-the-art superpixel methods'. In: *IEEE Trans. on Pattern Anal. and Mach. Intell.* (2012) (cit. on p. 98).

[2] Abien Fred Agarap. 'Deep learning using rectified linear units (ReLU)'. In: *arXiv preprint arXiv:1803.08375* (2018) (cit. on p. 81).

[3] Syed Ammar Abbas and Andrew Zisserman. 'A geometric approach to obtain a bird's eye view from an image'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2019, pp. 0–0 (cit. on p. 100).

[4] Herbert Bay, Tinne Tuytelaars and Luc Van Gool. 'Surf: Speeded up robust features'. In: *Lecture notes in computer science* 3951 (2006), pp. 404–417 (cit. on p. 20).

[5] Berta Bescos, Carlos Campos, Juan D Tardós and José Neira. 'DynaSLAM II: Tightly-coupled multi-object tracking and SLAM'. In: *IEEE Robotics and Automation Letters* (2021) (cit. on pp. 2, 31).

[6] Berta Bescos, José M Fácil, Javier Civera and José Neira. 'DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes'. In: *IEEE Robotics and Automation Letters* (2018) (cit. on p. 30).

[7] Michael Bloesch, Michael Burri, Hannes Sommer, Roland Siegwart and Marco Hutter. 'The two-state implicit filter recursive estimation for mobile robots'. In: *IEEE Robotics and Automation Letters* 3.1 (2017), pp. 573–580 (cit. on p. 3).

[8] Michael Bloesch, Sammy Omari, Marco Hutter and Roland Siegwart. 'Robust visual inertial odometry using a direct EKF-based approach'. In: *IEEE/RSJ international conference on intelligent robots and systems*. 2015 (cit. on p. 24).

[9] Alexey Bochkovskiy, Chien-Yao Wang and Hong-Yuan Mark Liao. 'YOLOv4: Optimal Speed and Accuracy of Object Detection'. In: *arXiv preprint arXiv:2004.10934* (2020) (cit. on p. 30).

[10] Berk Calli, Arjun Singh, James Bruce, Aaron Walsman, Kurt Konolige, Siddhartha Srinivasa, Pieter Abbeel and Aaron M Dollar. 'Yale-CMU-Berkeley dataset for robotic manipulation research'. In: *The International Journal of Robotics Research* 36.3 (2017), pp. 261–268 (cit. on p. 41).

[11] Carlos Campos, Richard Elvira, Juan J Gomez Rodriguez, Joss MM Montiel and Juan D Tardos. 'ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM'. In: *IEEE Transactions on Robotics* (2021) (cit. on pp. 1, 24, 77–80, 82, 84, 86–89, 95).

[12] Marco Camurri, Maurice Fallon, Stéphane Bazeille, Andreea Radulescu, Victor Barasuol, Darwin G Caldwell and Claudio Semini. 'Probabilistic contact estimation and impact detection for state estimation of quadruped robots'. In: *IEEE Robotics and Automation Letters* 2.2 (2017), pp. 1023–1030 (cit. on p. 3).

[13] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei and Yaser Sheikh. 'OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields'. In: *arXiv preprint arXiv:1812.08008* (2018) (cit. on p. 30).

[14] Jiyu Cheng, Hong Zhang and Max Q-H Meng. 'Improving Visual Localization Accuracy in Dynamic Environments Based on Dynamic Region Removal'. In: *IEEE Transactions on Automation Science and Engineering* (2020) (cit. on p. 31).

[15] Alejo Concha and Javier Civera. 'DPPTAM: Dense piecewise planar tracking and mapping from a monocular sequence'. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 5686–5693 (cit. on p. 23).

[16] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi and Christian Theobalt. 'Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration'. In: *ACM Transactions on Graphics (ToG)* 36.4 (2017), p. 1 (cit. on pp. 1, 22).

[17] Andrew J Davison, Ian D Reid, Nicholas D Molton and Olivier Stasse. 'MonoSLAM: Real-time single camera SLAM'. In: *IEEE Transactions on pattern analysis and machine intelligence* 29.6 (2007), pp. 1052–1067 (cit. on p. 19).

[18] Frank Dellaert, Michael Kaess et al. 'Factor graphs for robot perception'. In: *Foundations and Trends® in Robotics* 6.1-2 (2017), pp. 1–139 (cit. on p. 16).

[19] Richard Elvira, Juan D Tardós and Jose MM Montiel. 'ORBSLAM-Atlas: a robust and accurate multi-map system'. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 6253–6259 (cit. on p. 20).

[20] Jakob Engel, Thomas Schöps and Daniel Cremers. 'LSD-SLAM: Large-scale direct monocular SLAM'. In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13*. Springer. 2014, pp. 834–849 (cit. on p. 1).

[21] Maurice F Fallon, Matthew Antone, Nicholas Roy and Seth Teller. 'Drift-free humanoid state estimation fusing kinematic, inertial and lidar sensing'. In: *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE. 2014, pp. 112–119 (cit. on p. 24).

[22] Olivier D Faugeras. 'What can be seen in three dimensions with an uncalibrated stereo rig?' In: *Computer Vision—ECCV'92: Second European Conference on Computer Vision Santa Margherita Ligure, Italy, May 19–22, 1992 Proceedings 2*. Springer. 1992, pp. 563–578 (cit. on p. 26).

[23] Chen Feng, Yuichi Taguchi and Vineet R Kamat. 'Fast plane extraction in organized point clouds using agglomerative hierarchical clustering'. In: *IEEE International Conference on Robotics and Automation*. 2014. DOI: 10.1109/ICRA.2014.6907776 (cit. on pp. 22, 57, 58).

[24] Martin A Fischler and Robert C Bolles. 'Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography'. In: *Communications of the ACM* 24.6 (1981), pp. 381–395 (cit. on p. 26).

[25] Christian Forster, Luca Carlone, Frank Dellaert and Davide Scaramuzza. 'On-manifold preintegration for real-time visual-inertial odometry'. In: *IEEE Transactions on Robotics* 33.1 (2016), pp. 1–21 (cit. on pp. 79, 82).

[26] Dorian Gálvez-López and Juan D Tardos. 'Bags of binary words for fast place recognition in image sequences'. In: *IEEE Transactions on Robotics* 28.5 (2012), pp. 1188–1197 (cit. on p. 21).

[27] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton and Julien Valentin. 'FastNeRF: High-fidelity neural rendering at 200fps'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 14346–14355 (cit. on p. 99).

[28] Andreas Geiger, Philip Lenz and Raquel Urtasun. 'Are we ready for autonomous driving? the kitti vision benchmark suite'. In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 3354–3361 (cit. on p. 96).

[29] Clement Godard, Oisin Mac Aodha and Gabriel J. Brostow. 'Unsupervised Monocular Depth Estimation With Left-Right Consistency'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017 (cit. on p. 97).

[30] Aleksey Golovinskiy and Thomas Funkhouser. 'Min-cut based segmentation of point clouds'. In: *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. IEEE. 2009, pp. 39–46 (cit. on p. 30).

[31] Chris Harris, Mike Stephens et al. 'A combined corner and edge detector'. In: *Alvey vision conference*. Vol. 15. 50. Citeseer. 1988, pp. 10–5244 (cit. on p. 20).

[32] Richard I Hartley. 'Estimation of relative camera positions for uncalibrated cameras'. In: *Computer Vision—ECCV'92: Second European Conference on Computer Vision Santa Margherita Ligure, Italy, May 19–22, 1992 Proceedings 2*. Springer. 1992, pp. 579–587 (cit. on p. 26).

[33] Kaiming He, Georgia Gkioxari, Piotr Dollár and Ross Girshick. 'Mask R-CNN'. In: *Proceedings of the IEEE international conference on computer vision*. 2017 (cit. on pp. 30, 32, 66, 89).

[34] Charlie Houseago, Michael Bloesch and Stefan Leutenegger. 'KO-Fusion: dense visual SLAM with tightly-coupled kinematic and odometric tracking'. In: *IEEE International Conference on Robotics and Automation*. 2019 (cit. on pp. 24, 37).

[35] Ming Hsiao, Eric Westman, Guofeng Zhang and Michael Kaess. 'Keyframe-based dense planar SLAM'. In: *IEEE Int. Conf. on Robot. and Automat.* 2017. DOI: 10.1109/ICRA.2017.7989597 (cit. on p. 22).

[36] Guoquan P Huang, Anastasios I Mourikis and Stergios I Roumeliotis. 'On the complexity and consistency of UKF-based SLAM'. In: *2009 IEEE international conference on robotics and automation*. IEEE. 2009, pp. 4401–4408 (cit. on p. 19).

[37] Jiahui Huang, Sheng Yang, Tai-Jiang Mu and Shi-Min Hu. 'ClusterVO: Clustering moving instances and estimating visual odometry for self and surroundings'. In: *Proc. of the IEEE/CVF Conf. on Comput. Vis. and Pattern Recognit.* 2020 (cit. on pp. 31, 56).

[38] Mariano Jaimez, Christian Kerl, Javier Gonzalez-Jimenez and Daniel Cremers. 'Fast odometry and scene flow from RGB-D cameras based on geometric clustering'. In: *IEEE International Conference on Robotics and Automation*. 2017 (cit. on pp. 36, 44, 46, 65).

[39] Kevin M Judd, Jonathan D Gammell and Paul Newman. 'Multimotion visual odometry (MVO): Simultaneous estimation of camera and third-party motions'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2018 (cit. on pp. 27, 28, 33, 56–58, 60).

[40] Kevin Michael Judd and Jonathan D Gammell. 'The Oxford multimotion dataset: Multiple SE(3) motions with ground truth'. In: *IEEE Robotics and Automation Letters* (2019) (cit. on p. 53).

[41] Simon J Julier and Jeffrey K Uhlmann. 'New extension of the Kalman filter to nonlinear systems'. In: *Signal processing, sensor fusion, and target recognition VI*. Vol. 3068. Spie. 1997, pp. 182–193 (cit. on p. 19).

[42] Michael Kaess. 'Simultaneous localization and mapping with infinite planes'. In: *IEEE Int. Conf. on Robot. and Automat.* 2015 (cit. on p. 22).

[43] Michael Kaess, Ananth Ranganathan and Frank Dellaert. 'iSAM: Incremental smoothing and mapping'. In: *IEEE Transactions on Robotics* 24.6 (2008), pp. 1365–1378 (cit. on p. 16).

[44] Deok-Hwa Kim, Seung-Beom Han and Jong-Hwan 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 (cit. on p. 31).

[45] Deok-Hwa Kim and Jong-Hwan Kim. 'Effective background model-based RGB-D dense visual odometry in a dynamic environment'. In: *IEEE Transactions on Robotics* 32.6 (2016), pp. 1565–1573 (cit. on p. 28).

[46] Joon-Ha Kim, Seungwoo Hong, Gwanghyeon Ji, Seunghun Jeon, Jemin Hwangbo, Jun-Ho Oh and Hae-Won Park. 'Legged robot state estimation with dynamic contact event information'. In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 6733–6740 (cit. on p. 3).

[47] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo et al. 'Segment anything'. In: *arXiv preprint arXiv:2304.02643* (2023) (cit. on p. 98).

[48] Georg Klein and David Murray. 'Parallel tracking and mapping for small AR workspaces'. In: *2007 6th IEEE and ACM international symposium on mixed and augmented reality*. IEEE. 2007, pp. 225–234 (cit. on p. 20).

[49] Matthew Klingensmith, Siddartha S Sirinivasa and Michael Kaess. 'Articulated robot motion for simultaneous localization and mapping (ARM-SLAM)'. In: *IEEE robotics and automation letters* 1.2 (2016), pp. 1156–1163 (cit. on p. 24).

[50] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart and Paul Furgale. 'Keyframe-based visual-inertial odometry using nonlinear optimization'. In: *The International Journal of Robotics Research* (2015) (cit. on p. 24).

[51] Shile Li and Dongheui Lee. 'RGB-D SLAM in dynamic environments using static point weighting'. In: *IEEE Robotics and Automation Letters* (2017) (cit. on p. 29).

[52] Yanyan Li, Raza Yunus, Nikolas Brasch, Nassir Navab and Federico Tombari. 'RGB-D SLAM with structural regularities'. In: *2021 IEEE international conference on Robotics and automation (ICRA)*. IEEE. 2021, pp. 11581–11587 (cit. on p. 23).

[53] Yanyan Li, Raza Yunus, Nikolas Brasch, Nassir Navab and Federico Tombari. 'RGB-D SLAM with structural regularities'. In: *IEEE International Conference on Robotics and Automation*. 2021. DOI: 10.1109/ICRA48506.2021.9561560 (cit. on pp. 55, 59, 65).

[54] Jianheng Liu, Xuanfu Li, Yueqian Liu and Haoyao Chen. 'RGB-D Inertial Odometry for a Resource-Restricted Robot in Dynamic Environments'. In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 9573–9580. DOI: 10.1109/LRA.2022.3191193 (cit. on pp. 32, 87–89).

[55] Ran Long, Christian Rauch, Tianwei Zhang, Vladimir Ivan, Tin Lun Lam and Sethu Vijayakumar. 'RGB-D SLAM in Indoor Planar Environments With Multiple Large Dynamic Objects'. In: *IEEE Robotics and Automation Letters* 7.3 (2022), pp. 8209–8216. DOI: 10.1109/LRA.2022.3186091 (cit. on pp. 77, 82, 87–90).

[56] Ran Long, Christian Rauch, Tianwei Zhang, Vladimir Ivan and Sethu Vijayakumar. 'RigidFusion: Robot localisation and mapping in environments with large dynamic rigid objects'. In: *IEEE Robotics and Automation Letters* (2021). DOI: 10.1109/LRA.2021.3066375 (cit. on pp. 60, 61, 64, 66, 77, 98).

[57] Quan-Tuan Luong and Olivier D Faugeras. 'The fundamental matrix: Theory, algorithms, and stability analysis'. In: *International journal of computer vision* 17.1 (1996), pp. 43–75 (cit. on p. 26).

[58] Todd Lupton and Salah Sukkarieh. 'Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions'. In: *IEEE Transactions on Robotics* 28.1 (2011), pp. 61–76 (cit. on p. 79).

[59] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi and Ren Ng. 'NeRF: Representing scenes as neural radiance fields for view synthesis'. In: *Communications of the ACM* 65.1 (2021), pp. 99–106 (cit. on pp. 22, 99).

[60] Raul Mur-Artal, Jose Maria Martinez Montiel and Juan D Tardos. 'ORB-SLAM: a versatile and accurate monocular SLAM system'. In: *IEEE transactions on robotics* 31.5 (2015), pp. 1147–1163 (cit. on pp. 1, 20, 82).

[61] Raul Mur-Artal and Juan D Tardós. 'ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras'. In: *IEEE Transactions on Robotics* (2017) (cit. on pp. 1, 13, 20, 57, 59, 75, 80).

[62] Raúl Mur-Artal and Juan D Tardós. 'Fast relocalisation and loop closing in keyframe-based SLAM'. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 846–853 (cit. on p. 84).

[63] Armin Mustafa, Hansung Kim, Jean-Yves Guillemaut and Adrian Hilton. 'Temporally coherent 4d reconstruction of complex dynamic scenes'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4660–4669 (cit. on p. 99).

[64] Richard A Newcombe, Dieter Fox and Steven M Seitz. 'Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 343–352 (cit. on p. 99).

[65] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges and Andrew Fitzgibbon. 'KinectFusion: Real-time dense surface mapping and tracking'. In: *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. IEEE. 2011, pp. 127–136 (cit. on pp. 1, 21).

[66] Richard A Newcombe, Steven J Lovegrove and Andrew J Davison. 'DTAM: Dense tracking and mapping in real-time'. In: *2011 international conference on computer vision*. IEEE. 2011, pp. 2320–2327 (cit. on p. 21).

[67] Pauline C Ng and Steven Henikoff. 'SIFT: Predicting amino acid changes that affect protein function'. In: *Nucleic acids research* 31.13 (2003), pp. 3812–3814 (cit. on p. 20).

[68] Michael Niemeyer, Lars Mescheder, Michael Oechsle and Andreas Geiger. 'Occupancy flow: 4d reconstruction by learning particle dynamics'. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 5379–5389 (cit. on p. 99).

[69] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999 (cit. on p. 17).

[70] Kemal E Ozden, Konrad Schindler and Luc Van Gool. 'Multibody structure-from-motion in practice'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.6 (2010), pp. 1134–1141 (cit. on p. 26).

[71] Pedro F Proença and Yang Gao. 'Fast cylinder and plane extraction from depth cameras for visual odometry'. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 6813–6820 (cit. on p. 22).

[72] Albert Pumarola, Enric Corona, Gerard Pons-Moll and Francesc Moreno-Noguer. 'D-NeRF: Neural radiance fields for dynamic scenes'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 10318–10327 (cit. on p. 99).

[73] Mark Pupilli and Andrew Calway. 'Real-time visual slam with resilience to erratic motion'. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 1. IEEE. 2006, pp. 1244–1249 (cit. on p. 19).

[74] Tong Qin, Peiliang Li and Shaojie Shen. 'VINS-Mono: A robust and versatile monocular visual-inertial state estimator'. In: *IEEE Transactions on Robotics* (2018) (cit. on pp. 1, 3, 24, 86–89).

[75] Kejie Qiu, Tong Qin, Wenliang Gao and Shaojie Shen. 'Tracking 3-D motion of dynamic objects using monocular visual-inertial sensing'. In: *IEEE Transactions on Robotics* 35.4 (2019), pp. 799–816 (cit. on p. 32).

[76] Thinal Raj, Fazida Hanim Hashim, Aqilah Baseri Huddin, Mohd Faisal Ibrahim and Aini Hussain. 'A survey on LiDAR scanning mechanisms'. In: *Electronics* 9.5 (2020), p. 741 (cit. on p. 98).

[77] Christian Rauch, Ran Long, Vladimir Ivan and Sethu Vijayakumar. 'Sparse-Dense Motion Modelling and Tracking for Manipulation without Prior Object Models'. In: *IEEE Robotics and Automation Letters* (2022) (cit. on p. 72).

[78] Joseph Redmon, Santosh Divvala, Ross Girshick and Ali Farhadi. 'You only look once: Unified, real-time object detection'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788 (cit. on p. 30).

[79] Joseph Redmon and Ali Farhadi. 'Yolov3: An incremental improvement'. In: *arXiv preprint arXiv:1804.02767* (2018) (cit. on p. 32).

[80] Yifei Ren, Binbin Xu, Christopher L. Choi and Stefan Leutenegger. 'Visual-Inertial Multi-Instance Dynamic SLAM with Object-level Relocalisation'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2022. DOI: 10.1109/IROS47612.2022.9981795 (cit. on pp. 32, 77).

[81] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui and Cordelia Schmid. 'Epicflow: Edge-preserving interpolation of correspondences for optical flow'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1164–1172 (cit. on pp. 28, 29).

[82] *Robot Vacuums by Dyson online resource available at*. URL: https://www.dyson.com/robot-vacuums/dyson-360-eye-overview.html (cit. on p. 1).

[83] Ethan Rublee, Vincent Rabaud, Kurt Konolige and Gary Bradski. 'ORB: An efficient alternative to SIFT or SURF'. In: *IEEE International Conference on Computer Vision*. 2011 (cit. on pp. 20, 31, 58, 78, 79).

[84] Martin Rünz and Lourdes Agapito. 'Co-Fusion: Real-time segmentation, tracking and fusion of multiple objects'. In: *IEEE International Conference on Robotics and Automation*. 2017 (cit. on pp. 2, 27, 33, 44, 46–48, 56, 65, 75, 89, 90).

[85] Martin Rünz, Maud Buffier and Lourdes Agapito. 'MaskFusion: Real-time recognition, tracking and reconstruction of multiple moving objects'. In: *IEEE International Symposium on Mixed and Augmented Reality*. 2018 (cit. on pp. 2, 30, 33).

[86] Reza Sabzevari and Davide Scaramuzza. 'Monocular simultaneous multi-body motion segmentation and reconstruction from perspective views'. In: *IEEE International Conference on Robotics and Automation*. 2014 (cit. on pp. 26, 27).

[87] Reza Sabzevari and Davide Scaramuzza. 'Multi-body motion estimation from monocular vehicle-mounted cameras'. In: *IEEE Transactions on Robotics* (2016) (cit. on p. 27).

[88] Renato F Salas-Moreno, Ben Glocken, Paul HJ Kelly and Andrew J Davison. 'Dense planar SLAM'. In: *2014 IEEE international symposium on mixed and augmented reality (ISMAR)*. IEEE. 2014, pp. 157–164 (cit. on p. 22).

[89] Konrad Schindler and David Suter. 'Two-view multibody structure-and-motion with outliers through model selection'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.6 (2006), pp. 983–995 (cit. on p. 26).

[90] Konrad Schindler, David Suter and Hanzi Wang. 'A model-selection framework for multibody structure-and-motion of image sequences'. In: *International Journal of Computer Vision* 79 (2008), pp. 159–177 (cit. on p. 26).

[91] Konrad Schindler, James U and Hanzi Wang. 'Perspective n-view multibody structure-and-motion through model selection'. In: *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*. Springer. 2006, pp. 606–619 (cit. on p. 26).

[92] Thomas Schops, Torsten Sattler and Marc Pollefeys. 'Bad slam: Bundle adjusted direct rgb-d slam'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 134–144 (cit. on p. 22).

[93] David Schubert, Thore Goll, Nikolaus Demmel, Vladyslav Usenko, Jörg Stückler and Daniel Cremers. 'The TUM VI benchmark for evaluating visual-inertial odometry'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2018 (cit. on p. 24).

[94] Raluca Scona, Mariano Jaimez, Yvan R Petillot, Maurice Fallon and Daniel Cremers. 'StaticFusion: Background Reconstruction for Dense RGB-D SLAM in Dynamic Environments'. In: *IEEE International Conference on Robotics and Automation*. 2018 (cit. on pp. 2, 29, 33, 34, 36, 38, 39, 44, 46–48, 60, 61, 64, 65, 75, 81, 82, 84, 87–90, 95).

[95] Raluca Scona, Simona Nobili, Yvan R Petillot and Maurice Fallon. 'Direct visual SLAM fusing proprioception for a humanoid robot'. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 1419–1426 (cit. on p. 24).

[96] Zeyong Shan, Ruijian Li and Sören Schwertfeger. 'RGBD-inertial trajectory estimation and mapping for ground robots'. In: *Sensors* 19.10 (2019), p. 2251 (cit. on p. 89).

[97] Jianbo Shi and Jitendra Malik. 'Motion segmentation and tracking using normalized cuts'. In: *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*. IEEE. 1998, pp. 1154–1160 (cit. on p. 26).

[98] Xuesong Shi, Dongjiang Li... and Qi She. 'Are We Ready for Service Robots? The OpenLORIS-Scene Datasets for Lifelong SLAM'. In: *International Conference on Robotics and Automation (ICRA)*. 2020 (cit. on pp. 86, 88, 89).

[99] Seungwon Song, Hyungtae Lim, Alex Junho Lee and Hyun Myung. 'DynaVINS: A visual-inertial SLAM for dynamic environments'. In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 11523–11530 (cit. on pp. 77, 80).

[100] Theodoros Stouraitis, Iordanis Chatzinikolaidis, Michael Gienger and Sethu Vijayakumar. 'Online Hybrid Motion Planning for Dyadic Collaborative Manipulation via Bilevel Optimization'. In: *IEEE Transactions on Robotics* (2020) (cit. on p. 33).

[101] Hauke Strasdat, José MM Montiel and Andrew J Davison. 'Visual SLAM: why filter?' In: *Image and Vision Computing* 30.2 (2012), pp. 65–77 (cit. on p. 20).

[102] Michael Strecke and Jorg Stuckler. 'EM-fusion: Dynamic object-level SLAM with probabilistic data association'. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019 (cit. on p. 30).

[103] Michael Strecke and Joerg Stueckler. 'EM-Fusion: Dynamic object-level SLAM With probabilistic data association'. In: *2019 IEEE/CVF International Conference on Computer Vision*. 2019. DOI: 10.1109/ICCV.2019.00596 (cit. on pp. 2, 31, 33, 65, 73).

[104] Lukas von Stumberg and Daniel Cremers. 'DM-VIO: Delayed marginalization visual-inertial odometry'. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 1408–1415. DOI: 10.1109/LRA.2021.3140129 (cit. on pp. 1, 24, 77).

[105] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard and Daniel Cremers. 'A benchmark for the evaluation of RGB-D SLAM systems'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012 (cit. on pp. 14, 43, 65, 66, 69, 73, 86).

[106] Edgar Sucar, Shikun Liu, Joseph Ortiz and Andrew J Davison. 'iMAP: Implicit mapping and positioning in real-time'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 6229–6238 (cit. on p. 22).

[107] Deqing Sun, Xiaodong Yang, Ming-Yu Liu and Jan Kautz. 'PWC-Net: Cnns for optical flow using pyramid, warping, and cost volume'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8934–8943 (cit. on p. 29).

[108] Yuxiang Sun, Ming Liu and Max Q-H Meng. 'Improving RGB-D SLAM in dynamic environments: A motion removal approach'. In: *Robotics and Autonomous Systems* (2017) (cit. on p. 28).

[109] Yuxiang Sun, Ming Liu and Max Q-H Meng. 'Motion removal for reliable RGB-D SLAM in dynamic environments'. In: *Robotics and Autonomous Systems* (2018) (cit. on p. 28).

[110] Yuichi Taguchi, Yong-Dian Jian, Srikumar Ramalingam and Chen Feng. 'Point-plane SLAM for hand-held 3D sensors'. In: *2013 IEEE international conference on robotics and automation*. IEEE. 2013, pp. 5182–5189 (cit. on p. 22).

[111] Jiapeng Tang, Dan Xu, Kui Jia and Lei Zhang. 'Learning parallel dense correspondence from spatio-temporal descriptors for efficient and robust 4d reconstruction'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 6022–6031 (cit. on p. 99).

[112] Philip HS Torr. 'Geometric motion segmentation and model selection'. In: *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 356.1740 (1998), pp. 1321–1340 (cit. on p. 26).

[113] Philip HS Torr and David William Murray. 'Stochastic motion clustering'. In: *Computer Vision—ECCV'94: Third European Conference on Computer Vision Stockholm, Sweden, May 2–6 1994 Proceedings, Volume II 3*. Springer. 1994, pp. 328–337 (cit. on pp. 25, 26).

[114] Philip HS Torr, Andrew Zisserman and Stephen J Maybank. 'Robust detection of degenerate configurations while estimating the fundamental matrix'. In: *Computer vision and image understanding* 71.3 (1998), pp. 312–333 (cit. on p. 26).

[115] PHS Torr, A Zisserman and DW Murray. 'Motion clustering using the trilinear constraint over three views'. In: *Proceedings of the Europe-China Workshop on Geometrical Modelling and Invariants for Computer Vision*. 1995, pp. 118–125 (cit. on p. 26).

[116] Bill Triggs, Philip F McLauchlan, Richard I Hartley and Andrew W Fitzgibbon. 'Bundle adjustment—a modern synthesis'. In: *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*. Springer. 2000, pp. 298–372 (cit. on p. 20).

[117] Vladyslav Usenko, Jakob Engel, Jörg Stückler and Daniel Cremers. 'Direct visual-inertial odometry with stereo cameras'. In: *IEEE International Conference on Robotics and Automation*. 2016 (cit. on p. 24).

[118] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger and Bastian Leibe. 'MOTS: Multi-object tracking and segmentation'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7942–7951 (cit. on p. 30).

[119] Chenjie Wang, Bin Luo, Yun Zhang, Qing Zhao, Lu Yin, Wei Wang, Xin Su, Yajun Wang and Chengyuan Li. 'DymSLAM: 4D dynamic scene reconstruction based on geometrical motion segmentation'. In: *IEEE Robotics and Automation Letters* (2020) (cit. on p. 64).

[120] Xi Wang, Marc Christie and Eric Marchand. 'Tt-slam: Dense monocular slam for planar environments'. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 11690–11696 (cit. on p. 23).

[121] Yunfeng Wang and Gregory S Chirikjian. 'Nonparametric second-order theory of error propagation on motion groups'. In: *The International journal of robotics research* 27.11-12 (2008), pp. 1258–1273 (cit. on p. 12).

[122] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard and J. McDonald. 'Robust real-time visual odometry for dense RGB-D mapping'. In: *2013 IEEE International Conference on Robotics and Automation*. May 2013, pp. 5724–5731. DOI: 10.1109/ICRA.2013.6631400 (cit. on p. 21).

[123] Thomas Whelan, Stefan Leutenegger, Renato Salas-Moreno, Ben Glocker and Andrew Davison. 'ElasticFusion: Dense SLAM without a pose graph'. In: Robotics: Science and Systems. 2015 (cit. on pp. 1, 75).

[124] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison and Stefan Leutenegger. 'ElasticFusion: Real-time dense SLAM and light source estimation'. In: *The International Journal of Robotics Research* 35.14 (2016), pp. 1697–1716 (cit. on pp. 21, 38, 41).

[125] David Wisth, Marco Camurri and Maurice Fallon. 'Robust legged robot state estimation using factor graph optimization'. In: *IEEE Robotics and Automation Letters* (2019) (cit. on p. 24).

[126] David Wisth, Marco Camurri and Maurice Fallon. 'VILENS: Visual, inertial, lidar, and leg odometry for all-terrain legged robots'. In: *IEEE Transactions on Robotics* (2022) (cit. on pp. 3, 16).

[127] Yu-Shiang Wong, Changjian Li, Matthias Niessner and Niloy J Mitra. 'RigidFusion: RGB-D Scene Reconstruction with Rigidly-moving Objects'. In: *Computer Graphics Forum*. Vol. 40. 2. Wiley Online Library. 2021, pp. 511–522 (cit. on p. 99).

[128] Linhui Xiao, Jinge Wang, Xiaosong Qiu, Zheng Rong and Xudong Zou. 'Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment'. In: *Robotics and Autonomous Systems* 117 (2019) (cit. on p. 31).

[129] Binbin Xu, Wenbin Li, Dimos Tzoumanikas, Michael Bloesch, Andrew Davison and Stefan Leutenegger. 'Mid-fusion: Octree-based object-level multi-instance dynamic slam'. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 5231–5237 (cit. on pp. 2, 33).

[130] S. Yang, Y. Song, M. Kaess and S. Scherer. 'Pop-up SLAM: Semantic monocular plane SLAM for low-texture environments'. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, pp. 1222–1229. DOI: 10.1109/IROS.2016.7759204 (cit. on p. 23).

[131] Chao Yu, Zuxin Liu, Xin-Jun Liu, Fugui Xie, Yi Yang, Qi Wei and Qiao Fei. 'DS-SLAM: A semantic visual SLAM towards dynamic environments'. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 1168–1174 (cit. on p. 30).

[132] Tianwei Zhang and Yoshihiko Nakamura. 'PoseFusion: Dense RGB-D SLAM in Dynamic Human Environments'. In: *Proceedings of the 2018 International Symposium on Experimental Robotics*. Springer International Publishing, 2020. ISBN: 978-3-030-33950-0 (cit. on p. 30).

[133] Tianwei Zhang, Huayan Zhang, Yang Li, Yoshihiko Nakamura and Lei Zhang. 'FlowFusion: Dynamic Dense RGB-D SLAM Based on Optical Flow'. In: *arXiv preprint arXiv:2003.05102* (2020) (cit. on p. 29).

[134] Xiaoyu Zhang, Wei Wang, Xianyu Qi, Ziwei Liao and Ran Wei. 'Point-plane SLAM using supposed planes for indoor environments'. In: *Sensors* (2019) (cit. on pp. 55, 59).

[135] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald and Marc Pollefeys. 'Nice-SLAM: Neural implicit scalable encoding for slam'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12786–12796 (cit. on pp. 22, 99).