UNIVERSITY OF BIRMINGHAM

DOCTORAL THESIS

Dense RGB-D SLAM and Object Localisation for Robotics and Industrial Applications

Author: Feiying LAN

Supervisor: Marco CASTELLANI

A thesis submitted in fulfilment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in the

Department of Mechanical Engineering
University of Birmingham
UK, B15 2TT

September 22, 2022

Acknowledgements

I would like to thank my PhD supervisor, Dr Marco Castellani. Dr Marco Castellani's nomination and support for the PhD scholarship application instigated my PhD research starting in Oct 2018. Dr Marco Castellani's academic supervision and inspiration directed my research on the topics of computer vision, optimisation and robotics throughout the past four years. The working experience with Dr Marco Castellani has profound influence on my research career now and in the future.

I would like to thank Prof Duc Truong Pham for his supervision and guidance during my summer research internship (Jun 2017 - Sep 2017) and final year project (Sep 2017 - Jun 2018) in my BEng studies, as well as my PhD studies.

I would like to thank Dr Yongjing Wang for his support and for sharing his experience of academic research during my BEng and PhD studies. I would like to thank Dr Senjing Zheng. We had many discussions about academic research and exchanged opinions throughout the PhD studies.

I would like to thank the School of Engineering, University of Birmingham for granting me the Postgraduate EPSRC School Scholarship - School of Engineering for three years. During my PhD studies, this research is also supported by the UK Engineering and Physical Sciences Research Council (EPSRC), Grant No.EP/N018524/1 - Autonomous Remanufacturing (AutoReman) project.

Abstract

Dense reconstruction and object localisation are two critical steps in robotic and industrial applications. The former entails a joint estimation of camera egomotion and the structure of the surrounding environment, also known as Simultaneous Localisation and Mapping (SLAM), and the latter aims to locate the object in the reconstructed scenes. This thesis addresses the challenges of dense SLAM with RGB-D cameras and object localisation towards robotic and industrial applications.

Camera drift is an essential issue in camera egomotion estimation. Due to the accumulated error in camera pose estimation, the estimated camera trajectory is inaccurate, and the reconstruction of the environment is inconsistent. This thesis analyses camera drift in SLAM under the probabilistic inference framework and proposes an online map fusion strategy with standard deviation estimation based on frame-to-model camera tracking. The camera pose is estimated by aligning the input image with the global map model, and the global map merges the information in the images by weighted fusion with standard deviation modelling. In addition, a pre-screening step is applied before map fusion to preclude the adverse effect of accumulated errors and noises on camera egomotion estimation. Experimental results indicated that the proposed method mitigates camera drift and improves the global consistency of camera trajectories.

Another critical challenge for dense RGB-D SLAM in industrial scenarios is to handle mechanical and plastic components that usually have reflective and shiny surfaces. Photometric alignment in frame-to-model camera tracking tends to fail on such objects due to the inconsistency in intensity patterns of the images and the global map model. This thesis addresses this problem and proposes

RSO-SLAM, namely a SLAM approach to reflective and shiny object reconstruction. RSO-SLAM adopts frame-to-model camera tracking and combines local photometric alignment and global geometric registration. This study revealed the effectiveness and excellent performance of the proposed RSO-SLAM on both plastic and metallic objects. In addition, a case study involving the cover of a electric vehicle battery with metallic surface demonstrated the superior performance of the RSO-SLAM approach in the reconstruction of a common industrial product.

With the reconstructed point cloud model of the object, the problem of object localisation is tackled as point cloud registration in the thesis. Iterative Closest Point (ICP) is arguably the best-known method for point cloud registration, but it is susceptible to sub-optimal convergence due to the multimodal solution space. This thesis proposes the Bees Algorithm (BA) enhanced with the Singular Value Decomposition (SVD) procedure for point cloud registration. SVD accelerates the speed of the local search of the BA, helping the algorithm to rapidly identify the local optima. It also enhances the precision of the obtained solutions. At the same time, the global outlook of the BA ensures adequate exploration of the whole solution space. Experimental results demonstrated the remarkable performance of the SVD-enhanced BA in terms of consistency and precision. Additional tests on noisy datasets demonstrated the robustness of the proposed procedure to imprecision in the models.

Declaration

I, Feiying Lan, hereby declare that this Ph.D. thesis entitled "Dense RGB-D SLAM and Object Localisation for Robotics and Industrial Applications" was carried out by my own for the degree of Doctor of Philosophy in the University of Birmingham. I confirm that:

- The presented work has never been previously included in a thesis or dissertation submitted for a degree or other qualifications.
- Where the thesis is based on joint works done by myself with others, a clear statement has been made to illustrate how the contribution was exactly distributed.
- Except where states otherwise by reference or acknowledgement, the work presented is entirely composed by myself.

Signed: 

Date: September 22, 2022

Contents

List of Figures

List of Tables

Glossary

$SE(3)$ 3D Special Euclidean Group. 31, 138, 139, 145

$\mathfrak{se}(3)$ Lie algebra of $SE(3)$. 139, 140

AR Augmented Reality. 2

BA Bees Algorithm. viii, 6, 7, 15, 118, 120

CPU Central Processing Unit. 18

DHD Directed Hausdorff Distance. 6, 60, 61, 67, 74, 75

EA Evolutionary algorithm. 7, 118

F2F Frame-to-Frame. xv, 4–6, 10, 19, 32–37, 44–47, 49, 51, 52, 67, 74, 75, 82

F2M Frame-to-Model. xv, 5, 6, 10, 12, 15, 32, 36–38, 41, 44–47, 49–52, 55–58, 67, 74, 75, 82, 83, 117, 135

GPGPU General-Purpose Graphics Processing Unit. 9, 12

ICP Iterative Closest Point. viii, 6–8, 10, 19, 45, 118, 136

LiDAR Laser Detection and Ranging. 18

LM Levenberg-Marquardt. 6, 145, 146

MAP Maximum A Posteriori. 4, 5, 25–27, 29, 34, 35, 39

MSD Mean Surface Distance. 6, 60, 61, 67, 74, 75

PC Point Cloud. 4, 6

PSO Particle swarm optimisation. 7, 118

RGB-D 4-channel image format from RGB-D cameras. RGB are the colour channels: R is for red, G is for green, B is for blue. D is for depth channel which gives the depth measurement. vii, 3–6, 9–11, 13–15, 17–19, 25, 26, 28, 29, 41, 45–47, 49–52, 55, 57, 58, 62, 67, 74–76, 79, 82, 83, 117–120, 135, 136

RSO-SLAM An SLAM approach for Reflective and Shiny Object reconstruction. viii, 5, 6, 12, 15, 55, 57, 58, 60, 62, 67, 75, 76, 82, 83, 118, 119

SLAM Simultaneous localisation and mapping. vii, xv, 1–11, 13–15, 17–19, 22, 25, 26, 28, 29, 32, 40, 41, 44, 51, 55, 57, 58, 76, 79, 83, 117–120

SVD Singular value decomposition. viii, 6, 7, 15, 118, 120

VR Virtual Reality. 2

Chapter 1

Introduction

Remanufacturing is the process of restoring a used product to at least its original performance specifications via a combination of reuse, repair, and substitution of its components with new ones [1]. It is a key component of the circular economy [2] and sustainable development [3], due to its contribution to environmental protection, the economy, and society. Disassembly is the process of segregating mechanical assemblies into separate parts. It is arguably the most critical step in remanufacturing [4] due to its labour-intensive and time-consuming nature. Unlike assembly, which is a deterministic process and manipulates newly manufactured components of known shapes and features, the automation of disassembly operations is challenging due to the stochastic variations in the conditions, shape, and dimensions of end-of-life (EoL) products.

Robot perception and machine vision endow remanufacturing systems with the capability of tackling major uncertainties of EoL products in automatic disassembly processes. The first and overarching aim of robot perception is to acquire the texture and structure of the objects in the workspace, followed by the analysis of their uncertain features and conditions for decision-making. Usually, the analysis step makes use of standard CAD models obtained from the original equipment manufacturer (OEM) or reverse engineering.

Simultaneous Localisation and Mapping (SLAM) is a key enabler of object structure recon-

struction and sensor localisation in robotic applications, thanks to its real-time nature [5]. SLAM jointly estimates camera egomotion and the structure of the environment based on the input image sequence. The SLAM system acquires images sequentially from the camera. The system starts from the known camera pose of the first input image. The first image is used to initialise the global map model which represents the structure and appearance of the seen environment. For the following images, camera pose estimation and map fusion are executed alternatively. That is, each new input image is registered, viz. aligned with its precedent frame or the global map model via a rigid transformation, and the rigid transformation is used to estimate the camera pose. Once registered, the new image is merged with the existing global map model, viz. the global map model is built incrementally. In the end, the sequence of estimated camera poses gives the camera trajectory in the 3D space, whilst the incrementally merged map model provides the structure of the scene. The general field of SLAM includes a wide range of applications, such as robot perception and navigation [6, 7], Augmented Reality AR / Virtual RealityVR [8, 9], and high precision mapping for autonomous driving [10, 11, 12].

Monocular SLAM is a classic and lightweight implementation with a single 2D camera freely moving in the environment. Pioneering work by Davison et al. [13] on monocular SLAM featured a real-time system for camera tracking and sparse landmark mapping (*sparse* SLAM), using an uncontrolled 2D camera. Mur-Artal et al. [14], Mur-Artal and Tardós [15] proposed a framework for real-time monocular SLAM with efficient ORB (Oriented FAST and Rotated BRIEF) [16] feature extraction, which hybridises the FAST (Features from Accelerated Segment Test) [17] and BRIEF (Binary Robust Independent Elementary Features) [18] procedures, namely, ORB-SLAM.

Due to the scale loss caused by the projection nature of 2D cameras, monocular SLAM systems usually require an extra initialisation step, for example a stereo measurement [19, 20] to recover the physical dimensions of the structure. Also, scale drift occurs in camera egomotion estimation and structure reconstruction due to scale ambiguity [21, 22]. Binocular vision provides a solution to eliminate scale ambiguity and recover physical scale directly in SLAM using stereo cameras

[23, 24, 25].

Regular RGB images captured by 2D cameras consist of three channels of intensity information (i.e., red, green, and blue). RGB-D images add an extra depth channel (i.e. they are 3D), which directly generates the texture and structure of 3D objects by the inverse of the camera projection. Current RGB-D cameras combine the high frame rate and high resolution of standard RGB cameras with the direct range measurement of depth sensors. Application of RGB-D cameras is restricted to indoor scenarios due to multiple problems such as their limited detection range, issues arising from uncontrolled illumination, and the interference of sunlight with the depth scanner light beam. In indoor scenarios, RGB-D cameras can accurately measure depth up to a distance of 8.5 metres [26], removing the scale ambiguity problem of monocular RGB cameras. For this reason, RGB-D SLAM is a valuable technique for indoor robotic and industrial applications.

Newcombe et al. [5] proposed an innovative SLAM system with a commodity RGB-D sensor (Kinect [27]), namely, KinectFusion, and demonstrated its compelling performance for surface reconstruction and camera tracking. Compared to customary sparse SLAM in which the map consists of a few landmark pixels without the description of local surface information, a notable feature of KinectFusion is its dense representation (*dense* SLAM) [28], which depicts the surface of objects using the entire set of pixels in images. Whelan et al. [29] further extended KinectFusion using a point-based representation for map reconstruction and exploited real-time lightning source estimation without prior knowledge about the environment. Other environment-related issues of dense SLAM were discussed in the literature, such as pose estimation in dynamic environments [30] and specular reflections [31].

Dense SLAM with RGB-D cameras enables dense reconstruction in real-time and is thus an ideal tool for perception and major uncertainty handling in robotic disassembly systems. The work presented in this thesis contributes to the field of dense reconstruction and object localisation for robotic remanufacturing applications. Aside from questions on SLAM methodology and its adaptation to specific applications, some general issues such as camera drift and reflective object recon-

struction currently limit the adoption of dense SLAM in industrial applications. This thesis will address these issues.

Once completed the reconstruction process, the output of a dense SLAM system is a 3D model of the scene/object of interest. This model is given in the form of a point cloud (PC), viz. an unordered set of points. From the model, the location of the object of interest needs to be calculated.

Object localisation is customarily formulated as a point cloud registration problem. Point cloud registration amounts to an optimisation problem, where it is required to find the rigid transformation that minimises the alignment error between two PCs: the reconstructed scene model and a template model of the sought object. Unfortunately, due to the multimodality of the registration criterion, sub-optimal convergence of the registration algorithm may give unsatisfactory solutions. The final part of this thesis will address the problem of optimal point cloud registration.

In summary, three critical problems for object localisation via dense SLAM sensing were addressed in this study: 1) camera drift and 2) reflective and shiny object reconstruction in the context of dense SLAM, and 3) global optimisation for point cloud registration. The first two points concern the problem of acquiring a faithful 3D map of the environment, and the third point is how to use this map to localise an object of interest.

The first part of the thesis describes the problem of camera drift in dense RGB-D SLAM. Camera drift amounts to the fact that the estimated camera trajectory shifts away from the ground truth as the camera tracking process advances. The inaccurately estimated camera orbit subsequently degrades the consistency and precision of the map reconstruction process. Frame-to-frame (F2F) camera tracking recurrently estimates the relative camera pose between two consecutive images. In this case, the relative pose error accumulates and determines camera drift. Common causes of errors in the computation of the camera trajectory are sensor noise and pose estimation errors.

This study formulates the problem of camera egomotion estimation as Maximum a Posteriori (MAP) inference on a factor graph [32]. A nonlinear optimisation technique, the Gauss-Newton (GN) method [33] is adopted, and the pose estimation problem is solved by minimising the photo-

metric and geometric alignment errors. Based on a probabilistic inference framework and nonlinear optimisation, a frame-to-model (F2M) camera tracking with an online map fusion strategy is proposed. That is, each subsequent image is aligned to the global 3D map model of the environment. The new images are then merged into the global map model using the online fusion strategy with standard deviation estimation.

The two camera tracking methods (F2F and F2M) were experimentally tested using two types of image alignment methods (RGB and RGB-D) on two open-source datasets: the MS RGB-D dataset [34] and the TUM RGB-D dataset [35]. The results showed that the proposed F2M RGB-D based approach outperforms the others in terms of global consistency and local accuracy, i.e., the obtained camera trajectory consistently aligns with the ground truth orbit, and the relative rigid transformation between two adjacent frames is accurate. The study demonstrated the usefulness of the proposed low-drift dense SLAM system with an RGB-D camera.

The second part of the thesis approached the critical problem of reflective and shiny object reconstruction using dense RGB-D SLAM for robotic and industrial applications. For the using the dense RGB-D SLAM photometric approach developed in the first part of the thesis, an indispensable prerequisite is the assumption of photometric consistency, also known as the optical flow [36, 37]. Photometric consistency implies the stability of the light intensity of the objects displayed in the new image (which needs to be registered) and the global map model. However, reflective and shiny objects (e.g., metallic or plastic surfaces) break this assumption, since they display variable intensity patterns in multiview images. The work in this thesis addressed the fundamental problem of dense RGB-D SLAM for reflective and shiny object reconstruction, proposing a novel approach, RSO-SLAM, based on local photometric and global geometric alignment. That is, the camera pose for a new image is estimated using the geometric information from the global map model, and the intensity information from the previously registered image. Using a probabilistic inference framework (MAP inference) similar to the one adopted in the first part of the thesis, the camera trajectory is estimated by solving a nonlinear optimisation problem using the Levenberg-Marquardt

(LM) method [38]. The objective of the optimisation procedure is the minimisation of two terms: the geometric error between the depth image and the global map model, and the photometric error between the new image and the previously registered image.

Experimental tests were carried out to evaluate the performance of RSO-SLAM on 20 plastic and 10 metallic shapes using two error metrics: the Mean Surface Distance (MSD) and the Directed Hausdorff Distance (DHD), and the results were compared with the state-of-the-art methods (F2M and F2F camera tracking) analysed in the first part of the thesis. The quality of the reconstructed shapes indicated the effectiveness of RSO-SLAM, whilst the control methods failed to different degrees on plastic and metallic surfaces. Quantitative evaluation results demonstrated the precision of the proposed reconstruction process, and demonstrated that RSO-SLAM consistently outperformed the state-of-the-art in terms of both DHD and MSD evaluation metrics. A case study involving the cover of an electric vehicle battery verified the performance of RSO-SLAM on an industrial product in the real world, confirming that the method represents a valid tool for the use of dense RGB-D SLAM for model reconstruction for mechanical components.

The third part of the thesis aimed to devise an effective registration approach for locating an object in the reconstructed point cloud using a template shape. Iterative Closest Point (ICP) [39, 40] is arguably the best known local method for PC registration. It iteratively minimises the closest-point error metric between two PCs, starting from an initial estimate of the rigid transformation and using Singular Value Decomposition SVD [41]. ICP is efficient to solve the PC registration problem, but being based on the local SVD optimisation method, is prone to sub-optimal convergence due to the multimodal nature of the solution space.

This study adopted a global search approach based on the Bees Algorithm BA and enhanced it with the SVD procedure [41] for efficient refinement of the locally optimal solutions. The BA distributes the computing effort amongst the most promising areas discovered in the solution space, adaptively adjusting the scope and duration of the local search via the neighbourhood shrinking and site abandonment procedures.

The performance of the SVD-enhanced BA was tested on ten benchmark shapes, and compared with the following state-of-the-art methods: ICP [39], the standard BA [42, 43], an Evolutionary Algorithm (EA) [44], Particle Swarm Optimisation (PSO) [45], and SVD-enhanced versions of EA and PSO. The results showed the superiority of the proposed SVD-enhanced BA, which outperformed the other methods in terms of consistency and precision. In addition, tests on noisy datasets revealed the robustness of the proposed SVD-enhanced BA to data corruption. This feature is particularly significant for real-life industrial and especially remanufacturing applications, where sensor imprecision and environmental conditions (light, dust) affect the precision of the scans.

Overall, the work presented in this thesis covers the whole process of building a faithful model of the environment from multiple partial scans, and locating desired objects in the modelled scene.

1.1 Background

This section introduces the research background on the three main topics discussed in this thesis. Section 1.1.1 introduces SLAM systems for robot perception, whilst Section 1.1.2 discusses the problem of reflective and shiny object reconstruction in the field of machine vision. Section 1.1.3 formulates object localisation as a point cloud registration problem and frames its solution from the perspective of global optimisation.

1.1.1 Robot Perception with Dense SLAM

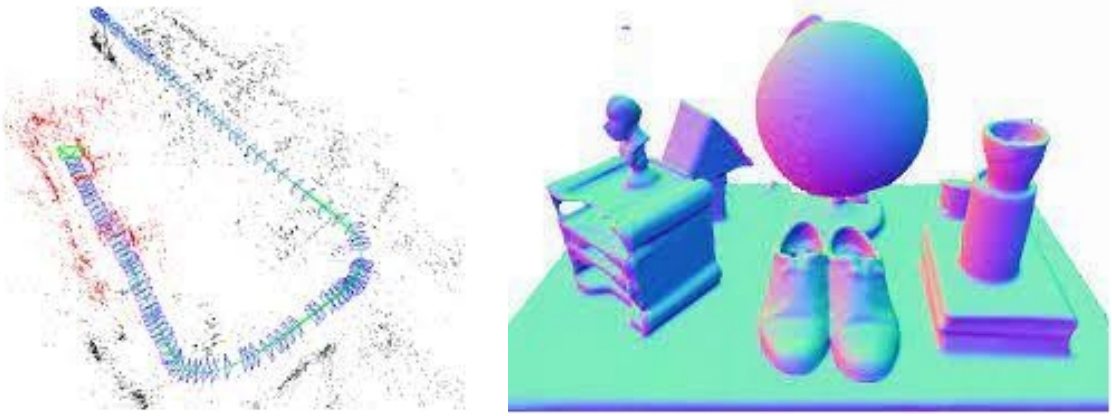
Perception empowers robots and machines with the capability of modelling and understanding the environment for reasoning and decision making. The intelligent system collects data from multiple sensors (e.g., cameras, microphones and inertial units), preprocesses them, and reorganises the processed information for environmental modelling [46]. The obtained environmental model then undergoes advanced pattern analysis and machine intelligence processing for inspection, interaction, and task-oriented manipulation. Vision is arguably the most important and powerful sensing

modality in the architecture of robot perception [47]. Environmental reconstruction and mapping involving stereo measurement technologies (e.g., laser scanning, structured light, and stereo vision) entails a process of multiview image registration. Due to the multimodal landscape of the solution space, efficient registration approaches based on local search procedures such as ICP [39, 40] are susceptible to sub-optimal convergence, whilst approaches based on global search procedures are usually offline as a result of their computational overheads. Current 3D scanning technology in industry involves fiducial markers or stickers for image registration [48], in addition to human intervention and an extra step to label or remove the stickers. More importantly, this approach might be inapplicable in robotic disassembly if the conditions of end-of-life products doesn't allow marker attachment, e.g., the object surface is rusty or cracked.

SLAM is a computational method for robot perception which simultaneously builds a 3D map of the environment and estimates camera egomotion from image sequences. Differently from standard local and global registration methods, a SLAM system aims to achieve real-time camera egomotion estimation and environmental mapping via simple camera motion linearisation with high frame-rate cameras. The system configuration for SLAM is also flexible, i.e., a freely moving camera in the 3D space without trajectory control and fiducial markers. SLAM technology is widely adopted [7, 8, 12], as it is suitable for online applications.

A SLAM system may include multiple routines to adapt to various system configurations and application demands. The map models in SLAM can be categorised into sparse and dense models according to the information usage of the image. Sparse SLAM focuses on camera egomotion estimation and aims to perform real-time camera motion tracking on general computing platforms which usually have limited computing resources (e.g., embedded systems on aerial crafts and autonomous vehicles). To achieve computational efficiency, sparse SLAM conducts camera tracking and builds a sparse map model using a portion of distinct image features with significant intensity change (e.g., edges and corners) extracted from the image. The sparse map model comprises a set of discrete points as shown in Figure 1.1a [49].

Dense SLAM takes both camera tracking and mapping into account and reconstructs the surfaces of visible objects by sensory data fusion as shown in Figure 1.1b [5]. Thanks to the development of the General-Purpose Graphics Processing Unit GPGPU and the regular-grid data structure of images, dense SLAM can be implemented in real-time via high-performance parallel computing, e.g., Nvidia CUDA programming [50]. More importantly, the information flow in dense SLAM is equivalent to data fusion, and its inverse operation is model rendering. That is, the global map model is built by fusing the images, and the images can be recovered by rendering the map model at the specified camera pose.



(a) The sparse map created by sparse SLAM. (b) The dense map created by dense SLAM.

Figure 1.1: The map models obtained by SLAM. (a): Sparse SLAM. (b): Dense SLAM.

Camera egomotion is estimated from the camera pose of the images. In sparse SLAM, the image is registered by matching the small number of points belonging to the image features (e.g., edges, corners, and textures in small image patches), also known as the *indirect* method for camera localisation. In dense SLAM, the camera pose is identified by aligning the raw image pixels (i.e., the intensity values or gradients) under the assumption of photometric consistency, also known as the *direct* method for camera localisation. The advantage of direct alignment is that it employs the entire set of pixels in the images for pose estimation, which makes it robust to sensor noise.

RGB-D is a class of sensors which generates coloured images with depth measurements. Common RGB-D cameras includes Microsoft Kinect [27], intel RealSense [51, 52], and other stereo

measurement technologies [53, 54, 55]. It is a popular type of sensor for dense SLAM, as it provides a pixel-level depth-sensing function for scale recovery.

Camera drift is a critical problem in RGB-D SLAM. It amounts to the fact that the estimated camera trajectory deviates from the true trajectory due to accumulated pose error. Inaccurate estimation of the camera pose in turn affects the quality of the geometric structure of the environment model, due to incorrect surface creation and fusion. F2F camera tracking is a popular approach in modern RGB-D SLAM systems [56, 57, 58]. It estimates camera egomotion by registering two consecutive frames from the sequence of captured images. The Markov assumption, a fundamental premise in the sequential process of F2F camera tracking, supposes that the camera pose of the new frame relies on the pose of the previously registered image. This assumption makes the method computationally efficient, since at any time only two of the whole sequence of images are considered. However, F2F camera tracking is liable to accuracy degradation when the new frame is not conditionally independent of the full sequence of historical poses. For example, when the camera moves in a closed loop and returns to previously taken positions.

Newcombe et al. [5] addressed the issue of camera drift and adopted a pure-geometric F2M camera tracking approach (ICP with projective data association [59]). Newcombe et al. [5] used only the geometric information in the images for camera egomotion estimation, namely the camera pose was identified by aligning the depth channel information of the RGB-D images with the global map model. Whelan et al. [29] extended the work of Newcombe et al. [5] to enable large-scale mapping for volumetric map representation [28]. More recently, modern dense SLAM frameworks addressed the issues of lightning source estimation [60] and online map integration [61].

This thesis addressed the problem of camera drift and proposed an online weighted map fusion strategy based on F2M camera tracking. This approach creates a point-based map representation and registers the input images against the global map model. The map fusion step allows the creation of new data points in the model from new observations, and merges similar points with uncertainty modelling.

1.1.2 Shiny and Reflective Object Reconstruction with Dense RGB-D SLAM

Mechanical components with plastic or metallic surfaces are common objects in many robotic and industrial applications. Robot perception involving these reflective and shiny objects is challenging due to the violation of the assumption of photometric consistency [36, 37] in multiview images. Reflective and shiny object reconstruction using dense RGB-D SLAM is still an open problem.

Unlike diffuse objects whose surfaces display constant colour patterns in multiview images, the intensity pattern of reflective and shiny surfaces varies at different angles. Pragmatic industrial practices include diminishing the reflective and shiny nature of the smooth surface by coating with white powder or opaque lacquer [62, 63]. However, this approach is limited in the fields of manufacturing and remanufacturing as the procedure involving surface treatment is complicated, the adhered coating layer requires additional removal operations, and the chemical material could cause corrosion to the metallic surfaces.

In the field of stereo vision with structured light scanning, proposed improvements for perceiving reflective shapes include enhancing the anti-reflection capability of the structured light pattern with a designed coding strategy [64, 65], preserving coded phase information in a single exposure image [66], and fusing multiview images captured from structured light systems [67, 68]. These approaches aim to build a structured light system for range detection, wherein a sophisticated apparatus and complicated hardware setup are required. In addition, pure-geometric range detection methods perform camera pose estimation without involving visual information. If the sensor enters a textureless scene, the observations captured by the measurement system cannot provide sufficient constraints to restrict the camera motion for this estimation problem, i.e., the problem of camera pose estimation becomes ill-posed.

Dense RGB-D SLAM entails a joint estimation of camera trajectory and the structure of the environment with RGB-D cameras. RGB-D cameras provide coloured 2D images with depth information. The advantages of dense RGB-D SLAM systems in industrial scenes are threefold: 1) direct depth-sensing information resolves scale ambiguity; 2) joint estimation with colour and

depth measurements avoids ill-posed problems for pose estimation in textureless or structureless environments, since it provides sufficient constraints from intensity and depth data; 3) real-time localisation and mapping are viable with GPGPU acceleration.

Reflections are problematic to deal in industrial fields. Objects with various surface roughness have different reflection characteristics. Diffuse objects with rough surfaces are well addressed as they conform the photometric consistency assumption. Reflective and smooth surfaces violate the photometric consistency assumption, and intensity-based features [16, 15, 69] tend to fail in this case. An extreme example is specular reflective objects (e.g., mirrors). Most of industrial products with shiny surfaces have properties that can be classified between specular and diffuse reflections, e.g., plastic or metallic objects. Unfortunately, the literature on real-life SLAM with specular reflections is scarce. Whelan et al. [31] developed a solution which attaches a tag to the camera rig for mirror surface identification, whilst other authors [70, 71] proposed methods for specular reflection detection of a laser beam. Studies covering metallic and plastic object perception in industrial scenarios are still lacking.

This study addresses the problem of reflective and shiny object reconstruction (typically, plastic and metallic surfaces) and proposes RSO-SLAM. The reflective nature of plastic or metallic surfaces violates photometric consistency assumed by the F2M camera tracking approach in Section 1.1.1. This work exploited the local photometric consistency in two consecutive images with high frame rate cameras [27, 51]. The solution entails a joint estimation of local photometric flow and global geometric alignment to avoid the inconsistent intensity patterns in multiview images. The combination of photometric and geometric data fusion effectively removes the problem of ill-posed estimation for camera poses.

1.1.3 Object Localisation via Point Cloud Registration

Object localisation entails the process of locating the position and orientation of objects in robot perception, which offers the foundation for further robotic manipulation, e.g., vision-guided robotic

grasping [72, 73] and pick-place operation [74]. Techniques used for object localisation task varies depending on the data representation for the perceived environment. In this research, the scene model is obtained in the form of a point cloud by using a dense RGB-D SLAM system. The task of object localisation is formulated as point cloud registration, which is the process of finding the 6 degrees of freedom (6 DoF) rigid transformation aligning the reconstructed model with the template shape. Point cloud registration is a fundamental problem in machine vision and has a wide range of industrial applications. For example, in the field of reverse engineering [75], 3D scanning reconstruction [76] builds a complete 3D model by aligning and merging the partial point clouds from range scans.

Customarily, point cloud registration is tackled as an optimisation problem, where the goal is the minimisation of the mismatch between the source point cloud (the template) and the reconstructed point cloud (the target). Iterative Closest Point (ICP) [39, 40] is arguably the most popular registration method. ICP is an iterative algorithm looking to optimise the closest point correspondence of the two models, using the L_2 norm as an error metric. At each step. the candidate solution is obtained by reducing the error metric via Singular Value Decomposition (SVD) [41]. This operation is repeated until a given convergence criterion is met.

The ICP convergence theorem [39] proves that the algorithm is capable of always converging to the nearest local minimum of the mean-square error metric. This metric is decreased monotonically during the iterative process owing to the closest point correspondence heuristics. However, the solution space of the point cloud registration problem is multimodal, and there exist many local minima. Given an unfavourable initial candidate solution (i.e. an estimate of the pose), ICP local search strategy would fail to find the globally optimal solution, leading to poor alignment of the two point clouds.

Metaheuristic algorithms are general-purpose strategies characterised by a global outlook in their search of the solution space. A popular instance of metaheuristic is the Bees Algorithm (BA) [42, 43], which simulates the foraging behaviour of honey bees to solve complex optimisation

problems. The BA balances random explorative and local exploitative searches by reproducing the behaviour of scouts and foragers in bee colonies. The BA site abandonment procedure prevents the algorithm from remaining stuck in the local minima of the error landscape.

This research proposes a novel optimisation method for 3D point cloud registration based on the BA. The original BA is hybridised with the problem-specific SVD operator to enhance the efficiency of the search. Because of the global nature of the BA, the proposed optimiser can avoid sub-optimal convergence to local error minima. At the same time, the SVD operator accelerates the descent of the local basins of attraction and refines the precision of the final solution.

1.2 Aims and Objectives of the Thesis

The research hypothesis of the thesis is that dense RGB-D SLAM enables real-time and efficient object reconstruction and localisation for robotic and industrial applications. The hypothesis will be proved by achieving the following aims and objectives:

- To implement a real-time dense RGB-D SLAM system for joint estimation of camera egomotion and the structure of the environment (Objective 1).
- Using the RGB-D SLAM implementation created in fulfilment of Objective 1, to devise an online camera tracking and mapping strategy for reducing camera drift and improving trajectory consistency in camera egomotion estimation (Objective 2).
- Based on the results of Objectives 1 and 2, to develop a system for the reconstruction of reflective and shiny objects using dense RGB-D SLAM system for industrial applications (Objective 3).
- To perform an experimental evaluation of the performance of the system developed in fulfilment of Objective 3, comparing its performance with state-of-the-art methods (Objective 4).

- To demonstrate the effectiveness of the devised dense RGB-D SLAM system using an industrial product (Objective 5).

- To investigate the task of object localisation using data representation in dense RGB-D SLAM, and devise an effective and reliable optimisation technique for the purpose (Objective 6).

1.3 Outline of the Thesis

The remaining sections of this thesis are structured as follows:

- Chapter 2 surveys the literature related to the three topics investigated in this thesis: dense RGB-D SLAM for robot perception, the challenge of reflective and shiny object reconstruction in industrial applications, and optimisation techniques for point cloud registration.

- Chapter 3 presents the online weighted map fusion strategy with the F2M camera tracking approach for low-drift camera egomotion estimation and accurate mapping (addressing Objective 1 and 2).

- Chapter 4 addresses the issue of reflective and shiny object reconstruction using dense RGB-D SLAM and proposes RSO-SLAM. The chapter presents a case study involving an industrial product, an electric vehicle battery, to demonstrate its effectiveness and reconstruction precision (addressing Objectives 3, 4, and 5).

- Chapter 5 describes object localisation as point cloud registration and proposes the SVD-enhanced BA for efficient precise object registration (addressing Objective 6).

Chapter 2

Literature Review

This chapter surveys the literature related to the three topics investigated in this thesis. Section 2.1 reviews the research related to dense visual SLAM systems with RGB-D cameras. Section 2.2 examines the advancements in reconstruction techniques for reflective and shiny objects, and highlights the scarcity of methods using dense RGB-D SLAM. Section 2.3 discusses various local and global optimisation techniques for point cloud registration.

2.1 Dense SLAM with RGB-D Cameras

Camera egomotion estimation is one of the two pillars of dense SLAM. Approaches for camera pose estimation in dense SLAM with RGB-D cameras can be mainly categorised into two types: *sparse* and *dense*. A third category, *semi-dense*, is used by some authors to group approaches that fall in between sparse and dense SLAM.

Feature extraction and matching are the backbones of sparse SLAM. The camera pose for each image is estimated by solving the epipolar constraints [77] of associated feature points in two consecutive images with outlier removal [78]. This step is followed by the build of the sparse landmark map with the extracted feature points. Popular feature extraction techniques for sparse SLAM include classic handcrafted features, e.g., SIFT (Scale-Invariant Feature Transform) [79],

SURF (Speed Up Robust Feature) [80], and ORB [14, 16], as well as data-driven features with learned priors [81]. Sarlin et al. [82] proposed a scheme of learning feature matching and enabled end-to-end SLAM. In this scheme, the local features are associated by solving a differentiable optimal transport problem [83] using a graph neural network.

Semi-dense SLAM involves more detail (pixels) than sparse SLAM without losing the feature of CPU-based real-time processing. Engel et al. [84] proposed a semi-dense visual odometry that extracts pixels with non-negligible image gradient for camera tracking and map reconstruction. Textureless regions are disregarded in this approach, due to the difficulty of building point associations for depth estimation in the textureless areas of two monocular camera images. Engel et al. [85, 24] extended this work to large-scale SLAM with loop closure [86] and pose graph optimisation [87]. Caruso et al. [88] adopted this approach for omni-directional (360°) cameras. A wide literature addressed the problems of long-term camera drift [89], convergence basin enlargement [90, 91], and sparse depth information exploitation by incorporating LiDAR sensors [92]. In summary, sparse and semi-dense approaches mainly focus on improving the consistency and accuracy of camera trajectory tracking, and are meant to run in real-time on a CPU.

Dense SLAM aims to obtain camera egomotion and build a dense model of the environment using the entire set of pixels in the images. The direct method of optical flow for camera pose estimation is a core component in dense SLAM. Unlike feature-based image alignment which generally gives a coarse camera pose, the direct method accurately refines the camera pose by minimising the pixel-level differences between two consecutive images [61].

The monocular camera is usually not the most suitable tool for high-precision dense SLAM reconstruction applications, since its accuracy is limited by problems such as scale loss [93, 19, 20], scale drift [21, 22], and depth estimation in textureless areas [94, 95, 96].

Depth estimation in textureless areas is problematic also in binocular SLAM systems. Dense SLAM with RGB-D cameras overcomes this drawback by introducing an auxiliary depth sensor for direct range detection. The structure in range images retains physical scale and obviates the

problems caused by scale ambiguity. The direct method for camera pose estimation using RGB-D usually includes photometric and geometric alignment routines. Photometric alignment assumes photometric consistency, and entails the estimation the motion field of the pixels of two images. This problem is also known as optical flow estimation [97, 36, 37] Geometric alignment is also considered an instantiation of optical flow estimation [57], i.e., it is interpreted as the direct alignment of the depth values of the pixels of two images. It is usually solved using the point-to-plane ICP algorithm [40]

Initial work focused on real-time dense SLAM systems with a single depth sensor [5, 98]. Kerl et al. [57] combined colour and depth information in an RGB-D camera, and used an F2F camera tracking approach, which aims at the joint optimisation of photometric and geometric errors between two consecutive images. Whelan et al. [60] provided an implementation of dense SLAM which was robust to various conditions of the lightning source. Concha and Civera [99] proposed semi-dense camera tracking with a keyframe scheme. The system examines the images and selects them as a keyframe if the percentage of overlapping area in the new image and the previous keyframe is below a preset threshold. Schops et al. [58] proposed a modern framework for RGB-D SLAM including F2F camera tracking and point-based map fusion. Their framework enables the online joint optimisation of the camera poses and map model using the entire set of images, also known as bundle adjustment [100]. Schops et al. [58] also indicated the necessity of using a global shutter RGB-D camera for synchronised colour and depth image acquisition. Cai et al. [101] exploited the accuracy of F2F camera tracking using bi-directional alignment with image pyramid estimation. The cost function for the minimisation of image misalignment combines forward and backward motion estimation.

2.2 Reflective and Shiny Object Reconstruction

Reflective and shiny object reconstruction is a challenging task in machine vision and robot perception, and was approached from various perspectives in the literature.

A number of studies focused on reconstructing specular reflective objects using a moving camera. *Specular flow* is a key component of reflective and shiny object reconstruction. It defines a dense motion field on the image plane that is jointly induced by the specular surfaces and the camera movement [102]. Object reconstruction with specular flow estimation is known as *shape from specular flow*. Specular flow estimation entails an important assumption of distant illumination (i.e., the object is placed sufficiently far from the surrounding lighting sources), so that the specular features and patterns keep constant across multiple image frames [103, 104, 105]. Roth and Black [103] modelled specular flow by a mixture model of diffuse and specular components in two images, with the constraint that the specular and diffuse reflectance must be in distinct areas. Adato et al. [104] formulated the problem of *shape from specular flow* as the solution of partial differential equations, analysing the related numerical issues [106]. Although prior knowledge of the environment is not necessary, *shape from specular flow* approaches are still sensitive to textureless environments, and have difficulties in handling inter-reflections and complex shapes with occlusion. The requirement for a distant surrounding environment may restrict the domain of applicability in industrial scenarios.

Shape from distortion is another method for specular surface reconstruction. It can be done with either controlled or natural illumination conditions. *Shape from distortion* with controlled illumination techniques require the use of a known chessboard pattern to compute the distorted pattern. Bonfort and Sturm [107] presented a reconstruction algorithm for mirror-like objects (e.g., polished metal) which uses a pre-calibrated board. Savarese et al. [108] recovered the sparse model of a reflective object from a single image using a calibrated chessboard pattern. The limited area of the chessboard pattern is an issue for holistic object reconstruction, viz. only a portion of the object

surface reflects the distorted pattern of the chessboard. Balzer et al. [109, 110] addressed this problem and proposed a multiview reconstruction method for large-size shapes. Liu et al. [111] devised a mirror surface reconstruction system based on a known chessboard pattern and pose. Their approach is compatible with both dense and sparse correspondences between the reference board and the distorted pattern. *Shape from distortion* approaches based on controlled illumination are not flexible, since they need careful hardware setup and the use of the pre-calibrated reference pattern. Furthermore, they have limited industrial applicability since most products have mixed reflective and shiny surfaces, whilst *shape from distortion* methods are tailored for high specular surfaces where the reference pattern can be clearly reflected.

Shape from distortion techniques with natural illumination relax the requirement of the known reference pattern. However, prior knowledge of the ambient lightning condition is necessary. Lambertian objects are characterised by diffusely reflecting surfaces. They can reflect incident light uniformly in all directions, and this property is customarily used to calibrate the environment lighting conditions. Johnson and Adelson [112] demonstrated that the surface normals of a single-colour Lambertian object can be estimated from a single image under natural illumination. This method requires that the reflectance map of the natural illumination is calibrated by a Lambertian sphere. Oxholm and Nishino [113, 114] extended the approach of Johnson and Adelson [112] to the case of rough reflective objects with diffuse components, using multiview calibrated images. Godard et al. [105] extended the method of Oxholm and Nishino [114] to highly specular objects using multiview images, handcrafted silhouette images of the object of interest, and the a panoramic image of the surrounding environment. The need of manually-made object silhouettes complicate the automatic reconstruction process, and require the intervention of human operators.

For reflective objects with strong diffuse components, the specularity in images is handled in the literature under the assumption of photometric consistency. Nehab et al. [115] developed a stereo measurement framework to reconstruct specular objects. Their process of specular triangulation relies on the calibrated dual camera and a temporal-encoded lighting source system. Tunwattanapong

et al. [116] presented a solution to reconstruct objects with variable reflectance. In their approach, the diffuse and specular reflectance is separated by a encoded pattern of illumination, and the shape is reconstructed by multiview stereo matching. Other studies focused on adopting a photometric stereo algorithm for non-Lambertian objects [117], specular reflective objects [118], and glossy objects [119]. The configuration of the lighting source and the multiview camera calibration is essential for these approaches, as controlled illumination is the prerequisite.

The research on reflective and shiny object reconstruction with SLAM is scarce. A few studies addressed the challenge of specular reflection in indoor mapping [31, 70, 71]. Investigation of the applicability of dense SLAM in industrial scenarios is still lacking.

In summary,reflective object reconstruction systems require the fulfilment of various requisites, and this limits their deployment in industrial scenarios. Dense SLAM provides a flexible system configuration (i.e. a freely moving camera) for unstructured environment (i.e. no prior information for environmental illumination and workspace layout). This study will address the reconstruction of reflective and shiny objects via dense SLAM.

2.3 Global Optimisation of Point Cloud Registration

Object localisation via template matching aims to estimate the rigid transformation which aligns the reconstructed object with the template shape. The problem is formulated as a point cloud registration task. Point cloud registration is usually tackled as an optimisation problem. That is, it is sought the rigid transformation that minimises a given metric of the alignment error between the reconstructed and template shape.

ICP [39, 40] is the best known algorithm for point cloud registration. It defines a *cost* function based on closest point correspondence via the root mean square metric. Least squares minimisation of this cost function is achieved via iterative application of SVD [41]. The effectiveness of ICP is demonstrated by its convergence theorem [39].

Rusinkiewicz and Levoy [120] investigated various variants of ICP featuring different sampling, matching, and reweighting strategies. Other studies focused on the solution of various problems affecting ICP. For instance, techniques were developed to mitigate the effect of outliers, such as trimming [121], distance measurements [122], M-estimation [123, 124], and soft rejection functions [125, 126]. Babin et al. [127] conducted a comparative study on popular outlier handling methods with various robust estimators. Other authors aimed to increase the computational efficiency of ICP, using techniques such as data pre-processing and k-d tree search modification [128, 129], linearisation with projective data association [59], and Anderson acceleration [130].

Unfortunately, due to the local nature of the optimisation strategy, all ICP variants are liable to get stuck into local minima of the error function. Alternative optimisation methods based on global search strategies have been investigated by several authors. For example, geometric Branch and Bound (BnB) was adopted for 2D image registration [131, 132]. Li and Hartley [133] applied the BnB strategy to a 3D point cloud registration problem, although under the constraint that the two point clouds have orientation misalignment only. The main challenge in the application of BnB to 3D problems is its computational complexity, which increases exponentially with the number of variables due to the curse of dimensionality [134]. Yang et al. [135] proposed a nested BnB strategy in rotation and translation space, to avoid directly searching in the 6D space.

Metaheuristics are optimisation procedures aiming to provide acceptable solutions when exhaustive search of the solution space is computationally infeasible. Evolutionary [136] and swarm [137] algorithms are two popular classes of such procedures. Usually inspired by biological systems, they implement global search strategies, and are generally amenable to implementation in fast parallel computation schemes.

Various instances of evolutionary algorithms (EAs) have been used for point cloud registration. Brunnstrom and Stoddart [44] employed a genetic algorithm (GA) to implement a coarse-to-fine optimisation strategy. A similar strategy was devised by Silva et al. [138], who first used a GA for coarse alignment of the point clouds, and then stochastic hill-climbing for quick refinement of

the solutions. The authors reported occasional failures of the registration algorithm, which were probably caused by premature convergence of the GA search.

Robertson and Fisher [139] applied a parallel EA to avoid premature convergence to local minima. Zhu et al. [140] introduced a centre alignment technique to compress the search space, and used the Trimmed ICP algorithm [121] to accelerate the evolution of the GA population. Yan et al. [141] applied a GA to the registration of TLS-TLS (Terrestrial LiDAR Scanning) and TLS-MLS (Mobile LiDAR Scanning) point clouds. Sahillioğlu [142] and Edelstein et al. [143] identified the correspondence between two isometric shapes with a GA approach, while Zhang et al. [144] and Li and Dian [145] applied differential evolution for aligning partially overlapping point clouds.

Swarm intelligence found application in several image (2D) and point cloud (3D) registration problems. Ant colony optimisation [146] is one of the earliest and most popular swarm algorithms, and found application mainly in the 2D registration domain [147, 148, 149]. In the 3D domain, Yu and Wang [150] and Ge et al. [151] adopted particle swarm optimisation (PSO) [45] for point cloud registration. Zhan et al. [152] added to the PSO optimiser a pre-processing stage based on the mean filter for noise suppression. PSO was also used by Wongkhuenkaew et al. [153] in a hierarchical registration framework for tooth model reconstruction.

This study will evaluate the suitability of the BA as a global optimisation method for point cloud registration.

Chapter 3

On Reducing Camera Drift for Dense RGB-D SLAM

This chapter addresses the problem of camera drift in dense RGB-D SLAM. In Section 3.1, SLAM is framed as a Maximum A Posteriori (MAP) inference on a factor graph and solved as a non-linear optimisation problem with numerical techniques. In Section 3.2, the groundwork for dense SLAM is laid, including the mathematical foundation for the camera model, map data representation, and camera pose. In Section 3.3, the nonlinear optimisation strategy for camera egomotion estimation and online fusion is elaborated. The on-manifold optimisation and solution updating scheme for camera egomotion estimation is detailed in Appendix A. In Section 3.4, the performance of the proposed method is evaluated on two RGB-D data sets: MS and TUM. Two error metrics are used for the evaluation: the absolute translational error and the relative pose error.

3.1 Problem Formulation: Probabilistic Inference to Numeric Optimisation

SLAM is the process of simultaneously estimating the camera trajectory and structure of the surrounding environment. In this chapter, it is formulated as a probabilistic inference problem performed on two categories of data structures: random variables and observations. Random variables include camera poses and the global map model. The set of camera poses $\mathcal{T} = \{\boldsymbol{T}_1, \boldsymbol{T}_2, ..., \boldsymbol{T}_n\}$ for the input sequence of images consists of a sequence of rigid transformations $\boldsymbol{T}_i$, each defining the position and orientation of the camera in the 3D space. The mathematical representation for the rigid transformation is detailed in Section 3.2.2. The global map model $\mathcal{M} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_m\}$ of the environment consists of the overall collection of the coordinates of the points obtained from the set $\mathcal{Z}$ of individual observations. The set of observations $\mathcal{Z} = \{\boldsymbol{I}_1, \boldsymbol{I}_2, ..., \boldsymbol{I}_n\}$ is the sequence of images captured by the sensor (i.e., RGB-D camera), where $\boldsymbol{I}_i \in \mathcal{Z}$ is the individual image.

The random variables $\mathcal{T}$ and $\mathcal{M}$ are estimated by maximising the posterior probability density $p(\mathcal{T}, \mathcal{M}|\mathcal{Z})$ conditioned on the set of observations $\mathcal{Z}$. However, the direct solution to the MAP inference is computationally intractable due to the large number of random variables and lack of correlation relationships between the variables and observations.

Camera tracking in SLAM transforms the probabilistic inference for all random variables conditioned on the entire set of observations $p(\mathcal{T}, \mathcal{M}|\mathcal{Z})$, into the sequential maximisation of the posterior probability density $\mathcal{P}$ of the camera pose $\boldsymbol{T}_n$ given a new input image $\boldsymbol{I}_n$:

$$\boldsymbol{T}_n^* = \arg\max_{\boldsymbol{T}_n} \ \mathcal{P} = \arg\max_{\boldsymbol{T}_n} \ p(\boldsymbol{T}_n|\mathcal{M}_{n-1}, \mathcal{T}_{1:n-1}, \mathcal{Z}_{1:n}) \tag{3.1}$$

where $\boldsymbol{T}_n$ is the camera pose for the new input image in $\mathcal{T}$, $\boldsymbol{T}_n^*$ is the optimal solution for $\boldsymbol{T}_n$, $\mathcal{Z}_{1:n}$ is the set of images (i.e., $\mathcal{Z}_{1:n} = \{\boldsymbol{I}_1, \boldsymbol{I}_2, ..., \boldsymbol{I}_n\}$), and $\mathcal{T}_{1:n-1}$ is the subset of $\mathcal{T}$ consisting of the first $n-1$ camera poses (i.e., $\mathcal{T}_{1:n-1} = \{\boldsymbol{T}_1, \boldsymbol{T}_2, ..., \boldsymbol{T}_{n-1}\}$). The map $\mathcal{M}_{n-1}$ is defined as the partial

map model created using the set $\mathcal{Z}_{1:n-1}$ (the first $n-1$ observations).

Equation (3.1) is transformed using Bayes' rule for MAP inference:

$$\begin{aligned}\boldsymbol{T}_n^* &= \arg\max_{\boldsymbol{T}_n} \ \mathcal{P} \\ &= \arg\max_{\boldsymbol{T}_n} \ p(\mathcal{Z}_{1:n}|\mathcal{M}_{n-1}, \mathcal{T}_{1:n-1}, \boldsymbol{T}_n) p(\mathcal{M}_{n-1}|\mathcal{T}_{1:n-1}, \boldsymbol{T}_n) p(\mathcal{T}_{1:n-1}|\boldsymbol{T}_n) p(\boldsymbol{T}_n)\end{aligned} \tag{3.2}$$

Since the global map $\mathcal{M}_{n-1}$ and the camera poses $(\mathcal{T}_{1:n-1}, \boldsymbol{T}_n)$ are independent and identically distributed, Equation (3.2) can be rewritten as:

$$\boldsymbol{T}_n^* = \arg\max_{\boldsymbol{T}_n} \ \mathcal{P} = \arg\max_{\boldsymbol{T}_n} \ p(\mathcal{Z}_{1:n}|\mathcal{M}_{n-1}, \mathcal{T}_{1:n-1}, \boldsymbol{T}_n) p(\mathcal{T}_{1:n-1}|\boldsymbol{T}_n) p(\boldsymbol{T}_n) \tag{3.3}$$

To find the MAP for camera egomotion estimation in Equation (3.3), the maximisation of the posterior probability density is reformulated as the minimisation of the negative logarithmic probability:

$$\begin{aligned}\boldsymbol{T}_n^* &= \arg\min_{\boldsymbol{T}_n} [-\log(\mathcal{P})] \\ &= \arg\min_{\boldsymbol{T}_n} [-\log(p(\mathcal{Z}_{1:n}|\mathcal{M}_{n-1}, \mathcal{T}_{1:n-1}, \boldsymbol{T}_n)) - \log(p(\mathcal{T}_{1:n-1}|\boldsymbol{T}_n)) - \log(p(\boldsymbol{T}_n))]\end{aligned} \tag{3.4}$$

Assuming Gaussian noise, the expression of $\arg\min_{\boldsymbol{T}_n}[-\log(\mathcal{P})]$ in Equation (3.4) can be rewritten as a least squares problem (i.e., $\min \mathcal{F}$)):

$$\begin{aligned}\boldsymbol{T}_n^* &= \arg\min_{\boldsymbol{T}_n} \mathcal{F} \\ &= \arg\min_{\boldsymbol{T}_n} \frac{1}{2}\|\mathcal{Z}_{1:n} - h_1(\mathcal{M}_{n-1}, \mathcal{T}_{1:n-1}, \boldsymbol{T}_n)\|^2_{\Sigma_1} + \frac{1}{2}\|\mathcal{T}_{1:n-1} - h_2(\boldsymbol{T}_n)\|^2_{\Sigma_2} - \|h_3(\boldsymbol{T}_n)\|^2_{\Sigma_3}\end{aligned} \tag{3.5}$$

where $\mathcal{F}$ is the objective function for the least square minimisation problem, Σ is the covariance matrix, and $\|\cdot\|^2_\Sigma$ is the Mahalanobis distance [154] given covariance matrix Σ, i.e., $\|X\|^2_\Sigma = x^T\Sigma^{-1}x$.

The first term of Equation (3.5) is the likelihood of the new camera pose $\boldsymbol{T}_n$, given the global map model $\mathcal{M}_{n-1}$, the past $n-1$ camera poses $\mathcal{T}_{1:n-1}$, and the set of observations $\mathcal{Z}_{1:n}$. $h_1(\cdot)$ is the observation function in the SLAM system. For RGB-D cameras, it entails the operation of point projection and pixel indexing in images. The second term is the state transition of $\boldsymbol{T}_n$, and $h_2(\cdot)$ is the state transition function. This term depicts the motion of the camera under controlled input. The third term is the prior probability for $\boldsymbol{T}_n$, and can be considered as a regularisation term.

Since the functions h_1, h_2 and h_3 in Equation (3.4) are nonlinear in the SLAM system, the least square minimisation task is a nonlinear optimisation problem and can be solved via numerical techniques. The first critical step for the solution of the optimisation problem is the linearisation of the functions $h_{[\cdot]}(\cdot)$ in Equation (3.6). Here, $\boldsymbol{T}$ is used to denote the variable $\boldsymbol{T}_n$ for the sake of clarity, and Equation (3.5) becomes:

$$\begin{aligned}\mathcal{F}(\boldsymbol{T}) = \; & \frac{1}{2}\left\|\mathcal{Z}_{1:n} - h_1(\mathcal{M}_{n-1}, \mathcal{T}_{1:n-1}, \boldsymbol{T}_0) - \frac{\partial h_1}{\partial \boldsymbol{T}}\Delta\boldsymbol{T}\right\|^2_{\Sigma_1} \\ & + \frac{1}{2}\left\|\mathcal{T}_{1:n-1} - h_2(\boldsymbol{T}_0) - \frac{\partial h_2}{\partial \boldsymbol{T}}\Delta\boldsymbol{T}\right\|^2_{\Sigma_2} \\ & + \left\|h_3(\boldsymbol{T}_0) + \frac{\partial h_3}{\partial \boldsymbol{T}}\Delta\boldsymbol{T}\right\|^2_{\Sigma_3}\end{aligned} \tag{3.6}$$

where $\boldsymbol{T}_0$ is the given initial parameter (i.e., known initial camera pose), and $\partial h_{[\cdot]}/\partial \boldsymbol{T}$ is the Jacobian at $\boldsymbol{T} = \boldsymbol{T}_0$.

The Mahalanobis distance $\|\cdot\|^2_\Sigma$ in Equation (3.6) is converted to the more familiar Euclidean distance $\|\cdot\|^2_2$ via a process known as *whitening* [32]:

$$\|x\|^2_\Sigma = x^T\Sigma^{-1}x = \left(\Sigma^{-1/2}x\right)^T\left(\Sigma^{-1/2}x\right) = \|\Sigma^{-1/2}x\|^2_2 \tag{3.7}$$

Equation (3.6) is then rewritten in form of weighted squares:

$$\begin{aligned}\mathcal{F} = \; & \frac{1}{2}\left\|\Sigma_1^{-1/2}\left\{\frac{\partial h_1}{\partial \boldsymbol{T}}\Delta\boldsymbol{T} - [\mathcal{Z}_{1:n} - h_1(\mathcal{M}_{n-1}, \mathcal{T}_{1:n-1}, \boldsymbol{T}_0]\right\}\right\|_2^2 \\ & + \frac{1}{2}\left\|\Sigma^{-1/2}\left\{\frac{\partial h_2}{\partial \boldsymbol{T}}\Delta\boldsymbol{T} - [\mathcal{T}_{1:n-1} - h_2(\boldsymbol{T}_0)]\right\}\right\|_2^2 \\ & + \left\|\Sigma^{-1/2}\left[\frac{\partial h_3}{\partial \boldsymbol{T}}\Delta\boldsymbol{T} - h_3(\boldsymbol{T}_0)\right]\right\|_2^2 \\ = \; & \frac{1}{2}\left\|J_1\Delta\boldsymbol{T} - b_1\right\|_2^2 + \frac{1}{2}\left\|J_2\Delta\boldsymbol{T} - b2\right\|_2^2 + \left\|J_3\Delta\boldsymbol{T} - b_3\right\|_2^2\end{aligned} \tag{3.8}$$

where:

$$\begin{aligned} J_1 &= \Sigma^{-1/2}\cdot\frac{\partial h_1}{\partial \boldsymbol{T}}, \quad b_1 = \Sigma^{-1/2}\cdot[\mathcal{Z}_{1:n} - h_1(\mathcal{M}_{n-1}, \mathcal{T}_{1:n-1}, \boldsymbol{T}_0)]\,; \\ J_2 &= \Sigma^{-1/2}\cdot\frac{\partial h_2}{\partial \boldsymbol{T}}, \quad b_2 = \Sigma^{-1/2}\cdot[\mathcal{T}_{1:n-1} - h_2(\boldsymbol{T}_0)]\,; \\ J_3 &= \Sigma^{-1/2}\cdot\frac{\partial h_3}{\partial \boldsymbol{T}}, \quad b_3 = -\Sigma^{-1/2}\cdot h_3(\boldsymbol{T}_0). \end{aligned} \tag{3.9}$$

Once obtained the Jacobians in Equation (3.9), the least square problem can be solved via numerical optimisation techniques [33]. The on-manifold optimisation technique used for this problem is explained in detail in Appendix A.

Similarly, the MAP inference for environmental mapping can be formulated as in Equation (3.10), incrementally building the model of the environment with the sequentially registered images from camera tracking:

$$\mathcal{M}_{1:n}^* = \arg\max_{\mathcal{M}_n}\; \mathcal{P}_{map} = \arg\max_{\mathcal{M}_n}\; p(\mathcal{M}_n|\mathcal{T}_{1:n}, \mathcal{Z}_{1:n}) \tag{3.10}$$

where $\mathcal{M}_n$ is the environmental structure estimated from the sequence of observations $\mathcal{Z}_{1:n}$ and estimated camera poses $\mathcal{T}_{1:n}$. However, differently from the camera tracking procedure, in incremental mapping the conditions for the environmental mapping are various, e.g., the map representation and its use in dynamic or static environments. Thus, the direct method for transforming this second MAP problem into a least squares problem relies on the environmental conditions.

In Section 3.2, the groundwork for the camera model and map representation is laid. In Section 3.3, the mathematical formulation of the MAP problem for RGB-D SLAM is detailed. This

formulation regards the variables and observations as constituting a probabilistic graph, and camera egomotion with online mapping is performed via a probabilistic inference on the graph.

3.2 SLAM Groundwork

This section introduces the mathematical models for camera egomotion estimation in Sections 3.2.1 and 3.2.2, and the data representation for environmental mapping Section 3.2.3.

3.2.1 Camera Model and Warping Function

The pin-hole camera model is the common mathematical representation for regular cameras. The pin-hole camera model is illustrated in Figure 3.1.

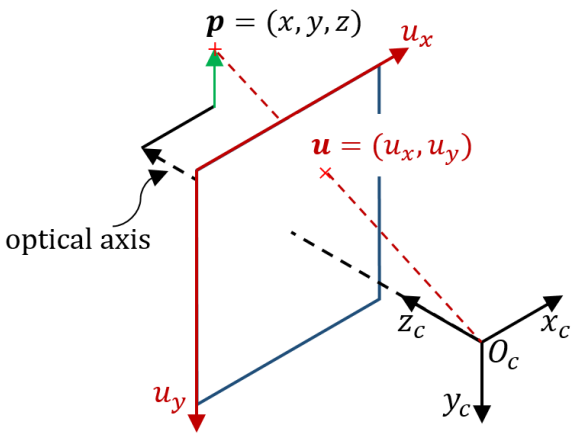


Figure 3.1: Illustration of the pin-hole camera model.

Figure 3.1 depicts the association of the 3D point $\boldsymbol{p}$ in the camera coordinate system O_c with the coordinate of the corresponding pixel $\boldsymbol{u}$ in the image. The optical axis is the line along axis z_c which passes through the camera centre and is perpendicular to the image plane.

A pin-hole camera is parameterised by four intrinsic parameters f_x, f_y, c_x, and c_y, where f_x and f_y are the focal lengths, and (c_x, c_y) is known as the principal point. The camera intrinsic

parameters constitute an intrinsic matrix $\boldsymbol{K}$:

$$\boldsymbol{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.11}$$

The camera projection of a 3D point $\mathfrak{p} = [x, y, z]^T$ into a pixel point $\boldsymbol{u}$ is described as below:

$$\boldsymbol{u} = \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} = \frac{1}{z} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{z} \boldsymbol{K} \mathfrak{p} \tag{3.12}$$

where $\boldsymbol{u}$ is the pixel coordinate of the 3D point in the image.

The process of projection in Equation (3.12) is encapsulated as a warping function π such as $\boldsymbol{u} = \pi(\mathfrak{p})$. The inverse of projection π is the function π^{-1} such as $\mathfrak{p} = \pi^{-1}(\boldsymbol{u}, d)$. The inverse 'unprojects' a pixel $\boldsymbol{u}$ at measured depth value d, and computes the coordinates of the 3D point $\mathfrak{p}$ in camera coordinate system:

$$\mathfrak{p} = \pi^{-1}(\boldsymbol{u}, d) = d\boldsymbol{K}^{-1}\boldsymbol{u} \tag{3.13}$$

3.2.2 Representations for Camera Pose and Rigid Transformation

Camera pose $\boldsymbol{T}$ refers to the position and orientation of the camera coordinate system in the global map. The mathematical representation for a camera pose $\boldsymbol{T}$ entails a rigid transformation in the Special Euclidean Group ($SE(3)$):

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix}, \ \boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}, det(\boldsymbol{R}) = 1, \ \boldsymbol{t} \in \mathbb{R}^3 \tag{3.14}$$

where $\boldsymbol{R}$ is a rotation matrix (orientation of the camera), and $\boldsymbol{t}$ is a translation vector (position of the camera).

The camera pose $\boldsymbol{T}$ describes the 6D pose of a camera coordinate system $\boldsymbol{\tau}$ by transforming the global map coordinate system using the rigid transformation $\boldsymbol{T}$. Meanwhile, $\boldsymbol{T}$ also represents the transformation of points from the camera coordinate system to the global map coordinate system.

3.2.3 Map Data Representation

The data representation for the environmental map in SLAM can be categorised into two classes: voxel-based [28, 5] and point-based [56, 155]. This thesis adopts the point-based data representation for global map reconstruction due to its advantages in terms of flexible amendments, low complexity for rendering, and efficient memory occupation.

The map model $\mathcal{M}$ is an unordered set of surfels (namely, element of surface): $\mathcal{M} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_m\}$. A surfel $\boldsymbol{s}$ is an oriented disc characterised by the following parameters: its position $\boldsymbol{p} \in \mathbb{R}^3$, normal vector $\boldsymbol{n} \in \mathbb{R}^3$, color information $\boldsymbol{c} \in \mathbb{N}^3$, radius $r \in \mathbb{R}$, and weight $w \in \mathbb{R}$.

$$\boldsymbol{s} = \begin{bmatrix} \boldsymbol{p}^T & \boldsymbol{n}^T & \boldsymbol{c}^T & r & w \end{bmatrix}^T \tag{3.15}$$

3.3 Camera Egomotion Estimation and Mapping

This section presents the original contributions of this thesis: i) the analysis of camera drift for camera egomotion estimation under the probabilistic framework with factor graph, and ii) the proposed online map fusion strategy with standard deviation estimation using the F2M camera tracking method.

Section 3.3.1 and Section 3.3.2 describe the camera tracking inference on probabilistic graphs for F2F and F2M, respectively. Section 3.3.3 details the online weighted map fusion with estimated covariances.

3.3.1 Factor Graph for Frame-to-Frame Camera Tracking

F2F camera tracking sequentially estimates the relative pose between two consecutive frames in the image sequence using image registration. The camera trajectory in the global reference frame is obtained by accumulating the relative poses.

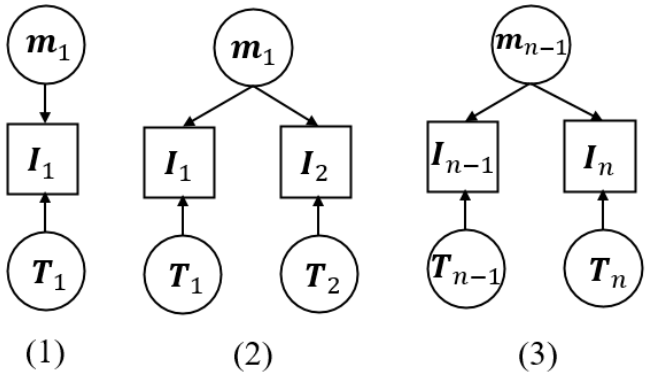
Given the image sequence $\mathcal{Z} = \{\boldsymbol{I}_1, \boldsymbol{I}_2, ..., \boldsymbol{I}_n\}$, and the initial camera pose which is set to identity $\boldsymbol{T}_1 = \boldsymbol{I}$, the goal of the SLAM procedure is to estimate the sequence of camera poses $\mathcal{T} = \{\boldsymbol{T}_1, \boldsymbol{T}_2, ..., \boldsymbol{T}_n\}$ associated to the images of $\mathcal{Z}$. F2F tracking gives the relative pose $\Delta\boldsymbol{T}_i$ between the position where the camera captured ith image $\boldsymbol{I}_i$ and the position where it captured image $\boldsymbol{I}_{i+1}$. That is, $\Delta\boldsymbol{T}_i$ defines a movement of the camera from $\boldsymbol{T}_i$ to $\boldsymbol{T}_{i+1}$. The recurrent expression for the pose of $\boldsymbol{I}_{i+1}$ is computed as:

$$\boldsymbol{T}_{i+1} = \Delta\boldsymbol{T}_i \cdot \boldsymbol{T}_i \tag{3.16}$$

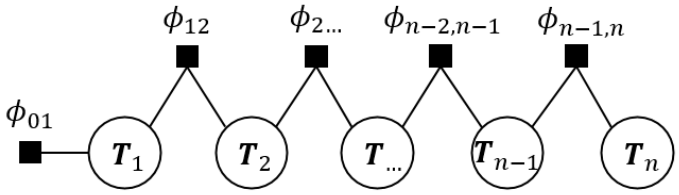
Given the initial camera pose $\boldsymbol{T}_1$ and the sequentially estimated relative camera poses $\{\Delta\boldsymbol{T}_1, \Delta\boldsymbol{T}_2, ..., \Delta\boldsymbol{T}_n\}$, the camera pose $\boldsymbol{T}_{i+1}$ is calculated as:

$$\boldsymbol{T}_{i+1} = \Delta\boldsymbol{T}_i \cdot \Delta\boldsymbol{T}_{i-1} \cdots \Delta\boldsymbol{T}_2 \cdot \Delta\boldsymbol{T}_1 \cdot \boldsymbol{T}_1 \tag{3.17}$$

In F2F camera tracking, the estimation of $\mathcal{T}$ doesn't entail the use of the global map model $\mathcal{M}$. Thus, $\mathcal{M}$ is independent of the random variable $\mathcal{T}$ and the set of observations $\mathcal{Z}$. As a result, the variable $\mathcal{M}$ is eliminated from the posterior probability density in Equation (3.3). In addition, the state transition of the camera pose is omitted because there is no active control input for camera motion. Hence, the relationships between the random variables and observations were built as shown in the probabilistic graphs of Figure 3.2.



(a) Bayesian network for F2F camera tracking. The local maps $\boldsymbol{m}_{[\cdot]}$ represent the associated images $\boldsymbol{I}_{[\cdot]}$ in the form of a point cloud.



(b) Factor graph for F2F camera tracking. The shaded squares $\phi_{[\cdot]}$ are the factors associated to the factorisation of the joint probability distribution of the random variables.

Figure 3.2: Bayesian network (a) and factor graph (b) for F2F camera tracking. In the Bayesian network, the arrows represent the dependencies of the random variables. In the Factor graph, the circles are random variables, whilst the blocks are probabilistic factors.

Figure 3.2a shows the Bayesian network which describes the recurrent process for camera pose estimation in F2F camera tracking. The random variables are represented as circled elements, the observations are shown as squares, and the dependencies are illustrated as arrows. The variables $\boldsymbol{I}_{[\cdot]}$ are the observation set (a collection of images). The variables $\boldsymbol{T}_{[\cdot]}$ are the camera poses associated to the images. The variable $\boldsymbol{m}_i$ is the local map model (point clouds) extracted from the corresponding image $\boldsymbol{I}_i$. The local map model $\boldsymbol{m}_{i-1}$ is used temporarily for estimating the camera pose $\boldsymbol{T}_n$.

In Step (1) of the procedure the camera pose $\boldsymbol{T}_1$ of the first input image $\boldsymbol{I}_1$ is initialised (Figure 3.2a). The associated local map model $\boldsymbol{m}_1$ is estimated via MAP inference from the Bayesian

network:

$$\boldsymbol{m}_1^* = \arg\max_{\boldsymbol{m}_1} p(\boldsymbol{I}_1|\boldsymbol{T}_1, \boldsymbol{m}_1) \tag{3.18}$$

In step (2), $\boldsymbol{I}_1$ and the new image $\boldsymbol{I}_2$ constitute a Bayesian network for MAP inference. The camera pose $\boldsymbol{T}_2$ is estimated via MAP inference on the network, i.e., maximising the probability:

$$\boldsymbol{T}_2^* = \arg\max_{\boldsymbol{T}_2} p(\boldsymbol{I}_1|\boldsymbol{T}_1, \boldsymbol{m}_1) \cdot p(\boldsymbol{I}_2|\boldsymbol{T}_2, \boldsymbol{m}_1) \tag{3.19}$$

In Equation (3.19), the two probability functions $p(\boldsymbol{I}_1|\boldsymbol{T}_1, \boldsymbol{m}_1)$ and $p(\boldsymbol{I}_2|\boldsymbol{T}_2, \boldsymbol{m}_1)$ are decoupled during the maximisation process. The former probability is maximised first, and gives the estimated $\boldsymbol{m}_1$, viz. the local map $\boldsymbol{m}_1$ is extracted from $\boldsymbol{I}_1$ given $\boldsymbol{T}_1$. The latter is maximised to obtain the estimated $\boldsymbol{T}_2$ given $\boldsymbol{m}_1$.

Similarly to step (2), step (n) builds a Bayesian network for $\boldsymbol{T}_{n-1}$ and $\boldsymbol{T}_n$. The MAP inference entails the maximisation of the following probability:

$$\boldsymbol{T}_n^* = \arg\max_{\boldsymbol{T}_n} \; p(\boldsymbol{I}_{n-1}|\boldsymbol{T}_{n-1}, \boldsymbol{m}_{n-1}) \cdot p(\boldsymbol{I}_n|\boldsymbol{T}_n, \boldsymbol{m}_{n-1}) \tag{3.20}$$

The two probability functions $p(\boldsymbol{I}_{n-1}|\boldsymbol{T}_{n-1}, \boldsymbol{m}_{n-1})$ and $p(\boldsymbol{I}_n|\boldsymbol{T}_n, \boldsymbol{m}_{n-1})$ in Equation (3.20) are also decoupled and maximised separately. The maximisation of the former estimates the local map $\boldsymbol{m}_{n-1}$, and the latter estimates $\boldsymbol{T}_n$ given $\boldsymbol{m}_{n-1}$.

Figure 3.2b illustrates the factor graph for the entire process of F2F camera tracking associated to the Bayesian network in Figure 3.2a. A factor graph is a bipartite graph with factors and random variables. The factors are visualised as shaded squares ($\phi_{[\cdot]}$), and represent the factorisation of the joint probability distributions of the random variables. The circles $\boldsymbol{T}_{[\cdot]}$ are the camera poses to be estimated.

In Figure 3.2b, the first factor $\phi_{01}(\boldsymbol{T}_1)$ relates to the initialisation of the first camera pose $\boldsymbol{T}_1$ in step (1). The recurrent factors $\phi_{n-1,n}(\boldsymbol{T}_{n-1}, \boldsymbol{T}_n)$ represent the joint probability distributions

$p(\boldsymbol{I}_{n-1}|\boldsymbol{T}_{n-1}, \boldsymbol{m}_{n-1})$ and $p(\boldsymbol{I}_n|\boldsymbol{T}_n, \boldsymbol{m}_{n-1})$ of $\boldsymbol{T}_{n-1}$ and $\boldsymbol{T}_n$ used in Equation (3.20). The factor graph shows that the sequential camera pose estimation for $\boldsymbol{T}_n$ is conducted by maximising the recurrent factor function $\phi_{n-1,n}(\boldsymbol{T}_{n-1}, \boldsymbol{T}_n)$ as in Equation (3.21), i.e., the Markov assumption.

$$\boldsymbol{T}_n^* = \arg\max_{\boldsymbol{T}_n} \ \phi_{n-1,n}(\boldsymbol{T}_{n-1}, \boldsymbol{T}_n) \tag{3.21}$$

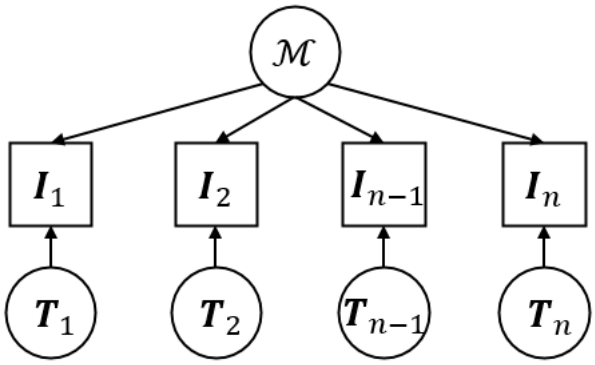
As discussed above, camera egomotion estimation with F2F camera tracking preserves the Markov assumption that the state for the new image is determined by the previous image alone, and the historical information is omitted. With this simplified assumption, the camera tracking is memory-efficient and computationally cheap for resource-limited computing platforms, e.g., embedded systems and microprocessors in unmanned robots or aircraft.

3.3.2 Factor Graph for Frame-to-Model Camera Tracking

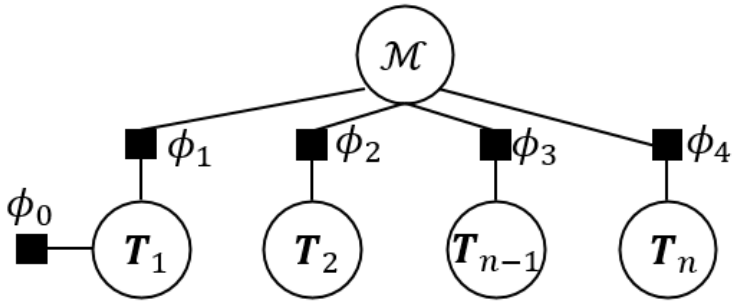
F2M camera tracking maintains a global map model and performs camera egomotion estimation by aligning the new image with the global map model.

F2M camera tracking initialises the first camera pose $\boldsymbol{T}_1$ for $\boldsymbol{I}_1$ and creates the global map $\mathcal{M}$ as done in the F2F method described in Section 3.3.1. Given a new input image $\boldsymbol{I}_n$, the associated camera pose $\boldsymbol{T}_n$ is estimated directly by aligning $\boldsymbol{I}_n$ with $\mathcal{M}$. This process doesn't involve relative pose accumulation like the F2F method.

The probabilistic graphs for the F2M approach are shown in Figure 3.3. Figure 3.3a depicts the Bayesian network, whilst Figure 3.3b shows the factor graph.



(a) Bayesian network for F2M camera tracking.



(b) Factor graph for F2M camera tracking.

Figure 3.3: Bayesian network (a) and factor graph (b) for F2M camera tracking. The circled elements are random variables (sub-figures (a) and (b)). The rectangular elements are observations (sub-figures (a) and (b)). The shaded blocks are probabilistic factors (sub-figures (a) and (b)).

The procedure for F2M camera tracking is detailed as follows.

First, the camera pose $\boldsymbol{T}_1$ is initialised to identity as did for the initialisation of the F2F method. The initial map $\mathcal{M}$ is created from the first image $\boldsymbol{I}_1$ by maximising the probability:

$$\mathcal{M}^* = \arg\max_{\mathcal{M}} \; p(\boldsymbol{I}_1|\boldsymbol{T}_1, \mathcal{M}) \tag{3.22}$$

Given a new image $\boldsymbol{I}_n$, the new camera pose $\boldsymbol{T}_n$ is estimated using the Bayesian network in Figure 3.3a.

$$\boldsymbol{T}_n^* = \arg\max_{\boldsymbol{T}_n} \left\{ \left[\prod_{i=1}^{n-1} p(\boldsymbol{I}_i|\boldsymbol{T}_i, \mathcal{M}) \right] \cdot p(\boldsymbol{I}_n|\boldsymbol{T}_n, \mathcal{M}) \right\} \tag{3.23}$$

Equation (3.23) shows that the probability function contains two decoupled terms. F2M camera

tracking involves two steps: map fusion and camera pose estimation.

The first term $\prod\ p(\boldsymbol{I}_i|\boldsymbol{T}_i, \mathcal{M})$ doesn't involve the camera pose $\boldsymbol{T}_n$. The maximisation of the first term entails a process known as *global model reconstruction and fusion* which is defined in Equation (3.10). That is, the map $\mathcal{M}$ is merged using the previously registered camera poses $\mathcal{T}_{1:n-1}$ and the images $\mathcal{Z}_{1:n-1}$, as will be discussed in Section 3.3.3.

The second term $p(\boldsymbol{I}_n|\boldsymbol{T}_n, \mathcal{M})$ in Equation (3.23) is the likelihood probability of the variables $(\boldsymbol{T}_n, \mathcal{M})$ given the observation $\boldsymbol{I}_n$, known as *camera pose estimation*. In summary, Given the fused map $\mathcal{M}$ by the first term, the camera pose $\boldsymbol{T}_n$ is estimated by maximising the following probability:

$$\boldsymbol{T}_n^* = \arg\max_{\boldsymbol{T}_n} p(\boldsymbol{I}_n|\boldsymbol{T}_n, \mathcal{M}) \tag{3.24}$$

The Bayesian network in Figure 3.3a is converted into the factor graph in Figure 3.3b. The factor graph factorises the joint probability distributions of the global map $\mathcal{M}$ and the involved camera poses $\boldsymbol{T}_{[.]}$. The first factor ϕ_0 refers to the initialisation of $\boldsymbol{T}_1$. The rest of the factors ϕ_i represent the probability function in Equation (3.23).

Therefore, the factor graph in Figure 3.3b implies that F2M camera pose estimation relaxes the Markov assumption by entailing a step of map fusion which collects the historical observations.

3.3.3 MAP Inference and Online Fusion for Mapping

Map fusion is the operation of merging and fusing the set of observations $\mathcal{Z}$ with the registered camera poses $\mathcal{T}_{1:n}$ into the global map model according to Equation (3.10).

Since the camera poses $\mathcal{T}_{1:n}$ and the map $\mathcal{M}$ are independent and identically distributed without prior information (i.e., the camera moves freely, and the environment is unknown), the posterior probability $\mathcal{P}_{map}$ in Equation (3.10) is transformed into the likelihood probability function in Equation (3.25), further rewritten as the product of probabilistic factors according to the probabilistic

graph of Figure 3.3.

$$\mathcal{M}_n^* = \arg\max_{\mathcal{M}_n} \ \mathcal{P}_{map} = \arg\max_{\mathcal{M}_n} \ p(\mathcal{Z}_{1:n}|\mathcal{T}_{1:n}, \mathcal{M}_n) = \arg\max_{\mathcal{M}_n} \ \left[\prod_{i=1}^{n} \ p(\boldsymbol{I}_i|\mathcal{M}_n, \boldsymbol{T}_i)\right] \tag{3.25}$$

Similarly to Equations (3.4) and (3.5), the MAP inference problem in Equation (3.25) is transformed into a nonlinear least square minimisation problem with the assumption of Gaussian noise:

$$\mathcal{M}_n^* = \arg\min_{\mathcal{M}_n} \frac{1}{2} \sum_{i=1}^{n} \|\boldsymbol{\varepsilon}_i(\boldsymbol{I}_i, \mathcal{M}_n, \boldsymbol{T}_i)\|_{\boldsymbol{\Sigma}_i}^2 = \arg\min_{\mathcal{M}_n} \frac{1}{2} \sum_{i=1}^{n} \|\boldsymbol{I}_i - h_1(\mathcal{M}_n, \boldsymbol{T}_i)\|_{\boldsymbol{\Sigma}_i}^2 \tag{3.26}$$

where $h_1(\mathcal{M}_n, \boldsymbol{T}_i)$ is the observation function involving the camera projection, which renders the map model $\mathcal{M}_n$ at the specified camera pose $\boldsymbol{T}_i$, and $\boldsymbol{\Sigma}_i$ is the covariance matrix.

Since the map model $\mathcal{M}$ is an unordered point set as in Section 3.2.3, and the image $\boldsymbol{I}_i$ consists of many pixel observations $z_k \in \boldsymbol{I}_i$, each error term $\boldsymbol{\varepsilon}_i$ in Equation (3.26) forms a vector:

$$\boldsymbol{\varepsilon}_i = \boldsymbol{I}_i - h_1(\mathcal{M}_n, \boldsymbol{T}_i) = \begin{bmatrix} z_1 - h_1(\boldsymbol{s}_1, \boldsymbol{T}_i) \\ z_2 - h_1(\boldsymbol{s}_2, \boldsymbol{T}_i) \\ \vdots \\ z_j - h_1(\boldsymbol{s}_j, \boldsymbol{T}_i) \\ \vdots \\ z_m - h_1(\boldsymbol{s}_m, \boldsymbol{T}_i) \end{bmatrix}, \ \boldsymbol{s}_j \in \mathcal{M}, z_j \in \boldsymbol{I}_i \tag{3.27}$$

where $h_1(\boldsymbol{s}_j, \boldsymbol{T}_i)$ projects the 3D point $\boldsymbol{s}_j$ into the pixel space of the camera at pose $\boldsymbol{T}_i$, and z_j is the corresponding pixel observation. The individual least square error term $\boldsymbol{\varepsilon}_i(\boldsymbol{I}_i, \mathcal{M}_n, \boldsymbol{T}_i)$ describes the residual error between the observed image $\boldsymbol{I}_i$ and the prediction of the image $h_1(\cdot)$ using the variables $\mathcal{M}_n$ and $\boldsymbol{T}_i$.

The observation function h_1 is detailed as follows:

$$h_1(\boldsymbol{s}_j, \boldsymbol{T}_i) = \boldsymbol{I}_i[\pi(\boldsymbol{T}_i^{-1} \cdot \boldsymbol{s}_j)] \tag{3.28}$$

where $\boldsymbol{T}_i^{-1}\boldsymbol{s}_j$ describes the operation of transforming the surfel $\boldsymbol{s}_j$ into the camera coordinate system at pose $\boldsymbol{\tau}_i$, $\pi(\boldsymbol{x})$ is the camera projection function which identifies the pixel coordinates of the 3D point $\boldsymbol{x}$, and $\boldsymbol{I}_i(\boldsymbol{u})$ is the image function which gives the pixel $\boldsymbol{u}$ in $\boldsymbol{I}_i$.

This study addresses the problem of covariance estimation for the least square problem stated in Equation (3.26). In the literature, it is reported that the depth measurement error is proportional to the square of the depth d [156]. The covariance matrix $\boldsymbol{\Sigma}_i$ for the ith error term in Equation (3.26) reflects the reliability of the depth measurements for the pixels. Here, the covariance $\boldsymbol{\Sigma}_i$ is estimated as an $m \times m$ diagonal matrix (m is the number of surfels in $\mathcal{M}$):

$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \sigma_{11}^2 & & & & \\ & \ddots & & & \\ & & \sigma_{kk}^2 & & \\ & & & \ddots & \\ & & & & \sigma_{mm}^2 \end{bmatrix} \tag{3.29}$$

The standard deviation σ_{kk} is modelled as quadratic to the depth measurement $d_{k,i}$: $\sigma_{kk} \propto d_{k,i}^2$. It summarises the mapping error between the kth surfel in $\mathcal{M}$ and the associated pixel in $\boldsymbol{I}_i$.

The least-square minimisation of the right-hand term in Equation (3.26) is usually performed utilising the entire set of data as an off-line process. The SLAM system enables real-time mapping by exploiting incremental nature of the map reconstruction process. It does so by an online weighted fusion strategy which is equivalent to the original least square minimisation in Equation (3.26).

The procedure for the online map fusion is detailed as follows. First, given an input image $\boldsymbol{I}_n$, a local surfel model is created $\mathcal{L} = \{\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, ..., \boldsymbol{\ell}_a\}$, which contains the positions, normal vectors, radius, and colour information of each element. For each local surfel $\boldsymbol{\ell}_i$ and its depth measurement d_i, the weight w_i is initialised as $w_i = 1/d_i^2$.

During the process of local surfel model creation, a pre-screening step is applied to remove

unnecessary points. Namely, for each pixel $\boldsymbol{u}$ in $\boldsymbol{I}_n$, if it associates to an existing surfel in $\mathcal{M}$ by projective data association, it won't create a surfel $\boldsymbol{s}$ in the local surfel model $\mathcal{L}$. Otherwise, the pixel which has no associated surfel in $\mathcal{M}$ will create a surfel in $\mathcal{L}$ for map fusion.

The created surfels in $\mathcal{L}$ are incrementally fused into $\mathcal{M}$, and their position and normal vectors are updated according to the following weighted fusion strategy:

$$\boldsymbol{p}_n = \frac{w_{n-1}\boldsymbol{p}_{n-1} + w_n\boldsymbol{p}_n}{w_{n-1} + w_n} \tag{3.30}$$

$$\boldsymbol{n}_n = \frac{w_{n-1}\boldsymbol{n}_{n-1} + w_n\boldsymbol{n}_n}{w_{n-1} + w_n},\ \boldsymbol{n}_n = \frac{\boldsymbol{n}_n}{\|\boldsymbol{n}_n\|} \tag{3.31}$$

$$w_n = w_{n-1} + w_n \tag{3.32}$$

The colour $\boldsymbol{c}$ and radius r can be updated in the same way.

3.4 Results and Discussion

This section reports the results of the experimental evaluation of the proposed F2M SLAM method with map fusion. F2M SLAM was implemented using two image alignment techniques: the first employing RGB (intensity information only) images, and the second RGB-D (intensity and geometry) images as proposed in this section.

Two benchmarks were used for the evaluation of the SLAM methods: the MS RGB-D dataset [34] and the TUM RGB-D dataset [35]. Two metrics were employed to measure the performance of the SLAM methods: the Absolute Translational Error (ATE) and the Relative Pose Error (RPE) [157, 158].

The two metrics are introduced in Section 3.4.1. Section 3.4.2 introduces the datasets and the parameter settings for the algorithms. Section 3.4.3 and Section 3.4.4 present the results of the evaluations using the ATE and RPE metrics, respectively.

3.4.1 Evaluation Metrics

The Absolute Translational Error (ATE) measures the difference in distance between the estimated and the ground-truth trajectories, also known as the *global consistency* of camera egomotion estimation.

Given the estimated camera trajectory $\mathcal{P} = \{\boldsymbol{P}_1, \boldsymbol{P}_2, ..., \boldsymbol{P}_n\}$ and the ground truth $\mathcal{Q} = \{\boldsymbol{Q}_1, \boldsymbol{Q}_2, ..., \boldsymbol{Q}_n\}$, the first step is to identify a rigid transformation $\boldsymbol{S}$ which aligns $\mathcal{P}$ with $\mathcal{Q}$ [159]. The absolute trajectory error $\boldsymbol{E}_a$ for two associated camera poses $\boldsymbol{P}_i$ and $\boldsymbol{Q}_i$ is calculated as $\boldsymbol{E}_{a,i} = \boldsymbol{Q}_i^{-1}\boldsymbol{S}\boldsymbol{P}_i$.

For all associated elements in $\mathcal{P}$ and $\mathcal{Q}$, the absolute trajectory errors are computed as a collection of errors:

$$\mathcal{E}_a = \{\boldsymbol{E}_{a,1}, \boldsymbol{E}_{a,2}, ..., \boldsymbol{E}_{a,n}\} \tag{3.33}$$

The ATE is the root mean squared error (RMSE) over the translational components of all elements in the list of absolute errors $\mathcal{E}_a$:

$$ATE_{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} \|trans(\boldsymbol{E}_{a,i})\|_2^2}{n}},\ \boldsymbol{E}_{a,i} \in \mathcal{E}_a \tag{3.34}$$

where $trans(\boldsymbol{T})$ extracts the translational vector of the given transformation $\boldsymbol{T}$.

The computation for the ATE is illustrated in Figure 3.4a. The estimated trajectory $\mathcal{P}$ (in grey) is aligned and transformed as $\mathcal{P}'$ (in black) first, where $\boldsymbol{P}_i' = \boldsymbol{S}\boldsymbol{P}_i$. The ATE is obtained according to Equation (3.34).

The Relative Pose Error (RPE) measures the relative pose difference between the estimated and the ground-truth camera trajectories over a given trajectory segment. It is also known as the *local accuracy* of camera egomotion estimation. This metric has two components: the translational error and the rotational error.

Given a fixed camera trajectory interval δ, the associated estimated and ground-truth segments

are $\{\boldsymbol{P}_i, \boldsymbol{P}_{i+\delta}\}$ and $\{\boldsymbol{Q}_i, \boldsymbol{Q}_{i+\delta}\}$, respectively. The RPE $\boldsymbol{E}_r$ for the two trajectory segments is calculated as:

$$\boldsymbol{E}_{r,i} = (\boldsymbol{Q}_i^{-1}\boldsymbol{Q}_{i+\delta})^{-1}(\boldsymbol{P}_i^{-1}\boldsymbol{P}_{i+\delta}) \tag{3.35}$$

where $\boldsymbol{Q}_i^{-1}\boldsymbol{Q}_{i+\delta}$ and $\boldsymbol{P}_i^{-1}\boldsymbol{P}_{i+\delta}$ are the relative poses of the trajectory segment leading from pose i to $i+\delta$ for the estimated and ground-truth trajectories, respectively. $\boldsymbol{E}_{r,i}$ is the difference of the relative poses between the estimated and ground-truth trajectories.

For all trajectory segments, the relative pose errors are computed as a collection of errors:

$$\mathcal{E}_r = \{\boldsymbol{E}_{r,1}, \boldsymbol{E}_{r,2}, ..., \boldsymbol{E}_{r,n-\delta}\} \tag{3.36}$$

The translational and rotational errors are obtained by calculating the RMSE values of the translational and rotational components in $\mathcal{E}_r$:

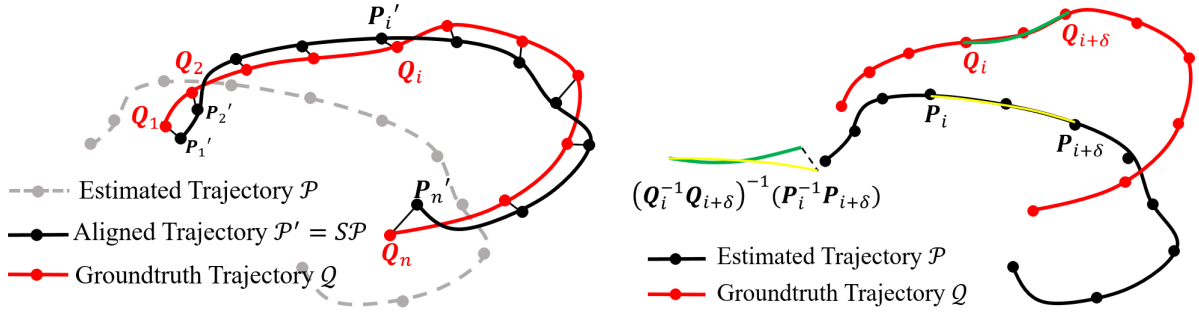
$$RPE_{trans} = \sqrt{\frac{1}{n-\delta}\sum_{i=1}^{n-\delta} \|trans(\boldsymbol{E}_{r,i})\|_2^2}, \ \boldsymbol{E}_{r,i} \in \mathcal{E}_r \tag{3.37}$$

$$RPE_{\angle rot} = \sqrt{\frac{1}{n-\delta}\sum_{i=1}^{n-\delta} \|\angle rot(\boldsymbol{E}_{r,i})\|_2^2}, \ \boldsymbol{E}_{r,i} \in \mathcal{E}_r \tag{3.38}$$

where $trans(\boldsymbol{E}_r)$ extracts the translational component of $\boldsymbol{E}_r$, and $\angle rot(\boldsymbol{E}_r)$ computes the rotation angle for the rigid transformation $\boldsymbol{E}_r$ as:

$$\angle rot(\boldsymbol{E}_{r,i}) = \arccos\frac{tr(\boldsymbol{E}_{r,i}) - 1}{2}, \ \boldsymbol{E}_{r,i} \in \mathcal{E}_r \tag{3.39}$$

The computation of the relative rotational error is illustrated in Figure 3.4b. The relative pose error over all trajectory segments is calculated according to Equation (3.37) and Equation (3.38). In summary, $\mathcal{E}_r$ is the list of relative errors, whilst RPE_{trans} and RPE_{rot} are the RMSE values over all the translational and rotational components of the elements in $\mathcal{E}_r$.



(a) absolute translational error (b) relative pose error

Figure 3.4: Two evaluation metrics for SLAM.

3.4.2 Datasets and Parameter Settings

The F2F and F2M methods for dense RGB-D SLAM were evaluated on two open-source datasets: the TUM RGB-D dataset[35] and the Microsoft RGB-D dataset[34]. The two RGB-D datasets provide RGB-D image sequences with ground-truth camera trajectories. The camera parameters for both datasets are detailed in Table 3.1.

Table 3.1: Parameters of the two datasets.

MS RGB-D Dataset			
resolution	640×480		
depth scale	0.001		
camera intrinsics			
f_x	f_y	c_x	c_y
585.0	585.0	320.0	240.0
TUM RGB-D Dataset			
resolution	640×480		
depth scale	0.0002		
f1 camera intrinsics			
f_x	f_y	c_x	c_y
517.3	516.5	318.6	255.3
f3 camera intrinsics			
f_x	f_y	c_x	c_y
535.4	539.2	320.1	247.6

This study compared the standard F2F with the proposed F2M camera tracking method. Two image alignment methods (i.e., RGB and RGB-D) are employed for camera egomotion estimation. A purely geometric alignment method (ICP) was tested for both the F2F and F2M methods. However, it failed to track the camera pose in most of RGB-D image sequences, and due to its poor performance , it was not included in the comparison. Finally, for the evaluation of the RGB-D image alignment, the photometric and geometric standard deviations (eq. (A.2)) were set empirically to $\sigma_p = 1000.0$ and $\sigma_g = 1.0$, respectively.

3.4.3 Absolute Translational Error

The estimated camera trajectories on MS and TUM RGB-D datasets are evaluated through the ATE metric. The ATEs (RMSE) results are detailed in Table 3.2.

Table 3.2: ATE results on MS and TUM RGB-D Datasets (unit: m). The best result is highlighted in bold.

	F2F RGB	F2F RGB-D	F2M RGB	F2M RGB-D
MS chess	0.117965	0.094436	0.077865	**0.062278**
MS heads	0.135951	0.118754	0.053585	**0.036631**
MS office	0.298758	0.169617	0.156492	**0.060558**
MS pumpkin	0.278099	0.234330	0.172938	**0.169448**
MS red kitchen	0.084734	0.078133	**0.047079**	0.061986
TUM f1 desk	0.127189	0.109843	0.057866	**0.025276**
TUM f1 rpy	0.071191	0.051325	0.269734	**0.031373**
TUM f1 xyz	0.050705	0.031251	0.010801	**0.010603**
TUM f3 structure texture far	0.029211	0.043203	**0.010394**	0.012644
TUM f3 structure texture near	0.030226	0.030246	**0.013621**	0.029085

Overall, Table 3.2 shows that the proposed F2M camera tracking with map fusion strategy outperformed the F2F method. The ATE values attained using the F2M method are smaller than those obtained using the F2F method for all image sequences (MS and TUM). The results indicate that the global consistency of camera trajectory tracking is improved when matching new images against the globally fused model.

The trajectories obtained using the F2M RGB-D camera tracking method show lower ATE values than those obtained using the F2M RGB approach on 7 out of 10 image sequences. This result shows that the joint alignment of photometric and geometric information using F2M tracking and RGB-D images promotes consistent camera tracking.

Figure 3.5 visualises the statistical distribution of the list $\mathcal{E}_a$ of absolute translational errors described in Equation (3.33). Appendix B numerically summarises the statistics of the error list $\mathcal{E}_a$ in Table B.1. Figure 3.5 shows that the F2M method achieved the best performance for the median values on all image sequences, a result which is mirrored in the numeric results in Table B.1. Similarly to the RMSE results in Table 3.2, the F2M camera tracking with RGB-D image alignment gives stably low median values of translational errors over most datasets, whilst the F2M RGB method gives less consistent results, and in particular fails to obtain an acceptable average on the TUM f1 rpy image sequence.

The spread of translational errors in the boxplot assesses the robustness of the methods, i.e., their ability to track the camera trajectory consistently with the ground-truth orbit. The F2F method shows: a) a large spread of the translational errors for most of the datasets b) a particularly high median value of translational errors on three image sequences (e.g., MS red kitchen, TUM structure texture far and near).

The F2M RGB-D method shows a narrow spread of the ATEs on all image sequences. In consideration also of the low median ATE values obtained, it can be concluded that the F2M RGB-D method outperforms the other methods in terms of global consistency and trajectory accuracy.

The estimated trajectories for two example image sequences (TUM f1 desk and MS chess) are

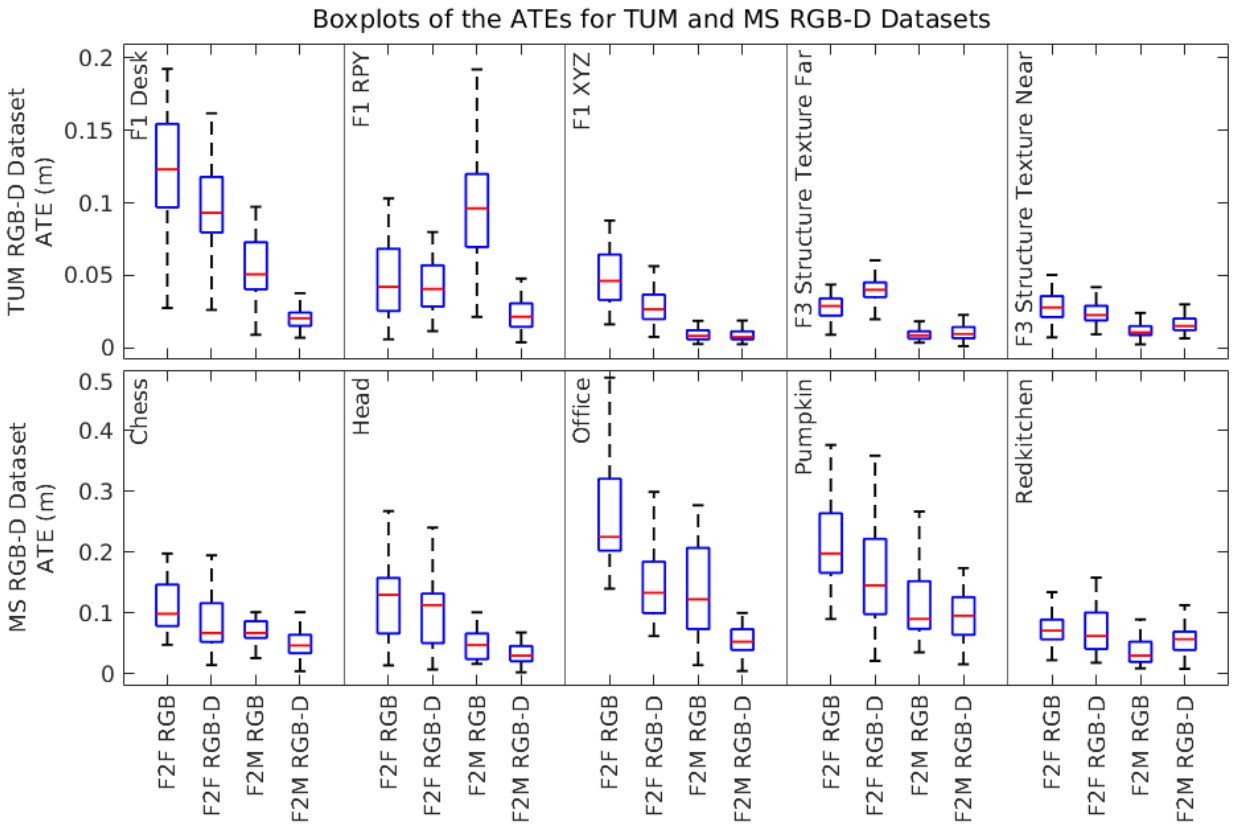
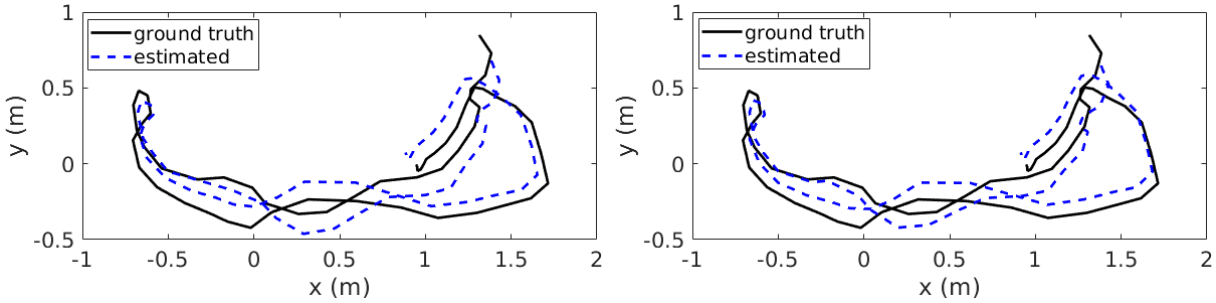


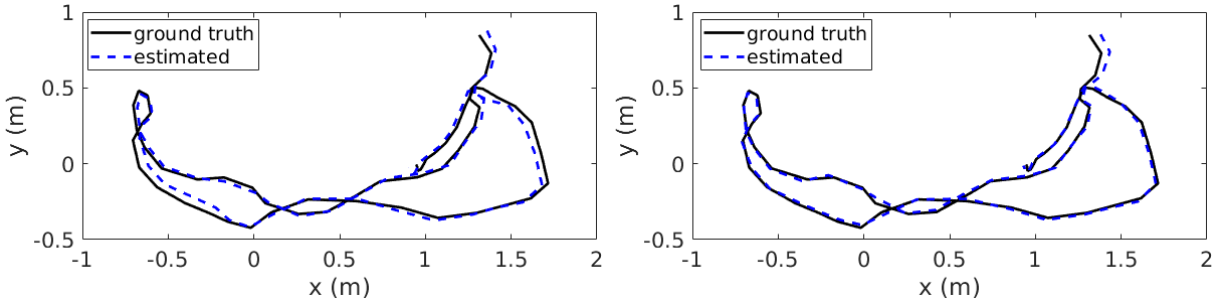
Figure 3.5: The distributions of translational errors for all camera poses.

visualised in Figure 3.6 and Figure 3.7, respectively. The translational component of the estimated camera trajectories is plotted in the 2D xOy plane. The estimated trajectories are plotted in blue dashed line, whilst the ground-truth trajectories are plotted in black solid line. The curves obtained using the F2M RGB-D method best fit the ground truth. F2M RGB tracking gives overall well-aligned trajectories but still has inconsistent trajectory segments. Regarding the F2F methods, there are significant shifts and misalignments between the estimated trajectories and the ground truth. Figure 3.6 and Figure 3.7 demonstrate the global consistency of the proposed method.



(a) TUM f1 desk F2F RGB

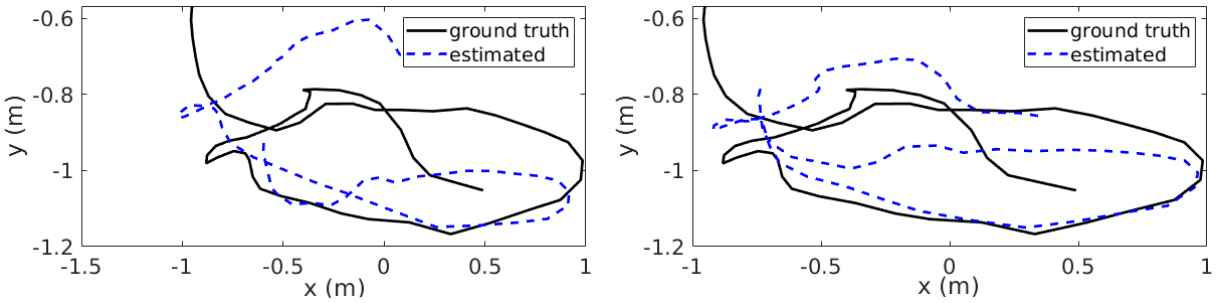
(b) TUM f1 desk F2F RGB-D



(c) TUM f1 desk F2M RGB

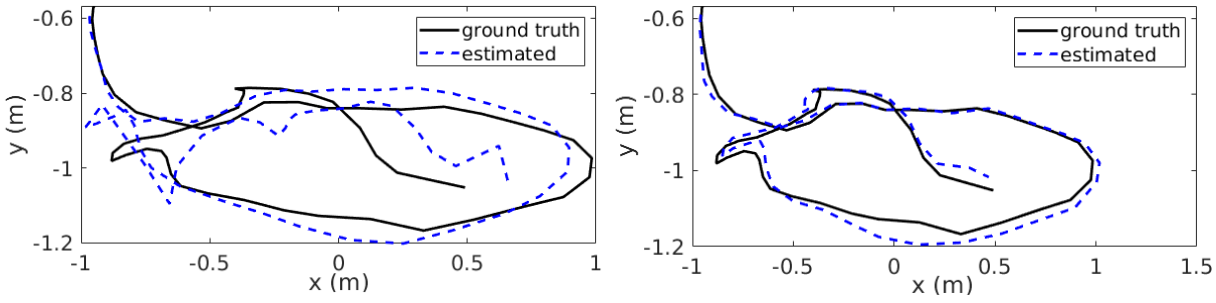
(d) TUM f1 desk F2M RGB-D

Figure 3.6: The estimated and ground-truth trajectories on TUM f1 desk image sequence.



(a) MS office F2F RGB

(b) MS office F2F RGB-D



(c) MS office F2M RGB

(d) MS office F2M RGB-D

Figure 3.7: The estimated and ground-truth trajectories on MS office image sequence.

3.4.4 Relative Pose Error

The RPE corresponds to the local accuracy of the relative pose estimation, and reflects camera drift in the camera egomotion estimation. It consists of two components: translational and rotational errors. In this section, the RPE of the estimated trajectories with a fixed interval of 10 keyframes is measured. The translational and rotational components of the RPE are presented in Table 3.3 and Table 3.4, respectively.

Overall, Table 3.3 and Table 3.4 show that the proposed F2M methods outperformed the F2F methods in terms of both relative translational and rotational errors. Tracking images using the global map model avoids accumulating relative pose error between two consecutive images, thus, the local accuracy in camera egomotion estimation is improved. As the local accuracy is an indicator of camera drift, the F2M method excels at lower camera drift camera egomotion estimation.

Table 3.3 presents the results of relative translational error (RTE). Like for the ATE results in Table 3.2, the F2M RGB-D method obtained more accurate results than that of F2M RGB approach on 7 image sequences. For the remaining three image sequences (MS red kitchen, TUM structure texture far and near), the image sequences contain rich texture and reliable photometric information for RGB image alignment, whilst the depth measurements are too noisy.

Table 3.4 presents the results of the relative rotational error (RRE). The F2M RGB-D tracking method gives the lowest RRE over 8 image sequences. For the remaining two image sequences (TUM f3 structure texture far and near), F2M RGB tracking achieves the best performance due to the rich texture information and reliable photometric measurements in the two image sequences. F2M RGB-D tracking still outperforms the F2F method on these two sequences.

Table 3.3: The relative translational errors on MS and TUM RGB-D Datasets (unit: m).

	F2F RGB	F2F RGB-D	F2M RGB	F2M RGB-D
MS chess	0.122512	0.093629	0.094239	**0.058999**
MS heads	0.170734	0.151938	0.071276	**0.049573**
MS office	0.249302	0.150348	0.226902	**0.094885**
MS pumpkin	0.266132	0.236460	0.227191	**0.219893**
MS red kitchen	0.094603	0.086118	**0.068123**	0.081947
TUM f1 desk	0.042679	0.028394	0.015066	**0.014728**
TUM f1 rpy	0.066308	0.053075	0.314476	**0.038376**
TUM f1 xyz	0.146482	0.119416	0.085471	**0.046727**
TUM f3 structure texture far	0.030008	0.026835	**0.014102**	0.014737
TUM f3 structure texture near	0.026011	0.034726	**0.018049**	0.029814

Table 3.4: The relative rotational errors on MS and TUM RGB-D Datasets (unit: deg).

	F2F RGB	F2F RGB-D	F2M RGB	F2M RGB-D
MS chess	5.361452	4.229916	4.084109	**2.205814**
MS heads	8.911519	7.678661	3.774171	**2.979665**
MS office	8.617817	5.908766	8.710466	**3.087342**
MS pumpkin	5.954713	5.551905	5.874146	**3.307544**
MS red kitchen	4.254842	3.992756	4.151391	**2.389606**
TUM f1 desk	2.513023	1.859462	0.826138	**0.736385**
TUM f1 rpy	4.038709	3.701973	5.756568	**2.108423**
TUM f1 xyz	6.549196	5.605098	4.731415	**2.449999**
TUM f3 structure texture far	0.900952	0.780440	**0.480852**	0.533322
TUM f3 structure texture near	1.193095	1.972002	**0.803939**	1.523902

The distribution of the relative pose errors defined in Equation (3.36) is visualised in Figure 3.8 and Figure 3.9 for the translational and rotational components of the elements of list $\mathcal{E}_r$.

Both figures show that the F2M RGB-D method obtains the lowest median values for both the RTE and RRE. This result is numerically detailed in Table B.2 and Table B.3. The RGB-D based image alignment method presents a more stable performance than the RGB method for F2M

camera tracking, and in particular the F2M RGB-D method achieves the lowest median values of the RTE and RRE on all image sequences. The performance of F2M RGB approach failed in some cases, e.g. the MS office image sequence.

The F2F method shows poor ability to obtain consistently accurate relative poses, as indicated by the large IQR values in Table B.2 and Table B.3. The F2M RGB approach obtained smaller spreads of RTE and RRE compated to the F2F implementations, but still struggled on 5 image sequences, e.g., TUM f1 rpy, MS chess, MS head, MS office and MS pumpkin.

In summary, the proposed online map fusion with F2M camera tracking method effectively improves the global consistency of camera trajectory and reduces camera drift for dense SLAM. Furthermore, RGB-D image alignment allows robust and stable camera tracking with photometric and geometric data fusion.

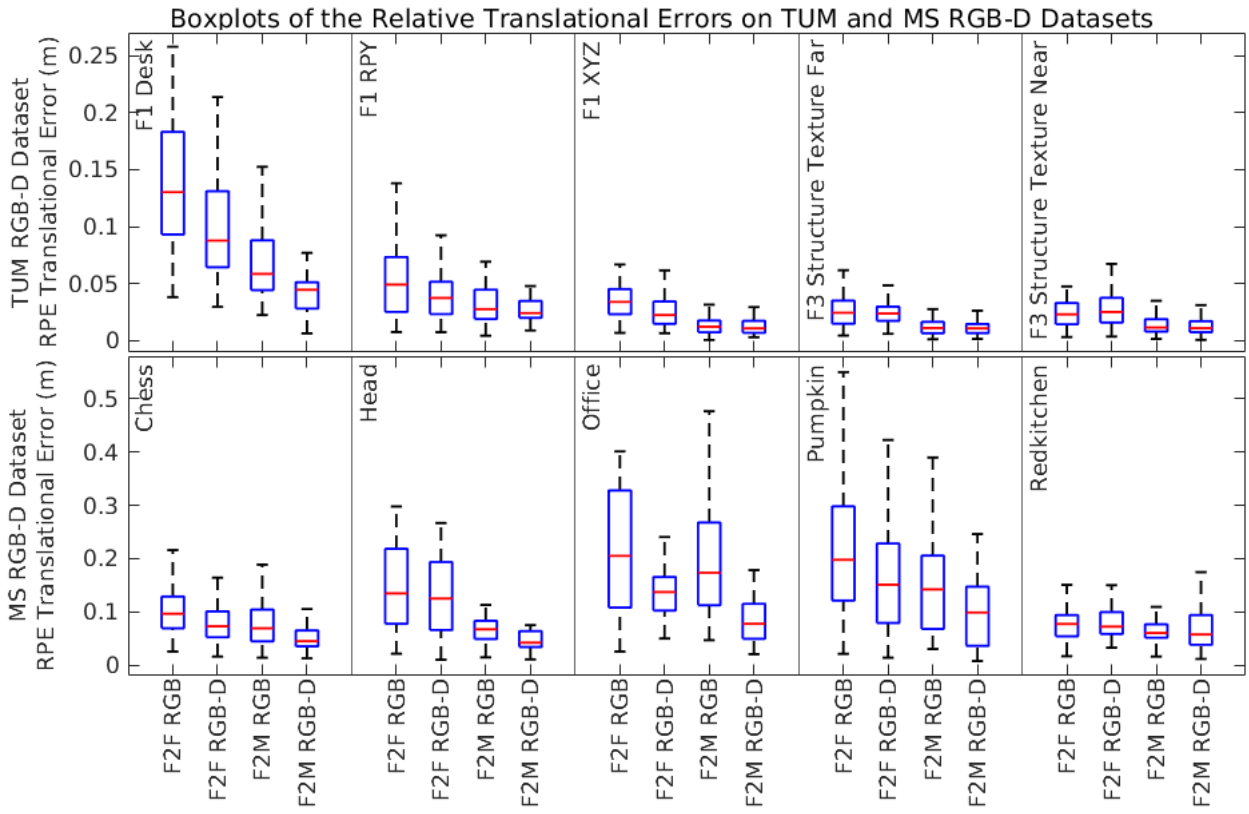


Figure 3.8: The distributions of the relative translational errors on TUM and MS RGB-D datasets.

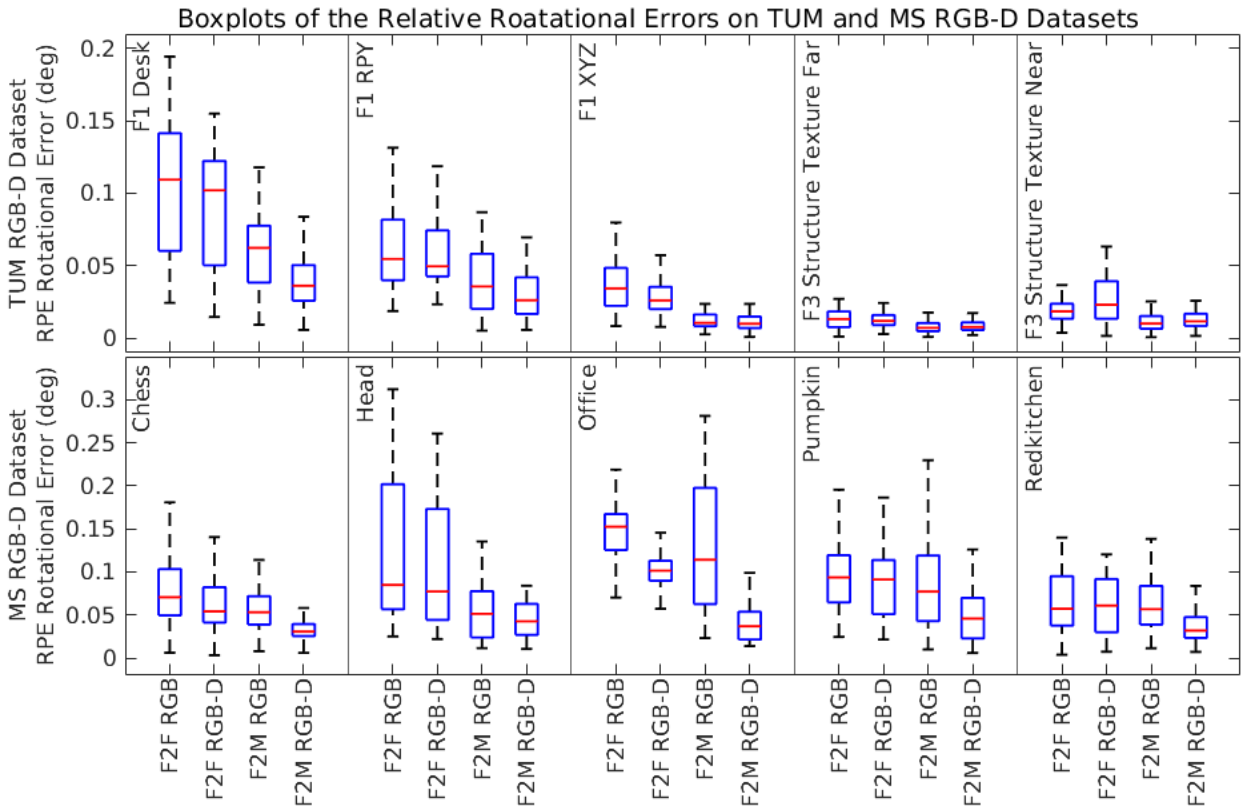


Figure 3.9: The distributions of the relative rotational errors on TUM and MS RGB-D datasets.

Finally, the statistical summary of the distributions visualised in Figure 3.8 and Figure 3.9 is detailed in Table B.2 and Table B.3, respectively.

3.5 Conclusion

This chapter analyses camera drift under the probabilistic framework with factor graph and proposes online map fusion with standard estimation for F2M camera tracking. The performance of the proposed F2M method was compared with that of the F2F camera tracking method on two open-source datasets (TUM RGB-D dataset and MS RGB-D dataset) using two metrics: the absolute translational error and relative pose error. The results show that F2M method outperforms the F2F approach in global consistency and local accuracy. Additionally, RGB-D image alignment is

more reliable and robust than RGB-only alignment in camera egomotion estimation.

Chapter 4

Dense RGB-D SLAM for Reflective and Shiny Objects

It is a challenging task to reconstruct mechanical products with reflective and shiny surfaces in industrial scenarios. This chapter addressed this issue and proposes RSO-SLAM.

Section 4.1 explains the photometric inconsistency for reflective and shiny objects in multiview images. Section 4.2 proposes RSO-SLAM with RGB-D cameras, where local photometric and global geometric alignment are integrated in F2M camera tracking. Section 4.3 evaluates RSO-SLAM and compares it with state-of-the-art methods on two datasets: a plastic object dataset and a metallic object dataset. A case study involving an electric vehicle battery demonstrated the compelling performance of RSO-SLAM: the reconstructed 3D model precisely and consistently replicated the structure of the battery cover.

4.1 Photometric Inconsistency in Multiview Images

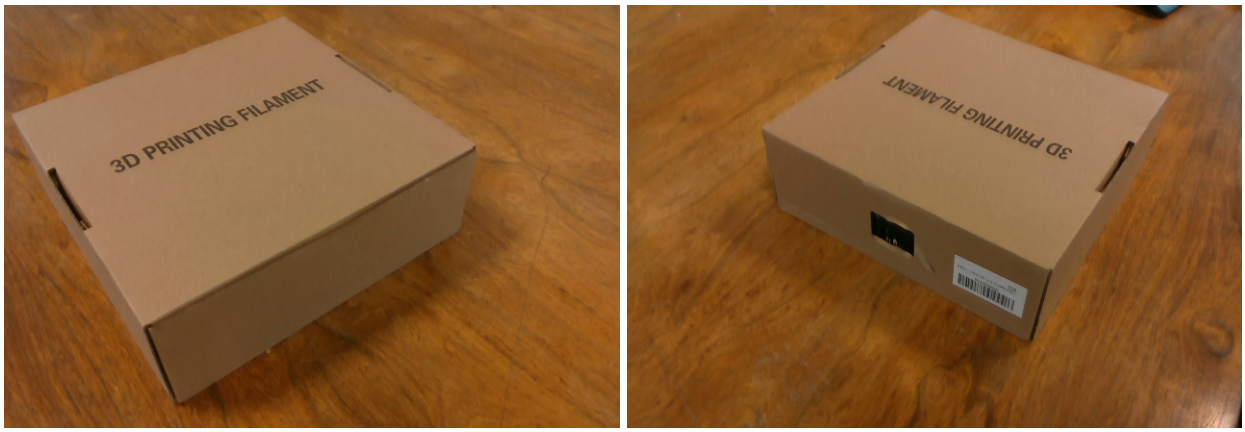
Camera tracking entails the joint photometric and geometric alignment of images in dense RGB-D SLAM. With the linearisation of camera motion, camera pose estimation is solved via direct

alignment of pixel measurements between two consecutive images in the input image sequence, also known as optical flow [36, 37, 57].

Optical flow defines a dense motion field of the intensity pattern of objects between two consecutive images. The estimation of this motion field on the image plane is the foundation for camera pose estimation. Photometric consistency is the fundamental assumption of optical flow, i.e., the objects display a constant light intensity pattern in multiview images. However, common mechanical products (i.e., plastic or metallic surfaces) are reflective and shiny, and this causes a variable intensity pattern in multiview images, i.e., the assumption of photometric consistency is broken.

In F2M camera tracking, the system starts mapping the intensity pattern of objects into the global map model whilst the camera is at its initial position. As the camera moves, the system tracks the camera poses by aligning the intensity pattern in the new images with that of the global map model. This procedure usually results in successful camera egomotion estimation as long as the object has diffuse surfaces, which display consistent intensity patterns in multiview images.

Figure 4.1 shows a box in two multiview images. The intensity pattern of the box in the two images is constant, even if the images are captured at significantly different angles.

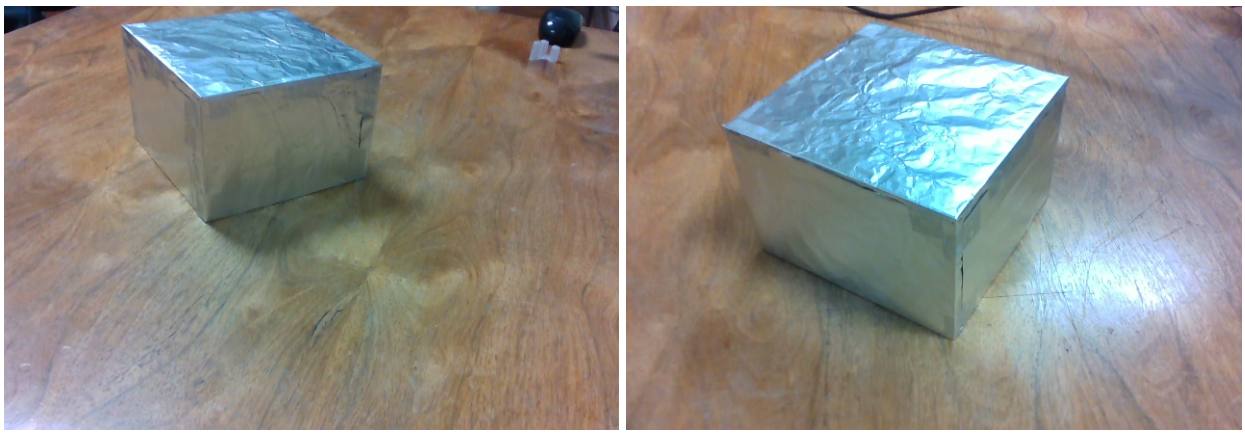


(a) A diffuse box captured at the first viewpoint. (b) A diffuse box captured at the second viewpoint.

Figure 4.1: Consistent intensity pattern of the diffuse box in multiview images

Reflective and shiny objects display variable intensity patterns in multiview images due to their reflective nature. Figure 4.2 shows a shiny cube with metallic surfaces in two images captured by

a camera at different angles. The F2M camera tracking procedure would fail in this case since the assumption of photometric consistency is violated.



(a) A metallic cube captured at the first viewpoint. (b) A metallic cube captured at the second viewpoint.

Figure 4.2: Inconsistent intensity pattern of the shiny metallic object in multiview images

Mechanical products with reflective and shiny surfaces are commonly used in industrial scenarios. This study addressed the problem of reflective and shiny object reconstruction using dense RGB-D SLAM for industrial applications. The proposed RSO-SLAM is detailed in Section 4.2.

4.2 Methodology

This section addresses the problem of photometric inconsistency detailed in Section 4.1, and proposes a novel solution with RGB-D cameras, RSO-SLAM, for reflective and shiny object reconstruction.

Section 4.2.1 formulates camera egomotion estimation as a nonlinear optimisation problem, registering the RGB-D images via local photometric and global geometric alignment. Section 4.2.2 details the residual error functions for local photometric and global geometric alignment. Appendix C illustrates the on-manifold minimisation of the photometric and geometric errors using the Levenberg-Marquardt method.

4.2.1 Camera Egomotion Estimation with Shiny Objects

This chapter introduces a novel F2M camera tracking method for reflective and shiny object reconstruction, i.e., RSO-SLAM. The problem of camera pose estimation is solved via nonlinear optimisation. The on-manifold optimisation strategy with Levenberg-Marquardt method is similar to the strategy with Gauss-Newton method described in Appendix A. The proposed system uses dense RGB-D SLAM and was designed for reflective and shiny object reconstruction in industrial scenarios.

The camera pose for the input image is estimated by jointly minimising the local photometric error and global geometric error:

$$\boldsymbol{T}_n^* = \arg\min_{\boldsymbol{T}_n} \mathcal{F}(\boldsymbol{T}_n) = \arg\min_{\boldsymbol{T}_n} \frac{1}{2} \sum_{i=1}^{m} \|\boldsymbol{\epsilon}_i(\boldsymbol{T}_n)\|^2_{\boldsymbol{\Sigma}_i} = \arg\min_{\boldsymbol{T}_n} \frac{1}{2} \sum_{i=1}^{m} \left\| \begin{bmatrix} \epsilon_{g,i}(\boldsymbol{T}_n) \\ \epsilon_{p,i}(\boldsymbol{T}_n) \end{bmatrix} \right\|^2_{\boldsymbol{\Sigma}_i} \tag{4.1}$$

where $\mathcal{F}$ is the objective function, $\boldsymbol{T}_n$ is the camera pose, $\boldsymbol{\Sigma}_i$ is the covariance matrix, $\boldsymbol{\epsilon}(\cdot)$ is a 2D error vector such as $\boldsymbol{\epsilon} = [\epsilon_g, \epsilon_p]^T$, and $\epsilon_g(\cdot)$ and $\epsilon_p(\cdot)$ are respectively geometric and photometric error functions.

As discussed in Section 4.1, shiny objects (typically, metallic or plastic surfaces) violate the assumption of photometric consistency in multiview images (i.e., *photometric inconsistency*). In this study, the photometric error $\epsilon_p(\cdot)$ in Equation (4.1) exploits the linearisation of camera motion with high frame rate of RGB-D cameras and estimates the camera pose by aligning the intensity information between the previous image and the new image, i.e., *local photometric alignment* for RGB-D camera egomotion estimation. The geometric error $\epsilon_g(\cdot)$ represents the goodness of the alignment of the new depth image with the structure of the global map model, i.e., *global geometric alignment* for RGB-D camera egomotion estimation.

The error functions $\epsilon_p(\cdot)$ and $\epsilon_g(\cdot)$ for local photometric and global geometric alignment in Equation (4.1) are detailed in Section 4.2.2.

4.2.2 Local Photometric and Global Geometric Alignment Error Functions

This subsection describes in detail the local photometric and geometric errors.

Local Photometric Error

Local Photometric alignment is the process of aligning the intensity of the new image $\boldsymbol{I}_n$ with the previously registered image $\boldsymbol{I}_{n-1}$.

Before local photometric alignment, the global map model renders a virtual depth image $\mathfrak{D}_{n-1}$ at the camera pose of the previously registered frame. A synthesised image $\mathfrak{I}_{n-1}$ combines the intensity channel of the previous image $\boldsymbol{I}_{n-1}$ and the rendered depth image $\mathfrak{D}_{n-1}$. That is, the pixel in $\mathfrak{I}_{n-1}$ contains the intensity value provided by $\boldsymbol{I}_{n-1}$ and the depth value obtained from $\mathfrak{D}_{n-1}$.

Local photometric alignment computes and minimises the intensity difference between the associated pixels in $\mathfrak{I}_{n-1}$ and $\boldsymbol{I}_n$.

For each pixel $\boldsymbol{u} \in \mathfrak{I}_{n-1}$, a 3D point $\mathfrak{p}$ is computed by mapping the pixel $\boldsymbol{u}$ into the camera coordinate system τ_{n-1}: $\mathfrak{p} = \pi^{-1}(\boldsymbol{u}, d)$ where d is the depth value of $\boldsymbol{u} \in \mathfrak{I}_{n-1}$, and $\pi^{-1}(\cdot)$ is the inverse projection function. The coordinates of the 3D point $\mathfrak{p}$ are further transformed into the camera coordinate system τ_n of the new image $\boldsymbol{I}_{n-1}$: $\mathfrak{q} = \boldsymbol{T}_n^{-1} \cdot \boldsymbol{T}_{n-1} \cdot \mathfrak{p}$. The 3D point $\mathfrak{q}$ is associated to a pixel $\mathfrak{u} \in \boldsymbol{I}_n$ by projecting $\mathfrak{q}$ onto the image plane of $\boldsymbol{I}_n$: $\mathfrak{u} = \pi(\mathfrak{q})$. The local photometric alignment aims to find the camera pose $\boldsymbol{T}_n$ by minimising the error between the intensity values at $\boldsymbol{u} \in \mathfrak{I}_{n-1}$ and $\mathfrak{u} \in \boldsymbol{I}_n$.

The full expression for local photometric error function is detailed in Equation (4.2).

$$\epsilon_p(\boldsymbol{T}_n) = \mathfrak{I}_{n-1}(\boldsymbol{u}) - \boldsymbol{I}_n \left\{ \pi \left[\boldsymbol{T}_n^{-1} \boldsymbol{T}_{n-1} \cdot \pi^{-1}(\boldsymbol{u}, d) \right] \right\} \tag{4.2}$$

where $\mathfrak{I}_{n-1}(\boldsymbol{u})$ extracts the intensity value of the pixel $\boldsymbol{u}$ in the image $\mathfrak{I}_{n-1}$, and $\boldsymbol{I}_n($ extracts the intensity value of the pixel $\mathfrak{u}$ in the image $\boldsymbol{I}_n$.

Global Geometric Error

Global geometric alignment is the process of registering the depth channel of the new image $\boldsymbol{I}_n$ to the global map model $\mathcal{M}$. It computes and minimises the point-to-plane ICP error [40, 59] between the surfel in $\mathcal{M}$ and the associated pixel point in $\boldsymbol{I}_n$.

For each surfel $\boldsymbol{s} \in \mathcal{M}$, its position $\boldsymbol{p}$ is transformed into a 3D point $\boldsymbol{p}'$ in the camera coordinate system τ_n: $\boldsymbol{p}' = \boldsymbol{T}_n^{-1}\boldsymbol{p}$. A pixel $\boldsymbol{u}'$ is associated to the point $\boldsymbol{p}'$ by projection data association: $\boldsymbol{u}' = \pi(\boldsymbol{p}')$. The pixel $\boldsymbol{u}'$ is mapped into a 3D point $\boldsymbol{p}'' = \pi^{-1}(\boldsymbol{u}', d')$ where d' is the corresponding depth value at $\boldsymbol{u}'$. The geometric error is computed by the point-to-plane ICP error: $\boldsymbol{n}_f^T \cdot (\boldsymbol{p}' - \boldsymbol{p}'')$.

The full expression for geometric error function is detailed in Equation (4.3).

$$\epsilon_g(\boldsymbol{T}_n) = \boldsymbol{n}_f^T \cdot \left\{ \boldsymbol{T}_n^{-1}\boldsymbol{p} - \pi^{-1}\left[\pi(\boldsymbol{T}_n^{-1}\boldsymbol{p}), d'\right] \right\} \tag{4.3}$$

where $\boldsymbol{n}_f$ is the normal vector of the mapped pixel point $\boldsymbol{p}''$.

4.3 Results and Discussion

This section presents the experimental results obtained using RSO-SLAM and compares them to the results obtained using state-of-the-art methods. Section 4.3.1 introduces the two error metrics adopted in the quantitative evaluation. Section 4.3.2 presents the two datasets and the parameter settings used for the quantitative evaluation. Section 4.3.3 and Section 4.3.4 show the reconstruction results on 20 black plastic shapes and 10 shiny metallic shapes, respectively. Finally, Section 4.3.5 presents a case study involving the reconstruction of an industrial product.

4.3.1 Evaluation Metrics

This study evaluates the precision of the reconstructed point cloud using two error metrics: the *Directed Hausdorff Distance* (DHD) [160] and the *Mean Surface Distance* (MSD, also known as

average surface distance) [161].

The *Directed Hausdorff Distance* measures the dissimilarity between two point sets using a given error metric. In this study, the DHD measures the maximum distance between the reconstructed model (the result of the F2M dense SLAM procedure) and the standard shape (the CAD model of the test shape). Given the reconstructed point set $\mathfrak{S} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_n\}$ and the standard model $\mathfrak{T} = \{\boldsymbol{q}_1, \boldsymbol{q}_2, ..., \boldsymbol{q}_m\}$, the DHD is defined as:

$$DHD(\mathfrak{S}, \mathfrak{T}) = \max_{\boldsymbol{p} \in \mathfrak{S}} \left[\min_{\boldsymbol{q} \in \mathfrak{T}} \left(\, d(\boldsymbol{p}, \boldsymbol{q}) \, \right) \right], \; \boldsymbol{p} \in \mathfrak{S}, \boldsymbol{q} \in \mathfrak{T} \tag{4.4}$$

where $d(\boldsymbol{p}, \boldsymbol{q})$ is the Euclidean distance between the two points $\boldsymbol{p}$ and $\boldsymbol{q}$.

The computation of DHD can be described as follows. For each point $\boldsymbol{p} \in \mathfrak{S}$, its closest point $\boldsymbol{p} \in \mathfrak{T}$ is located by nearest neighbour search, and their Euclidean distance $d(\boldsymbol{p}, \boldsymbol{q})$ is computed. The distance list $\mathcal{D} = \{d_1, d_2, ..., d_n\}$ is created, and the $DHD(\mathfrak{S}, \mathfrak{T})$ is calculated as the largest element in the list $\mathcal{D}$. To obviate the effect of outliers in the reconstructed model, the 95% DHD was used. That is, the distance list $\mathcal{D}$ was ranked, and the largest 5% percentile was ignored. The largest element amongst the remaining 95% elements of $\mathcal{D}$ is the 95% DHD.

The *Mean Surface Distance* is the mean value of the distances between the points in $\mathfrak{S}$ and their closest point in $\mathfrak{T}$. This metric indicates the average surface difference between the reconstructed point cloud and the standard shape. The MSD is defined as:

$$MSD(\mathfrak{S}, \mathfrak{T}) = \frac{1}{n} \sum_{i}^{n} d(\boldsymbol{p}_i, \boldsymbol{q}_j), \; \boldsymbol{p}_i \in \mathfrak{S}, \boldsymbol{q}_i \in \mathfrak{T} \tag{4.5}$$

where $\boldsymbol{q}_j$ is the closest point in $\mathfrak{T}$ to $\boldsymbol{p}_i$ in $\mathfrak{S}$, and the point index j is identified by nearest neighbour search:

$$j^* = \arg\min_{j} d(\boldsymbol{p}_i, \boldsymbol{q}_j), \; \boldsymbol{p}_i \in \mathfrak{S}, \boldsymbol{q}_i \in \mathfrak{T} \tag{4.6}$$

4.3.2 Datasets and Parameter Settings

This study tested RSO-SLAM on plastic and metallic datasets.

The image sequences were acquired using an Intel RealSense D435 camera [51]. The Realsense D435 camera captures synchronised RGB-D images (i.e., the RGB and depth images are captured simultaneously) with a high frame rate (30 FPS). The front view and module layout of the camera is shown in Figure 4.3.

(a) Front view. (b) Modules view.

Figure 4.3: The intel Realsense D435 camera contains RGB module and depth module (IR projector, left & right imagers).n

The important parameters in an RGB-D machine vision system include the working distance, resolution, frame rate, intrinsic parameters, and the depth scale. They are given in Table 4.1.

Table 4.1: Parameter setting of the Realsense D435 camera.

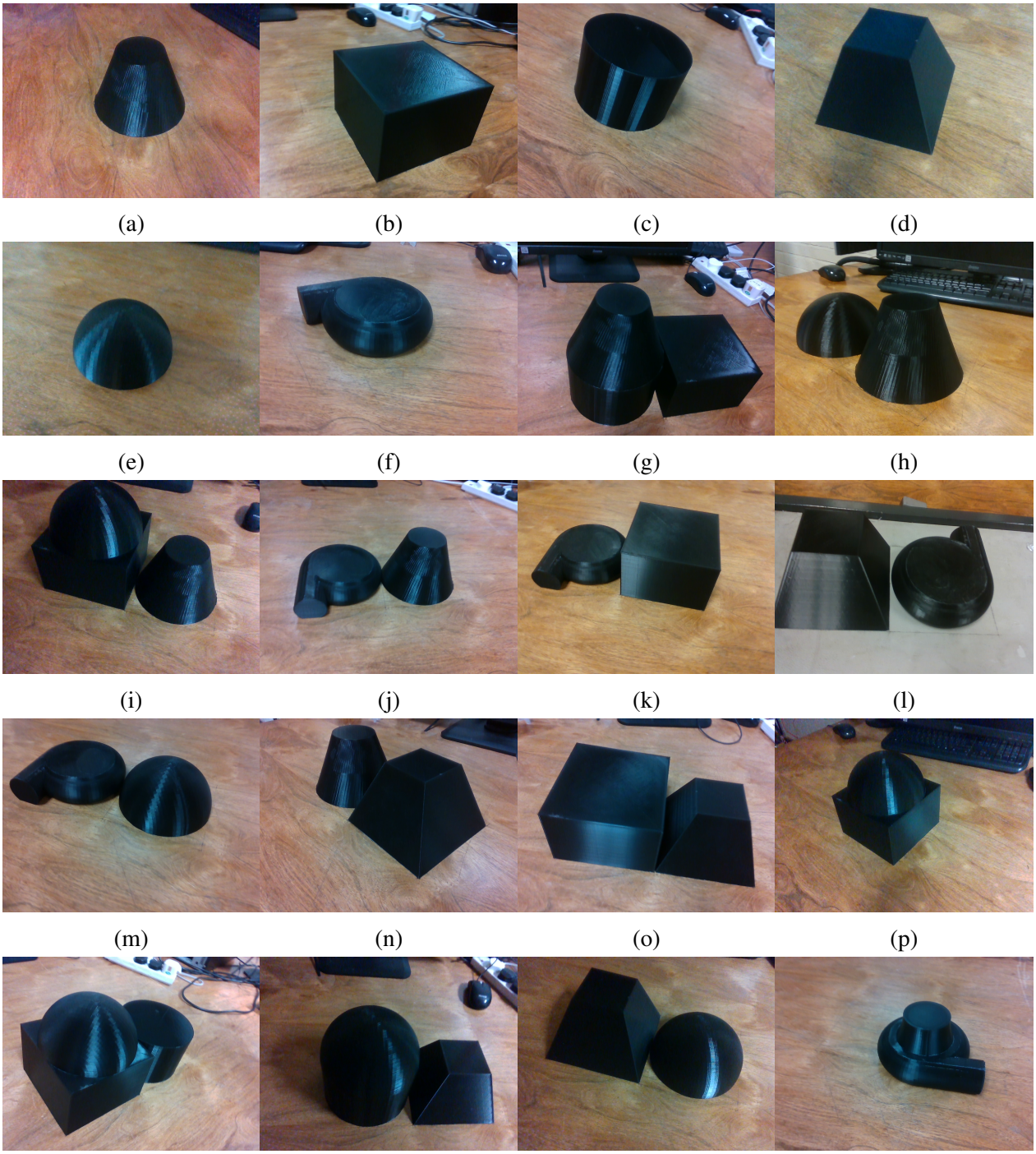
Parameters of intel Realsense D435 camera				
working distance	0.3-3m			
resolution	640×480			
frame rate (FPS)	30			
depth scale	0.001			
intrinsics	f_x	f_y	c_x	c_y
	615.847	616.085	326.006	235.272

The plastic object dataset consists of 20 shapes: 5 primitive shapes (cube, cylinder, semisphere, pyramidal frustum and conical frustum) and 15 complex shapes. It is representative of many industrial components made of plastic materials. Under common ambient lighting conditions, the

strong reflective regions on the objects are shiny, whilst the remaining areas are textureless and lack photometric information.

The metallic object dataset contains 10 shapes: 5 primitive shapes and 5 complex shapes coated with aluminium foil. The metallic object dataset mimics the shiny surface of many mechanical objects. The intensity pattern for the metallic surfaces is complex, and changes drastically with the camera viewpoint. Even with a stable ambient lightning system, photometric consistency doesn't hold for the objects of the metallic dataset.

The objects of the plastic and metallic datasets are shown in Figure 4.4 and Figure 4.5, respectively. The names of their shapes are listed in Table 4.2 and Table 4.3, respectively.



(a) (b) (c) (d)

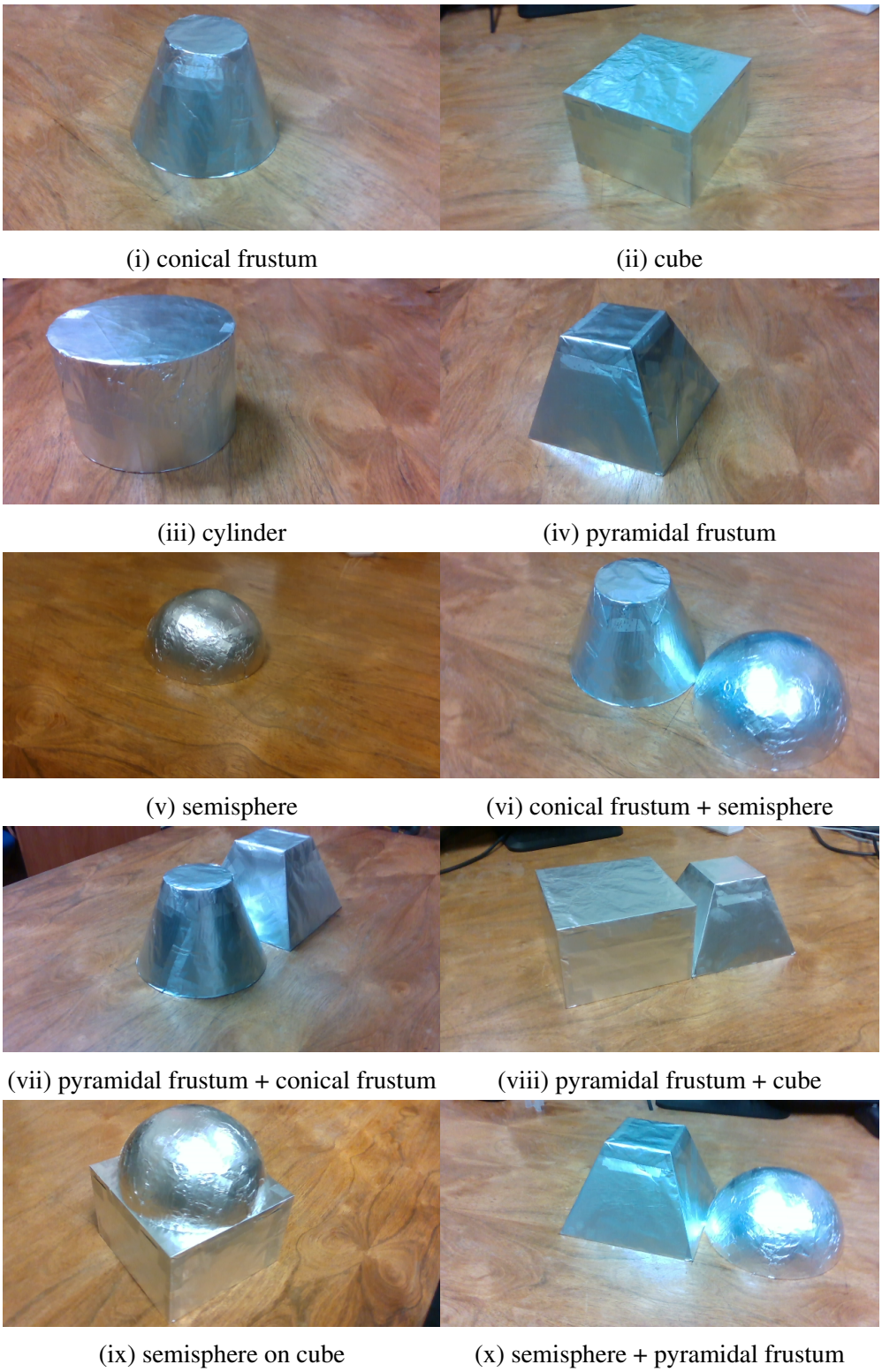
(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

Figure 4.4: The plastic object dataset consists of 20 shapes, including 5 primitive shapes and 15 complex shapes. The names of the shapes are given in Table 4.2



(i) conical frustum (ii) cube

(iii) cylinder (iv) pyramidal frustum

(v) semisphere (vi) conical frustum + semisphere

(vii) pyramidal frustum + conical frustum (viii) pyramidal frustum + cube

(ix) semisphere on cube (x) semisphere + pyramidal frustum

Figure 4.5: The metallic object dataset consists of 10 shapes, including 5 primitive shapes and 5 combined shapes. The names of the shapes are listed in Table 4.3

Table 4.2: Names of the shapes (a)-(t) in the plastic object dataset.

(a)	conical frustum	(k)	housing + cube
(b)	cube	(l)	housing + pyramidal frustum
(c)	cylinder	(m)	housing + semisphere
(d)	pyramidal frustum	(n)	pyramidal frustum + conical frustum
(e)	semisphere	(o)	pyramidal frustum + cube
(f)	housing	(p)	semisphere + cube
(g)	conical frustum + cylinder + cube	(q)	semisphere + cube + cylinder
(h)	conical frustum + semisphere	(r)	semisphere + cylinder + pyramidal frustum
(i)	conical frustum + semisphere + cube	(s)	semisphere + pyramidal frustum
(j)	housing +conical frustum	(t)	small conical frustum + housing

Table 4.3: Names of the shapes (i)-(x) in the metallic object dataset.

(i)	conical frustum	(vi)	conical frustum + semisphere
(ii)	cube	(vii)	pyramidal frustum + conical frustum
(iii)	cylinder	(viii)	pyramidal frustum + cube
(iv)	pyramidal frustum	(ix)	semisphere + cube
(v)	semisphere	(x)	semisphere +pyramidal frustum

4.3.3 Experimental Tests on Reflective Plastic Objects

The RSO-SLAM approach is tested on the plastic object dataset, and the experimental results are compared with those obtained using state-of-the-art F2F and F2M tracking methods, viz. the same control methods used in Chapter 3. The camera tracking approach involves three types of image alignment: RGB (photometric), ICP (geometric), and RGB-D (joint photometric and geometric alignment).

The reconstructed model $\mathcal{M}$ contained the object and the background (the desk). First, the point-based model $\mathcal{M}$ was manually segmented to remove the background (i.e. the desk). The segmented object point cloud is then aligned with the template shape (the CAD model of the object) via the ICP method. Finally, the accuracy of the object reconstruction procedure is quantitatively evaluated using the DHD and MSD metrics.

The results of the plastic dataset are summarised in Table 4.4 (DHD metric) and Table 4.5 (MSD metric). For each object, the best reconstruction result is highlighted in bold. The reconstructed point clouds are visualised in Tables 4.6 to 4.9.

Table 4.4: The results of 95% DHD (m) for the 20 plastic shapes (a)-(t).

		(a)	(b)	(c)	(d)	(e)
F2F	ICP	0.026263	0.015433	0.030957	0.035989	0.026752
	RGB	0.037155	0.019017	0.019073	0.013822	0.026724
	RGB-D	0.017869	0.012039	0.014619	0.01047	0.010719
F2M	ICP	0.02542	**0.00717**	**0.0099**	0.016697	0.011374
	RGB	0.101428	0.01311	0.088712	0.020528	0.037909
	RGB-D	0.021983	0.013906	0.054671	0.013894	0.182116
RSO-SLAM	RGB-D	**0.00914**	**0.00789**	**0.00997**	**0.00677**	**0.00815**

		(f)	(g)	(h)	(i)	(j)
F2F	ICP	0.013641	0.024496	0.030685	0.014668	0.032197
	RGB	0.007112	0.028716	0.043625	0.017533	0.015113
	RGB-D	0.00754	0.020985	0.026486	0.018615	0.011506
F2M	ICP	0.167794	0.018807	0.041831	0.025363	0.02766
	RGB	0.020632	0.083615	0.057836	0.040447	0.050014
	RGB-D	0.028787	0.084795	0.01274	0.106507	0.016637
RSO-SLAM	RGB-D	**0.00669406**	**0.01601**	**0.01005**	**0.00973**	**0.00849**

		(k)	(l)	(m)	(n)	(o)
F2F	ICP	0.025489	0.019005	0.029033	0.033979	0.013365
	RGB	0.017383	0.013448	0.018467	0.032525	0.017521
	RGB-D	0.015746	0.012753	0.013791	0.029621	0.011268
F2M	ICP	0.018539	0.019227	0.013115	0.028699	0.012546
	RGB	0.12286	0.023206	0.040695	0.050843	0.03071
	RGB-D	0.023931	0.010841	0.02891	0.043033	0.008795
RSO-SLAM	RGB-D	**0.01476**	**0.00720559**	**0.00973**	**0.01471**	**0.00802**

		(p)	(q)	(r)	(s)	(t)
F2F	ICP	0.027642	0.04157	0.01641	0.043231	0.014216
	RGB	0.027398	0.033457	0.014833	0.011551	0.009494
	RGB-D	0.022301	0.028799	0.011376	0.011535	0.007394
F2M	ICP	0.020608	0.027054	**0.00843**	0.047724	0.015843
	RGB	0.102406	0.034966	0.034584	0.040352	0.076007
	RGB-D	0.042238	0.015094	0.013676	0.011537	0.0436
RSO-SLAM	RGB-D	**0.01652**	**0.0126**	**0.00902**	**0.00861**	**0.00547**

Table 4.5: The results of MSD (m) for the 20 plastic shapes (a)-(t).

		(a)	(b)	(c)	(d)	(e)
F2F	ICP	0.009879	0.004979	0.012065	0.014729	0.012671
	RGB	0.014163	0.006997	0.007504	0.005365	0.010392
	RGB-D	0.007065	0.004499	0.00611	0.004042	0.004179
F2M	ICP	0.009123	0.003002	0.004713	0.004414	0.003868
	RGB	0.048178	0.004677	0.038913	0.006525	0.015332
	RGB-D	0.007724	0.004379	0.019199	0.005944	0.073023
RSO-SLAM	RGB-D	**0.00337**	**0.0027**	**0.0039**	**0.00244**	**0.00313**
		(f)	(g)	(h)	(i)	(j)
F2F	ICP	0.004704	0.007194	0.0096	0.004664	0.010812
	RGB	0.002916	0.009794	0.013796	0.005836	0.006045
	RGB-D	0.002962	0.007139	0.010989	0.005447	0.004531
F2M	ICP	0.031278	0.0064	0.0167	0.006024	0.007255
	RGB	0.007728	0.03744	0.025314	0.014009	0.01333
	RGB-D	0.009776	0.028787	0.004864	0.040076	0.004736
RSO-SLAM	RGB-D	**0.00251235**	**0.00553**	**0.00406**	**0.00388**	**0.00328**
		(k)	(l)	(m)	(n)	(o)
F2F	ICP	0.008064	0.004964	0.008764	0.015754	0.005365
	RGB	0.005921	0.004207	0.007243	0.014529	0.006856
	RGB-D	0.004748	0.003861	0.005527	0.012598	0.004678
F2M	ICP	0.006163	0.004545	0.00436	0.013003	0.004294
	RGB	0.057261	0.007309	0.01279	0.021929	0.009359
	RGB-D	0.007413	0.00347	0.008608	0.018855	0.003087
RSO-SLAM	RGB-D	**0.00427**	**0.00259115**	**0.00364**	**0.00667**	**0.00304**
		(p)	(q)	(r)	(s)	(t)
F2F	ICP	0.010273	0.012695	0.005844	0.019183	0.004425
	RGB	0.008576	0.011433	0.005332	0.004812	0.003336
	RGB-D	0.007269	0.009772	0.004387	0.004429	0.002797
F2M	ICP	0.006739	0.008436	0.003725	0.019213	0.005491
	RGB	0.039506	0.012277	0.008376	0.012405	0.026088
	RGB-D	0.016624	0.00571	0.003989	0.00416	0.017722
RSO-SLAM	RGB-D	**0.00443**	**0.00507**	**0.00339**	**0.00331**	**0.00212**

Table 4.6: Results of the reconstruction procedure - objects (a)-(e).

		(a)	(b)	(c)	(d)	(e)
F2F	ICP					
	RGB					
	RGB-D					
F2M	ICP					
	RGB					
	RGB-D					
RSO-SLAM	RGB-D					

Table 4.7: Results of the reconstruction procedure - objects (f)-(j).

		(f)	(g)	(h)	(i)	(j)
F2F	ICP					
	RGB					
	RGB-D					
F2M	ICP					
	RGB					
	RGB-D					
RSO-SLAM	RGB-D					

Table 4.8: Results of the reconstruction procedure - objects (k)-(o).

Table 4.9: Results of the reconstruction procedure - objects (p)-(t).

		(p)	(q)	(r)	(s)	(t)
F2F	ICP					
	RGB					
	RGBD					
F2M	ICP					
	RGB					
	RGBD					
RSO-SLAM	RGBD					

The F2F ICP method obtains poor reconstruction results on the plastic object dataset, as indicated by the high DHD (95%) and MSD values in Table 4.4 and 4.5. The poor performance of the F2F ICP method is also confirmed by the inconsistent and sometimes erroneous geometric structure of the reconstructed objects in Tables 4.6 to 4.9. Poor geometric reconstruction results usually also affect the quality of the camera trajectory estimation, due to the synergy between camera trajectory and reconstructed structure. This result could not be verified due to the lack of a high-precision motion capture system to accurately capture the ground-truth trajectory. Overall, the poor results obtained using the F2F ICP method confirm the limitations of pure-geometric alignment (ICP algorithm [40, 59]) in absence of clear textural elements that help the registration of the partial views [162].

The F2F RGB camera tracking method shows similar results to those obtained by the F2F ICP. Tables 4.6 to 4.9 show that the F2F RGB failed to correctly reconstruct 14 objects, e.g., (a), (c)-(e), (g)-(h), (k), and (m)-(s), due to the unreliability of its photometric-only alignment. F2F RGB-D method improves the performance of the F2F RGB method thanks to the use of depth information. However, the performance of the F2F RGB-D method is still limited, as shown by the high values of DHD and MSDin Table 4.4 and 4.5.

The F2M ICP method tracks the camera poses by aligning the depth image with the structure of the global map model via the ICP algorithm with projective data association. This approach adopts pure geometric alignment to avoid the *photometric inconsistency* caused by the reflective and shiny objects. Similarly to the F2F ICP method, the F2M ICP struggles to reconstruct non-structured scenes due to its pure geometric nature. Its reconstruction accuracy is still poor, as indicated by the high DHD and MSD values obtained in Table 4.4 and 4.5. The F2M ICP method fails most notably on the reconstruction of the shape of three plastic objects: (f), (h), and (s).

The F2M RGB and RGB-D methods (presented in Chapter 3) are based on photometric alignment and cannot handle photometric inconsistency. Hence, they fail to attain acceptable DHD and MSD values on most of the shapes of the plastic object dataset.

Finally, the RSO-SLAM approach demonstrates the effectiveness of the local photometric and global geometric alignment approach for camera tracking. It succeeds in reconstructing the object structures for all the shapes in the plastic object dataset, and obtains the lowest DHD and MSD values in the results of Table 4.4 and 4.5.

4.3.4 Evaluation on Reflective Metallic Objects

This section tests the RSO-SLAM approach on the metallic object dataset. The results of the quantitative evaluation are reported in the form of the DHD and MSD metrics in Table 4.10. The reconstructed point clouds for the ten metallic shapes are visualised in Table 4.11 and Table 4.12.

Overall, it can be said that the results obtained by the various methods on the metallic objects are consistent with those attained on the plastic object dataset.

The F2F ICP method failed to reconstruct the object structures as shown in Table 4.11 and Table 4.12, and indicated by the high DHD and MSD values in Table 4.10. The causes of this failure are the ill-posed task of pure geometric alignment in structureless scenes, and likely camera drift in F2F tracking. Likewise, the F2M ICP approach based on pure geometric alignment fails to obtain acceptable results.

Like in the case of plastic objects, the reconstruction results obtained on the shiny object set using the F2F RGB and RGB-D methods are characterised by large errors (DHD and MSD values).

The F2M RGB and RGB-D methods fail on most of metallic shapes due to the photometric inconsistency caused by the shiny surfaces. The corrupted structures suggest that the camera tracking was lost during the alignment due to the photometric inconsistency between the global map model and the images.

The RSO-SLAM demonstrates its effectiveness in reconstructing metallic shapes in this experiment, as indicated by the correctly reconstructed shapes in in Table 4.11 and Table 4.12. The RSO-SLAM also outperforms the control algorithms in terms of DHD and MSD values.

In summary, this experiment demonstrated the effectiveness of the novel RSO-SLAM for shiny

object reconstruction. The RSO-SLAM method represents thus a practical solution using dense RGB-D SLAM for industrial applications, where reconstruction of reflective and shiny objects is problematic.

Table 4.10: The 95% DHD (m) and MSD (m) results for metallic object dataset.

DHD 95% (m)						
		(i)	(ii)	(iii)	(iv)	(v)
F2F	ICP	0.021336	0.0224527	0.021336	0.030116	0.0272505
	RGB	0.007655	0.0092218	0.007655	0.0229543	0.0110119
	RGB-D	0.008754	0.00841932	0.008754	0.0105228	0.00812235
F2M	ICP	0.016845	0.0190973	0.016845	0.0189532	0.0684275
	RGB	0.031874	0.0986946	0.031874	0.028615	0.0753718
	RGB-D	0.085538	0.191169	0.085538	0.162	0.0334283
RSO-SLAM	RGB-D	**0.005362**	**0.00697608**	**0.005362**	**0.00596662**	**0.00577169**
		(vi)	(vii)	(viii)	(ix)	(x)
F2F	ICP	0.048109	0.0198746	0.0308096	0.0168487	0.019507
	RGB	0.0110385	0.00765861	0.0127435	0.023215	0.008672
	RGB-D	0.00843284	0.00616733	0.00954607	0.0128774	0.00674
F2M	ICP	0.0864617	0.0132094	0.0124602	0.0204451	0.021279
	RGB	0.0532087	0.0414908	0.0461073	0.0480687	0.060448
	RGB-D	0.0920755	0.0156055	0.0879676	0.0390895	0.043082
RSO-SLAM	RGB-D	**0.00721699**	**0.00475508**	**0.00849157**	**0.00743441**	**0.005581**

MSD (m)						
		(i)	(ii)	(iii)	(iv)	(v)
F2F	ICP	0.005942	0.0077564	0.005942	0.0125659	0.00945766
	RGB	0.003142	0.00347618	0.003142	0.0106516	0.00500317
	RGB-D	0.003368	0.00306696	0.003368	0.00451081	0.00353009
F2M	ICP	0.005236	0.0063098	0.005236	0.0077813	0.030558
	RGB	0.010917	0.0357534	0.010917	0.00842847	0.0176336
	RGB-D	0.036577	0.0710295	0.036577	0.0635211	0.0126965
RSO-SLAM	RGB-D	**0.002067**	**0.00275289**	**0.002067**	**0.00233407**	**0.00240821**
		(vi)	(vii)	(viii)	(ix)	(x)
F2F	ICP	0.0168978	0.00665205	0.00855945	0.00598901	0.007346
	RGB	0.003762	0.00317626	0.00503065	0.00874526	0.003567
	RGB-D	0.00303186	0.00251161	0.00349407	0.00466455	0.002817
F2M	ICP	0.0366566	0.00395702	0.00491499	0.00506884	0.007512
	RGB	0.02135	0.0166113	0.0154166	0.0178795	0.025813
	RGB-D	0.0390955	0.00544273	0.0317302	0.0115266	0.017047
RSO-SLAM	RGB-D	**0.00277063**	**0.00222332**	**0.00303634**	**0.0028288**	**0.002287**

Table 4.11: Visual images for the 5 reconstructed metallic shapes (i)-(v) presented in this table.

		(i)	(ii)	(iii)	(iv)	(v)
F2F	ICP					
	RGB					
	RGB-D					
F2M	ICP					
	RGB					
	RGB-D					
RSO-SLAM	RGB-D					

Table 4.12: Visual images for the 5 reconstructed metallic shapes (vi)-(x) presented in this table.

		(vi)	(vii)	(viii)	(ix)	(x)
F2F	ICP					
	RGB					
	RGB-D					
F2M	ICP					
	RGB					
	RGB-D					
RSO-SLAM	RGB-D					

4.3.5 Case Study on an Industrial Product: Electric Vehicle Battery Reconstruction

This study presents a case study involving an industrial product, the cover of an electric vehicle battery. The cover is a common industrial object with metallic surfaces, and its reconstruction is relevant to the field of remanufacturing in the rapidly expanding electric vehicle market. A picture of the battery cover is shown in Figure 4.6.

Figure 4.6: The battery cover used in this case study.

From Figure 4.6, it can be said that the reconstruction task presents three main challenges using dense RGB-D SLAM. 1) the metallic surface violates the assumption of photometric consistency: as the camera moves, the bright and dark areas are not the same in the image sequence. That is, the reconstruction system needs to handle inconsistent photometric patterns. 2) Camera drift and loop closing of the camera trajectory: due to the large size of the battery cover and the limited field

of view of the camera, the camera has to move circularly around the object to capture the whole battery cover surface. The reconstruction system needs to close the camera trajectory correctly and reduce camera drift. 3) non-structured information for camera pose estimation: due to the limited field of view of the camera, some partial views of the object will include only the planar structure of the battery cover. Thus, camera pose estimation might easily become ill-posed due to the insufficient geometric constraints in the depth images. The reconstruction system needs to handle the non-structured depth images in the image sequence.

The reconstructed shapes of the battery cover are visualised in Table 4.13 and 4.14.

Table 4.13: Reconstructed metallic battery cover images using *F2F* tracking methods.

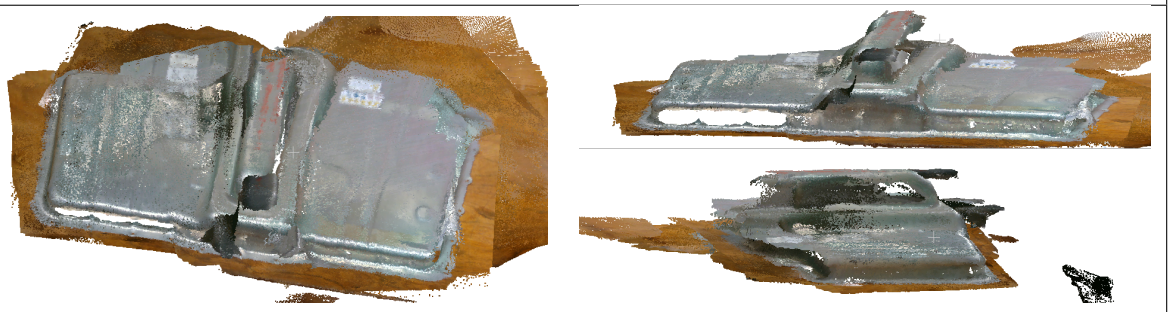
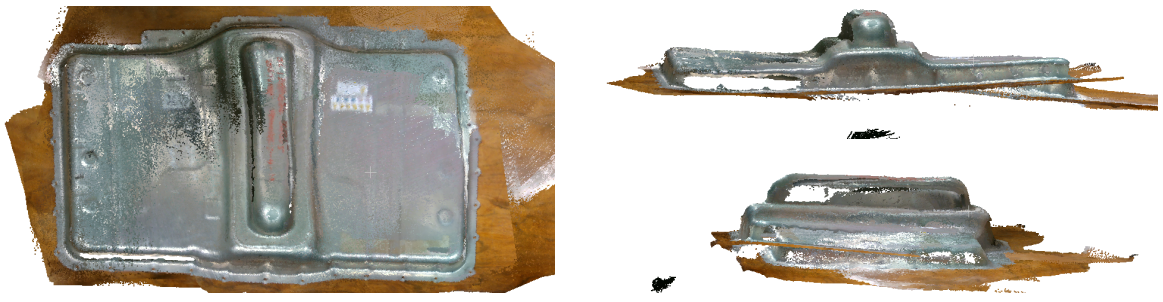
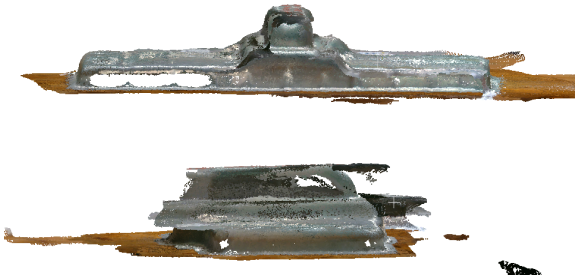
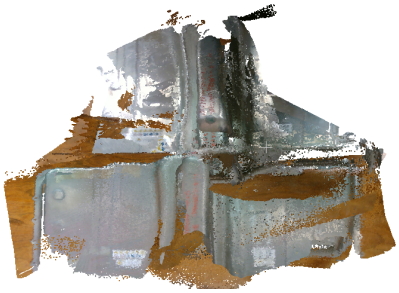
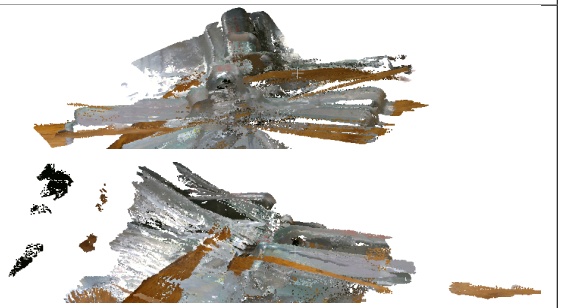
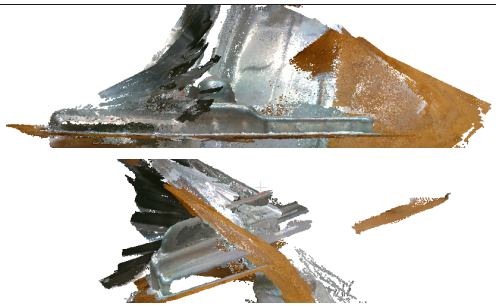
	Top view	Side view
F2F ICP		
F2F RGB		
F2F RGB-D		

Table 4.14: Visual images of the reconstructed metallic battery cover using *F2M* methods.

	Top view	Side view
F2M ICP		
F2M RGB		
F2M RGB-D		
our F2M RGB-D		

The pure geometric alignment approaches (i.e., F2F ICP and F2M ICP) fail to reconstruct the shape of the battery cover. The two approaches are unable to handle non-structured information (*Challenge 3*).

The F2F RGB and RGB-D approaches correctly reconstruct the structure of the battery cover overall. However, as seen from the side views of the reconstructed object in Table 4.13, the structure of the cover is not correctly closed. The inconsistent geometry indicates that camera drift occurs in camera egomotion, thus, the camera trajectory is not accurately closed. Namely, the reconstruction system fails to address *Challenge 2*.

The F2M RGB and RGB-D methods fail on the metallic battery cover and give erroneous structures as seen in Table 4.14. The failure of the two approaches is mainly caused by their inability to address *Challenge 1*, namely the inconsistent photometric patterns in the sequence of images.

The RSO-SLAM approach addresses all the three challenges in this task. The geometric structure of the battery cover is correctly reconstructed, since the local photometric alignment provides auxiliary constraints for camera pose estimation in textureless scenes (addressing *Challenge 3*). Moreover, the joint local photometric and global geometric alignment scheme effectively handles shiny metallic surfaces (addressing *Challenge 1*). That is, te shape of the battery cover is reconstructed consistently and precisely, which indicates the accurate estimation of the camera trajectory. The integrated F2M camera tracking reduces camera drift and tackles camera orbit closing (addressing *Challenge 2*).

In summary, the industrial case study proves the excellence of the RSO-SLAM approach for handling reflective and shiny objects.

4.4 Conclusion

This chapter addressed the challenges of reflective and shiny object reconstruction using dense RGB-D SLAM in industrial applications. The study generated a novel approach, RSO-SLAM, for reflective and shiny object reconstruction, which performs F2M camera tracking approach with joint local photometric and global geometric alignment. The RSO-SLAM was evaluated on a plastic object dataset and a metallic object dataset, and compared with the state-of-the-art in the literature. The experimental results show that the RSO-SLAM approach outperforms the control methods, and excels in obtaining consistent and precise geometric structures of plastic and metallic objects. A case study involving an electric vehicle battery cover demonstrated the superior performance of the RSO-SLAM approach in the reconstruction of a common industrial product.

Chapter 5

The SVD-Enhanced Bees Algorithm, a Novel Procedure for Point Cloud Registration

This chapter tackles object localisation as the problem of point cloud registration, and proposes a novel procedure using the SVD-enhanced Bees Algorithm. Section 5.1 formulates the problem of point cloud registration as a process of minimising an objective function. Section 5.2 describes the encoding scheme and assessment of the candidate solutions. Section 5.3 introduces the classic ICP algorithm for point cloud registration. Section 5.4 presents the standard Bees Algorithm and the SVD-enhanced global optimisation mechanism. Section 5.5 introduces the control algorithms, including Evolutionary Algorithm and Particle Swarm Optimisation that will be used for comparison. Section 5.6 and Section 5.7 present the experimental set-up and results. Section 5.8 discusses the findings of the experimental results.

5.1 Problem Formulation

Point cloud registration is the problem of estimating the spatial transformation that aligns two point clouds.

Formally, given a point cloud $\mathcal{X} = \{\boldsymbol{x}_i \mid i = 1, \ldots, N\}$ (the source) and a point cloud $\mathcal{Y} = \{\boldsymbol{y}_j \mid j = 1, \ldots, M\}$ (the target), the objective of 3D registration is to find the rigid transformation $\boldsymbol{T} = \boldsymbol{T_\mu}$, namely the rotation matrix $\boldsymbol{R} = \boldsymbol{R_\mu}$ and translation vector $\boldsymbol{t} = t_\mu$, that minimises function $\mathcal{E}$:

$$\boldsymbol{T_\mu} = \arg\min_{\boldsymbol{T}} \mathcal{E}(\boldsymbol{T}, \mathcal{X}, \mathcal{Y}) \tag{5.1}$$

where $\mathcal{E}$ is the $L2$-norm point-to-point mean square error function

$$\mathcal{E}(\boldsymbol{T}, \mathcal{X}, \mathcal{Y}) = \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{cp}_i - (\boldsymbol{R} \cdot \boldsymbol{x}_i + \boldsymbol{t})\|_2^2 \tag{5.2}$$

and $\boldsymbol{cp}_i$ is the closest point in $\mathcal{Y}$ to the transformation $\boldsymbol{y_i'} = (\boldsymbol{R} \cdot \boldsymbol{x}_i + \boldsymbol{t})$ of vector $\boldsymbol{x}_i$. That is:

$$\boldsymbol{cp}_i = \arg\min_{y_j \in \mathcal{Y}} \|\boldsymbol{y}_j - \boldsymbol{y_i'}\|_2^2 = \arg\min_{y_j \in \mathcal{Y}} \|\boldsymbol{y}_j - (\boldsymbol{R}\boldsymbol{x}_i + \boldsymbol{t})\|_2^2 \tag{5.3}$$

Equation (5.3) is used to drive the nearest neighbour search process and find the closest point correspondence between $\mathcal{X}$ and $\mathcal{Y}$. This is a well designed method for local registration and has been adopted widely for the ICP algorithm and its variants. It has been shown that the error function $\mathcal{E}$ described in Equation (5.2) is non-convex [135].

5.2 Candidate Solutions

This section describes how the candidate solutions are encoded and evaluated.

5.2.1 Encoding Scheme

A transformation $\boldsymbol{T}$ for point cloud registration contains a 3×3 rotation matrix $\boldsymbol{R}$ and a 3D translation vector $\boldsymbol{t} = (t_1, t_2, t_3)^T$. It is usually represented by the 3D Special Euclidean group $SE(3)$:

$$SE(3) = \{\boldsymbol{T} | \boldsymbol{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}_3^T & 1 \end{bmatrix}, \boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}, \, det(\boldsymbol{R}) = 1\} \tag{5.4}$$

The rotation matrix is difficult to use directly in optimisation problems, since it has 3×3 elements, but only 3 degrees of freedom following the application of the two constraints in Equation (5.4). It is important to encode the rotation component of $SE(3)$ into a vector without constraints for the BA.

In this study, angle-axis vector encoding is adopted to represent the rotation matrix. The angle-axis vector is an unconstrained 3D vector $\boldsymbol{r} = (r_1, r_2, r_3)^T$. The direction of this vector $\boldsymbol{r}$ represents the rotation axis of $\boldsymbol{R}$, which is a unit vector $\boldsymbol{n} = \boldsymbol{r}/|\boldsymbol{r}|$. The modulus of $\boldsymbol{r}$ represents the rotation angle of $\boldsymbol{R}$, which is a scalar $\theta = |\boldsymbol{r}|$. The transformation between the encoded rotation vector $\boldsymbol{r}$, and the rotation matrix $\boldsymbol{R}$ is implemented by Rodrigues' rotation formula [163].

Equation (5.5) shows the transformation from $\boldsymbol{r}$ to $\boldsymbol{R}$:

$$\boldsymbol{R} = \cos\theta \boldsymbol{I} + (1 - \cos\theta)\boldsymbol{r}\boldsymbol{r}^T + \sin\theta \boldsymbol{r}^\wedge \tag{5.5}$$

where $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{r}^\wedge$ is the skew-symmetric matrix of vector $\boldsymbol{r}$, that is, $\boldsymbol{r}^\wedge = [\boldsymbol{r}]_\times$.

The rotation vector $\boldsymbol{r}$ can be computed from $\boldsymbol{R}$:

$$\theta = \arccos\left(\frac{tr(\boldsymbol{R}) - 1}{2}\right), \quad \boldsymbol{R}\boldsymbol{n} = \boldsymbol{n} \tag{5.6}$$

where $\boldsymbol{n}$ is the eigenvector of $\boldsymbol{R}$ corresponding to the eigenvalue 1. Thus, $\boldsymbol{r} = \theta\boldsymbol{n}$.

A candidate solution $\boldsymbol{\xi} \in \mathbb{R}^6$ (i.e. a rigid transformation $\boldsymbol{T} = \boldsymbol{T_\xi}$) is encoded by concatenating its rotation vector $r = \boldsymbol{r_\xi} \in \mathbb{R}^3$ and translation vector $t = \boldsymbol{t_\xi} \in \mathbb{R}^3$:

$$\boldsymbol{\xi} = [\boldsymbol{r}_{\boldsymbol{\xi}}^T, \boldsymbol{t}_{\boldsymbol{\xi}}^T]^T = [r_{\xi 1}, r_{\xi 2}, r_{\xi 3}, t_{\xi 1}, t_{\xi 2}, t_{\xi 3}]^T \tag{5.7}$$

The main advantage of the proposed encoding scheme is the possibility of modifying $\boldsymbol{\xi}$ without considering the constraints on the rotation matrix $\boldsymbol{R}$.

5.2.2 Assessment of the Candidate Solutions - The Cost Function

The cost associated to a candidate solution $\boldsymbol{\xi}$ is defined as the point-to-point mean square error ($L2$ norm error metric) between the rigid transformation $\boldsymbol{T_\xi}$ of the source point cloud $\mathcal{X} = \{\boldsymbol{x}_i\}$ and the target point cloud $\mathcal{Y} = \{\boldsymbol{y}_i\}$. It is calculated using Equation (5.2):

$$\mathcal{F}(\xi, \mathcal{X}, \mathcal{Y}) = \mathcal{E}(\boldsymbol{T_\xi}, \mathcal{X}, \mathcal{Y}) \tag{5.8}$$

As stated in Section 5.1, the goal of the registration process is to find the solution μ that minimises function $\mathcal{F}$.

5.3 Point Cloud Registration via ICP

This section describes the standard ICP algorithm for point cloud registration.

5.3.1 SVD Procedure for 3D Point Cloud Registration

SVD is used in ICP to obtain a closed-form solution to the least squares fitting problem for two 3D point sets under given point correspondences [41].

Given a source point cloud $\mathcal{X} = \{\boldsymbol{x}_i\}$ and a target point cloud $\mathcal{Y} = \{\boldsymbol{y}_i\}$ ($1 \leq i \leq N$), using the cost function formulated in Equation (5.8), the least squares fitting of the two point clouds under

given point correspondences is equal to:

$$\boldsymbol{T}^* = \arg\min_{\boldsymbol{T}} \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{y}_i - \boldsymbol{T}\boldsymbol{x}_i\|_2^2,\ \boldsymbol{T} \in SE(3) \tag{5.9}$$

The first step is to find the centroid of the source and target point clouds:

$$\bar{\boldsymbol{x}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i,\ \bar{\boldsymbol{y}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{y}_i \tag{5.10}$$

Once the two centroids have been calculated, the correlation matrix between the two point clouds is calculated as follows:

$$\boldsymbol{H} = \sum_{i=1}^{N} (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{y}_i - \bar{\boldsymbol{y}})^T \tag{5.11}$$

The correlation matrix $\boldsymbol{H}$ is decomposed into $\boldsymbol{H} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}$, where $\boldsymbol{\Sigma}$ is a diagonal matrix, and $\boldsymbol{U}$ and $\boldsymbol{V}$ are two unitary matrices. The SVD operation gives the closed-form solution for the rotation component of the transformation $\boldsymbol{R} = \boldsymbol{V}\boldsymbol{U}^T$. The translation component is computed as $\boldsymbol{t} = \bar{\boldsymbol{y}} - \boldsymbol{R}\bar{\boldsymbol{x}}$. The two components $\boldsymbol{R}$ and $\boldsymbol{t}$ form the rigid transformation $\boldsymbol{T}$ described in Equation (5.4).

If $det(\boldsymbol{R}) = -1$, the algorithm fails. Otherwise, the algorithm returns the least squares estimation of the rigid transformation for two input point sets.

5.3.2 The Iterative Closest Point (ICP) Algorithm

The ICP algorithm [39, 40] is a local optimisation method for point cloud registration. The procedure is based on the iterative estimation of the rigid transformation that best aligns two given point clouds $\mathcal{X}$ (the source) and $\mathcal{Y}$ (the target). The ICP is terminated when either a solution of residual error smaller than a pre-set threshold ϵ is found, or a pre-set number τ of cycles has elapsed.

The flowchart of the ICP algorithm is visualised in Figure 5.1.

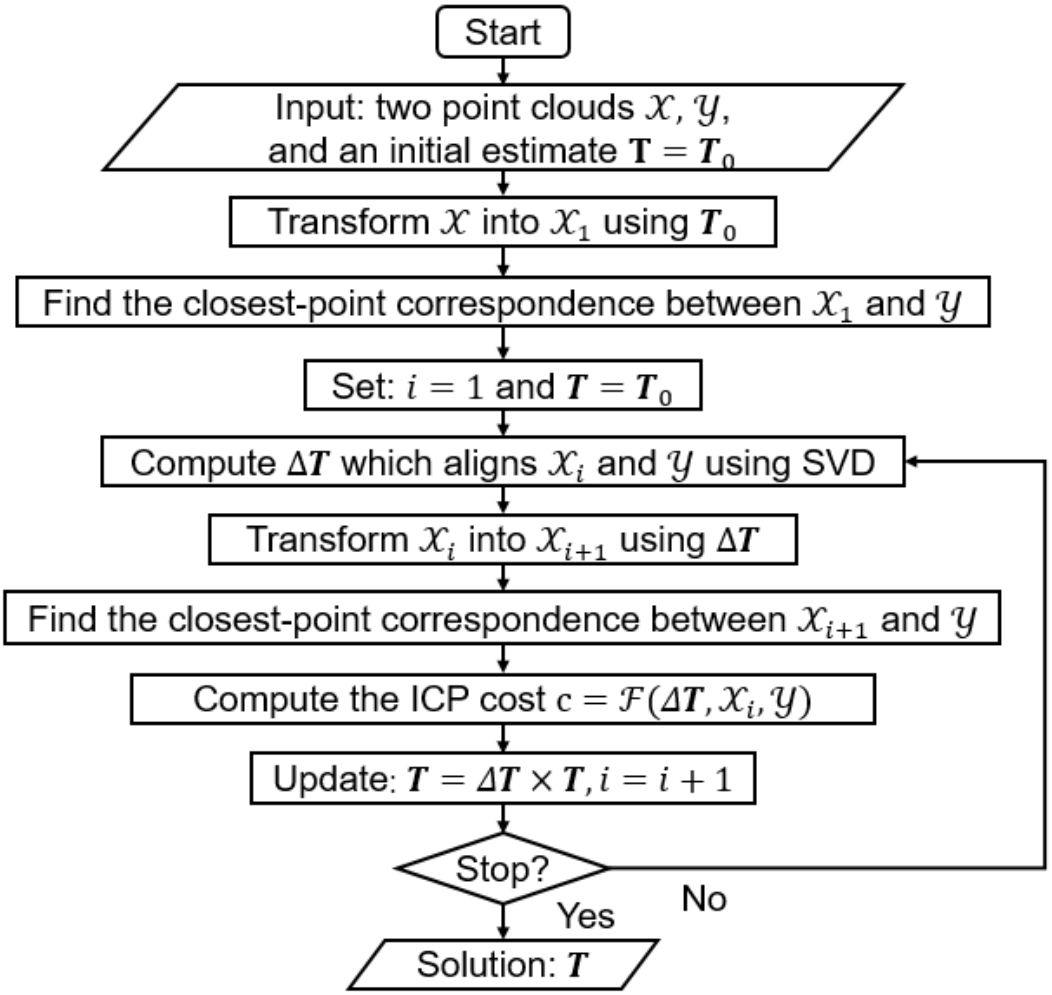


Figure 5.1: Flowchart of the ICP algorithm.

The procedure in Figure 5.1 is described by the following pseudocode:

1. Given two point clouds $\mathcal{X}$, $\mathcal{Y}$, and an initial estimate $\boldsymbol{T}_0$ of the rigid transformation that aligns them;
2. Compute $\mathcal{X}_1 = \boldsymbol{T}_0 \times \mathcal{X}$;
3. Find the closest-point correspondence between $\mathcal{X}_1$ and $\mathcal{Y}$;
4. Set $\boldsymbol{T} = \boldsymbol{T}_0$;
5. Set $i = 1$;
6. Set $stop = false$
7. do until $stop = false$
   (a) Compute the transformation $\Delta\boldsymbol{T}$ which aligns $\mathcal{X}_i$ and $\mathcal{Y}$ using SVD;
   (b) Compute $\mathcal{X}_{i+1} = \Delta\boldsymbol{T} \times \mathcal{X}_i$;
   (c) Find the closest-point correspondence between $\mathcal{X}_{i+1}$ and $\mathcal{Y}$ using Equation (5.3);
   (d) Compute the cost $c = \mathcal{F}(\Delta\boldsymbol{T}, \mathcal{X}_i, \mathcal{Y})$ using Equation (5.8);
   (e) Set $\boldsymbol{T} = \Delta\boldsymbol{T} \times \boldsymbol{T}$;
   (f) Set $i = i + 1$;
   (g) IF (($c > \epsilon$) OR ($i < \tau$)) THEN $stop = true$
   (h) go to step 7
8. Output the solution $\boldsymbol{T}$ and terminate.

5.4 Point Cloud Registration Using the Bees Algorithm

The BA is a popular metaheuristic for optimisation problems [164] that simulates the food foraging behaviour of honeybee colonies in nature. The BA was first proposed by Pham et al. [42], and its standard version was described by Pham and Castellani [43]. The behaviour of the BA was experimentally studied by Pham and Castellani [165, 166] and mathematically analysed by Baronti et al. [167].

5.4.1 Bee Foraging Mechanism in Nature

A biological bee colony uses a portion of the population as scouts to randomly explore the environment looking for food sources (flower patches). The scout bees assess the quality of the found food sources in terms of the availability, nutritional content, and ease of extraction of their pollen or nectar content. Once returned to the hive, scouts who found a profitable food source share this information with idle mates via the waggle dance. The waggle dance communicates the position and distance of the find, and stimulates onlookers to join the dancer in harvesting the advertised flower patch. High quality food sources elicit long and vigorous dances, which will be noticed by a large number of idle bees. Following this mechanism, the most profitable food sources are harvested by the largest share of the bee population.

5.4.2 The Standard Bees Algorithm

In this study, the BA uses the encoding method and cost function described in Section 5.2.1 and Section 5.2.2. Each artificial bee lands on a candidate solution $\boldsymbol{\xi}$ represented by the 6-dimensional vector described in Equation (5.7). The goodness of this solution is evaluated based on the residual error in the registration of the two point clouds. This error is calculated using the cost function described in Equation (5.8). The standard BA is summarised in the below flowchart.

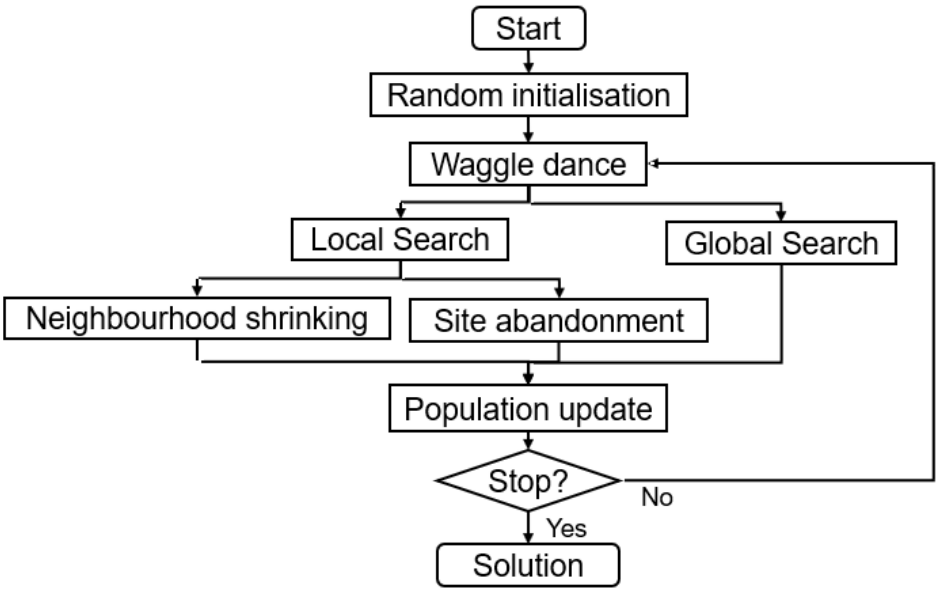


Figure 5.2: The standard Bees Algorithm.

The artificial bee colony is initialised with n scout bees (the whole colony) randomly scattered with uniform probability onto the solution space. The solutions visited by these scouts are evaluated and ranked by their residual errors (cost value). The BA then enters the main cycle, which begins with the *waggle dance* procedure. The waggle dance is implemented as follows: the scouts who found the top ranked solutions recruit foragers for local exploitative search, whilst the remaining scouts keep on performing random explorative search. The top nb solutions are defined as the *best sites* and undergo local exploitative search. Local search is carried out in a neighbourhood (customarily a hypercube) centred on the solution found by the scout. In analogy with the biological model, the local neighbourhood is often called a *flower patch*. Amongst the best sites, the best $ne \leq nb$ are called the *elite sites*. Each scout that visited an elite site recruits nre foragers for local search, whilst the scouts that visited the remaining $nb - ne$ best sites recruit $nrb \leq nre$ foragers. In this study, all sites are allocated the same sampling opportunities, that is $ne = 0$.

During the local search, if a forager lands on a solution of lower residual error than the solution found by the scout, that forager will replace the scout in the next recruitment cycle (waggle dance).

If more than one forager lands on a solution of lower error than the solution found by the scout, the forager that found best solution (lowest error) will replace the scout. If all foragers in a flower patch failed to find a better solution, the size of the neighbourhood is shrunk by a factor c (*neighbourhood shrinking* procedure) [43]. That is:

$$ngh_{k+1} = c \cdot ngh_k,\ 0 < c < 1 \tag{5.12}$$

where ngh_k is the size of the neighbourhood at BA cycle k. Also, a stagnation counter keeps track of the number of consecutive cycles where local search fails to yield better solutions in a flower patch. After a pre-defined number of consecutive cycles of stagnation ($stlim$), the local search is deemed to have found the local error minimum, and the flower patch is abandoned (*site abandonment* procedure) [43]. If the local optimum is the best-so-far, it will be kept in memory as the current solution. Site abandonment prevents the algorithm from remaining stuck in local minima of error.

The ns scout bees not taking part in the recruitment process are used for explorative search. Explorative search is implemented through random uniform sampling of the solution space. In summary, during the main cycle of the BA $ne \times nre + (nb - ne) \times nrb$ artificial bees (the foragers) are employed for local exploitative search, and the remaining ns artificial bees (the scouts) are employed for random explorative search. The total bee colony population is thus:

$$n = ne \times nre + (nb - ne) \times nrb + ns \tag{5.13}$$

The BA repeats cycles of local and random search until a given stopping criterion is met. In this study, the algorithm is terminated when either a solution of residual error smaller than a pre-set threshold ϵ is found, or a pre-set number T of cycles has elapsed. The first condition describes the level of point cloud registration accuracy that is deemed acceptable. The second condition limits the extent of the computational run time. At the end of the procedure, the best solution found during

the search is returned as the final solution to the optimisation problem.

5.4.3 The SVD-Enhanced Bees Algorithm for Point Cloud Registration

In this study, the SVD algorithm is used as a problem-specific operator to speed up the convergence of neighbourhood search to the local minima, increasing the efficiency of the BA optimisation procedure. It is also used to improve the accuracy of the solutions found via global search. In this latter case, the use of SVD is particularly helpful, since randomly generated solutions via global search are unlikely to be competitive against the results of consecutive cycles of local search.

In detail, one cycle of the SVD procedure is applied to all the solutions visited by the foragers. That is, SVD is applied to all the $ne \times nre + (nb - ne) \times nrb$ solutions visited in the local search step, and all ns solutions found via random global search. The flowchart of the SVD-enhanced BA is shown in Figure 5.3.

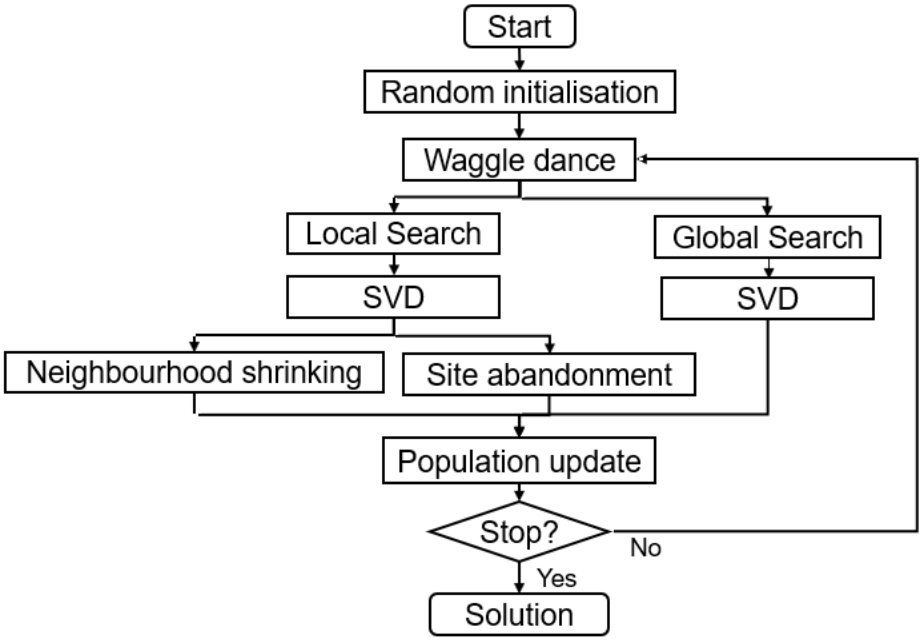


Figure 5.3: Bees Algorithm with SVD operation. At each cycle of the procedure, the solutions found via local and are improved by one cycle of SVD.

5.5 Control Algorithms

This section describes the EA and PSO procedures that will be used to benchmark the performance of the Bees Algorithm.

5.5.1 Evolutionary Algorithm for Point Cloud Registration

EAs are a class of metaheuristics inspired by the natural adaptation of species in nature. EA terminology heavily borrows from the biological model. Differently from the BA, EAs do not use artificial agents to probe the solution space, but directly work on the candidate solutions. These solutions are often referred to as *individuals*, and are typically represented as strings of variables called *chromosomes*. Each variable (*gene*) is defined within a discrete or continuous range of values (*alleles*), and represents a feature of the solution. More complex encodings are possible, according to the problem domain and EA paradigm used. For a detailed description of the evolutionary schemes and operators used in this study, the reader is referred to the specialised literature [136, 168]

New candidate solutions (*offspring*) are generated by mixing and modifying the features of existing solutions (*parents*), using operators modelled on biological mechanisms of genetic recombination and mutation. These operators act on the chromosomes of the parents. Namely, recombination (*crossover*) creates new individuals by randomly mixing the features (alleles) of two parents, whilst mutation randomly modifies one or more individual features. Mimicking biological competition for mating, EAs allocate higher chances of reproduction to the best performing solutions.

The EA used in this study initialises the population of candidate solutions by randomly sampling with uniform probability the search space. The population is encoded using the vector representation described in Section 5.2.1, and evaluated employing the cost function defined in Section 5.2.2. After the initialisation phase, the algorithm enters the main cycle. The population is evaluated and ranked in ascending order of cost (error). The individuals are allocated reproduc-

tion opportunities proportionally to their position in the ranking (*fitness ranking* selection scheme). According to this scheme, the best elements in the population can be mated to several other individuals.

The mated parents reproduce via one-point crossover (80% probability) or cloning. The chromosomes of the offspring may also undergo genetic mutation. Mutations are implemented by changing the allele of one randomly picked gene of an amount randomly sampled with uniform probability in a small interval $[-\delta, \delta]$. The width 2δ of the interval is defined for each variable i as follows:

$$\delta_i = \phi * w_i \tag{5.14}$$

where ϕ is a system parameter and $w_i = [min_i, max_i]$ is the width of the interval where variable i is defined.

At the end of every iteration (*generation*) of the main evolutionary cycle, the current population is completely replaced by the new individuals (offspring) generated during the reproduction process (*generational replacement* scheme), except for the best individual which is copied into the new population (*elitism*). Cycles of selection, reproduction, and population replacement are repeated until the stopping criterion is met. To ensure the comparability of the results, the EA uses the same stopping criterion employed by the BA (Section 5.4.2). The flowchart of the EA used in this study is shown in Figure 5.4.

Silva et al. [138] proposed a GA with added local search heuristics. According to this scheme, at the beginning of each evolutionary cycle the alignment of the best individual in the population is refined via a number of stochastic hill-climbing steps. Given a candidate solution x, hill-climbing creates a new solution x' by randomly perturbing the features (alleles) of x of a small offset, randomly sampled with uniform probability in the interval $[-\delta_{hc}, \delta_{hc}]$. If the cost of x' is smaller than the cost of x ($\mathcal{F}(x') < \mathcal{F}(x)$), x' replaces x as the current solution. A more computational efficient refinement method was devised by Zhu et al. [140], who used a few steps of the SVD procedure to improve the alignment of newly generated offspring. The aim of these hybrid algorithms is to

speed up the evolutionary search with some cycles of local exploitative search.

In this study, two hybrid EAs are tested, the first employing hill-climbing (EA with HC) and the second the SVD operator (EA with SVD). Both local search procedures are performed on the newly generated offspring as an additional operator after mutation (see Figure 5.4). For each offspring, ten cycles of hill-climbing are performed, whilst SVD is performed only once.

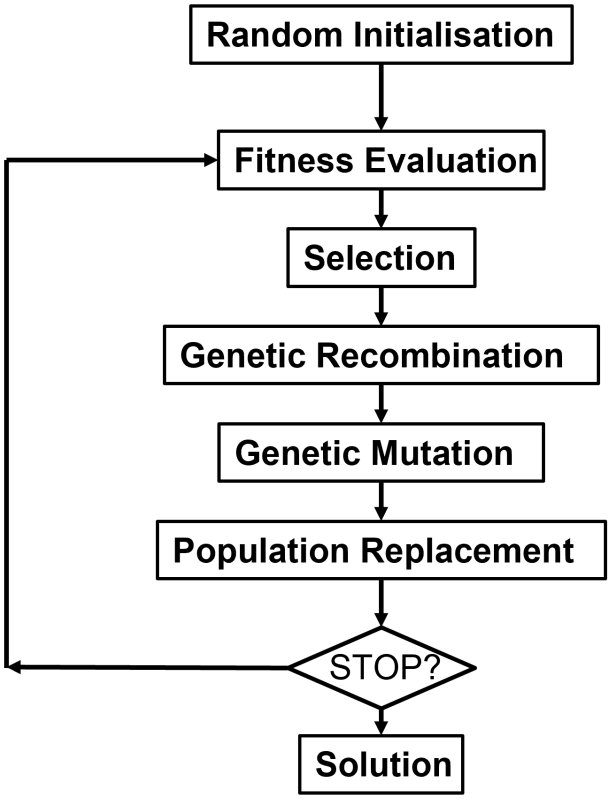


Figure 5.4: Standard EA.

5.5.2 Particle Swarm Optimisation for Point Cloud Registration

PSO is arguably the best known swarm intelligence algorithm for continuous optimisation problems. Inspired by the behaviour of bird flocks, PSO simulates a simplified social model [45] where each agent is thought of as a particle. Particles move in the solution space based on their past movements (*persistence* term), past experience (*individual* term), and their social neighbour's experience (*social* term). These three terms are represented in the equation determining the velocity

of a particle at iteration t of the algorithm main cycle:

$$v_i(t+1) = w(t) \cdot v_i(t) + c_p \cdot r_p \cdot (p_i - x_i(t)) + c_g \cdot r_g \cdot (g_i - x_i(t)) \tag{5.15}$$

where $v_i(t)$ is the component in the i^{th} variable of the particle velocity at time t, c_p and c_g are two system parameters weighting the contribution of respectively the individual and global term, and r_p and r_g are random numbers drawn with uniform probability in the interval $[0, 1]$. The position at cycle t of the particle in the n-dimensional search space is defined by the vector $\boldsymbol{x} = [x_1(t), \ldots, x_n(t)]$, whilst the best-so-far solution visited by the particle is indicated by the vector $\boldsymbol{p} = [p_1(t), \ldots, p_n(t)]$ (*personal best*). Analogously, the best-so-far solution visited by the social neighbours of the particle is indicated by the vector $\boldsymbol{g} = [g_1(t), \ldots, g_n(t)]$ (*neighbourhood best*). Finally, the parameter $w(t)$ is the *inertial weight* of the particle, and is linearly decreased with time according to the strategy devised by Wongkhuenkaew et al. [153].

In summary, the three terms on the right-hand side of Equation (5.15) describe how the velocity of the individuals is determined by the tendency to maintain the current direction (persistence term), tendency to return to the best-so-far found (individual term), and tendency to move towards the best solution found by the social neighbours (social term). In this study, the social neighbourhood of the particles correspond to the whole swarm. That is, the swarm is fully connected. The weight of the persistence term is decreased to facilitate the convergence of the swarm towards the global best towards the end of the search.

Once the velocity has been determined for each particle, the position is updated according to traditional Newton's mechanics ($\Delta t = 1$):

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{5.16}$$

PSO randomly initialises the position and velocity of the particles in the solution space. The algorithm enters then the main loop where the cost of the solutions where the particles currently lie is

evaluated. The personal and neighbourhood bests are then updated, and the new position and velocity of the particles is calculated. The speed of the particles is unbounded for the linear components of the vector in Equation (5.7), and limited to 2π per cycle for the rotational components.

PSO uses the same stopping criterion employed by the BA (Section 5.4.2) and EA. The flowchart of the standard algorithm is shown in Figure 5.5. In point cloud registration applications, also PSO was hybridised with local search algorithms such as SVD [153]. This approach was replicated in this study: one cycle of the SVD procedure was run on all particles after their position was updated.

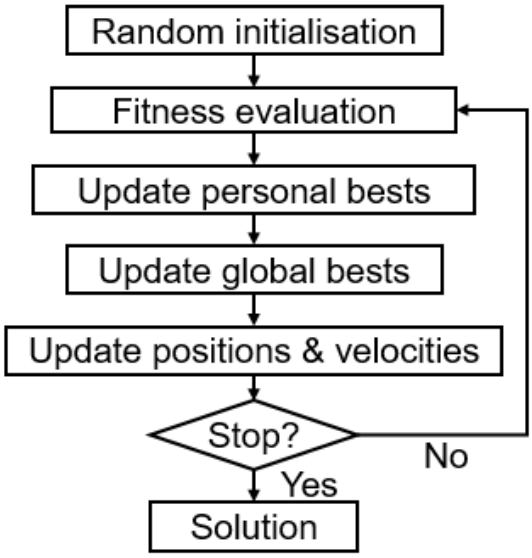


Figure 5.5: Standard PSO algorithm.

5.6 Experimental Set-Up

The performance of the proposed SVD-enhanced BA was compared to the performance of a number of state-of-the-art optimisation procedures and hybrids: the ICP algorithm [39, 40], the standard BA [43], an EA [136], an hill-climbing enhanced EA-HC as proposed by Silva et al. [138], an SVD enhanced EA-SVD as proposed by Zhu et al. [140], the standard PSO algorithm [45], and an SVD enhanced PSO-SVD [153].

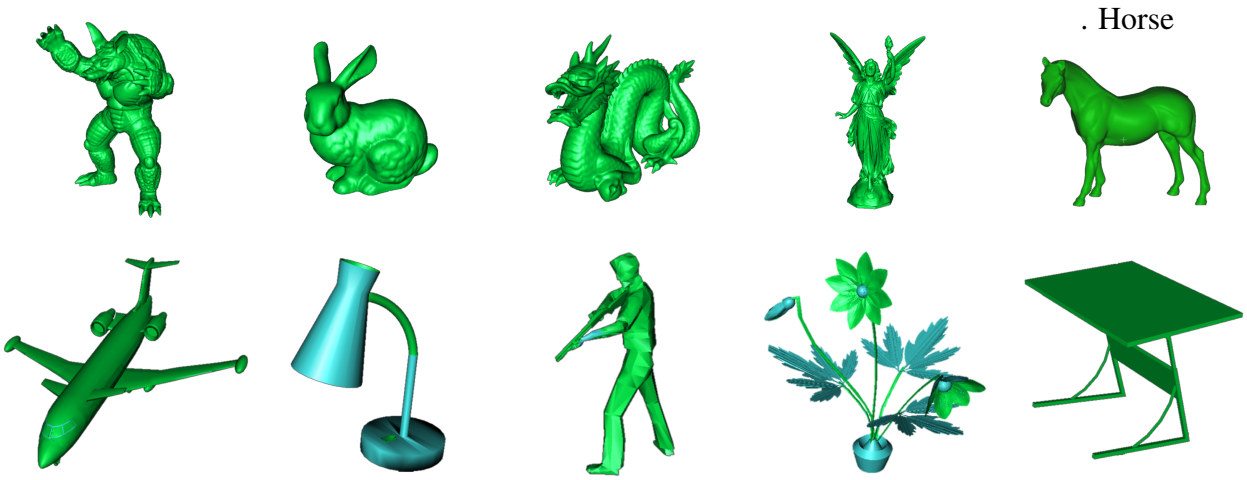


Figure 5.6: Shapes used in the experiments to test the performance of the registration algorithms. Shapes 1-4 were taken from the Stanford 3D Scanning Repository, Shape 5 from the Large Geometric Models Archive at Georgia Tech, Shapes 6-10 from the ModelNet repository.

5.6.1 Datasets

Ten shapes were used to evaluate the performance of the registration algorithms. They are shown in Figure 5.6. The Armadillo, Bunny, Dragon and Lucy statue shapes were taken from the Stanford 3D scanning repository [169]. The Horse shape was taken from the Large Geometric Models Archive at Georgia Tech, and the rest of the shapes were taken from the popular ModelNet repository [170].

A target model of 10^4 points was sampled from the surface of each shape. This target point cloud was used to generate 100 source models of 10^4 data points via random rigid transformations. All point clouds were bounded in a cube of size $[-1000, 1000]^3$ units, and their centre was placed at the origin of the Cartesian reference system.

The goal for the registration algorithms was to find the rigid transformation aligning the source to the target point cloud. In total, the set of point clouds employed in the experiments consisted of 10 shapes $\times$ 100 models $=$ 1000 elements, each element being composed of 10^4 points. In addition to this set of noise-free point clouds (*clean* data set), five more data sets were generated to test the robustness of the algorithms to noisy models (*noisy* sets). These latter sets were created to take into account that real data scans are affected by some level of sensor imprecision. The noisy sets

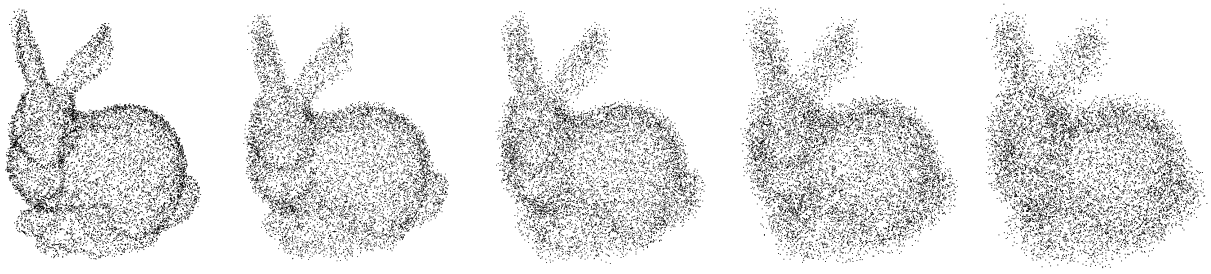


Figure 5.7: The Bunny point cloud corrupted with various levels noise, from 3% to 15% (left to right) in steps of 3%.

included corrupt duplicates of all the point clouds of the *clean* set, that is 1000 elements.

The noisy models were generated by randomly perturbing the position of the points in the cloud by a small amount δ_n. This amount was sampled for each variable (gene) i with uniform probability in a preset interval:

$$\delta_{ni} \sim \rho \cdot [min_i, max_i] \tag{5.17}$$

where ρ is a system parameter (noise level), and $w_i = [min_i, max_i]$ is the interval where variable i is defined. Five noisy sets were generated, one for each of the following values of the noise level ρ: 0.03 (3% noise), 0.06, 0.09, 0.12, and 0.15. An example of point cloud corrupted with various levels of noise is shown in Figure 5.7.

5.6.2 Parameter Settings

The hyperparameters defining the behaviour of the metaheuristics were optimised via extensive trial and error, and kept fixed across all the sets of experiments. They are listed for each algorithm in Table 5.1.

To ensure that all the metaheuristics were given the same sampling opportunities, the BA, EA, and PSO used the same number n of agents (henceforth referred to as the *population size*), and their duration was limited by the same termination criterion. The hybrid algorithms had the advantage

of the additional extra step of local exploitative search, which was performed on all solutions. This extra step was deterministic in the case of the SVD-based procedures, or relied on extra sampling of the solution space for the EA-HC hybrid.

Table 5.1: Hyperparameter setting of the 3D point cloud registration algorithms.

Common parameters					
maximum number of iterations T			100		
convergence threshold (ϵ) - clean set			0.01		
convergence threshold (ϵ) - noisy sets			100		
population size n			10, 20, 30, 40		
SVD cycles in hybrid algorithms			1		
BA hyperparameters					
stagnation limit ($stlim$)			10		
neighbourhood shrinking rate (c)			0.8		
n	ne	nb	nre	nrb	ns
2	-	1	-	1	1
4	-	2	-	1	2
6	-	2	-	2	2
8	-	2	-	2	4
10	-	2	-	2	6
20	-	4	-	3	8
30	-	5	-	4	10
40	-	6	-	5	10
EA and hillclimbing hyperparameters					
mutation rate			0.4		
mutation width ϕ			0.1		
crossover rate			0.8		
selection strategy			fitness ranking		
replacement scheme			generational replacement with elitism		
hillclimbing cycles			10		
hillcimbing scope δ_{hc}			0.5		
PSO hyperparameters					
inertial weight strategy			linear decrease		
initial inertial weight (w_{max})			0.7298		
personal learning rate (c_p)			1.0		
global learning rate (c_g)			1.0		
swarm connectivity			fully connected		
max component of particle velocity in variable i			$2\pi/cycle$		

5.7 Experimental Results

In this study, the consistency, precision, and robustness to noise of the SVD-enhanced BA was investigated, and the results compared to those obtained using ICP, EA, EA-HC, EA-SVD, PSO, PSO-SVD and the standard BA. This section presents the results of the experiments.

5.7.1 Consistency

Consistency relates to the success rate of the point cloud registration procedure. That is, the registration process may return an erroneous transformation with high residual error if the search converges to a local minimum. In this case, the erroneous transformation is marked as a failure. For the *clean* dataset, a registration attempt was considered successful if the cost $\mathcal{F}$ defined in Equation (5.8) was smaller than the empirically set threshold $\mathcal{F}_t < 0.01$.

In summary, the consistency of a registration algorithm depends on its search capability and susceptibility to get trapped into local optima, and is defined as the percentage of successful registration trials:

$$consistency = \frac{number\ of\ successful\ runs}{number\ of\ total\ runs} \times 100 \tag{5.18}$$

where the number of total runs is equal to 1000, that is the size of the *clean* data set.

In this study, the consistency of the algorithms was evaluated versus the population size. Given a fixed number of main cycle iterations, the population size defines how intensively the algorithm samples the solution space. It also defines the computational effort and hence the algorithm execution time. In the experiments, the population size for the EA, EA-HC, EA-SVD, PSO, PSO-SVD, BA and BA-SVD was increased from 10 to 40 individuals in steps of 10. To provide additional information on the capabilities of the SVD-enhanced BA, the consistency of the algorithm was also tested at smaller population sizes ranging from 2 to 10 in steps of 2. Table 5.2 details the results obtained by the tested metaheuristics and ICP. The results are also visualised in Figure 5.8.

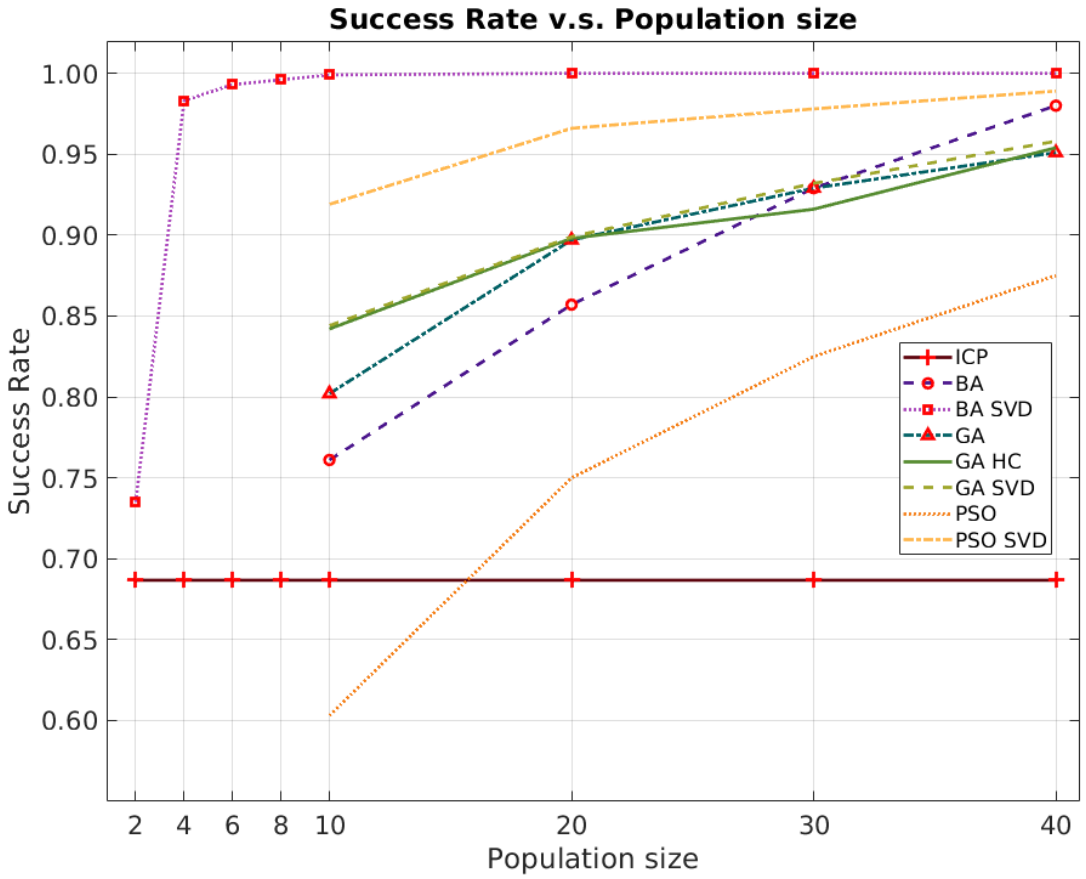


Figure 5.8: Success rate of the algorithms versus the population size.

Table 5.2: Success rates of the algorithms at different population sizes.

ICP	70.7%							
	Population size							
	2	4	6	8	10	20	30	40
EA	-	-	-	-	80.2%	89.7%	92.9%	95.1%
EA HC	-	-	-	-	84.2%	89.8%	91.6%	95.4%
EA SVD	-	-	-	-	84.4%	89.9%	93.2%	95.8%
PSO	-	-	-	-	60.3%	75.0%	82.5%	87.5%
PSO SVD	-	-	-	-	91.9%	96.6%	97.8%	98.9%
BA	-	-	-	-	76.1%	85.7%	92.9%	98.0%
BA with SVD	73.5%	98.3%	99.3%	**99.6%**	**99.9%**	**100%**	**100%**	**100%**

Overall, the results of the tests confirm that ICP is prone to suboptimal convergence. In terms of consistency, ICP was outperformed by all the metaheuristics, particularly when the latter were configured to use the highest values of population size. Population size was thus pivotal in determining the exploration capability of the algorithms, and consequently their ability to find the global optimum.

The standard EA performed comparatively well, particularly at low population sizes. Hybridisation with hill-climbing and SVD did not appear to significantly improve the performance of the EA. The standard PSO performed poorly compared to the standard EA and BA. Hybridisation with SVD dramatically improved its performance, lifting the success rate of PSO to nearly 100% for population sizes of 20 or more particles.

The performance of the standard BA was comparable to the performance of the standard EA, and superior to the standard PSO algorithm. Hybridisation had a major effect on the performance of the BA. The SVD-enhanced BA achieved a success rate very close to 100% using as few as 8 artificial bees, and a 100% success rate using 20 or more artificial bees. For the remaining of the experiments, 20 individuals will be set as the common value for the population size for all metaheuristics. Employing as few as 6 individuals, the SVD-enhanced BA achieved a higher success rate (99.3%) than any other algorithm regardless of its population size.

The significance of the differences in performance (Table 5.2) between the proposed SVD-enhanced BA and the other algorithms was statistically analysed. Given the categorical nature of the variable (success/failure), pairwise chi-square (χ^2) tests were used for the analysis. The results (p-values) of the χ^2 tests are reported in Tables 5.3 to 5.5.

Table 5.3 reports the significance of the differences in performance between the standard BA and the SVD-enhanced BA. The table shows that the hybrid algorithm significantly outperformed (i.e. the null hypothesis is rejected) the standard version for any value of the colony size.

Table 5.3: BA-SVD vs. standard BA consistency: results of the pairwise χ^2 tests. The null hypothesis (no difference) is rejected for p-values smaller than 0.05.

		BA			
		10	20	30	40
BA-SVD	10	0	0	0	0
	20	0	0	0	0
	30	0	0	0	0
	40	0	0	0	0

Table 5.4: Consistency of BA vs. ICP, PSO and EA: results of the pairwise χ^2 tests. The null hypothesis (no difference) is rejected for p-values smaller than 0.05.

		ICP	PSO				EA			
			10	20	30	40	10	20	30	40
BA	10	0.0073	0	0.6029	0.0005	0	0.0304	0	0	0
	20	0	0	0	0.0580	0.2645	0.0013	0.0079	0	0
	30	0	0	0	0	0	0	0.0139	1	0.0480
	40	0	0	0	0	0	0	0	0	0.0006

Table 5.5: Consistency of BA-SVD vs. EA and PSO hybrid algorithms: results of the pairwise χ^2 tests. The null hypothesis (no difference) is rejected for p-values smaller than 0.05.

		EA-HC				EA-SVD				PSO-SVD			
		10	20	30	40	10	20	30	40	10	20	30	40
BA-SVD	2	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0.0003	0	0	0	0.0015	0	0.2323	0.5177	0.3413
	6	0	0	0	0	0	0	0	0	0	0	0.0088	0.4775
	8	0	0	0	0	0	0	0	0	0	0	0.0008	0.1199
	10	0	0	0	0	0	0	0	0	0	0	0	0.0091
	20	0	0	0	0	0	0	0	0	0	0	0	0.0025
	30	0	0	0	0	0	0	0	0	0	0	0	0.0025
	40	0	0	0	0	0	0	0	0	0	0	0	0.0025

Table 5.4 reports the differences in performance between the standard BA versus ICP, the standard EA, and PSO. At a 5% level of significance, the performance of the ICP algorithm is significantly inferior to the performance of the BA. The BA also generally outperformed PSO, unless the latter used a larger population size. No PSO configuration was competitive with the BA when the latter employed at least 30 artificial bees. The results were more mixed in the comparison between the standard BA and the EA, with the former excelling at high and the latter at low population sizes. In general, the p-values reported in Table 5.4 indicate that the differences in the results shown in Table 5.2 are in most cases significant.

Finally, Table 5.5 reports the significance of the differences in performance between the SVD-enhanced BA and the other hybrid algorithms. For colony sizes of 10 or more, the BA-SVD performance is significantly superior to the performance of the other hybrids, regardless of the population size the latter used.

5.7.2 Precision

The precision of the algorithms was evaluated as the average residual error of the final solutions, where the residual error is the cost value $\mathcal{F}$ defined in Equation (5.8). A low residual error indicates that the two point clouds are well aligned, whilst a high residual error indicates a failure of the algorithm to align the two point clouds.

The 8 algorithms were run on the *clean* dataset, using a common population size of 20 individuals (see Section 5.7.1). For each run, the cost of the final solution was recorded and the overall results statistically analysed. The descriptive statistics of the experiments are plotted on a logarithmic scale in Figure 5.9, and tabulated in Table 5.6.

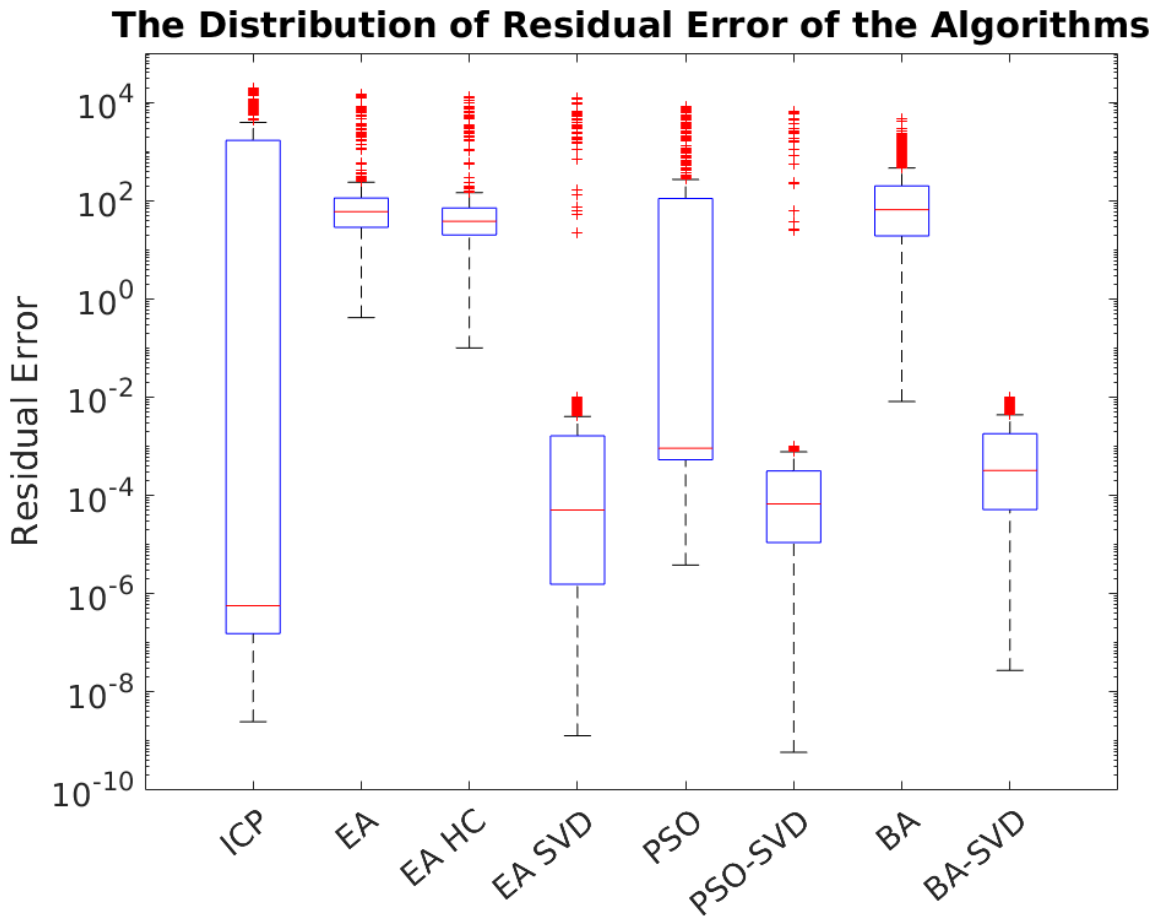


Figure 5.9: The distribution of the residual error of the solutions obtained by the registration algorithms for the *clean* set. The residual error (Y-axis) is plotted on a logarithmic scale.

Table 5.6: Spread of the residual errors obtained by the registration algorithms. From left to right: minimum, median, maximum value, and inter-quartile range (IQR).

	min.	median	max.	IQR
ICP	2.49×10^{-9}	5.70×10^{-7}	1.93×10^{4}	1.68×10^{3}
EA	0.42	59.33	1.43×10^{4}	83.84
EA-HC	0.10	37.82	1.34×10^{4}	50.53
EA-SVD	1.29×10^{-9}	5.06×10^{-5}	1.23×10^{4}	1.6×10^{-3}
PSO	3.83×10^{-6}	9.14×10^{-4}	8.28×10^{3}	109.86
PSO-SVD	5.94×10^{-10}	6.72×10^{-5}	6.55×10^{3}	3.04×10^{-4}
BA	8.20×10^{-3}	65.17	4.75×10^{3}	179.59
BA with SVD	2.76×10^{-8}	3.23×10^{-4}	9.96×10^{-3}	1.8×10^{-3}

As Figure 5.9 shows, the performance of the ICP algorithm is characterised by a large spread (a long tail) in the distribution of the results. This spread was due to the numerous runs where the algorithm converged to a local minimum. When it found the global optimum, thanks to its greedy search strategy ICP was able to align very accurately the two point clouds. This ability is reflected in the very small (10^{-7}) magnitude of the average (median) error.

The standard metaheuristics obtained far more consistent results than ICP. This ability was evidenced by the small inter-quartile range (IQR) of the average cost of the final solutions, which is one order of magnitude smaller than the range obtained by ICP. However, due to the stochastic nature of the search, on average the standard metaheuristics were not able to align the point clouds with the same precision as ICP. In general, the EA and BA were less precise (greater average cost) but more consistent (smaller spread) than PSO. It should also be remarked that below the $\epsilon = 10^{-2}$ convergence threshold, differences in precision are not visually appreciable.

Also, although the hybrid EA-HC was more precise than the standard EA, it was still not able to compete with ICP in terms of precision of alignment. This shortcoming was due to the stochastic nature of hill-climbing.

The SVD-enhanced metaheuristics confirmed the validity of the hybrid approach, combining the consistency of the population-based global search (small IQR) with the accuracy of the least squares method (low average cost). The plot in Figure 5.9 shows that all the final solutions obtained by the BA-SVD have a residual error inferior to 10^{-2}, that is the algorithm obtains a 100% success rate.

The statistical significance of the differences in the precision results was statistically analysed. In this case, due to the numerical nature of the variables, Mann-Whitney significance tests were used. The results (p-values) of the Mann-Whitney tests are reported in Table 5.7. Considering a 5% level of significance, the performance of the BA-SVD algorithm is clearly superior to the performance of all the other algorithms.

Table 5.7: Precision of the registration algorithms: results of pairwise Mann-Whitney significance tests. The null hypothesis (no difference) is rejected for p-values smaller than 0.05

	EA	EA-HC	EA-SVD	PSO	PSO-SVD	BA	BA-SVD
ICP	0	0	0	0	0	0	0
EA	-	0	0	0	0	0.5461	0
EA-HC	-	-	0	0	0	0	0
EA-SVD	-	-	-	0	0.6127	0	0
PSO	-	-	-	-	0	0	0
PSO-SVD	-	-	-	-	-	0	0
BA	-	-	-	-	-	-	0

5.7.3 Robustness to Noise

The registration algorithms were tested on the five *noisy* data sets defined in Section 5.6.1. As discussed in Section 5.7.1, the population size of the metaheuristics was fixed to 20 individuals. The success rates of the algorithms on the noisy data sets are detailed in Table 5.8 and visualised in Figure 5.10.

The introduction of noise had a major effect on the success rate of ICP, which was degraded to less than 50%. Conversely, all the metaheuristics showed a remarkable resilience to noise, obtaining performances substantially similar to those obtained on the noise-free *clean* set. However, it should be remembered that in this last set of tests, in order to take into account the impossibility of perfectly aligning a noisy point cloud, the maximum error threshold to consider an alignment trial successful was raised from 0.01 to 100. The success rates obtained on the *clean* and *noisy* data sets should thus be compared with care.

As the noise level was increased, the success rate of ICP slightly decreased, whilst the performance of the standard PSO and BA slightly improved. This perhaps counter-intuitive latter result might be explained with the fact that noise could have smoothed the error landscape. The success

rate of the other algorithms, and in particular the hybrid routines, showed little or no variation as the noise level was increased.

The significance of the differences between the results obtained on the noisy data sets by the BA-SVD, and those obtained by the other algorithms, was evaluated through χ^2 tests. The results (p-values) of the significance tests are reported in Table 5.9. They are all equal to zero (smaller than 10^{-4}), giving s strong indication of the significance of the superior performance of the BA-SVD documented in Table 5.8.

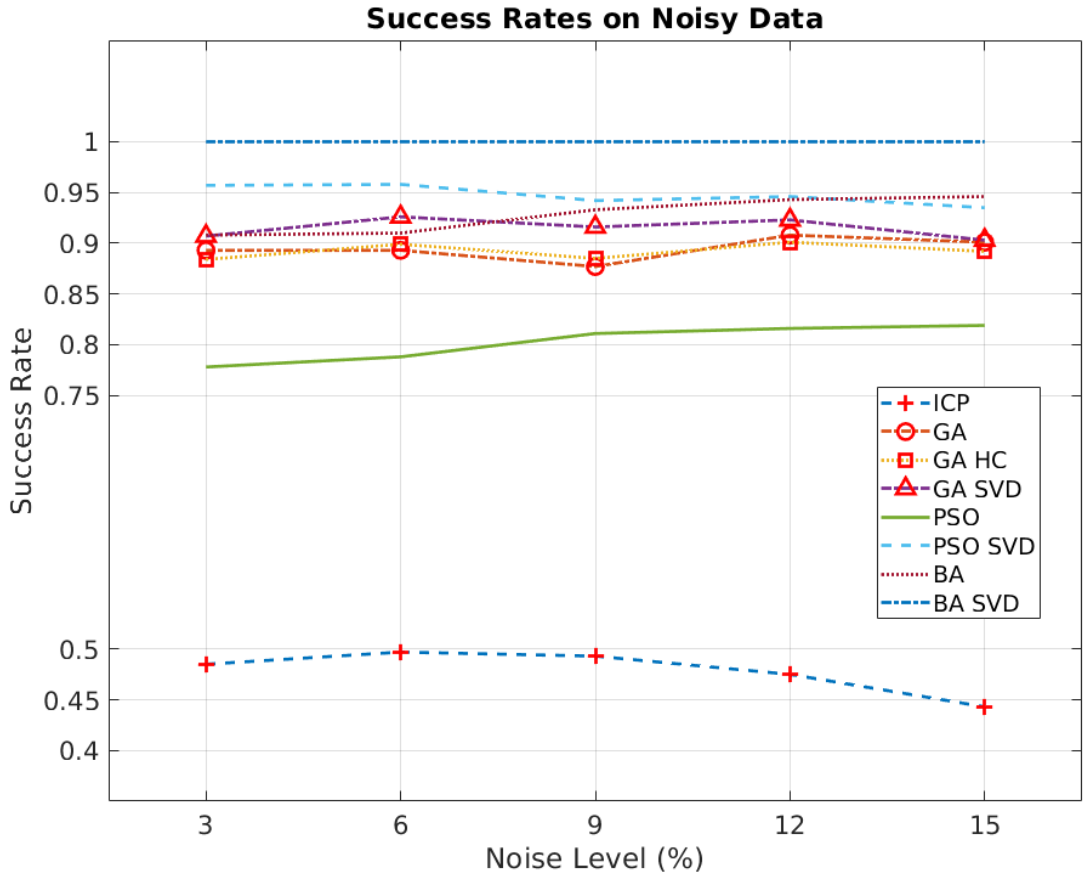


Figure 5.10: Success rate of the registration algorithms at incremental noise levels.

Table 5.8: Success rate of the registration algorithms on model sets of increasing level of noise.

	Noise Level				
	3	6	9	12	15
ICP	48.5%	49.7%	49.3%	47.5%	44.3%
EA	89.3%	89.3%	87.7%	90.8%	90.1%
EA-HC	88.4%	89.9%	88.5%	90.1%	89.2%
EA-SVD	90.7%	92.6%	91.6%	92.3%	90.3%
PSO	77.8%	78.8%	81.1%	81.6%	81.9%
PSO-SVD	95.7%	95.8%	94.2%	94.6%	93.5%
BA	90.8%	91.0%	93.3%	94.3%	94.6%
BA-SVD	100%	100%	100%	100%	100%

Table 5.9: Results (p-values) of pairwise χ^2 tests to evaluate the significance of the differences between the results obtained by the BA-SVD, and those obtained by the other algorithms. The null hypothesis (no difference) is rejected for p-values smaller than 0.05

		ICP	EA	EA-HC	EA-SVD	PSO	PSO-SVD	BA
BA-SVD	3	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0
	12	0	0	0	0	0	0	0
	15	0	0	0	0	0	0	0

5.8 Discussion of Results

The ICP algorithm aligns two point clouds by minimising their closest point correspondence error. ICP is a local search technique based on least squares minimisation of the alignment error via SVD. As shown in the tests, ICP is capable to align two point clouds with great accuracy. However, due to the local search strategy, ICP is prone to converge to local minima of the alignment error. Noise can also lead astray the algorithm, and the results of Section 5.7.3 showed a marked drop in ICP success rate even for modest levels of noise. This limitation is of particular concern, considering the imprecision of many real-life scanners.

Standard metaheuristics like EAs and PSO are able to minimise the likelihood of sub-optimal convergence thanks to their global search approach. However, as demonstrated in Section 5.7.2, due to the stochastic nature of the local search, they are less efficient at descending the error landscape.

Hybridising standard metaheuristics with a stochastic local search algorithm like hill-climbing improves only partially the efficiency of the search. The local search SVD procedure is based on analytical minimisation of the closest point correspondence mean-square error metric. This kind of deterministic descent of the error surface is very efficient, and combined with the global search capability of metaheuristics can provide very consistent and precise point cloud registration results.

Of the three metaheuristics tested, the proposed SVD-enhanced BA obtained the most consistent alignment results. This results is probably due to the parallel local search of the BA, which distributes the exploitation effort between nrb flower patches. Namely, the BA search is based on the interaction of nrb fairly independent sub-swarms. For the success of the registration process, it is sufficient that one of these sub-swarms finds the basin of attraction of the global optimum. Once in this region, SVD will efficiently drive the local search to the global optimum. The experiments reported in Section 5.7.1 showed this method being successful even with minimal bee colony sizes.

By helping the local search to quickly locate the minimum of the local basin of attraction, SVD also enhances the BA robustness to sub-optimal convergence. That is, once a sub-optimal basin has

been found, site abandonment frees the sub-population and restarts the local search elsewhere on the fitness landscape. By minimising the time taken by local search to find a local minimum, SVD thus increases the exploration capability of the BA.

EAs and PSO search happens at the whole population level. When the EA or PSO population has converged to a sub-optimal basin of attraction, the algorithm is trapped and the registration procedure fails. Moreover, the standard EA and PSO algorithms do not have an equivalent of the BA site abandonment procedure, which acts as a further policy against sub-optimal convergence. Further work should investigate the use of multi-swarm [171] or niching techniques [172] to increase the consistency of EA- and PSO-based point cloud registration methods.

Finally, it should be remarked that the results presented in this section are relevant only within the strict domain of point cloud registration. They can not be used to support any claim about a general superiority of the BA over the EA and PSO metaheuristics. Indeed, such claim would be in contradiction with the *No Free Lunch Theorem* [173].

5.9 Conclusions

This chapter proposed an SVD-enhanced Bees Algorithm for the solution of the 3D registration problem. The algorithm combines the robust global search approach of the Bees Algorithm metaheuristics with the fast local search of SVD.

The proposed algorithm was benchmarked against the standard ICP registration algorithm, various standard metaheuristics (EA, PSO, BA), and metaheuristics similarly enhanced with local optimisers such as hill-climbing (EA-HC) and SVD (EA-SVD, PSO-SVD). Compared on a range of point cloud registration problems, the proposed approach excelled in terms of consistency and precision. Experimental evidence also demonstrated that the SVD-enhanced BA is highly resilient to noisy data. This latter feature makes the SVD-enhanced BA an ideal candidate for industrial applications, where sensor noise is an issue.

Chapter 6

Conclusions

This thesis addressed three key robotics problems related to object perception and localisation using dense RGB-D SLAM. The three problems are: a) camera drift for camera egomotion estimation, b) reflective and shiny object reconstruction, and c) optimal point cloud registration for object localisation. The focus of the thesis is on applications in industrial scenarios, where metallic and plastic shiny objects abound, and particularly remanufacturing applications where sensing is crucial.

In the first part of this thesis, a dense RGB-D SLAM system was implemented, and an online weighted map fusion strategy with F2M camera tracking was proposed for reducing camera drift during camera egomotion estimation. The environmental structure is rebuilt in the form of a 3D model, merging the data gathered from the image sequence. The 3D model is initialised using the data from the first image, and successively updated and completed adding new data from the following images. For each image, the partial model is used as reference for camera pose estimation.

Camera egomotion estimation was tackled as a nonlinear optimisation problem, tracking for each image the camera pose with respect to the globally fused 3D model. Each tracked image is merged into the global map model simultaneously using an online weighted fusion strategy with standard deviation estimation. The proposed approach was demonstrated to minimise camera drift, since the global merging of information in the 3D model helps reducing the accumulation of error

in sequential image alignment. The results showed the proposed method achieved high consistency in camera trajectory estimation, and improved the accuracy of the relative camera pose estimation.

In the second part of this thesis, the practical challenge of implementing a dense RGB-D SLAM systems for reflective and shiny objects was approached. Due to the variation in photometric information on smooth surfaces, the standard F2M camera tracking of Chapter 3 fails on reflective and shiny mechanical parts. This thesis proposed RSO-SLAM which combines local photometric alignment (F2F) with global geometric alignment (F2M) for consistent joint estimation of camera trajectory and environmental structure mapping. Experimental tests were carried out on a purpose-built set of plastic and metallic objects, where the performance of the RSO-SLAM method was compared with that of state-of-the-art F2F methods, and the standard F2M camera tracking method proposed in Chapter 3. The results of the quantitative evaluation demonstrated the effectiveness and accuracy of the RSO-SLAM method. A real-life case study involving the cover of an electric vehicle battery demonstrated the accuracy of the RSO-SLAM method, and verified its applicability to industrial scenarios.

In the third part of this thesis, object localisation was formulated as a point cloud registration optimisation problem, and solved using the Bees Algorithm. To enhance the local search capability of the proposed procedure, the BA was hybridised with the standard SVD procedure. Thanks to the enhancement of local search via SVD, the BA can rapidly identify the minima of the multimodal solution space. Experimental tests on 10 benchmark shapes showed that the standard and the SVD enhanced BA outperformed the current state-of-the-art, including ICP and two popular standard and SVD enhanced metaheuristics: EA and PSO. The proposed SVD enhanced BA obtained top results even using micro bee colony sizes, reaching nearly 100% success rates with as few as 6 agents. Experimental tests on noisy versions of the 10 benchmark shapes indicated that the proposed SVD-enhanced BA is robust to data corruption, and is thus an ideal tool to handle the noisy sensory data that are typical of industrial applications.

6.1 Summary of Achievements

The research work presented in this thesis yielded the following scientific contributions:

- Improved understanding of camera drift in dense RGB-D SLAM applications, and a novel solution to minimise the accumulated error in sequential camera pose estimation by using frame-to-model camera tracking with online weighted map fusion. The global consistency and local accuracy of the proposed solution was evaluated.

- A new solution (i.e., *RSO-SLAM*) to the problem of global photometric inconsistency in the reconstruction of reflective and shiny objects. The RSO-SLAM method is based on the joint optimisation of the local photometric alignment and global geometric alignment in dense RGB-D SLAM systems.

- Assessment of the feasibility of adopting the RSO-SLAM system in industrial scenarios via a case study involving an electric car battery (reflective surfaces).

- Viewing object localisation as a point cloud registration problem, the Bees Algorithm was proposed as a novel solution for the optimisation of point cloud alignment.

- The Bees Algorithm was hybridised with the SVD operator for faster and accurate local search in point cloud registration.

6.2 Future Work

This thesis investigated the system of dense reconstruction and object localisation for robotics and industrial applications. Three critical challenges were addressed in this study: camera drift, the reconstruction of reflective and shiny objects, and global optimisation for point cloud registration. The results of this work enable the adoption of dense RGB-D SLAM systems in industrial scenarios, and provide the foundation for future works in robot perception, localisation, and decision-making.

Chapter 3 proposed a solution to reduce the drift and improve the global consistency of camera egomotion estimation. The inertial measurement unit (e.g., accelerometer and gyro) provides the kinematic measurements for the inertial state estimation and navigation. The kinematic parameters of the joints for industrial robots provide another channel of sensory measurement for camera egomotion. The work of this thesis could be extended by adding inertial and kinematic data to the RGB-D SLAM information, providing reliable state estimation even when the camera enters a poor- structured and textured environment.

Chapter 4 addressed the problem of reconstructing reflective and shiny objects. This work is a key enabler of dense RGB-D SLAM for mechanical components with plastic or metallic surfaces. Future work on this topic could extend this study to the dense reconstruction of specular reflective and transparent objects.

Chapter 5 proposed a novel BA with SVD enhancement for point cloud registration. Future work could investigate the process of registering partial point clouds if the reconstruction system gives incomplete structure for the objects.

6.3 Publications Arisen from This Thesis

Part of the work undertaken in this thesis has already been presented in the following conferences:

- Lan F., Castellani M. and Wang Y., 2019. Bees Algorithm with SVD Optimisation for 3D Registration. III International Workshop on Autonomous Remanufacturing (IWAR).

Part of this work also resulted in the following book chapter:

- Lan F., Castellani M., Wang Y. and Zheng S. 2022. Global Optimisation for Point Cloud Registration with the Bees Algorithm. In Pham D.T. and Hartono N. (eds) Intelligent Manufacturing and Production Optimisation - The Bees Algorithm Approach. Springer Series in Advanced Manufacturing.

Part of this work also submitted to the following journals:

- Lan F., Castellani M., Zheng S., and Wang Y. The SVD-Enhanced Bees Algorithm, a Novel Procedure for Point Cloud Registration. Submitted to International Journal of Computer Vision.

Bibliography

[1] M. R. Johnson, I. P. McCarthy, Product recovery decisions within the context of extended producer responsibility, Journal of Engineering and Technology Management 34 (2014) 9–28.

[2] E. Sundin, The role of remanufacturing in a circular economy, Remanufacturing in the Circular Economy: Operations, Engineering and Logistics (2019) 31–60.

[3] H. Vasudevan, V. Kalamkar, R. Terkar, Remanufacturing for sustainable development: Key-challenges, elements, and benefits, International Journal of Innovation, Management and Technology 3 (2012) 84.

[4] Y. Wang, F. Lan, J. Liu, J. Huang, S. Su, C. Ji, D. T. Pham, W. Xu, Q. Liu, Z. Zhou, Interlocking problems in disassembly sequence planning, International Journal of Production Research 59 (2021) 4723–4735.

[5] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, A. 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.

[6] A. Kim, R. M. Eustice, Perception-driven navigation: Active visual slam for robotic area coverage, in: 2013 IEEE International Conference on Robotics and Automation, IEEE, 2013, pp. 3196–3203.

[7] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, J. J. Leonard, Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age, IEEE Transactions on robotics 32 (2016) 1309–1332.

[8] J.-C. Piao, S.-D. Kim, Real-time visual–inertial slam based on adaptive keyframe selection for mobile ar applications, IEEE Transactions on Multimedia 21 (2019) 2827–2836.

[9] L. Jinyu, Y. Bangbang, C. Danpeng, W. Nan, Z. Guofeng, B. Hujun, Survey and evaluation of monocular visual-inertial slam algorithms for augmented reality, Virtual Reality & Intelligent Hardware 1 (2019) 386–410.

[10] F. Moosmann, C. Stiller, Velodyne slam, in: 2011 ieee intelligent vehicles symposium (iv), IEEE, 2011, pp. 393–398.

[11] R. Mur-Artal, J. D. Tardós, Visual-inertial monocular slam with map reuse, IEEE Robotics and Automation Letters 2 (2017) 796–803.

[12] G. Bresson, Z. Alsayed, L. Yu, S. Glaser, Simultaneous localization and mapping: A survey of current trends in autonomous driving, IEEE Transactions on Intelligent Vehicles 2 (2017) 194–220.

[13] A. J. Davison, I. D. Reid, N. D. Molton, O. Stasse, Monoslam: Real-time single camera slam, IEEE transactions on pattern analysis and machine intelligence 29 (2007) 1052–1067.

[14] R. Mur-Artal, J. M. M. Montiel, J. D. Tardos, Orb-slam: a versatile and accurate monocular slam system, IEEE transactions on robotics 31 (2015) 1147–1163.

[15] R. Mur-Artal, J. D. Tardós, Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras, IEEE transactions on robotics 33 (2017) 1255–1262.

[16] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, Orb: An efficient alternative to sift or surf, in: 2011 International conference on computer vision, Ieee, 2011, pp. 2564–2571.

[17] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, in: European conference on computer vision, Springer, 2006, pp. 430–443.

[18] M. Calonder, V. Lepetit, C. Strecha, P. Fua, Brief: Binary robust independent elementary features, in: European conference on computer vision, Springer, 2010, pp. 778–792.

[19] M. Okutomi, T. Kanade, A multiple-baseline stereo, IEEE Transactions on pattern analysis and machine intelligence 15 (1993) 353–363.

[20] Y. Zhou, F. Yan, Z. Zhou, Handling pure camera rotation in semi-dense monocular slam, The Visual Computer 35 (2019) 123–132.

[21] H. Strasdat, J. Montiel, A. J. Davison, Scale drift-aware large scale monocular slam, Robotics: Science and Systems VI 2 (2010) 7.

[22] E. Sucar, J.-B. Hayet, Bayesian scale estimation for monocular slam based on generic object detection for correcting scale drift, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 5152–5158.

[23] M. Tomono, Robust 3d slam with a stereo camera based on an edge-point icp algorithm, in: 2009 IEEE international conference on robotics and automation, IEEE, 2009, pp. 4306–4311.

[24] J. Engel, J. Stückler, D. Cremers, Large-scale direct slam with stereo cameras, in: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE, 2015, pp. 1935–1942.

[25] R. Gomez-Ojeda, F.-A. Moreno, D. Zuniga-Noël, D. Scaramuzza, J. Gonzalez-Jimenez, Pl-slam: A stereo slam system through the combination of points and line segments, IEEE Transactions on Robotics 35 (2019) 734–746.

[26] A. Grunnet-Jepsen, J. N. Sweetser, Intel® realsense™ depth cameras for mobile phones, New Technologies Group, Intel Corporation: Santa Clara, CA, USA (2019).

[27] Z. Zhang, Microsoft kinect sensor and its effect, IEEE multimedia 19 (2012) 4–10.

[28] B. Curless, M. Levoy, A volumetric method for building complex models from range im-

ages, in: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, 1996, pp. 303–312.

[29] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, J. McDonald, Real-time large-scale dense rgb-d slam with volumetric fusion, The International Journal of Robotics Research 34 (2015) 598–626.

[30] D. Sun, F. Geißer, B. Nebel, Towards effective localization in dynamic environments, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2016, pp. 4517–4523.

[31] T. Whelan, M. Goesele, S. J. Lovegrove, J. Straub, S. Green, R. Szeliski, S. Butterfield, S. Verma, R. A. Newcombe, M. Goesele, et al., Reconstructing scenes with mirror and glass surfaces., ACM Trans. Graph. 37 (2018) 102–1.

[32] F. Dellaert, M. Kaess, et al., Factor graphs for robot perception, Foundations and Trends® in Robotics 6 (2017) 1–139.

[33] J. Nocedal, S. J. Wright (Eds.), Nonlinear Least-Squares Problems, Springer New York, New York, NY, 1999, pp. 250–275. doi:10.1007/0-387-22742-3_10.

[34] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, A. Fitzgibbon, Scene coordinate regression forests for camera relocalization in rgb-d images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2930–2937.

[35] J. Sturm, N. Engelhard, F. Endres, W. Burgard, D. Cremers, A benchmark for the evaluation of rgb-d slam systems, in: 2012 IEEE/RSJ international conference on intelligent robots and systems, IEEE, 2012, pp. 573–580.

[36] N. Sharmin, R. Brad, Optimal filter estimation for lucas-kanade optical flow, Sensors 12 (2012) 12694–12709.

[37] E. Meinhardt-Llopis, J. Sánchez, Horn-schunck optical flow with a multi-scale strategy, Image Processing on line (2013).

[38] H. P. Gavin, The levenberg-marquardt algorithm for nonlinear least squares curve-fitting problems, Department of Civil and Environmental Engineering, Duke University 19 (2019).

[39] P. J. Besl, N. D. McKay, Method for registration of 3-d shapes, in: Sensor Fusion IV: Control Paradigms and Data Structures, volume 1611, International Society for Optics and Photonics, 1992, pp. 586–607.

[40] Y. Chen, G. Medioni, Object modelling by registration of multiple range images, Image and vision computing 10 (1992) 145–155.

[41] K. S. Arun, T. S. Huang, S. D. Blostein, Least-squares fitting of two 3-d point sets, IEEE Transactions on pattern analysis and machine intelligence (1987) 698–700.

[42] D. T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri, S. Rahim, M. Zaidi, The bees algorithm—a novel tool for complex optimisation problems, in: Intelligent Production Machines and Systems, Elsevier, 2006, pp. 454–459.

[43] D. T. Pham, M. Castellani, The bees algorithm: modelling foraging behaviour to solve

continuous optimization problems, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 223 (2009) 2919–2938.

[44] K. Brunnstrom, A. J. Stoddart, Genetic algorithms for free-form surface matching, in: Proceedings of 13th International Conference on Pattern Recognition, volume 4, IEEE, 1996, pp. 689–693.

[45] R. Eberhart, J. Kennedy, Particle swarm optimization, in: Proceedings of the IEEE international conference on neural networks, volume 4, Citeseer, 1995, pp. 1942–1948.

[46] C. Premebida, R. Ambrus, Z.-C. Marton, Intelligent robotic perception systems, in: Applications of Mobile Robots, IntechOpen London, UK, 2018.

[47] U. Frese, H. Hirschmüller, Special issue on robot vision: what is robot vision?, Journal of Real-Time Image Processing 10 (2015) 597–598.

[48] S. Barone, A. Paoli, A. V. Razionale, Three-dimensional point cloud alignment detecting fiducial markers by structured light stereo imaging, Machine Vision and Applications 23 (2012) 217–229.

[49] E. Dong, J. Xu, C. Wu, Y. Liu, Z. Yang, Pair-navi: Peer-to-peer indoor navigation with mobile visual slam, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, IEEE, 2019, pp. 1189–1197.

[50] D. Luebke, Cuda: Scalable parallel programming for high-performance scientific computing, in: 2008 5th IEEE international symposium on biomedical imaging: from nano to macro, IEEE, 2008, pp. 836–838.

[51] A. Grunnet-Jepsen, J. N. Sweetser, P. Winer, A. Takagi, J. Woodfill, Projectors for intel® realsense™ depth cameras d4xx, Intel Support, Interl Corporation: Santa Clara, CA, USA (2018).

[52] L. Keselman, J. Iselin Woodfill, A. Grunnet-Jepsen, A. Bhowmik, Intel realsense stereoscopic depth cameras, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 1–10.

[53] F. Rong, D. Xie, W. Zhu, H. Shang, L. Song, A survey of multi view stereo, in: 2021 International Conference on Networking Systems of AI (INSAI), IEEE, 2021, pp. 129–135.

[54] R. Horaud, M. Hansard, G. Evangelidis, C. Ménier, An overview of depth cameras and range scanners based on time-of-flight technologies, Machine vision and applications 27 (2016) 1005–1020.

[55] S. Barone, P. Neri, A. Paoli, A. Razionale, 3d acquisition and stereo-camera calibration by active devices: A unique structured light encoding framework, Optics and Lasers in Engineering 127 (2020) 105989.

[56] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, A. Kolb, Real-time 3d reconstruction in dynamic scenes using point-based fusion, in: 2013 International Conference on 3D Vision-3DV 2013, IEEE, 2013, pp. 1–8.

[57] C. Kerl, J. Sturm, D. Cremers, Dense visual slam for rgb-d cameras, in: 2013 IEEE/RSJ

International Conference on Intelligent Robots and Systems, IEEE, 2013, pp. 2100–2106.

[58] T. Schops, T. Sattler, M. 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.

[59] K.-L. Low, Linear least-squares optimization for point-to-plane icp surface registration, Chapel Hill, University of North Carolina 4 (2004) 1–3.

[60] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, S. Leutenegger, Elasticfusion: Real-time dense slam and light source estimation, The International Journal of Robotics Research 35 (2016) 1697–1716.

[61] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, C. Theobalt, Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration, ACM Transactions on Graphics (ToG) 36 (2017) 1.

[62] B. Valinasab, Atomization-based spray coating for improved 3D scanning, Ph.D. thesis, University of Victoria, 2014.

[63] D. Hruboš, T. Kouteckỳ, D. Paloušek, An experimental study for determination of an application method and tio2 powder to ensure the thinnest matte coating layer for 3d optical scanning, Measurement 136 (2019) 42–49.

[64] Z. Song, R. Chung, X.-T. Zhang, An accurate and robust strip-edge-based structured light means for shiny surface micromeasurement in 3-d, IEEE Transactions on Industrial Electronics 60 (2012) 1023–1032.

[65] Z. He, P. Li, X. Zhao, L. Kang, S. Zhang, J. Tan, Chessboard-like high-frequency patterns for 3d measurement of reflective surface, IEEE Transactions on Instrumentation and Measurement 70 (2021) 1–12.

[66] X. Liu, W. Chen, H. Madhusudanan, J. Ge, C. Ru, Y. Sun, Optical measurement of highly reflective surfaces from a single exposure, IEEE Transactions on Industrial Informatics 17 (2020) 1882–1891.

[67] J. Park, A. Kak, 3d modeling of optically challenging objects, IEEE Transactions on Visualization and Computer Graphics 14 (2008) 246–262.

[68] K. He, C. Sui, C. Lyu, Z. Wang, Y. Liu, 3d reconstruction of objects with occlusion and surface reflection using a dual monocular structured light system, Applied Optics 59 (2020) 9259–9271.

[69] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, J. D. Tardós, Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam, IEEE Transactions on Robotics 37 (2021) 1874–1890.

[70] R. Koch, S. May, P. Koch, M. Kühn, A. Nüchter, Detection of specular reflections in range measurements for faultless robotic slam, in: Robot 2015: Second Iberian Robotics Conference, Springer, 2016, pp. 133–145.

[71] X. Wang, J. Wang, Detecting glass in simultaneous localisation and mapping, Robotics and

Autonomous Systems 88 (2017) 97–103.

[72] G. Du, K. Wang, S. Lian, K. Zhao, Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review, Artificial Intelligence Review 54 (2021) 1677–1734.

[73] L. Tian, N. M. Thalmann, D. Thalmann, Z. Fang, J. Zheng, Object grasping of humanoid robot based on yolo, in: Computer Graphics International Conference, Springer, 2019, pp. 476–482.

[74] H.-I. Lin, Y.-Y. Chen, Y.-Y. Chen, Robot vision to recognize both object and rotation for robot pick-and-place operation, in: 2015 international conference on advanced robotics and intelligent systems (aris), IEEE, 2015, pp. 1–6.

[75] S. Son, H. Park, K. H. Lee, Automated laser scanning system for reverse engineering and inspection, International Journal of machine tools and manufacture 42 (2002) 889–897.

[76] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, et al., The digital michelangelo project: 3d scanning of large statues, in: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, 2000, pp. 131–144.

[77] Z. Zhang, Determining the epipolar geometry and its uncertainty: A review, International journal of computer vision 27 (1998) 161–195.

[78] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (1981) 381–395.

[79] D. G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the seventh IEEE international conference on computer vision, volume 2, Ieee, 1999, pp. 1150–1157.

[80] H. Bay, T. Tuytelaars, L. V. Gool, Surf: Speeded up robust features, in: European conference on computer vision, Springer, 2006, pp. 404–417.

[81] D. DeTone, T. Malisiewicz, A. Rabinovich, Superpoint: Self-supervised interest point detection and description, in: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2018, pp. 224–236.

[82] P.-E. Sarlin, D. DeTone, T. Malisiewicz, A. Rabinovich, Superglue: Learning feature matching with graph neural networks, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 4938–4947.

[83] G. Peyré, M. Cuturi, et al., Computational optimal transport: With applications to data science, Foundations and Trends® in Machine Learning 11 (2019) 355–607.

[84] J. Engel, J. Sturm, D. Cremers, Semi-dense visual odometry for a monocular camera, in: Proceedings of the IEEE international conference on computer vision, 2013, pp. 1449–1456.

[85] J. Engel, T. Schöps, D. Cremers, Lsd-slam: Large-scale direct monocular slam, in: European conference on computer vision, Springer, 2014, pp. 834–849.

[86] A. Glover, W. Maddern, M. Warren, S. Reid, M. Milford, G. Wyeth, Openfabmap: An open source toolbox for appearance-based loop closure detection, in: 2012 IEEE International Conference on Robotics and Automation, IEEE, 2012, pp. 4730–4735.

[87] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard, g 2 o: A general framework for graph optimization, in: 2011 IEEE International Conference on Robotics and Automation, IEEE, 2011, pp. 3607–3613.

[88] D. Caruso, J. Engel, D. Cremers, Large-scale direct slam for omnidirectional cameras, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015, pp. 141–148.

[89] X. Gao, R. Wang, N. Demmel, D. Cremers, Ldso: Direct sparse odometry with loop closure, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 2198–2204. doi:10.1109/IROS.2018.8593376.

[90] J. Lu, Z. Fang, Y. Gao, J. Chen, Line-based visual odometry using local gradient fitting, Journal of Visual Communication and Image Representation 77 (2021) 103071.

[91] S. Yang, S. Scherer, Direct monocular odometry using points and lines, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 3871–3877. doi:10.1109/ICRA.2017.7989446.

[92] Y.-S. Shin, Y. S. Park, A. Kim, Dvl-slam: sparse depth enhanced direct visual-lidar slam, Autonomous Robots 44 (2020) 115–130.

[93] R. A. Newcombe, S. J. Lovegrove, A. J. Davison, Dtam: Dense tracking and mapping in real-time, in: 2011 international conference on computer vision, IEEE, 2011, pp. 2320–2327.

[94] J. Stühmer, S. Gumhold, D. Cremers, Real-time dense geometry from a handheld camera, in: Joint Pattern Recognition Symposium, Springer, 2010, pp. 11–20.

[95] X. Wang, H. Zhang, X. Yin, M. Du, Q. Chen, Monocular visual odometry scale recovery using geometrical constraint, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 988–995. doi:10.1109/ICRA.2018.8462902.

[96] C. Su, G. Tan, Y. Luo, Research on stereo matching technology based on binocular vision, Open Access Library Journal 6 (2019) 1–10.

[97] S. Baker, I. Matthews, Lucas-kanade 20 years on: A unifying framework, International journal of computer vision 56 (2004) 221–255.

[98] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, J. McDonald, Kintinuous: Spatially extended kinectfusion (2012).

[99] A. Concha, J. Civera, Rgbdtam: A cost-effective and accurate rgb-d tracking and mapping system, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 6756–6763. doi:10.1109/IROS.2017.8206593.

[100] B. Triggs, P. F. McLauchlan, R. I. Hartley, A. W. Fitzgibbon, Bundle adjustment—a modern synthesis, in: International workshop on vision algorithms, Springer, 1999, pp. 298–372.

[101] J. Cai, L. Luo, S. Hu, Bi-direction direct rgb-d visual odometry, Applied Artificial Intelligence 34 (2020) 1137–1158.

[102] Y. Vasilyev, T. Zickler, S. Gortler, O. Ben-Shahar, Shape from specular flow: Is one flow enough?, in: CVPR 2011, IEEE, 2011, pp. 2561–2568.

[103] S. Roth, M. J. Black, Specular flow and the recovery of surface structure, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 2, IEEE, 2006, pp. 1869–1876.

[104] Y. Adato, Y. Vasilyev, O. Ben-Shahar, T. Zickler, Toward a theory of shape from specular flow, in: 2007 IEEE 11th International Conference on Computer Vision, IEEE, 2007, pp. 1–8.

[105] C. Godard, P. Hedman, W. Li, G. J. Brostow, Multi-view reconstruction of highly specular surfaces in uncontrolled environments, in: 2015 International Conference on 3D Vision, IEEE, 2015, pp. 19–27.

[106] Y. Adato, O. Ben-Shahar, Specular flow and shape in one shot:., in: BMVC, Citeseer, 2011, pp. 1–11.

[107] T. Bonfort, P. Sturm, Voxel carving for specular surfaces, in: 9th IEEE International Conference on Computer Vision (ICCV'03), volume 1, IEEE Computer Society, 2003, pp. 691–696.

[108] S. Savarese, M. Chen, P. Perona, Local shape from mirror reflections, International Journal of Computer Vision 64 (2005) 31–67.

[109] J. Balzer, S. Hofer, J. Beyerer, Multiview specular stereo reconstruction of large mirror surfaces, in: CVPR 2011, IEEE, 2011, pp. 2537–2544.

[110] J. Balzer, D. Acevedo-Feliz, S. Soatto, S. Höfer, M. Hadwiger, J. Beyerer, Cavlectometry: Towards holistic reconstruction of large mirror objects, in: 2014 2nd International Conference on 3D Vision, volume 1, IEEE, 2014, pp. 448–455.

[111] M. Liu, R. Hartley, M. Salzmann, Mirror surface reconstruction from a single image, IEEE Transactions on Pattern Analysis and Machine Intelligence 4 (2015) 760–773.

[112] M. K. Johnson, E. H. Adelson, Shape estimation in natural illumination, in: CVPR 2011, IEEE, 2011, pp. 2553–2560.

[113] G. Oxholm, K. Nishino, Shape and reflectance from natural illumination, in: European Conference on Computer Vision, Springer, 2012, pp. 528–541.

[114] G. Oxholm, K. Nishino, Multiview shape and reflectance from natural illumination, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2155–2162.

[115] D. Nehab, T. Weyrich, S. Rusinkiewicz, Dense 3d reconstruction from specularity consistency, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.

[116] B. Tunwattanapong, G. Fyffe, P. Graham, J. Busch, X. Yu, A. Ghosh, P. Debevec, Acquiring

reflectance and shape from continuous spherical harmonic illumination, ACM Transactions on graphics (TOG) 32 (2013) 1–12.

[117] H. Schultz, Retrieving shape information from multiple images of a specular surface, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (1994) 195–201.

[118] Z. Zheng, M. Lizhuang, L. Zhong, Z. Chen, An extended photometric stereo algorithm for recovering specular object shape and its reflectance properties, Computer Science and Information Systems 7 (2010) 1–12.

[119] H.-S. Chung, J. Jia, Efficient photometric stereo on glossy surfaces with wide specular lobes, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.

[120] S. Rusinkiewicz, M. Levoy, Efficient variants of the ICP algorithm., in: 3dim, volume 1, 2001, pp. 145–152.

[121] D. Chetverikov, D. Svirko, D. Stepanov, P. Krsek, The trimmed iterative closest point algorithm, in: Object recognition supported by user interaction for service robots, volume 3, IEEE, 2002, pp. 545–548.

[122] J. M. Phillips, R. Liu, C. Tomasi, Outlier robust icp for minimizing fractional rmsd, in: Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007), IEEE, 2007, pp. 427–434.

[123] M. Bosse, G. Agamennoni, I. Gilitschenski, et al., Robust estimation and applications in robotics, Now Publishers, 2016.

[124] P. Bergström, O. Edlund, Robust registration of surfaces using a refined iterative closest point algorithm with a trust region approach, Numerical Algorithms 74 (2017) 755–779.

[125] S. Bouaziz, A. Tagliasacchi, M. Pauly, Sparse iterative closest point, in: Computer graphics forum, volume 32, Wiley Online Library, 2013, pp. 113–123.

[126] G. Agamennoni, S. Fontana, R. Y. Siegwart, D. G. Sorrenti, Point clouds registration with probabilistic data association, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2016, pp. 4092–4098.

[127] P. Babin, P. Giguere, F. Pomerleau, Analysis of robust functions for registration algorithms, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 1451–1457.

[128] M. Greenspan, M. Yurick, Approximate kd tree search for efficient icp, in: Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings., IEEE, 2003, pp. 442–448.

[129] A. Nuchter, K. Lingemann, J. Hertzberg, Cached kd tree search for icp algorithms, in: Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007), IEEE, 2007, pp. 419–426.

[130] A. L. Pavlov, G. W. V. Ovchinnikov, D. Y. Derbyshev, D. Tsetserukou, I. V. Oseledets, AA-ICP: Iterative closest point with Anderson acceleration, in: 2018 IEEE International

Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 1–6.

[131] D. M. Mount, N. S. Netanyahu, J. Le Moigne, Efficient algorithms for robust feature matching, Pattern recognition 32 (1999) 17–38.

[132] T. M. Breuel, Implementation techniques for geometric branch-and-bound matching methods, Computer Vision and Image Understanding 90 (2003) 258–294.

[133] H. Li, R. Hartley, The 3d-3d registration problem revisited, in: 2007 IEEE 11th international conference on computer vision, IEEE, 2007, pp. 1–8.

[134] C. Olsson, F. Kahl, M. Oskarsson, Branch-and-bound methods for euclidean registration problems, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2008) 783–794.

[135] J. Yang, H. Li, D. Campbell, Y. Jia, Go-icp: A globally optimal solution to 3d icp point-set registration, IEEE transactions on pattern analysis and machine intelligence 38 (2015) 2241–2254.

[136] D. B. Fogel, Evolutionary algorithms in theory and practice, John Wiley & Sons, Inc. New York, 1997.

[137] C. Blum, D. Merkle, Swarm intelligence: introduction and applications, Springer Science & Business Media, 2008.

[138] L. Silva, O. R. P. Bellon, K. L. Boyer, Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms, IEEE transactions on pattern analysis and machine intelligence 27 (2005) 762–776.

[139] C. Robertson, R. B. Fisher, Parallel evolutionary registration of range data, Computer Vision and Image Understanding 87 (2002) 39–50.

[140] J. Zhu, D. Meng, Z. Li, S. Du, Z. Yuan, Robust registration of partially overlapping point sets via genetic algorithm with growth operator, IET Image Processing 8 (2014) 582–590.

[141] L. Yan, J. Tan, H. Liu, H. Xie, C. Chen, Automatic registration of tls-tls and tls-mls point clouds using a genetic algorithm, Sensors 17 (2017) 1979.

[142] Y. Sahillioğlu, A genetic isometric shape correspondence algorithm with adaptive sampling, ACM Transactions on Graphics (TOG) 37 (2018) 175.

[143] M. Edelstein, D. Ezuz, M. Ben-Chen, Enigma: Evolutionary non-isometric geometry matching, arXiv preprint arXiv:1905.10763 (2019).

[144] X. Zhang, B. Yang, Y. Li, C. Zuo, X. Wang, W. Zhang, A method of partially overlapping point clouds registration based on differential evolution algorithm, PloS one 13 (2018) e0209227.

[145] C. Li, S. Dian, Dynamic differential evolution algorithm applied in point cloud registration, in: IOP Conference Series: Materials Science and Engineering, volume 428, IOP Publishing, 2018, p. 012032.

[146] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating

agents, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 26 (1996) 29–41.

[147] L. Hong-yan, Study on mutual information medical image registration based on ant algorithm, International Journal of Hybrid Information Technology 8 (2015) 353–360.

[148] S. Gupta, N. Grover, et al., A new optimization approach using smoothed images based on aco for medical image registration., International Journal of Information Engineering & Electronic Business 8 (2016).

[149] Y. Wu, W. Ma, Q. Miao, S. Wang, Multimodal continuous ant colony optimization for multisensor remote sensing image registration with local search, Swarm and Evolutionary Computation 47 (2019) 89–95.

[150] Q. Yu, K. Wang, A hybrid point cloud alignment method combining particle swarm optimization and iterative closest point method, Advances in Manufacturing 2 (2014) 32–38.

[151] Y. Ge, B. Wang, J. Nie, B. Sun, A point cloud registration method combining enhanced particle swarm optimization and iterative closest point method, in: 2016 Chinese Control and Decision Conference (CCDC), IEEE, 2016, pp. 2810–2815.

[152] X. Zhan, Y. Cai, P. He, A three-dimensional point cloud registration based on entropy and particle swarm optimization, Advances in Mechanical Engineering 10 (2018) 1687814018814330.

[153] R. Wongkhuenkaew, S. Auephanwiriyakul, M. Chaiworawitkul, N. Theera-Umpon, Three-dimensional tooth model reconstruction using statistical randomization-based particle swarm optimization, Applied Sciences 11 (2021) 2363.

[154] R. De Maesschalck, D. Jouan-Rimbaud, D. L. Massart, The mahalanobis distance, Chemometrics and intelligent laboratory systems 50 (2000) 1–18.

[155] D. Lefloch, T. Weyrich, A. Kolb, Anisotropic point-based fusion, in: 2015 18th International Conference on Information Fusion (Fusion), 2015, pp. 2121–2128.

[156] D. Gallup, J.-M. Frahm, P. Mordohai, M. Pollefeys, Variable baseline/resolution stereo, in: 2008 IEEE conference on computer vision and pattern recognition, IEEE, 2008, pp. 1–8.

[157] J. Sturm, W. Burgard, D. Cremers, Evaluating egomotion and structure-from-motion approaches using the tum rgb-d benchmark, in: Proc. of the Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RJS International Conference on Intelligent Robot Systems (IROS), volume 13, 2012.

[158] Z. Zhang, D. Scaramuzza, A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2018, pp. 7244–7251.

[159] B. K. Horn, Closed-form solution of absolute orientation using unit quaternions, Josa a 4 (1987) 629–642.

[160] K. S. Sim, M. E. Nia, C. P. Tso, D. T. K. Kho, Brain ventricle detection using hausdorff distance, Emerging Trends in Applications and Infrastructures for Computational Biology,

Bioinformatics, and Systems Biology (2016) 523–531.

[161] V. Yeghiazaryan, I. Voiculescu, An overview of current evaluation methods used in medical image segmentation, Department of Computer Science, University of Oxford (2015).

[162] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, M. Levoy, Geometrically stable sampling for the icp algorithm, in: Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings., IEEE, 2003, pp. 260–267.

[163] R. M. Murray, Z. Li, S. S. Sastry, A mathematical introduction to robotic manipulation, CRC press, 2017.

[164] D. T. Pham, L. Baronti, B. Zhang, M. Castellani, Optimisation of engineering systems with the bees algorithm, International Journal of Artificial Life Research (IJALR) 8 (2018) 1–15.

[165] D. T. Pham, M. Castellani, Benchmarking and comparison of nature-inspired population-based continuous optimisation algorithms, Soft Computing 18 (2014) 871–903.

[166] D. T. Pham, M. Castellani, A comparative study of the bees algorithm as a tool for function optimisation, Cogent Engineering 2 (2015) 1091540.

[167] L. Baronti, M. Castellani, D. T. Pham, An analysis of the search mechanisms of the bees algorithm, Swarm and Evolutionary Computation 59 (2020) 100746.

[168] D. Dasgupta, Z. Michalewicz, Evolutionary algorithms in engineering applications, Springer Science & Business Media, 2013.

[169] M. Levoy, J. Gerth, B. Curless, K. Pull, The stanford 3d scanning repository, URL http://www-graphics. stanford. edu/data/3dscanrep 5 (2005).

[170] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3d shapenets: A deep representation for volumetric shapes, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1912–1920.

[171] T. Blackwell, J. Branke, Multi-swarm optimization in dynamic environments, in: Workshops on Applications of Evolutionary Computation, Springer, 2004, pp. 489–500.

[172] X. Li, M. G. Epitropakis, K. Deb, A. Engelbrecht, Seeking multiple solutions: an updated survey on niching methods and their applications, IEEE Transactions on Evolutionary Computation 21 (2016) 518–538.

[173] D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, IEEE transactions on evolutionary computation 1 (1997) 67–82.

[174] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipes with Source Code CD-ROM 3rd Edition: The Art of Scientific Computing, Cambridge University Press, 2007.

Appendix A

On-manifold Optimisation with Gauss-Newton Method

This appendix elaborates upon the nonlinear optimisation method to solve Equation (3.24) for F2M camera pose estimation using an RGB-D camera. Equation (3.24) can be formulated as a least-square minimisation problem:

$$\boldsymbol{T}^* = \arg\min_{\boldsymbol{T}} \frac{1}{2} \left\| \boldsymbol{I} - h(\boldsymbol{T}, \mathcal{M}) \right\|_{\boldsymbol{\Sigma}}^2 = \arg\min_{\boldsymbol{T}} \frac{1}{2} \sum_{k=1}^{m} \left\| \boldsymbol{\epsilon}_i(z_k, \boldsymbol{T}, \boldsymbol{s}_k) \right\|_{\boldsymbol{\Sigma}}^2 \tag{A.1}$$

where $\boldsymbol{T}$ denotes the nth camera pose $\boldsymbol{T}_n$ (the subscript n is dropped), and $h(\cdot)$ is the observation function of an RGB-D camera which predicts the rendered image of $\mathcal{M}$ at $\boldsymbol{T}$. The right-hand term of Equation (A.1) is the sum of the pixel-wise errors $\boldsymbol{\epsilon}_i$, whilst $\boldsymbol{s}_k$ is the surfel point in $\mathcal{M}$, and z_k is the pixel measurement associated to $\boldsymbol{s}_k$.

The pixel-level error $\boldsymbol{\epsilon}$ relates to the measurements in the RGB-D image $\boldsymbol{I}$. An RGB-D image provides two measurement channels: colour and depth. The depth channel of a pixel directly gives the distance between the 3D point and the xOy plane of the camera coordinate system. The colour channel of a pixel gives either a grey-scale intensity value or the colour value (red, green, and blue). This study adopts the grey-scale intensity for photometric alignment. As a result, the pixel error $\boldsymbol{\epsilon}$

forms a 2D vector $\boldsymbol{\epsilon} = [\epsilon_g, \epsilon_p]^T$, where ϵ_g and ϵ_p are geometric and photometric errors, respectively. The two error components ϵ_g and ϵ_p will be dealt with in Section A.1 and Section A.2, respectively.

The objective function is formulated as a joint minimisation of geometric and photometric least squares:

$$\boldsymbol{T}^* = \arg\min_{\boldsymbol{T}} \frac{1}{2} \sum_{k=1}^{m} (\boldsymbol{\Sigma}_k^{-1/2} \boldsymbol{\epsilon}_k)^T (\boldsymbol{\Sigma}_k^{-1/2} \boldsymbol{\epsilon}_k) = \arg\min_{\boldsymbol{T}} \frac{1}{2} \sum_{k=1}^{m} \left(\sigma_{g,k}^{-2} \cdot \epsilon_{g,k}^2 + \sigma_{p,k}^{-2} \cdot \epsilon_{p,k}^2 \right) \tag{A.2}$$

where $\boldsymbol{\Sigma}$ is a 2×2 diagonal covariance matrix whose diagonal elements are σ_g^2 and σ_p^2. It was reported that the standard deviation modelling for photometric alignment is still difficult when not impossible to calculate [58]. In this study, the parameterisation for standard deviation was set empirically according to the error scale of the RGB-D camera: $\sigma_p / \sigma_g = 1000$.

A.1 Geometric Residual Error

For the calculation of the geometric residual error ϵ_g in Equation (A.2), the point-to-plane error of ICP algorithm with projective data association [59] is adopted. The geometric residual error amounts to the difference between each surfel point $\boldsymbol{s}$ in the global model $\mathcal{M}$ and the associated pixel in the RGB-D image. It is defined as follows:

$$\epsilon_g(\boldsymbol{T}) = \boldsymbol{n}_f^T \cdot \left[\pi^{-1}(\pi(\boldsymbol{T}^{-1}\boldsymbol{p}), d) - \boldsymbol{T}^{-1}\boldsymbol{p} \right] \tag{A.3}$$

where $\boldsymbol{p}$ is the 3D position coordinates of surfel $\boldsymbol{s}$, $\boldsymbol{n}_f$ is the normal vector at the pixel point, $\pi(\cdot)$ is the camera projection function, and $\pi^{-1}(\cdot)$ is the inverse projection.

The geometric error is calculated as below. Each surfel $\boldsymbol{s} \in \mathcal{M}$ is transformed into the camera coordinate system $\boldsymbol{\tau}$ by $\boldsymbol{T}^{-1}\boldsymbol{p}$. The transformed point $\boldsymbol{T}^{-1}\boldsymbol{p}$ maps into a pixel $\boldsymbol{u} = \pi(\boldsymbol{T}^{-1}\boldsymbol{p})$ via camera projection. The pixel is unprojected into a 3D point $\boldsymbol{q} = \pi^{-1}(u, d)$ with the depth value d obtained via inverse camera projection in Equation (3.13). Finally, the geometric error is obtained

by computing the point-to-plane error between $\boldsymbol{q}$ and $\boldsymbol{T}^{-1}\boldsymbol{p}$.

To avoid erroneous data association, the residual error term is rejected if

i) The z component of the surfel normal in the camera coordinate system $\boldsymbol{\tau}$ is positive, i.e. the surfel is not oriented towards the camera (invisible), .

ii) The z component of the surfel position in the camera coodinate system $\boldsymbol{\tau}$ is negative, i.e. the surfel is beyond the field of view of the camera.

iii) The distance between $\boldsymbol{T}^{-1}\boldsymbol{p}$ in the camera coodinate system and $\boldsymbol{q}$ is greater than a given threshold δ, i.e. the position of the measured pixel point is too far from the surfel in $\mathcal{M}$,

iv) The angle between the surfel normal $\boldsymbol{T}^{-1}\boldsymbol{p}$ in the camera coordinate system and $\boldsymbol{n}_f$ is greater than a threshold θ, i.e. the angle difference between the surfel normal in $\mathcal{M}$ and the measured pixel normal in the image is too large.

A.2 Photometric Residual Error

The photometric residual error ϵ_p is the difference of grey-scale intensity value between the surfel point $\boldsymbol{s} \in \mathcal{M}$ and the associated image pixel by projective data association [59].

The photometric error is presented below:

$$\epsilon_p = \boldsymbol{I}_{\mathcal{M}}(\boldsymbol{p}) - \boldsymbol{I}_f(\pi(\boldsymbol{T}^{-1}\boldsymbol{p})) \tag{A.4}$$

where $\boldsymbol{I}_{\mathcal{M}}(\boldsymbol{p})$ extracts the intensity value of $\boldsymbol{p}$ in the global map $\mathcal{M}$, and $\boldsymbol{I}(\boldsymbol{u})$ extracts the intensity value at the pixel point $\boldsymbol{u}$ in the image $\boldsymbol{I}$.

Similarly to the projective data association of the geometric residual error in Section A.1, each surfel $\boldsymbol{s} \in \mathcal{M}$ associates to a pixel $\boldsymbol{u} = \pi(\boldsymbol{T}^{-1}\boldsymbol{p})$ in the image $\boldsymbol{I}_f$. The difference between the intensity value of $\boldsymbol{s} \in \mathcal{M}$ and the pixel $\boldsymbol{u} \in \boldsymbol{I}_f$ gives the photometric error.

A.3 On-Manifold Optimisation

In this section, the nonlinear least-square minimisation in Equation (A.2) is solved using the Gauss-Newton method [33].

The Gauss-Newton method is a local gradient-based optimisation technique. It linearises the nonlinear least-square function in Equation (A.2) with an initial estimate of the camera pose $\boldsymbol{T}_0$ according to Section 3.1, and iteratively computes the incremental transformation $\Delta\boldsymbol{T}_k$ by minimising the following objective function:

$$\Delta\boldsymbol{T}^* = \arg\min_{\Delta\boldsymbol{T}} \mathcal{F}(\Delta\boldsymbol{T}) = \arg\min_{\Delta\boldsymbol{T}} \frac{1}{2}\sum_{k=1}^{m} \|\boldsymbol{\Sigma}_k^{-1/2}\boldsymbol{J}_k\Delta\boldsymbol{T} + \boldsymbol{\Sigma}_k^{-1/2}\boldsymbol{\epsilon}_k(\boldsymbol{T})\|_2^2,\ k = 1, 2, ... \quad \text{(A.5)}$$

where $\mathcal{F}$ is the objective function, $\boldsymbol{\Sigma}_i$ is the covariance matrix, and $\boldsymbol{J}_i$ is the Jacobian.

The incremental transformation $\Delta\boldsymbol{T}$ is obtained by solving the stationary point condition of the objective function $\mathcal{F}$, i.e., the derivative of the function is set to zero:

$$\frac{\partial\mathcal{F}}{\partial\Delta\boldsymbol{T}} = 0 \quad \text{(A.6)}$$

$$\sum \boldsymbol{J}_i\boldsymbol{\Sigma}_i^{-1}\boldsymbol{J}_i\Delta\boldsymbol{T} + \sum \boldsymbol{J}_i\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\epsilon}_i(\boldsymbol{T}) = 0 \quad \text{(A.7)}$$

The linear system in Equation (A.7) can be efficiently solved by Cholesky decomposition [174], and the incremental transformation $\Delta\boldsymbol{T}$ will be used to update the camera pose $\boldsymbol{T}_{n-1}$. However, unlike common optimisation problems in which the solution space is in a multi-dimensional Euclidean space, the solution space for the function in Equation (A.5) is a Lie Group, known as the Special Euclidean group in dimension 3 ($SE(3)$) as in Equation (3.14). The transformation $\boldsymbol{T}$ in $SE(3)$ is overparameterised since the matrix has 16 entries but only 6 degrees of freedom. The solution space is a 6D manifold embedded in the 4×4 matrix space. Trivial matrix addition for solution updating $\boldsymbol{T}^* = \boldsymbol{T} + \Delta\boldsymbol{T}$ will draw the new solution $\boldsymbol{T}^*$ away from the $SE(3)$ manifold,

i.e., $\boldsymbol{T}^*$ won't satisfy the constraints in Equation (3.14):

$$\boldsymbol{T}^* = \boldsymbol{T} + \Delta\boldsymbol{T},\ \boldsymbol{T}^* \notin SE(3) \tag{A.8}$$

On-manifold optimisation avoids the updating scheme of Equation (A.8) in matrix space. It computes the incremental transformation in the form of the tangent space $\mathfrak{se}(3)$ (also known as Lie Algebra) of $SE(3)$ instead.

Each rigid transformation $\boldsymbol{T}$ is mapped into its tangent space $\boldsymbol{\xi} \in \mathfrak{se}(3)$. The camera pose $\boldsymbol{\xi} \in \mathfrak{se}(3)$ is a 6D vector such as $\boldsymbol{\xi} = [\boldsymbol{\rho}, \boldsymbol{\omega}]^T = [\rho_x, \rho_y, \rho_z, \omega_x, \omega_y, \omega_z]^T$, where $\boldsymbol{\rho}$ and $\boldsymbol{\omega}$ are respectively translation and rotation vectors. The rotation component $\boldsymbol{\omega}$ is the axis-angle representation. The unit direction vector $\boldsymbol{n} = \boldsymbol{\omega}/\|\boldsymbol{\omega}\|$ represents the rotation axis, and the modulus $\theta = \|\boldsymbol{\omega}\|$ is the rotation angle. The corresponding global coordinate $\boldsymbol{T} \in SE(3)$ is computed by the exponential mapping:

$$\boldsymbol{T} = \exp(\boldsymbol{\xi}^\wedge) = \sum_{n=0}^{\infty} \frac{1}{n!}(\boldsymbol{\xi}^\wedge)^n = \begin{bmatrix} \exp(\boldsymbol{\omega}^\wedge) & \mathcal{J}\boldsymbol{\rho} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \tag{A.9}$$

where:

$$\boldsymbol{\xi}^\wedge = \begin{bmatrix} \boldsymbol{\omega}^\wedge & \boldsymbol{\rho} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix},\ \boldsymbol{\omega}^\wedge = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{A.10}$$

$$\mathcal{J} = \frac{\sin\theta}{\theta}\boldsymbol{I} + \left(1 - \frac{\sin\theta}{\theta}\right)\boldsymbol{n}\boldsymbol{n}^T + \frac{1-\cos\theta}{\theta}\boldsymbol{n}^\wedge \tag{A.11}$$

$$\boldsymbol{R} = \cos\theta\boldsymbol{I} + (1-\cos\theta)\boldsymbol{n}\boldsymbol{n}^T + \sin\theta\boldsymbol{n}^T \tag{A.12}$$

Equation (A.12) is also known as Rodrigues' formula [163].

Hence, the original least-square minimisation in Equation (A.5) can be rewritten in the domain of $\mathfrak{se}(3)$:

$$\mathcal{F} = \frac{1}{2}\sum_{i=1}^{m} \left\|\boldsymbol{\Sigma}^{-1/2} \cdot \boldsymbol{g}_i(\boldsymbol{\xi})\right\|_2^2 = \frac{1}{2}\sum_{i=1}^{m} \left\|\boldsymbol{\Sigma}^{-1/2} \cdot \boldsymbol{\epsilon}_i(\exp(\boldsymbol{\xi}^\wedge)^{-1})\right\|_2^2 \tag{A.13}$$

where $\boldsymbol{g}_i(\cdot)$ is the function of $\boldsymbol{\epsilon_i}(\cdot)$ in the domain of $\mathfrak{se}(3)$. The Jacobian $\boldsymbol{J}$ of the function $\boldsymbol{g}$ in Equation (A.13) is computed from the perturbed expression:

$$\boldsymbol{J} = \frac{\partial \boldsymbol{g}(\boldsymbol{\xi})}{\partial \Delta\boldsymbol{\xi}} = \lim_{\Delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\boldsymbol{g}(\boldsymbol{\xi} \boxplus \Delta\boldsymbol{\xi}) - \boldsymbol{g}(\boldsymbol{\xi})}{\Delta\boldsymbol{\xi}} = \frac{\partial \boldsymbol{g}(\boldsymbol{\xi})}{\partial[\exp(\boldsymbol{\xi}^\wedge)^{-1}\boldsymbol{p}]} \cdot \frac{\partial[\exp(\boldsymbol{\xi}^\wedge)^{-1}\boldsymbol{p}]}{\partial \Delta\boldsymbol{\xi}} \tag{A.14}$$

where $\boxplus$ is the plus operator for the left perturbation in $\mathfrak{se}(3)$. In this thesis, left perturbation was used, i.e., $\boldsymbol{\xi} \boxplus \Delta\boldsymbol{\xi} = \exp(\Delta\boldsymbol{\xi}^\wedge)\exp(\boldsymbol{\xi}^\wedge)$.

The Jacobian $\boldsymbol{J}$ consists of two terms by the chain rule of derivatives: $j_1 = \partial \boldsymbol{g}/\partial[\exp(\boldsymbol{\xi}^\wedge)^{-1}\boldsymbol{p}]$ and $j_2 = \partial[\exp(\boldsymbol{\xi}^\wedge)^{-1}\boldsymbol{p}]/\partial\Delta\boldsymbol{\xi}$. The first term j_1 is the partial derivative of the nonlinear function $\boldsymbol{g}$ with respect to the transformed point $\boldsymbol{p}$. The second term j_2 is the Jacobian of $\exp(\boldsymbol{\xi}^\wedge)^{-1}\boldsymbol{p}$ with respect to the perturbation $\Delta\boldsymbol{\xi}$.

The geometric and photometric Jacobian matrices are detailed in Equation (A.15) and Equation (A.16), respectively.

$$\boldsymbol{J}_g = \boldsymbol{n}_f^T \begin{bmatrix} \boldsymbol{R}^T & -\boldsymbol{R}^T\boldsymbol{p}^\wedge \\ \mathbf{0}_{3\times1}^T & \mathbf{0}_{3\times1}^T \end{bmatrix} \tag{A.15}$$

$$\boldsymbol{J}_p = \frac{\partial \boldsymbol{I}_f}{\partial \boldsymbol{u}} \cdot \begin{bmatrix} \frac{f_x}{z} & 0 & -\frac{x\cdot f_x}{z^2} \\ 0 & \frac{f_y}{z} & -\frac{y\cdot f_y}{z^2} \end{bmatrix} \cdot \begin{bmatrix} -\boldsymbol{R}^T & \boldsymbol{R}^T\boldsymbol{p}^\wedge \\ \mathbf{0}_{3\times1}^T & \mathbf{0}_{3\times1}^T \end{bmatrix} \tag{A.16}$$

where $\boldsymbol{R}$ is the rotation matrix of $\boldsymbol{T}$, $\boldsymbol{u}$ is the pixel obtained by the projection of $\pi(\boldsymbol{T}^{-1}\boldsymbol{p})$, $\partial \boldsymbol{I}_f/\partial \boldsymbol{u}$ is the intensity gradient of $\boldsymbol{I}_f$ at the pixel point $\boldsymbol{u}$, f_x and f_y are the focal lengths of the pin-hole camera model, and $(x, y, z)^T$ is the coordinates of the point $\boldsymbol{T}^{-1}\boldsymbol{p}$.

The camera pose is updated in Equation (A.17) using the calculated increment $\Delta\boldsymbol{\xi}$ in the tangent space $\mathfrak{se}(3)$:

$$\boldsymbol{T}^* = \exp((\boldsymbol{\xi}^*)^\wedge) = \exp(\Delta\boldsymbol{\xi}^\wedge)\exp(\boldsymbol{\xi}^\wedge) \tag{A.17}$$

Appendix B

Statistical Summary for the Estimated Camera Trajectories

Table B.1: Statistical summary of the ATEs for MS and TUM RGB-D datasets (unit: m). The median and extreme values are reported together with the inter-quartile range (IQR)

		Median	Min.	Max.	IQR
MS chess	F2F RGB	0.098691	0.047708	0.19792	0.068244
	F2F RGB-D	0.067239	0.014328	0.195	0.063675
	F2M RGB	0.067164	0.025738	0.21927	0.027467
	F2M RGB-D	0.046643	0.0044298	0.13361	0.030139
MS head	F2F RGB	0.12998	0.013952	0.2678	0.091313
	F2F RGB-D	0.11299	0.0070738	0.24058	0.08143
	F2M RGB	0.047473	0.01633	0.10125	0.042114
	F2M RGB-D	0.029832	0.0024324	0.06794	0.025021
MS office	F2F RGB	0.22531	0.14002	0.54901	0.11835
	F2F RGB-D	0.13339	0.062158	0.32311	0.08484
	F2M RGB	0.12269	0.014271	0.27747	0.13366
	F2M RGB-D	0.052747	0.0046541	0.099977	0.034225
MS pumpkin	F2F RGB	0.19796	0.090165	0.78172	0.098001
	F2F RGB-D	0.14547	0.021101	0.72542	0.12383
	F2M RGB	0.090241	0.035424	0.57086	0.078373
	F2M RGB-D	0.095582	0.015666	0.62885	0.062047
MS redkitchen	F2F RGB	0.071096	0.022448	0.16108	0.032551
	F2F RGB-D	0.06239	0.018216	0.15817	0.059917
	F2M RGB	0.029918	0.0089423	0.13063	0.033284
	F2M RGB-D	0.056943	0.0082035	0.13468	0.030237
TUM f1 desk	F2F RGB	0.12318	0.027541	0.19252	0.057601
	F2F RGB-D	0.093196	0.026194	0.21345	0.038267
	F2M RGB	0.050788	0.0090212	0.097361	0.032445
	F2M RGB-D	0.020482	0.0070146	0.066944	0.009176
TUM f1 rpy	F2F RGB	0.042137	0.0059101	0.21881	0.042915
	F2F RGB-D	0.040617	0.011576	0.11352	0.028411
	F2M RGB	0.096246	0.021387	0.96112	0.050415
	F2M RGB-D	0.021498	0.0038408	0.14239	0.016153
TUM f1 xyz	F2F RGB	0.04623	0.016258	0.087875	0.031355
	F2F RGB-D	0.026593	0.0075383	0.062561	0.016895
	F2M RGB	0.008281	0.0026348	0.025394	0.0063001
	F2M RGB-D	0.0075137	0.0025819	0.031694	0.005498
TUM f3 structure texture far	F2F RGB	0.028859	0.0090682	0.043649	0.011855
	F2F RGB-D	0.040087	0.019766	0.079581	0.010229
	F2M RGB	0.0086202	0.0035919	0.029935	0.0050712
	F2M RGB-D	0.0095589	0.0012205	0.034166	0.0077421
TUM f3 structure texture near	F2F RGB	0.027848	0.0072584	0.050278	0.014501
	F2F RGB-D	0.022574	0.0093403	0.074481	0.010096
	F2M RGB	0.010626	0.0024519	0.039063	0.0061792
	F2M RGB-D	0.015105	0.0065777	0.1119	0.0080243

Table B.2: Statistical summary of the relative translational errors on MS and TUM RGB-D datasets. The median and extreme values are reported together with the inter-quartile range (IQR)

		median	min.	max.	IQR
MS chess	F2F RGB	0.096143	0.025418	0.31869	0.059564
	F2F RGB-D	0.072819	0.015921	0.23741	0.048539
	F2M RGB	0.069053	0.013825	0.18847	0.059607
	F2M RGB-D	0.044686	0.012664	0.12757	0.029932
MS head	F2F RGB	0.13449	0.021262	0.29768	0.1408
	F2F RGB-D	0.12475	0.009891	0.26667	0.12766
	F2M RGB	0.06702	0.014313	0.11292	0.034197
	F2M RGB-D	0.041975	0.010672	0.074976	0.02967
MS office	F2F RGB	0.205	0.025362	0.4011	0.21996
	F2F RGB-D	0.13713	0.04977	0.29997	0.063089
	F2M RGB	0.17328	0.046776	0.47653	0.1553
	F2M RGB-D	0.077446	0.020259	0.17823	0.065691
MS pumpkin	F2F RGB	0.19758	0.020916	0.66288	0.17692
	F2F RGB-D	0.15094	0.013662	0.65669	0.14926
	F2M RGB	0.14197	0.030056	0.72205	0.13772
	F2M RGB-D	0.098585	0.007498	0.76706	0.1111
MS redkitchen	F2F RGB	0.077025	0.016543	0.22079	0.039877
	F2F RGB-D	0.072151	0.032583	0.16443	0.041177
	F2M RGB	0.060245	0.015971	0.12996	0.025166
	F2M RGB-D	0.057433	0.011379	0.19345	0.05572
TUM f1 desk	F2F RGB	0.13029	0.03823	0.2577	0.089902
	F2F RGB-D	0.087831	0.029816	0.23825	0.066699
	F2M RGB	0.058603	0.022602	0.22785	0.043698
	F2M RGB-D	0.044741	0.006481	0.10511	0.023121
TUM f1 rpy	F2F RGB	0.049281	0.007658	0.1699	0.048034
	F2F RGB-D	0.037488	0.007483	0.12925	0.028327
	F2M RGB	0.027721	0.004416	1.0673	0.025654
	F2M RGB-D	0.024265	0.009026	0.1527	0.01485
TUM f1 xyz	F2F RGB	0.03409	0.006956	0.095502	0.022024
	F2F RGB-D	0.02257	0.006568	0.065025	0.019586
	F2M RGB	0.012363	0.000683	0.033849	0.01035
	F2M RGB-D	0.010821	0.003069	0.033586	0.010441
TUM f3 structure texture far	F2F RGB	0.024556	0.004562	0.069013	0.020233
	F2F RGB-D	0.023925	0.006096	0.060869	0.012483
	F2M RGB	0.011114	0.001362	0.035849	0.009928
	F2M RGB-D	0.010902	0.00159	0.045209	0.0079155
TUM f3 structure texture near	F2F RGB	0.023178	0.003302	0.04767	0.018696
	F2F RGB-D	0.02523	0.003743	0.09094	0.021746
	F2M RGB	0.011546	0.001691	0.050887	0.010878
	F2M RGB-D	0.010999	0.000776	0.12653	0.009629

Table B.3: Statistical summary of the relative rotational errors on MS and TUM RGB-D datasets. The median and extreme values are reported together with the inter-quartile range (IQR)

		median	min.	max.	IQR
MS chess	F2F RGB	0.070619	0.006253	0.21942	0.053687
	F2F RGB-D	0.054148	0.003384	0.17358	0.040892
	F2M RGB	0.053246	0.00793	0.15785	0.032948
	F2M RGB-D	0.030782	0.006117	0.097783	0.013904
MS head	F2F RGB	0.08491	0.02519	0.3122	0.1452
	F2F RGB-D	0.077336	0.022087	0.26074	0.12876
	F2M RGB	0.051425	0.011519	0.1354	0.0537
	F2M RGB-D	0.042707	0.01063	0.08394	0.03595
MS office	F2F RGB	0.15248	0.040946	0.21879	0.04176
	F2F RGB-D	0.10151	0.025973	0.14566	0.023198
	F2M RGB	0.11427	0.023222	0.28118	0.13491
	F2M RGB-D	0.036969	0.014009	0.11877	0.032159
MS pumpkin	F2F RGB	0.093738	0.024645	0.19551	0.0548
	F2F RGB-D	0.091262	0.021619	0.1864	0.062763
	F2M RGB	0.077217	0.010157	0.2297	0.076108
	F2M RGB-D	0.045907	0.006035	0.14173	0.046993
MS redkitchen	F2F RGB	0.057339	0.003904	0.13989	0.057294
	F2F RGB-D	0.060918	0.007421	0.12037	0.061693
	F2M RGB	0.056826	0.011384	0.15466	0.044878
	F2M RGB-D	0.031886	0.007251	0.085167	0.024288
TUM f1 desk	F2F RGB	0.10942	0.02439	0.1944	0.081224
	F2F RGB-D	0.10206	0.014722	0.15508	0.071948
	F2M RGB	0.062289	0.00926	0.20077	0.039275
	F2M RGB-D	0.036191	0.005736	0.083874	0.024556
TUM f1 rpy	F2F RGB	0.054652	0.018713	0.1315	0.041835
	F2F RGB-D	0.049592	0.023359	0.12205	0.031692
	F2M RGB	0.03569	0.005193	0.28858	0.038018
	F2M RGB-D	0.026185	0.005724	0.092292	0.025268
TUM f1 xyz	F2F RGB	0.034371	0.008392	0.10456	0.026208
	F2F RGB-D	0.026096	0.007785	0.079924	0.015325
	F2M RGB	0.010512	0.002656	0.034165	0.0081585
	F2M RGB-D	0.009924	0.00091	0.028161	0.0079693
TUM f3 structure texture far	F2F RGB	0.01315	0.00111	0.036181	0.010816
	F2F RGB-D	0.01193	0.002827	0.026384	0.0068935
	F2M RGB	0.007138	0.000907	0.01763	0.0056142
	F2M RGB-D	0.007624	0.002174	0.022556	0.005262
TUM f3 structure texture near	F2F RGB	0.018439	0.003783	0.042065	0.010389
	F2F RGB-D	0.023047	0.001532	0.096124	0.025898
	F2M RGB	0.010098	0.00069	0.042753	0.008688
	F2M RGB-D	0.011636	0.001575	0.11476	0.008447

Appendix C

On-Manifold Optimisation with Levenberg-Marquardt Method

Appendix A explains that it is impossible to perform direct optimisation in $SE(3)$ matrix space for camera pose estimation. This section adopts the same on-manifold optimisation framework with Levenberg-Marquardt (LM) method.

The Levenberg-Marquardt method is a trust-region approach performing reliable and robust minimisation of the objective function. The objective function is approximated over a subset of the solution domain (i.e., trust region). The length of update steps within the trust region is adjusted adaptively. This approach interpolates between steepest descent and Gauss-Newton methods by adaptively controlling the size of trust region.

Given an objective function in Equation (4.1), the camera pose $\boldsymbol{T}_n \in SE(3)$ is parameterised by a 6D vector $\boldsymbol{\xi}_n \in \mathfrak{se}(3)$ in the tangent space of $SE(3)$ according to the formulation in Appendix A, and the error functions $\boldsymbol{\epsilon}_i(\boldsymbol{T}_n)$ in Equation (4.1) is reformed as $\boldsymbol{g}_i(\boldsymbol{\xi}_n)$. The objective function $\mathcal{F}(\boldsymbol{T}_n)$ is reorganised as $\mathfrak{F}(\boldsymbol{\xi}_n)$ as follows:

$$\boldsymbol{\xi}_n = \arg\min_{\boldsymbol{\xi}_n} \mathfrak{F}(\boldsymbol{\xi}_n) = \arg\min_{\boldsymbol{\xi}_n} \frac{1}{2} \| \boldsymbol{E}(\boldsymbol{\xi}_n) \|_2^2 \tag{C.1}$$

$$\boldsymbol{E}(\boldsymbol{\xi}_n) = \boldsymbol{\Sigma}_{\boldsymbol{E}}^{-1/2} \cdot \begin{bmatrix} \boldsymbol{g}_1^T \\ \boldsymbol{g}_2^T \\ \cdots \\ \boldsymbol{g}_m^T \end{bmatrix} = \begin{bmatrix} \sigma_{g,1}^{-1} & & & & & & & \\ & \sigma_{p,1}^{-1} & & & & & & \\ & & \sigma_{g,2}^{-1} & & & & & \\ & & & \sigma_{p,2}^{-1} & & & & \\ & & & & \cdots & & & \\ & & & & & \sigma_{g,m}^{-1} & & \\ & & & & & & \sigma_{p,m}^{-1} & \end{bmatrix} \cdot \begin{bmatrix} g_{g,1} \\ g_{p,1} \\ \cdots \\ g_{g,m} \\ g_{p,m} \end{bmatrix} \tag{C.2}$$

where $\boldsymbol{E}$ is the multidimensional vector which contains the error functions $\boldsymbol{g}_{1:m}$, $\boldsymbol{\Sigma}_{\boldsymbol{E}}$ is a diagonal covariance matrix for $\boldsymbol{E}$ as shown in Equation (C.2).

The LM method starts by linearising the error vector $\boldsymbol{E}$ with a damped constraint within the trust region:

$$\boldsymbol{\xi}_n = \arg\min_{\boldsymbol{\xi}_n} \mathfrak{F}(\boldsymbol{\xi}_n) = \arg\min_{\boldsymbol{\xi}_n} \frac{1}{2} \left(\|\boldsymbol{E}(\boldsymbol{\xi}_0) + \boldsymbol{J}\Delta\boldsymbol{\xi}\|^2 + \lambda \|\boldsymbol{D}\Delta\boldsymbol{\xi}\|^2 \right), \ \boldsymbol{\xi}_n, \boldsymbol{\xi}_0, \Delta\boldsymbol{\xi} \in \mathfrak{se}(3) \tag{C.3}$$

where $\boldsymbol{J}$ is the Jacobian of $\boldsymbol{E}$, λ is the damping factor, and $\boldsymbol{D}$ is a non-negative diagonal matrix either set as identity or $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{J}^T\boldsymbol{J})$. In this study, the form of $\boldsymbol{D}$ is set as an identity matrix.

The increment $\Delta\boldsymbol{T}$ is determined by identifying the stationary point of Equation (C.3):

$$(\boldsymbol{J}^T\boldsymbol{J} + \lambda\boldsymbol{I})\Delta\boldsymbol{\xi} = -\boldsymbol{J}^T\boldsymbol{E}(\boldsymbol{\xi}_0) \tag{C.4}$$

Equation (C.4) is a 6×6 linear system, and the increment $\Delta\boldsymbol{\xi}$ can be solved by matrix factorisation (e.g., Cholesky decomposition).

Before proceeding to the next iteration with the updated the camera pose, a quality factor ρ which measures the performance of the approximation is computed:

$$\rho = \frac{\|\boldsymbol{E}(\boldsymbol{\xi}_0 + \Delta\boldsymbol{\xi})\|^2 - \|\boldsymbol{E}(\boldsymbol{\xi}_0)\|^2}{\|\boldsymbol{J}\Delta\boldsymbol{\xi} + \boldsymbol{E}(\boldsymbol{\xi}_0)\|^2 - \|\boldsymbol{E}(\boldsymbol{\xi}_0)\|^2} \tag{C.5}$$

In Equation (C.5), the numerator is the exact descent of the quadratic model of $\boldsymbol{E}(\cdot)$, and the denominator is the decrease of the approximated quadratic model $(\boldsymbol{E}(\boldsymbol{\xi}_0) + \boldsymbol{J}\Delta\boldsymbol{\xi})$ around $\boldsymbol{\xi}_0$.

If $\rho > 0$, the updated camera pose will be accepted, and the trust region is expanded accordingly with the quality of ρ by reducing the damping factor λ. The optimisation process behaves closer to the steepest descent in this case. Otherwise, the approximated model is not reliable. The trust region is shrunk by increasing λ, giving the updating step closer to the Gauss-Newton method. The initialisation of λ and the adjustment strategy for the trust region is detailed in [38].

The updating scheme of this nonlinear optimisation problem applied the same strategy of on-manifold optimisation as elaborated in Appendix A.